



TIBCO® Product and Service Inventory

Installation and Configuration

*Version 2.1.0
August 2022*



Contents

Contents	2
Installation Overview	4
Preinstallation	4
Installation	4
Postinstallation	5
Operating System and Database Requirements	5
Operating System	5
Databases	5
Installation Guidelines	6
Required Products	6
Required TIBCO Products	6
Required Third-Party Products	7
Installing TIBCO Product and Service Inventory	9
Postinstallation Tasks	11
Postinstallation Task 1: Copying Dependencies	11
Postinstallation Task 2: Creating the Database	11
Creating a PostgreSQL Database for the Admin User	12
Creating an Oracle Database for the Admin User	13
Creating a PostgreSQL Database for the TIBCO PSI User	14
Creating an Oracle Database for the TIBCO PSI User	16
Postinstallation Task 3: Creating the TIBCO Enterprise Message Service Channel	17
Postinstallation Task 4: Configuring the Installation	17
Postinstallation Task 5: Creating an Admin User	24
Postinstallation Task 6: Uploading Seed Data	24
Postinstallation Task 7: Starting or Restarting the Services	25

Installing Helm Chart	26
TIBCO Product and Service Inventory as a Container on Kubernetes	28
TIBCO Product and Service Inventory Bulk Load	29
Bulk Load for Oracle Database	29
Configuring for the Bulk Load for Oracle Database	29
Bulk Load for PostgreSQL Database	36
Configuring for the Bulk Load for PostgreSQL Database	36
Triggering the Bulk Load	38
Bulk Load Logging for Oracle Database	39
Bulk Load Logging for PostgreSQL Database	40
TIBCO Documentation and Support Services	42
Product-Specific Documentation	42
Legal and Third-Party Notices	44

Installation Overview

This chapter provides information about the system requirements in terms of operating systems and disk space, recommended setup for TIBCO® Product and Service Inventory (formerly TIBCO® Fulfillment Subscriber Inventory) installation, variables and properties required, and important files necessary to perform the post-install log analysis.

Provided below are links to detailed information for preinstallation requirements and settings, installation, and the post-installation steps.

Preinstallation

To ensure that you have a good experience installing TIBCO Product and Service Inventory, it is always recommended to check whether your computer is ready for installation. The preinstallation section gives you the prerequisites to install TIBCO Product and Service Inventory:

1. [Operating System and Database Requirements](#): This section provides you with information about the platforms that TIBCO Product and Service Inventory supports.
2. [Installation Guidelines](#): This section provides details about the installation guidelines for TIBCO Product and Service Inventory.
3. [Required Products](#): TIBCO Product and Service Inventory requires some software components to be installed. For a complete list of versions and platforms supported, see the `TIB_psi_2.1.0_readme.txt` file. Install and configure them in the mentioned order.
4. [Installation Related Files](#): This section provides information regarding disk space requirements, installation registry, history, and log files.

Installation

- [Installing TIBCO Product and Service Inventory](#): Step-by-step instructions are

provided in this chapter to help you install TIBCO Product and Service Inventory using different installation modes.

Postinstallation

Post-Installation Tasks: After you complete the installation of TIBCO Product and Service Inventory, complete the postinstallation tasks and verify the installation.

Operating System and Database Requirements

TIBCO Product and Service Inventory supports the following platforms and databases:

Operating System

Operating System
Red Hat Enterprise Linux Server 8.X 64-bit on x86-64

Databases

Database
Oracle 19c Enterprise Edition (Oracle 19.3.0.0.0), single and RAC
PostgreSQL 12



Note: Oracle Databases use traditional (non-container) Enterprise Edition releases.

Installation Guidelines

Use the following installation guidelines when installing TIBCO Product and Service Inventory:

Installer Account	TIBCO Product and Service Inventory can be installed by all users, including regular (non-root) user and super-user (root).
Installing from a Network Drive	To install the product from a network drive, you must ensure that the account used for installation has permission to access the network drive.
Install Required Software First for All the Installations	TIBCO recommends that you install TIBCO Enterprise Messaging Service™ before installing TIBCO Product and Service Inventory.

Required Products

You must install the third-party products and TIBCO products listed in this section.

Required TIBCO Products

The following table lists the required TIBCO product for TIBCO Product and Service Inventory:

Required TIBCO Products for TIBCO Product and Service Inventory

Product & Version	Mandatory/Optional	Purpose	Reference
TIBCO Enterprise Message Service™ 8.3.x or 8.4.x	It is mandatory if notifications are to be enabled on Enterprise Message Service™, but else optional.	Standards-based messaging software that can serve as the backbone of an SOA by providing Java Message Service (JMS)-compliant communications across a wide range of platforms and	<i>TIBCO Enterprise Message Service™ Installation</i>

Product & Version	Mandatory/Optional	Purpose	Reference
-------------------	--------------------	---------	-----------

application technologies.

Note:

Enterprise Message Service™ license is not a part of the TIBCO Product and Service Inventory product license and must be acquired separately.

Required Third-Party Products

This section lists the required third-party products.

Jars	Download link
antlr-2.7.7	https://repo1.maven.org/maven2/antlr/antlr/2.7.7/antlr-2.7.7.jar
dom4j-2.1.1	https://repo1.maven.org/maven2/org/dom4j/dom4j/2.1.1/dom4j-2.1.1.jar
hibernate-commons-annotations-5.1.2.Final	https://repo1.maven.org/maven2/org/hibernate/common/hibernate-commons-annotations/5.1.2.Final/hibernate-commons-annotations-5.1.2.Final.jar
hibernate-core-5.6.7.Final	https://repo1.maven.org/maven2/org/hibernate/hibernate-core/5.6.7.Final/hibernate-core-5.6.7.Final.jar
hibernate-entitymanager-5.6.7.Final	https://repo1.maven.org/maven2/org/hibernate/hibernate-entitymanager/5.6.7.Final/hibernate-entitymanager-5.6.7.Final.jar
hibernate-jpa-2.1-api-1.0.0.Final	https://repo1.maven.org/maven2/org/hibernate/javax/persistence/hibernate-jpa-2.1-api/1.0.0.Final/hibernate-jpa-2.1-api-1.0.0.Final.jar

Jars	Download link
javax.servlet-api-4.0.1	https://repo1.maven.org/maven2/javax/servlet/javax.servlet-api/4.0.1/javax.servlet-api-4.0.1.jar

i Note: When you obtain third-party software or services, it is your responsibility to ensure you understand the license terms associated with such third-party software or services and comply with such terms.

JDK Environment Variables

Create the following environment variables after installing JDK:

- Set JAVA = /usr/java/jdk-11.0.4/bin/java
- Set JAVA_HOME = /usr/java/jdk-11.0.x
- In the PATH variable, add JAVA_HOME/bin.

Installing TIBCO Product and Service Inventory

This section describes the process for installing TIBCO Product and Service Inventory. Before you begin, ensure that the following requirements are met.

Requirements

All TIBCO PSI microservices configurations must be configured externally.

Downloading and Installing TIBCO Product and Service Inventory

Download the TIBCO Product and Service Inventory build from [TIBCO eDelivery](#). The following project artifacts are downloaded in the `PSI_HOME` folder.

- `TIB_psi_2.1.0.zip` - extract to the TIBCO PSI folder
- `jdk-11.0.4_linux-x64_bin.tar.gz` (You must use this version of jdk)

The following components are present in the `TIB_psi_2.1.0.zip` file.

Component	Details
Docker	Contains files and artifacts to create and run Docker containers for all TIBCO PSI microservices
database-scripts	Contains Oracle and PostgreSQL database scripts.
ems	Contains files related to the creation or deletion of queues or topics.
externalLib	The external jars required for all microservices need to be copied here.
roles	Contains the <code>copyLib.sh</code> script and the configuration files

Component	Details
	for all the microservices.
samples	Sample models, web service requests, and response files.
helm	Contains all files required to deploy the authorization service and all other TIBCO PSI services Helm charts.
schemas	Contains the wsdl and xsd's required to execute SOAP requests.
seed-data	In this directory, the configuration files and application properties are stored. For more details on the configuration, see the "Configurator UI" section in the <i>TIBCO® Product and Service Inventory User Guide</i> .
Kubernetes	Contains Kubernetes scripts for all TIBCO PSI services.



Note: For a complete list of versions and supported platforms, see the `TIB_psi_2.1.0_readme.txt` file.

Postinstallation Tasks

This section describes postinstallation tasks.

Postinstallation Task 1: Copying Dependencies

TIBCO PSI does not include all the required third-party libraries. Ensure that the third-party libraries are present at the `$PSI_HOME/externalLib` directory and then run the `copyLib.sh` script.

Postinstallation Task 2: Creating the Database

If this is your first time installing TIBCO Product and Service Inventory, then create the database by running the provided scripts.



Note: To run the SQL scripts, you must have appropriate permission.

TIBCO Product and Service Inventory contains the following types of database:

- Admin database: You can either use Oracle or PostgreSQL. For more information about creating an admin user in the Oracle and PostgreSQL database, see the following topics:
 - [Creating a PostgreSQL Database for the Admin User](#)
 - [Creating an Oracle Database for the Admin User](#)
- In-memory database: You can use Oracle/PostgreSQL (relational DB).

You can choose any combination of the supported in-memory database types or you can also choose one database for both PSI user and Admin data.

- [Creating a PostgreSQL Database for TIBCO PSI User](#)
- [Creating an Oracle Database for TIBCO PSI User](#)

i Note: After executing the database scripts, you must run the `$PSI_HOME/roles/copyLib.sh` script before modifying the properties files of any service.

Creating a PostgreSQL Database for the Admin User

Create a PostgreSQL database for the Admin User and Default Tenant by configuring and running the following script:

`$PSI_HOME/database-scripts/admin/postgreSQL/bin/db-setup.sh`

Procedure

1. Open the `$PSI_HOME/database-scripts/admin/postgreSQL/bin/postgres_admin_db.properties` file in a suitable editor and update the following values in the script:

Property	Value
PG_HOME	C:/Program Files/PostgreSQL/11/bin
PG_HOST	PostgreSQL database host
PG_PORT	PostgreSQL port
PG_SUPER_USER_NAME	PostgreSQL superuser name
PG_SUPER_USER_PASSWORD	PostgreSQL superuser password
PG_SUPER_USER_DATABASE	PostgreSQL superuser database
PG_ADMIN_USER	New admin user
PG_ADMIN_PASSWORD	Password for the admin user
PG_ADMIN_DATABASE	admin database name
PG_ADMIN_SCHEMA	admin schema

Property	Value
PG_ADMIN_TABLESPACE_LOCATION	admin tablespace location
PG_ADMIN_TABLESPACE	admin tablespace
IS_CLOUD_PLATFORM	false
EXECUTE_DDL_DML_ONLY	false
BASE_INSTALLATION_SCRIPTS	database_ddl.sql (End user must not modify this value)

2. Save and close the file.
3. Run the `db-setup.sh` script.

Creating an Oracle Database for the Admin User

Create an Oracle database for the admin user by running the `$PSI_HOME/database-scripts/admin/oracle/bin/db-setup.sh` script that is provided.

Procedure

1. Open the `$PSI_HOME/database-scripts/admin/oracle/bin/oracle_admin_db.properties` file in a suitable editor and update the following values in the script:

Property	Value
ORCL_HOST	Oracle database host
ORCL_PORT	Oracle port
ORCL_USERNAME	Oracle user name
ORCL_PASSWORD	Oracle user password

Property	Value
ORCL_SERVICENAME	Oracle service name
ORCL_ADMIN_USER	New admin user
ORCL_ADMIN_PASSWORD	Password for the admin user
ORCL_ADMIN_TABLESPACE	admin tablespace
ORCL_TABLESPACE_SIZE	admin tablespace size
ORCL_MINSIZE	admin table minimum size
ORCL_MAXSIZE	admin table maximum size
ORCL_DATAFILE_PATH	Oracle datafile path
IS_CLOUD_PLATFORM	false
EXECUTE_DDL_DML_ONLY	false
BASE_INSTALLATION_SCRIPTS	database_ddl.sql (Not to be modified by end user)

2. Save and close the file.
3. Run the `db-setup.sh` script.

Creating a PostgreSQL Database for the TIBCO PSI User

Create a PostgreSQL database for the TIBCO PSI user and default Tenant by configuring and running the following script:

```
$PSI_HOME/database-scripts/psi/postgreSQL/bin/db-setup.sh
```

Before you begin

In bulkload API, create internal tables on the psi server by using the `file_fdw` and `adminpack` PostgreSQL extensions. If these two extensions are not available by default,

install them manually.

Procedure

1. Open the `$PSI_HOME/database-scripts/psi/postgreSQL/bin/postgres_psi_db.properties` file in a suitable editor and update the following values in the script:

Property	Value
PG_HOME	C:/Program Files/PostgreSQL/11/bin
PG_HOST	PostgreSQL database host
PG_PORT	PostgreSQL port
PG_SUPER_USER_NAME	PostgreSQL superuser name
PG_SUPER_USER_PASSWORD	PostgreSQL superuser password
PG_SUPER_USER_DATABASE	PostgreSQL superuser database
PG_PSI_USER	New TIBCO PSI user
PG_PSI_PASSWORD	Password for the TIBCO PSI user
PG_PSI_DATABASE	TIBCO PSI database name
PG_PSI_SCHEMA	TIBCO PSI schema
PG_PSI_TABLESPACE_LOCATION	PSI tablespace location
PG_PSI_TABLESPACE	TIBCO PSI tablespace
IS_CLOUD_PLATFORM	false
EXECUTE_DDL_DML_ONLY	false
BASE_INSTALLATION_SCRIPTS	fos_psi_ddl.sql

2. Save and close the file.

3. Run the `db-setup.sh` script.

Creating an Oracle Database for the TIBCO PSI User

Create an Oracle database for the TIBCO PSI user by running the provided script `$PSI_HOME/database-scripts/psi/oracle/bin/db-setup.sh`.

Procedure

1. Open the `$PSI_HOME/database-scripts/psi/oracle/bin/oracle_psi_db.properties` file in a suitable editor and update the following values in the script:

Property	Value
ORCL_HOST	Oracle database host
ORCL_PORT	Oracle port
ORCL_USERNAME	Oracle user name
ORCL_PASSWORD	Oracle user password
ORCL_SERVICENAME	Oracle service name
ORCL_PSI_USER	New TIBCO PSI user
ORCL_PSI_PASSWORD	Password for the TIBCO PSI user
ORCL_PSI_TABLESPACE	TIBCO PSI tablespace
ORCL_TABLESPACE_SIZE	TIBCO PSI tablespace size
ORCL_MINSIZE	TIBCO PSI table minimum size
ORCL_MAXSIZE	TIBCO PSI table maximum size
ORCL_DATAFILE_PATH	Oracle datafile path

Property	Value
IS_CLOUD_PLATFORM	false
EXECUTE_DDL_DML_ONLY	false
BASE_INSTALLATION_SCRIPTS	fos_psi_ddl.sql

2. Save and close the file.
3. Run the `db-setup.sh` script.

Postinstallation Task 3: Creating the TIBCO Enterprise Message Service Channel

To create the TIBCO Enterprise Message Service channels, run the `tibemsadmin` command.

1. Go to `$EMS_HOME/bin` and run the following command:

```
$ tibemsadmin -server tcp://localhost:7222 -user admin -script $PSI_
HOME/ems/inventoryEMS.txt
```

Postinstallation Task 4: Configuring the Installation

Configuring the installation deals with tasks like configuring the database and messaging.

In the `application.properties` file of the configurator, set the following configurations:

- If you use a relational database, then set `pluggableCache=Relational` and configure the admin database details from the 'Relational Database Connection Properties' section accordingly.

Also, configure admin database details accordingly in the `application.properties` file.

Before running all the services, you have to configure the properties from the following table in the `$PSI_HOME/roles/<servicename>/standalone/config/application.properties` file.

Service	Element	Value
Authorization Service	server.port	9091

Service	Element	Value
	amPluggableCache	Relational (based on the DB configured)

Service	Element	Value
	allowedUserRoles	ROLE_ADMIN, ROLE_USER, ROLE_PARTY, and ROLE_ITEM
	datasourceDriverClassName	org.postgresql.Driver (Postgres) or oracle.jdbc.driver.OracleDriver (Oracle)
	adminDsUrl	jdbc:postgresql://localhost:5432/admindbll?current (PostgreSQL) or jdbc:oracle:thin:@//localhost:1521/pdb1 (Oracle)
	adminDsUsername	adminuserll
	adminDsPassword	adminuserll
	hibernateDialect	org.hibernate.dialect.PostgreSQLDialect (Postgres) or org.hibernate.dialect.Oracle10gDialect (Oracle)
	datasourceValidationQuery	SELECT 1 (Postgres) or select 1 from dual (Oracle)
	adminRedisHost	localhost
	adminRedisPort	6379
	default.tenant.id	TIBCO
	auth.superuser.appId	auth
	auth.superuser.appKey	ENC(P2yXphz4OVM=)
	hibernateDsDefaults	false
	adminHibernateShowSql	false

Service	Element	Value
	adminDsInitialSize	10
	adminDsMaxWait	30000
	adminDsMaxActive	100
	adminDsMaxIdle	100
	adminDsMinIdle	10
	adminDsTestOnBorrow	true
	adminDsValidationInterval	5000

Service	Element	Value
Configurator service	server.port	9090
	pluggableCache	Redis/Relational (based on the DB configured)
	adminDsUrl	jdbc:postgresql://localhost:5432/admindbll?current (PostgreSQL) or jdbc:oracle:thin:@//localhost:1521/pdb1 (Oracle)
	adminDsUsername	adminuserll
	adminDsPassword	adminuserll
	datasourceDriverClassName	org.postgresql.Driver (Postgres) or oracle.jdbc.driver.OracleDriver (Oracle)
	hibernateDialect	org.hibernate.dialect.PostgreSQLDialect (Postgres) or org.hibernate.dialect.Oracle10gDialect (Oracle)
	datasourceValidationQuery	SELECT 1 (Postgres) or select 1 from dual (Oracle)
	adminRedisHostPort	localhost:6379
	security.key	ENC(nSa0k6lmjPPN8ZA5SO6BpQ==)

Service	Element	Value
Configurator UI	server.port	9104
	configuratorServiceUrl	http://localhost:9090
	authorizationServiceTokenEndPoint	http://localhost:9091
	authorized.client.id	order-management-client
	authorized.client.secret	ENC(ggsmFvh5HBbeSD1j+l5Y0rP4qv0rJvEm)
PSI	server.port	8080
	configuratorServiceUrl	http://localhost:9090
	security.key	ENC(nSa0k6lmjPPN8ZA5SO6BpQ==)

Note: You must add the same key as mentioned in the `application.properties` file of configurator service.

Postinstallation Task 5: Creating an Admin User

You must create an admin user with TIBCO as the tenant id value. The tenant id value is not set by default. The value of the `userRoles` property must be `ROLE_ADMIN`. For more information about creating a user, see the "Authorization Service" section in *TIBCO® Product and Service Inventory User Guide*.

Postinstallation Task 6: Uploading Seed Data

Upload the seed data (application properties and configuration files) from the configurator API or configurator UI. See the "Configurator UI" section in *TIBCO® Product and Service Inventory User Guide* if you want to execute from UI.

For configurator API, you can proceed with the following steps:

Procedure

1. Save `$PSI_HOME/psi/2.1.0/seed-data/application_metadata.json` file properties from the `[POST] /v1/configuration/configFilesMetadata` configurator API.
2. Save the application properties (`$PSI_HOME/seed-data/app-properties`) from `[POST] /v1/configuration/{applicationId}` API.
3. Upload the configuration files (`$PSI_HOME/seed-data/config-files`) from `[POST] /v1/configuration/configFile/{applicationId}` API.

i Note: Before uploading the seed data, you must run the `application_metadata.json` script from the configurator UI or Application Metadata API.

Postinstallation Task 7: Starting or Restarting the Services

To start or restart all the services, perform the following procedure:

i Note: You can configure the authorization service by adding the details in the `$PSI_HOME/roles/authorization-service/standalone/config/application.properties` file.

1. To start or restart the Configurator service, navigate to the `$PSI_HOME/roles/configurator/standalone/bin` directory and run the `start.sh` script.
2. To start or restart all other services, navigate to the `$PSI_HOME/roles/<service-name>/standalone/config` directory and update `application.properties` for `configuratorServiceUrl`. Then start the service from `$PSI_HOME/roles/<service-name>/standalone/bin` by running the `start.sh` script.

i Note: You must start the Configurator and Authorization services first and then all the other services.

Installing Helm Chart

By using a Helm chart, you can deploy all the services and pods at once, instead of deploying each service and package manually. A Helm chart supports most of the cloud platforms.

Prerequisites


1. Helm Client version 3.5.0 or later must be installed on your Kubernetes cluster.
2. The EMS server must be running with all the required queues, topics, and bridges.
3. Create database users by running the scripts present in the `$PSI_HOME/database-scripts` directory.
4. Ensure that the third-party libraries are present in the `$PSI_HOME/externalLib` directory and then run the `$PSI_HOME/roles/copyLib.sh` script.
5. To seed data and upload configuration files, start the configurator service.
6. Run the `$PSI_HOME/docker/copy-required-files.sh` script.
7. Create docker images for all the services.

Procedure

1. Set the environment variables and Docker image names for all the required services in the `values.yaml` file that is present at the `$PSI_HOME/helm/psi_services` folder.
2. Run a helm chart from the location where the copied directories are present.
Example: `helm install psi ./psi_services`.

Result

A Helm chart is deployed with all the services present in the chart.

 **Note:** Component-specific installation is now available in helm charts.

The following new properties have been added in the `values.yaml` file:

```
# --- Select components to be deployed -  
-----
```

```
authorizationService: true
```

```
configuratorService: true
```

```
configuratorUIService: true
```

By default, these properties are true. If you do not wish to deploy a specific service, set the value as false.

TIBCO Product and Service Inventory as a Container on Kubernetes

TIBCO Product and Service Inventory can be deployed as a container application. To containerize the application, you must build and run the Docker images using the bundled Dockerfile. Kubernetes is used to deploy TIBCO PSI Inventory and also to orchestrate the containers. The deployment architecture and the detailed instructions are written in the Kubernetes context because Kubernetes is used to deploy TIBCO PSI. Deployment uses Kubernetes concepts such as Deployment, pod, kubectl, Kubernetes master, and service. For more information, see the [Kubernetes documentation](#).

Procedure

1. Upload the properties with correct values in the database for common and psi applications. See the [Postinstallation Tasks](#) for more details.
2. Run the `copyLib` script from the `<PSI_HOME>/roles` directory.
3. Provide all the required properties for each service in `<PSI_HOME>/roles/<service-name>/standalone/config/application.properties` files.
4. Create all the required docker images from `<PSI_HOME>/docker` directory by following the readme present under the same location.
5. Copy the Kubernetes scripts from `<PSI_HOME>/kubernetes` directory to the Kubernetes cluster and modify them according to the required changes.
6. Modify the location for images in `image: tibco/authorization-service:2.1.0` script, where the docker images are pushed.
7. Deploy the scripts using the following command from where the Kubernetes scripts are copied.
Example: `kubectl apply -f kubernetes-deploy-run-authorization-service.yml`

TIBCO Product and Service Inventory Bulk Load

The bulk load service refers to the action of mass importing existing inventory data into the TIBCO Product and Service Inventory database. This feature is designed to support migration from an existing system where the data is in the form of CSV files. This chapter explains the requirements for the bulk load, configuration for the bulk load, triggering the bulk load, and the bulk load logging.

Bulk Load for Oracle Database

This section explains the bulk loading for Oracle Database.

Configuring for the Bulk Load for Oracle Database

The bulk load service provides a way to mass provision the inventory database with existing parties and items. A configuration file has been provided which needs to be configured with appropriate values before the bulk load is initiated.

Before you begin

The data to load must be converted as CSV files if they are not already in that format. The Oracle database is only able to read CSV files and treats these as a table. Inventory then applies the stored procedures to create internal items, item characteristics, item relationships, orders, parties, characteristics, relationships, and orders. For details, see [CSV File and JSON Config File Samples](#).

Procedure

1. If you have not already, create CSV files containing items, item characteristics, orders, item relationships, parties, and party characteristics, and put these files in a directory on the Oracle database server so that Oracle can access them.

2. Define the CSV file directory accessible by the Oracle database by running the following command:

```
create or replace directory PSI_BULK_DIR as '[existing directory  
absolute path on oracle server]';  
grant read, write on directory PSI_BULK_DIR to [inventory oracle  
user];
```

Both the directory and file need to be readable and writable by the Oracle process user.

3. Create a bulk load JSON configuration file (bulkConfig.json) on the TIBCO Product and Service Inventory server machine. For details, see [CSV File and JSON Config File Samples](#). This file can be saved in the \$PSI_HOME/config/ file.
4. Configure the following parameters in the bulk load JSON configuration file:

Parameter	Description
oracleDirectoryName	This is the directory name for the Oracle server. The value must be "PSI_BULK_DIR". The data files containing the data to be loaded must be placed in this directory which must be created beforehand in the Oracle server.
timeStampFormat	This is the timestamp format for the date/time columns that are used in the CSV exported data. Timestamps here follow the Oracle notation. It is possible to customize the date/time format per entity for the date/time type of columns in the source data. One possible timestamp format is: "DD-MON-YYYY HH12.MI.SS PM". For example, "01-JAN-2014 05.40.12 PM".
commitSize	This is the frequency of commits when performing creations.
entities	This is the list of entities to load. Each entity to load is a configuration object containing the following parameters: <ul style="list-style-type: none">• type - This parameter defines the type of entity to load. It can be any one of the following types: PARTY, PARTY_CHARACTERISTIC, ITEM, ITEM_

Parameter	Description
	<p>CHARACTERISTIC, ITEM_RELATOINSHIP, ITEM_ORDER, ITEM_ORDER_COMMENT</p> <ul style="list-style-type: none"> • <code>dataSourceName</code> - This is the name of the CSV file relative to the inside of the <code>PSI_BULK_DIR</code> directory. • <code>dataSourceColumnList</code> - This parameter is for the names of the columns in the CSV file. The possible values depend on the type of entity to load. Some columns are mandatory and some are optional. For details, see Mandatory and Optional Columns.

5. Configure the path to the bulk load JSON config file in the `ConfigValues_PSIService.json` file. The value to configure is `bulk.configuration`.

CSV File and JSON Config File Samples for Oracle Database

The following are samples of the CSV files for items and item characteristics and the bulk load JSON configuration file:

items.csv File Sample

```
ID_0,ItemType,Active,owner,creator,01-JAN-2014 05.40.12 PM,0,0, PARTY_
ID_0, PRODUCT_ID_0,01-JAN-2014 05.40.12 PM
ID_1,ItemType,Active,owner,creator,01-JAN-2014 05.40.12 PM,1,0, PARTY_
ID_1, PRODUCT_ID_1,01-JAN-2014 05.40.12 PM
...
```

items_characs.csv File Sample

```
Charac_ID_0_0,ID_0,Name1,Value1
Charac_ID_0_1,ID_0,Name2,Value2
...
```

JSON Config File Sample

```
{
  {
    "oracleDirectoryName" : "PSI_BULK_DIR",

    "timeStampFormat" : "DD-MON-YYYY HH12.MI.SS PM",

    "commitSize" : 100,

    "entities" : [ {

      "type" : "PARTY",

      "dataSourceName" : "party.csv",

      "dataSourceColumnList" : [ "ID", "PARTY_REF", "NAME", "PARTY_TYPE", "SUB_TYPE",
        "STATUS", "PARENT_PARTY_ID", "OWNED_BY", "CREATED_BY", "CREATED_ON", "UPDATED_BY",
        "UPDATED_ON", "VERSION", "DELETED" ]

    }, {

      "type" : "PARTY_CHARACTERISTIC",

      "dataSourceName" : "party_characs.csv",

      "dataSourceColumnList" : [ "ID", "PARTY_ID", "CHARACTERISTIC_NAME",
        "CHARACTERISTIC_VALUE" ]

    }, {

      "type" : "ITEM",
```

```
"dataSourceName" : "items.csv",
```

```
    "dataSourceColumnList" : [ "ID", "ITEM_REF", "PRODUCT_ID", "PRODUCT_VERSION",
    "NAME", "ITEM_TYPE", "SUB_TYPE", "STATUS", "PARTY_ID", "START_DATE", "END_DATE", "OWNED_
    BY", "CREATED_BY", "CREATED_ON", "UPDATED_BY", "UPDATED_ON", "VERSION", "DELETED" ]
```

```
  }, {
```

```
    "type" : "ITEM_CHARACTERISTIC",
```

```
      "dataSourceName" : "item_characs.csv",
```

```
      "dataSourceColumnList" : [ "ID", "ITEM_ID", "CHARACTERISTIC_NAME",
    "CHARACTERISTIC_VALUE" ]
```

```
    }, {
```

```
      "type" : "ITEM_RELATIONSHIP",
```

```
        "dataSourceName" : "item_relationships.csv",
```

```
        "dataSourceColumnList" : [ "ID", "ITEM_ID", "FORWARD_TYPE", "REVERSE_TYPE",
    "CHILD_ITEM_ID" ]
```

```
      }, {
```

```
        "type" : "ITEM_ORDER",
```

```
          "dataSourceName" : "item_orders.csv",
```

```
          "dataSourceColumnList" : [ "ITEM_ID", "ID", "ORDER_REF", "ORDER_DATE", "LINE_
    NUMBER", "LINE_ACTION", "LINE_ACTION_MODE", "PLAN_ITEM_ID", "PLAN_ITEM_ACTION" ]
```

```
        }, {
```

```
          "type" : "ITEM_ORDER_COMMENTS",
```

```

        "dataSourceName" : "item_order_comments.csv",

        "dataSourceColumnList" : [ "ORDER_ID", "COMMENT_DETAIL" ]

    } ]

}

```

Mandatory and Optional Columns for Oracle Database

The following table indicates which columns are mandatory or optional:

Entity Type	Mandatory Columns	Optional Columns
PARTY	ID(VARCHAR2) PARTY_TYPE(VARCHAR2) STATUS(VARCHAR2) OWNED_BY(VARCHAR2) CREATED_BY(VARCHAR2) CREATED_ON(TIMESTAMP(6)) VERSION(NUMBER), DELETED(NUMBER) (0 for not deleted, 1 for deleted)	PARTY_REF (VARCHAR2) NAME(VARCHAR2) SUB_TYPE (VARCHAR2) PARENT_PARTY_ID (VARCHAR2) UPDATED_BY (VARCHAR2) UPDATED_ON (TIMESTAMP(6))
PARTY_CHARACTERISTIC	ID(VARCHAR2) PARTY_ID(VARCHAR2) (the party to relate to) CHARACTERISTIC_NAME(VARCHAR2)	N/A

Entity Type	Mandatory Columns	Optional Columns
	CHARACTERISTIC_VALUE(VARCHAR2)	
ITEM	ID(VARCHAR2) PRODUCT_ID(VARCHAR2) ITEM_TYPE(VARCHAR2) STATUS(VARCHAR2) PARTY_ID(VARCHAR2) START_DATE(TIMESTAMP(6)) OWNED_BY(VARCHAR2) CREATED_BY(VARCHAR2) CREATED_ON(TIMESTAMP(6)) VERSION(NUMBER) DELETED(NUMBER) (0 for not deleted, 1 for deleted)	ITEM_REF (VARCHAR2) PRODUCT_VERSION (VARCHAR2) NAME(VARCHAR2) SUB_TYPE (VARCHAR2) END_DATE (TIMESTAMP(6)) UPDATED_BY (VARCHAR2) UPDATED_ON (TIMESTAMP(6))
ITEM_CHARACTERISTIC	ID(VARCHAR2) ITEM_ID(VARCHAR2) (the item to relate to) CHARACTERISTIC_NAME(VARCHAR2) CHARACTERISTIC_VALUE(VARCHAR2)	N/A
ITEM_RELATIONSHIP	ID(VARCHAR2) ITEM_ID(VARCHAR2) (the item to relate to) FORWARD_TYPE(VARCHAR2) REVERSE_TYPE(VARCHAR2) CHILD_ITEM_ID(NUMBER) (the child item to relate to)	N/A
ITEM_ORDER	ITEM_ID(VARCHAR2) (the item to relate to)	LINE_ACTION_MODE

Entity Type	Mandatory Columns	Optional Columns
	ID(VARCHAR2)	(VARCHAR2)
	ORDER_REF(VARCHAR2)	PLAN_ITEM_ID (VARCHAR2)
	ORDER_DATE(TIMESTAMP(6))	
	LINE_NUMBER(VARCHAR2)	PLAN_ITEM_ACTION (VARCHAR2)
	LINE_ACTION(VARCHAR2)	
ITEM_ORDER_	ORDER_ID	N/A
COMMENTS	COMMENT_DETAIL	

Bulk Load for PostgreSQL Database

This section explains the bulk loading for PostgreSQL Database.

Configuring for the Bulk Load for PostgreSQL Database

The bulk load service provides a way to mass provision the inventory database with existing parties and items. A configuration file has been provided which needs to be configured with appropriate values before the bulk load is initiated.

Before you begin

The data to load must be converted as CSV files if they are not already in that format. The PostgreSQL database is only able to read CSV files and treats these as a table. Inventory then applies the stored procedures to create internal items, item characteristics, item relationships, orders, parties, characteristics, relationships, and orders. For details, see [CSV File and JSON Config File Samples](#) for examples of item.csv and items_characs.csv files.

Procedure

1. If you have not already, create CSV files containing items, item characteristics, orders,

item relationships, parties, and party characteristics, and put these files in a directory on the TIBCO Product and Service Inventory server machine.

2. Create a bulk load JSON configuration file (bulkConfig.json) on the TIBCO Product and Service Inventory server machine. For details, see [CSV File and JSON Config File Samples](#) for an example of how the JSON configuration file must be constructed. This file can be saved in the \$PSI_HOME/config/ file.
3. Configure the following parameters in the bulk load JSON configuration file:

Parameter	Description
oracleDirectoryName	This is the absolute directory path where the csv files containing the data are to be loaded.
timeStampFormat	This is the timestamp format for the date/time columns that are used in the CSV exported data. Timestamps here follow the PostgreSQL notation. It is possible to customize the date/time format per entity for the date/time type of columns in the source data. One possible timestamp format is: "DD-MON-YYYY HH12.MI.SS PM". An example value of this format is: "01-JAN-2014 05.40.12 PM".
commitSize	Unlike the Oracle database, the bulk load does not use this parameter. Keep the default.
entities	<p>This is the list of entities to load. Each entity to load is a configuration object containing the following parameters:</p> <ul style="list-style-type: none">• type - This parameter defines the type of entity to load. It can be any one of the following types: PARTY, PARTY_CHARACTERISTIC, ITEM, ITEM_CHARACTERISTIC, ITEM_RELATOINSHIP, ITEM_ORDER, ITEM_ORDER_COMMENT• dataSourceName - This is the name of the CSV file.• dataSourceColumnList - This parameter is for the names of the columns in the CSV file. The possible values depend on the type of entity to load. Some

Parameter	Description
	columns are mandatory and some are optional. For details, see Mandatory and Optional Columns .

4. Configure the path to the bulk load JSON config file in the ConfigValues_PSIService.json file. The value to configure is bulk.configuration.

CSV File and JSON Config File Samples for PostgreSQL Database

For details, see [CSV File and JSON Config File Samples for Oracle Database](#) for more information on the CSV and JSON config file samples of PostgreSQL Database.

Mandatory and Optional Columns for PostgreSQL Database

For details, see [Mandatory and Optional Columns for Oracle Database](#) for more information on mandatory and optional columns of the PostgreSQL database.

Triggering the Bulk Load

You must use the TIBCO Product and Service Inventory UI to trigger the bulk load. Logs are available on the UI and the Oracle server during and after the load.

Before you begin

Users must have administrator privileges to trigger the bulk load.

Procedure

1. Log in to TIBCO Product and Service Inventory.
2. Click **Bulk**.

You are directed to the **Bulk Loads** page. From this page, you can view previous bulk load logs.

3. Provide a bulk load ID and click **Load**.

Bulk Load Logging for Oracle Database

There are four types of log files created for each data input file. The bulk load process logs the various stages of execution. A unique ID is generated for each load.

The log files for the bulk load are created in the same directory where the bulk load files are placed (the directory created in step 2 in [Configuring for the Bulk Load for Oracle Database](#)).

The following log files are generated for the particular load process:

Log File

This file is used by Oracle to log information about the process used to create the external table. The log file name has the following format: Log_<EntityName>_<LoadId>.log (for example, Log_Item_001.log) This file contains the logs generated by the ACCESS driver and is created by the Oracle server. The ACCESS driver is used to perform data loads, and the data comes from text data files.

Bad File

This file contains records that cannot be loaded because of errors. For example, a record was written to the bad file because a field in the datafile could not be converted to the datatype of a column in the external table. The bad file name has the following format: Bad_<EntityName>_<LoadId>.bad (for example, Bad_Item_001.bad). This file is created only if any bad records are encountered. This file contains bad records as detected by the ACCESS driver as well as the insert process and is created by the Oracle server.

Discard File

This file contains records that fail the condition in the LOAD WHEN clause of the statement used to create the external table. This file is created only if any records that fail the LOAD WHEN condition. The discard file name has the following format: Discard_<EntityName>_<LoadId>.discard (For example, Bad_Item_001.bad).

Stored Procedure logs

This file is created by the procedures which copy the records from the external table to the actual table. This file displays errors encountered during the copying process. It also displays the current status of the load. The file name has the following format: Log_SP_<EntityName>_<LoadId>.log. The log file contains the following type of content:

```
05-Jan-2016 17:09:50.430000000 - Load has started with commit
frequency 10
05-Jan-2016 17:10:05.510000000 - Committed Inventory items after
processing 2000(1-2000) records
05-Jan-2016 17:10:11.698000000 - Committed Inventory items after
processing 2000(2001-4000) records
05-Jan-2016 17:10:12.712000000 - Committed Inventory items after
processing 2000(4001-6000) records
05-Jan-2016 17:10:38.226000000 - Committed Inventory items after
processing 2000(6001-8000) records
05-Jan-2016 17:10:39.492000000 - Committed Inventory items after
processing 2000(8001-10000) records
...
...
... (truncated for brevity)
...
...
05-Jan-2016 17:13:19.685000000 - Committed Inventory items after
processing 1156(102001-103156) records
05-Jan-2016 17:13:19.686000000 - Completed load of Inventory items
after loading 103156 records out of 103156 records.
05-Jan-2016 17:13:19.688000000 - Load has ended
```

Bulk Load Logging for PostgreSQL Database

There is one log file created for each data input file. The bulk load process logs the various stages of execution. A unique ID is generated for each load.

The log files for the bulk load are created in the data directory of PostgreSQL <PSI_HOME>/bin/pgsql/data/directory.

The following log files are generated for the particular load process:

Stored Procedure Log File

This file is created by the procedures which copy the records from the external table to the actual table. This file displays the errors encountered during the copying process. It also displays the current status of the load. The file name has the following format: Log-SP_<EntityName>_<LoadId>.log. For example,
25-Sept-201815:34:51 - Load has started 25-Sept-201815:34:51 - Completed load of inventory items after loading 1 record out of 1 record 25-Sept-201815:34:51 - Load has ended

TIBCO Documentation and Support Services

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

How to Access TIBCO Documentation

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

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

Product-Specific Documentation

The following documentation for this product is available on the [TIBCO® Product and Service Inventory](#) documentation page:

- *TIBCO® Product and Service Inventory Release Notes*
- *TIBCO® Product and Service Inventory Installation and Configuration*
- *TIBCO® Product and Service Inventory User Guide*
- *TIBCO® Product and Service Inventory REST Services Guide*
- *TIBCO® Product and Service Inventory Web Services Guide*

How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the [TIBCO Support](#) website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to [TIBCO Support](#) website. If you do not have a user name, you can request one by clicking **Register** on

the website.

How to Join TIBCO Community

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

Legal and Third-Party Notices

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

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

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

TIBCO, the TIBCO logo, and the TIBCO O logo 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 and/or its affiliates.

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

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

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

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

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

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. 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 © 2015-2022. TIBCO Software Inc. All Rights Reserved.