# TIBCO® Fulfillment Subscriber Inventory User's Guide

*Software Release 1.0*
*August 2016*

Two-Second Advantage®

## Important Information

# Contents

# Figures

# TIBCO Documentation and Support Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, please visit:

https://docs.tibco.com

### Product-Specific Documentation

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

- *TIBCO Fulfillment Subscriber Inventory Installation and Configuration Guide*
- *TIBCO Fulfillment Subscriber Inventory User's Guide*
- *TIBCO Fulfillment Subscriber Inventory Web Services Guide*
- *TIBCO Fulfillment Subscriber Inventory REST Services Guide*
- *TIBCO Fulfillment Subscriber Inventory Release Notes*

### How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

  http://www.tibco.com/services/support

- If you already have a valid maintenance or support contract, visit this site:

  https://support.tibco.com

  Entry to this site requires a user name and password. If you do not have a user name, you can request one.

### How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

https://www.tibcommunity.com

# Fulfillment Subscriber Inventory Overview

TIBCO Fulfillment Subscriber Inventory holds specific information about subscribers and the services or products that the subscribers have purchased, about to purchase, or being provisioned.

The function of the TIBCO Fulfillment Subscriber Inventory is to maintain a current image of customer installed products at any given point in time, which is capable of supporting fast concurrent read or write access while ensuring data consistency. TIBCO Fulfillment Subscriber Inventory also provides a rich, user-friendly web interface that can be used to explore and modify the contents of the system in a safe and secure manner.

# Uses of Fulfillment Subscriber Inventory

You can use Fulfillment Subscriber Inventory to:

- Store data (characteristics) related to the Parties
- Create hierarchical party support for coping with complex organization, such as an enterprise comprising of many groups or employees
- Create complex structure of item support to allow storage of complex relationships defined in Fulfillment Catalog (ProductComprisedOf or ProductDependsOn relationships stored with products)
- Store data (characteristics) related to the Items
- Store related orders for an item, so that you can walk through the item history
- Create relationship between party and item, or, if no party is used, adding a party name to an item for retrieval of subscriber products
- Implement CRUD operations for data management on both party and item
- Search both party and item
- Lock a resource to be protected against other concurrent updates
- Create a batch that allows multiple updates to be grouped together, either committed together, or rolled-back entirely
- Implement access management for user roles and users, so that the access can be restricted
- Implement multi-tenancy for cloud based deployment and operation for multiple customers

# Data Managed by Inventory

The data managed by Fulfillment Subscriber Inventory are as follows:

| | |
|---|---|
| **Party** | Stores customer, subscriber, or other party-related information. See the Parties topic for more details. |
| **Item** | Stores product, service, or other tangible or intangible entities that have been ordered by a party, and are present as components of that party's image. See the Items topic for more details. |

# Object Locking

Party and item entities provide the capability for locking and lock management. This is useful in a context where multiple concurrent processes are all working on the same object but it is necessary to control the order of access to the object in a defined sequence.

It is important to note that the Fulfillment Subscriber Inventory locking functionality does not provide any logic or rule-based checking of operation invoke against an object. It will only register a list of locks of interest, and publish the lock notifications in a controlled sequence. If clients do not respect the object locks, there are no hard rules within the Fulfillment Subscriber Inventory system to prevent the clients from accessing or updating the object out of sequence. Failure to respect the locking functionality may violate business requirements and result in an inconsistent state within the Inventory system.

## Lock Requests

The Lock requests will provide support to the following criteria:

- A key identifying the lock.
- A timeout in seconds:
  - "0", to request immediate lock, which means the lock request is either successful and grants the lock right away, or returns an error.
  - Non-zero, to request a lock that can be delayed. If the lock cannot be granted right away, it is delayed by the entity. If the entity is released before getting timed out, the lock will be granted to the requester. Otherwise, the lock request will get timed out and the request will be discarded.
- A priority (higher the value, higher the priority). When a lock is released and there are delayed lock requests, the system will grant the lock to the request with the higher priority. If the priority value is the same for many requests then the priority placed earlier will get the lock.

## Lock Notifications

The granting of a lock or the expiry of a lock is asynchronous. It is necessary for client applications to have channels to receive lock notifications. Fulfillment Subscriber Inventory uses JMS topics to receive lock notifications. There are three topics per channel. They are:

| Topics | Description |
|---|---|
| com.tibco.inventory.notification.lock.obtained.topic.[tenantId] | Used for messages, which indicate that a lock was granted. |
| com.tibco.inventory.notification.lock.removed.topic.[tenantId] | Used for messages which indicate that a lock request was removed by user or an API interaction.<br>• The message is not sent when the current lock is "unlocked" on an item or party. It is only meant for removal.<br>• New lock request to acquire a lock on the item or the party will get a lock obtained notification. |

| Topics | Description |
|---|---|
| com.tibco.inventory.notification.lock.expired.topic.[tenantId] | Used for messages, which indicate that a lock request expired. |

## Lock Modes

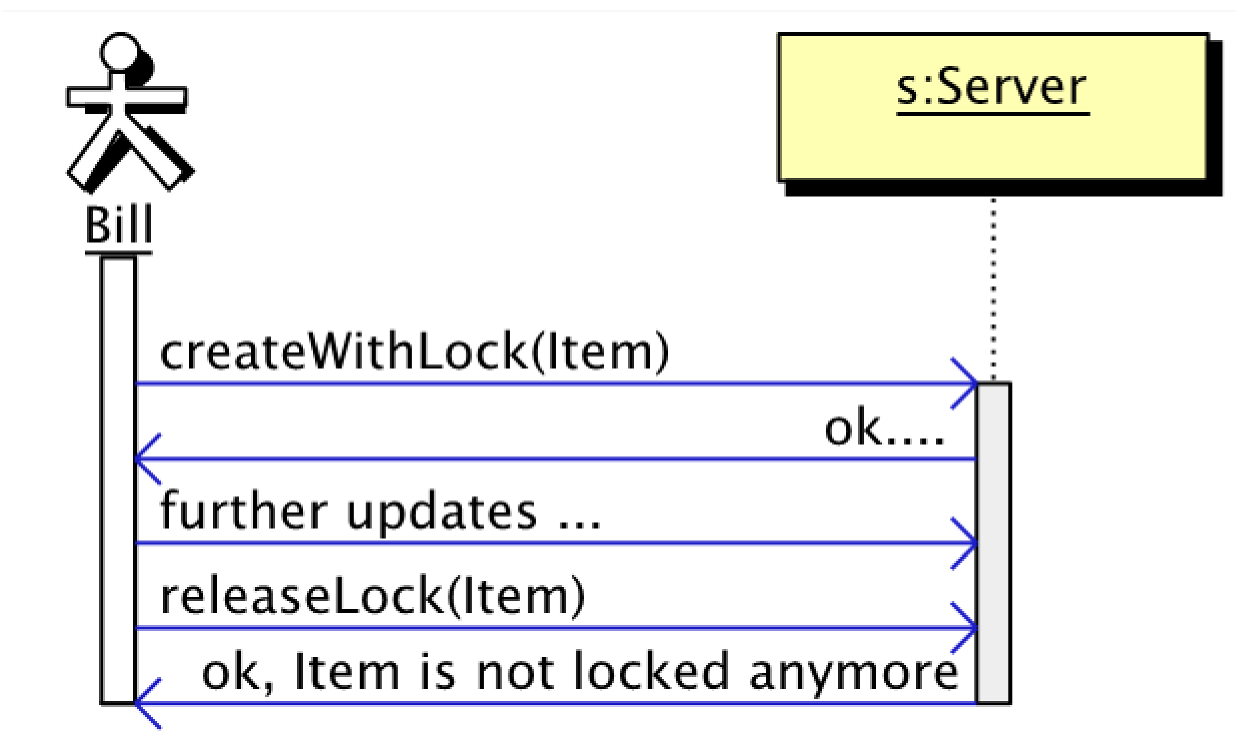
Fulfillment Subscriber Inventory supports two modes of locking. You can configure the modes of locking as per your requirements.

| | |
|---|---|
| Locks are Advisory | You can always perform an update on an item or a party. You have to check if you need to perform an update or not. |
| Locks are Hard | You cannot perform an update if the entity is locked and you were not the one to implement the lock. The default lock mode for all users is hard lock, except the user whose role is admin. The admin user role, by default, has the `PERM_UPDATE_WHEN_NOT_LOCK_OWNER` permission. |

## Create With Lock

You can use SOAP API to create an entity with immediate Lock in place.

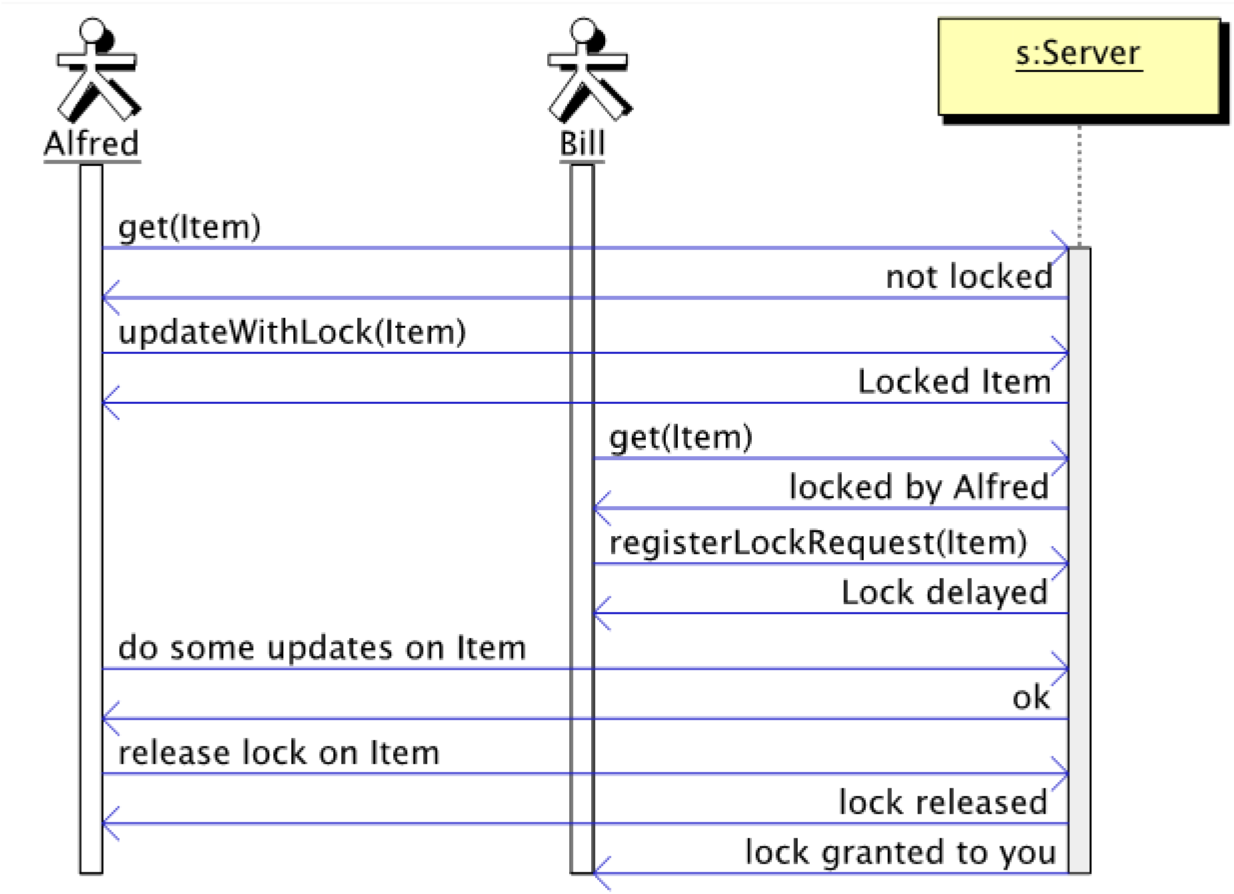
*Create With Lock*



## Prevent Concurrent Update

The following diagram illustrates the concurrent aspect of locks:

*Prevent Concurrent Update*



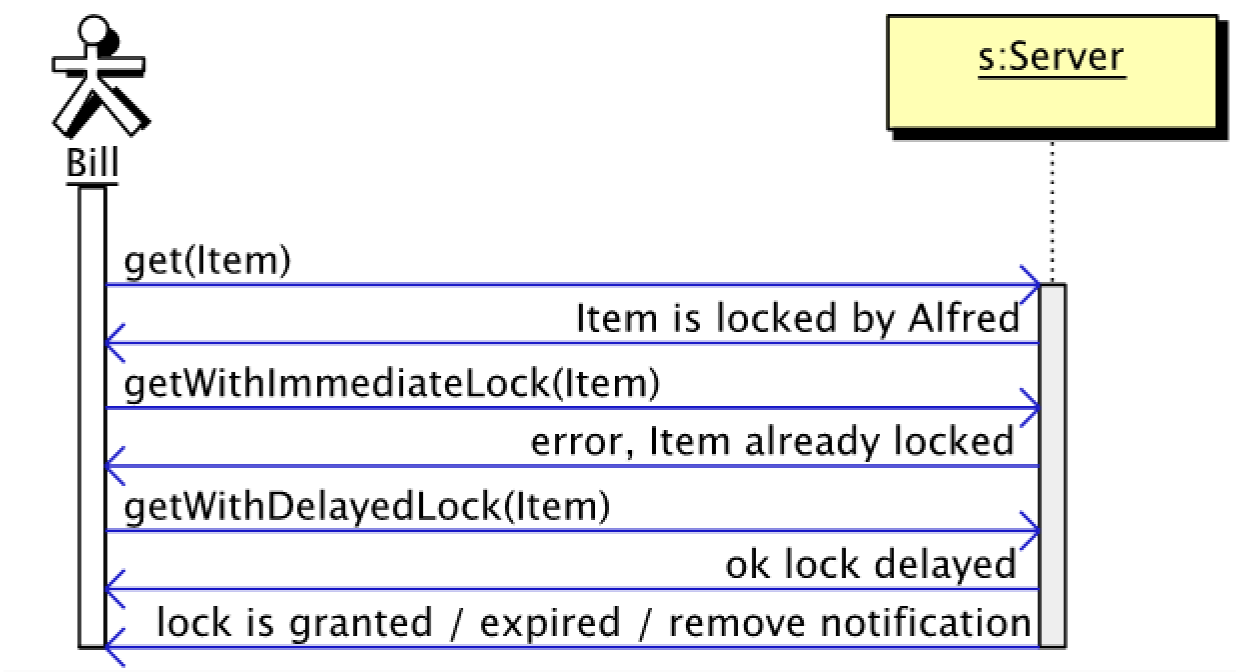

Whether locks are hard or advisory, you should always read an Item before trying to lock the item.

If you request a lock with timeout and the lock gets delayed, the lock will be granted, and a notification will be published on the JMS topic to indicate the granting of the lock.

## Get with Lock Delayed

The following diagram provides additional details related to the immediate or delayed locking mechanism:

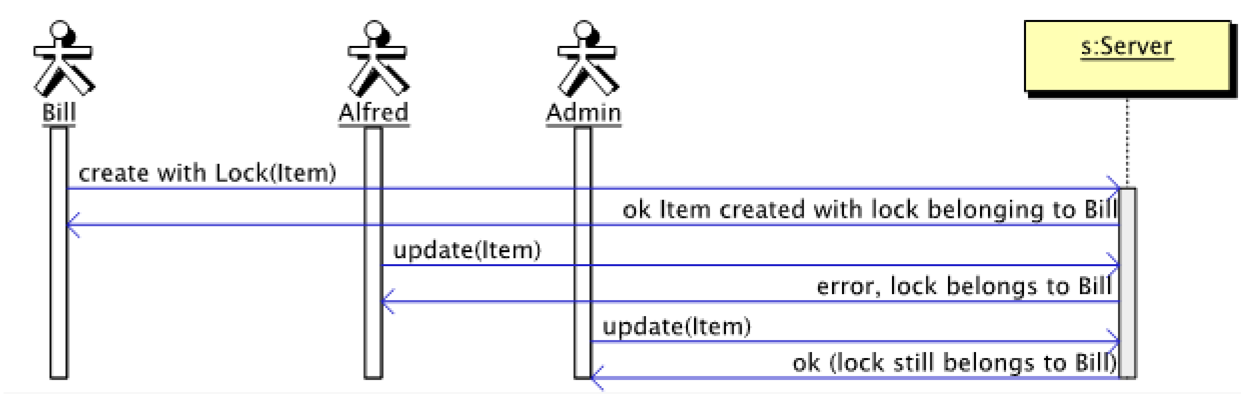
*Get with Lock Delayed*




## Admin Update When Not Lock Owner Permission

If you are a user with administrator privileges you need not own a lock to update the entity.

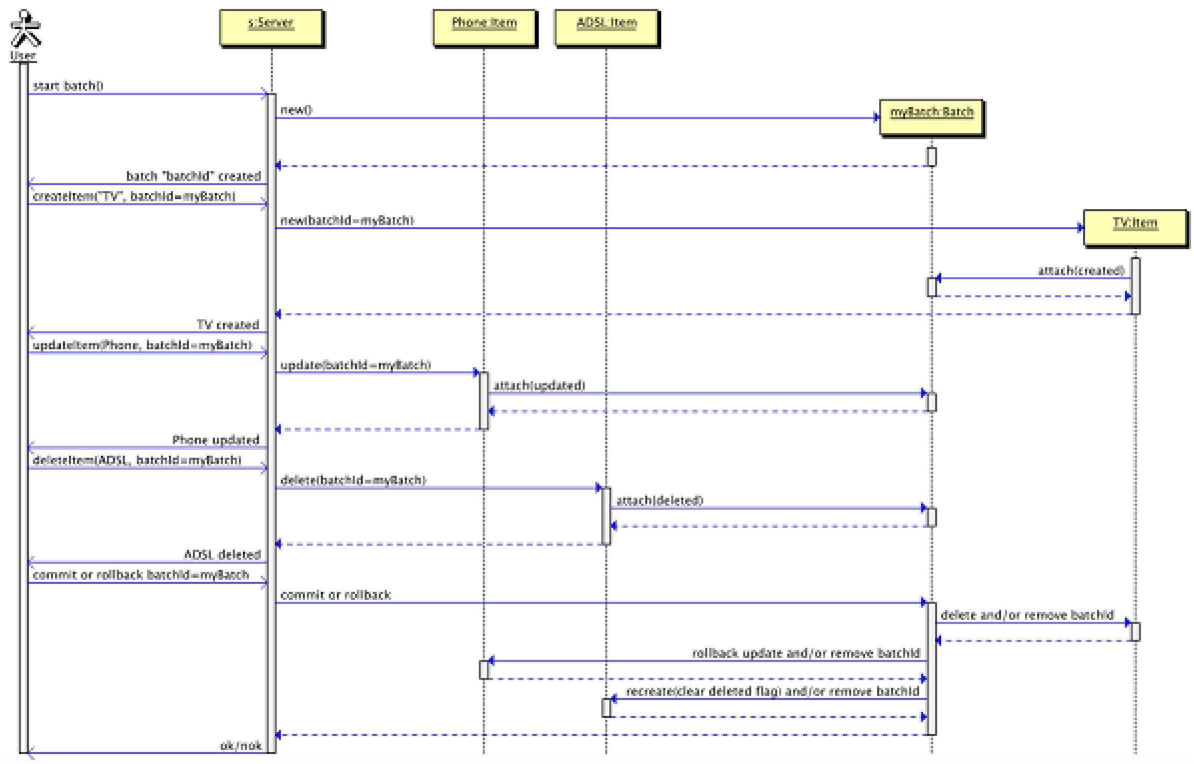
*Admin Update When Not Lock Owner Permission*



# Batch Functionality

The Batch functionality on server side is the ability to group the commands under a common batchId. The group of commands can be either committed as a whole or rolled back as a whole.

*Batch Functionality*




The feature works as follows on the SOAP interface:

- Call the StartBatch() to get a new batch with a batchId.
- Modify (update, delete, or create) entities, sending the batch id for each request. This will attach the entities to the batch.
- Either call completeBatch() with a commit or rollback parameter.
  - Commit will remove the batchId marker on entities.
  - Rollback will remove the batchId marker on entities and rollback changes on the entities (delete ones created during the batch, recreate ones deleted during the batch, or just get data back to the values they had initially).
  - Both will delete the batch.

An entity that is attached to a batch cannot be modified outside of it. Trying to update, or delete an entity outside of the batch will result in an error.

# Configuration Values

The configuration values are located in the <FSI_HOME>/config/ConfigValues_FSI.xml file.

The configuration values are as follows:

| Property Name | Description | Default Value | Comments |
|---|---|---|---|
| com.tibco.fos.fsi.pooledDataSource.driverClassName | Pooled Data Source Driver Class Name | oracle.jdbc.driver.OracleDriver | |
| com.tibco.fos.fsi.pooledDataSource.host | Pooled Data Source Host | localhost | |
| com.tibco.fos.fsi.pooledDataSource.port | Pooled Data Source Port | 1521 | |
| com.tibco.fos.fsi.pooledDataSource.database | Pooled Data Source Database | orcl | |
| com.tibco.fos.fsi.pooledDataSource.username | Pooled Data Source Username | FSI_APP_MAC | |
| com.tibco.fos.fsi.hibernate.default_catalog | Hibernate Default Catalog | FSI_APP_MAC | Should be the same as the pooled data source Username. |
| com.tibco.fos.fsi.pooledDataSource.password | Pooled Data Source Password | FSI_APP_MAC | |
| com.tibco.fos.fsi.pooledDataSource.url | Pooled Data Source URL | jdbc:oracle:thin:@//<br>${com.tibco.fos.inv.pooledDataSource.host}:<br>${com.tibco.fos.inv.pooledDataSource.port}/<br>${com.tibco.fos.inv.pooledDataSource.database} | |
| com.tibco.fos.fsi.pooledDataSource.initializeSize | Pooled Data Source Initialize Size | 2 | |
| com.tibco.fos.fsi.pooledDataSource.maxIdle | Pooled Data Source Max Idle | 11 | |

| Property Name | Description | Default Value | Comments |
| --- | --- | --- | --- |
| com.tibco.fos.fsi.pooledDataSource.maxActive | Pooled Data Source Max Active | 12 | |
| com.tibco.fos.fsi.pooledDataSource.maxWait | Pooled Data Source Max Wait | 10000 | |
| com.tibco.fos.fsi.pooledDataSource.validationQuery | Pooled Data Source Validation Query | select 1 from dual | |
| com.tibco.fos.fsi.pooledDataSource.testOnBorrow | Pooled Data Source Test OnBorrow | FALSE | |
| com.tibco.fos.fsi.pooledDataSource.testWhileIdle | Pooled Data Source Test WhileIdle | TRUE | |
| com.tibco.fos.fsi.pooledDataSource.timeBetweenEvictionRunsMillis | Pooled Data Source Eviction Interval | 5000 | |
| com.tibco.fos.fsi.pooledDataSource.minEvictableIdleTimeMillis | Pooled Data Source Minimum Evictable Idle Time | 5000 | |
| com.tibco.fos.fsi.pooledDataSource.numTestsPerEvictionRun | Pooled Data Source Tests Per Eviction Run | 5 | |
| com.tibco.fos.fsi.hibernate.dialect | Hibernate dialect | org.hibernate.dialect.Oracle10gDialect | |
| com.tibco.fos.fsi.hibernate.cache.use_second_level_cache | Hibernate Second Level Cache Usage | FALSE | |
| com.tibco.fos.fsi.hibernate.cache.provider_class | Hibernate Cache Provider Class | org.hibernate.cache.NoCacheProvider | |
| com.tibco.fos.fsi.hibernate.jdbc.batch_size | Hibernate JDBC Batch size | 30 | |
| com.tibco.fos.fsi.hibernate.show_sql | Hibernate Show SQL | FALSE | |
| com.tibco.fos.fsi.jms.jndi.url | JNDI URL for JMS Service | tibjmsnaming://localhost:7222 | Used for lock notification or JMS SOAP API. |

| Property Name | Description | Default Value | Comments |
|---|---|---|---|
| com.tibco.fos.fsi.jms.jndi.security.principal | JNDI Username | admin | |
| com.tibco.fos.fsi.jms.jndi.security.credentials | JNDI Password | admin | |
| com.tibco.fos.fsi.soapService.jms.targetDestination | JMS Queue to process incoming SOAP API requests | com.tibco.inventory.soap.api.jms.queue | |
| com.tibco.fos.fsi.soapService.jms.concurrentConsumers | Initial count of consuners that process the incoming SOAP request | 5 | |
| com.tibco.fos.fsi.soapService.jms.maxConcurrentConsumers | Max count of consuners that process the incoming SOAP request | 10 | |
| com.tibco.fos.fsi.expiry.check.delay | Number of milliseconds between 2 expiry checks | 60000 | |
| com.tibco.fos.fsi.bulk.configuration | Bulk load configuration file path (on the inventory server machine) | /tmp/bulkconfig.json | |
| com.tibco.fos.fsi.jwt.rest.token.validity | Number of seconds for which JWT token is valid for REST API | 3600 | |
| com.tibco.fos.fsi.jwt.soap.token.validity | Number of seconds for which JWT token is valid for SOAP API | 3600 | |

# Oracle RAC Configuration

Connecting to oracle RAC (Real Application Cluster) using JDBC with thin driver, the classic url: `jdbc:oracle:thin:@<HOST>:1521:<SID>` will not work.

The correct URL is as follows:

```
jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=on)
(ADDRESS=(PROTOCOL=TCP)(HOST=host1) (PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=host2) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=service)))
```

The mentioned URL can be obtained from the `tnsnames.ora` file in the Oracle installation

# Fulfillment Subscriber Inventory User Interface

You can access Fulfillment Subscriber Inventory by logging in to the application using the provided credentials.

The homepage or the main page of the application provides access to the entire Fulfillment Subscriber Inventory application.

## Basic Access Controls of Fulfillment Subscriber Inventory

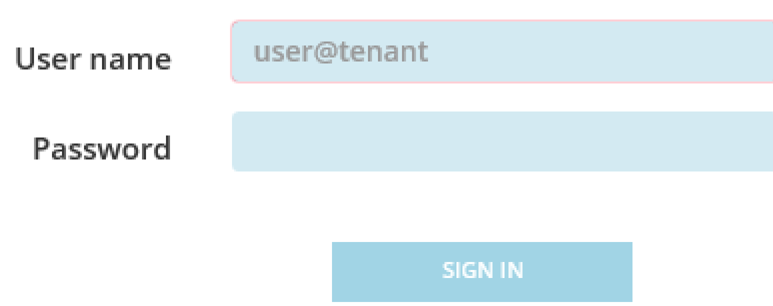
The basic access controls of Fulfillment Subscriber Inventory includes:

| | |
|---|---|
| Log In | To access Fulfillment Subscriber Inventory system using the provided credentials. |
| Main Page | Provides access to all the features provided by the Fulfillment Subscriber Inventory system. |
| Search Bar | Search Items and Parties by their name. |
| Log Out | To exit from the Fulfillment Subscriber Inventory system. |

### Logging in to Fulfillment Subscriber Inventory

Use the provided credentials to access the Fulfillment Subscriber Inventory application. To log in to the Fulfillment Subscriber Inventory application, perform the following steps:

***Fulfillment Subscriber Inventory Login Page***




**Procedure**

1. Enter the **User name**.

   The username format is `username@tenant_name`.

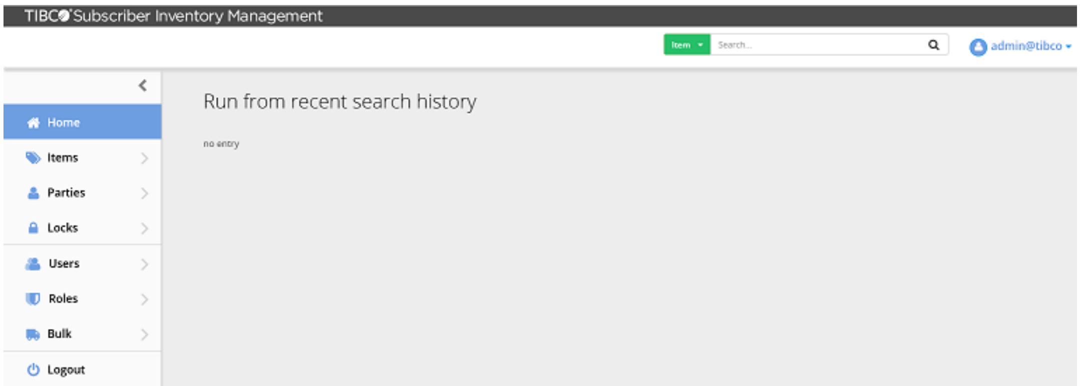
2. Enter the **Password**.

3. Click the **Sign In** button.
   Upon successful verification of credentials, a session will be established and the control will be passed to the Fulfillment Subscriber Inventory main page.

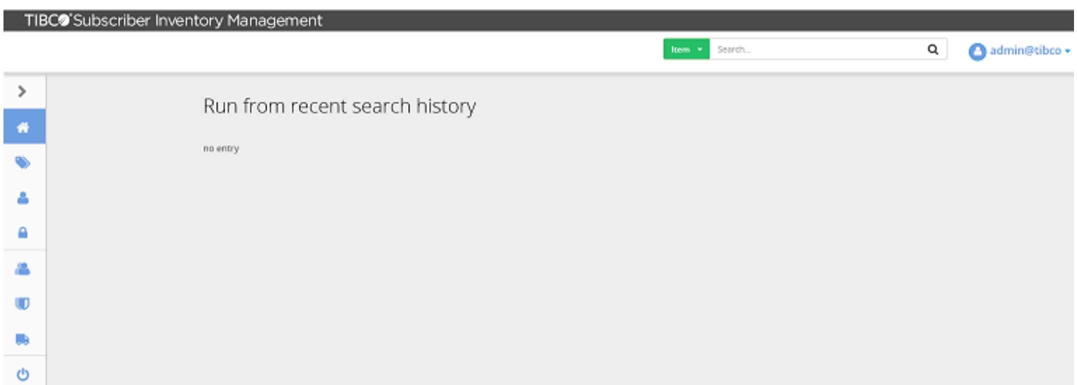
## Fulfillment Subscriber Inventory Main Page

Fulfillment Subscriber Inventory main page provides access to all the features of the application. The collapsible menu on the left side provides access to:

*Fulfillment Subscriber Inventory Main Page*




*Fulfillment Subscriber Inventory Main Page with Menu Names Collapsed*




| | |
|---|---|
| **Home** | To pass the control back to the main page |
| **Items** | To view and add items |
| **Parties** | To view and add parties |
| **Locks** | To view the locked items and locked parties |
| **Users** | To create users who can use the Fulfillment Subscriber Inventory application |
| **Roles** | To create roles that can be assigned to users |
| **Bulk** | To perform Bulk Load operation |
| **Logout** | To log out from the Fulfillment Subscriber Inventory application |

The access to menus is dependent on the role of the user. Access to all the menus are only available to the administrator. To know about roles and permissions, see List of Permissions for a Role topic for more details.

The top bar of the Fulfillment Subscriber Inventory application provides access to:

| | |
|---|---|
| Search | To search a particular item or party |
| Profile Name | Displays the profile that is logged in. |

## Searching Items and Parties

To search for an item and party, perform the following steps:

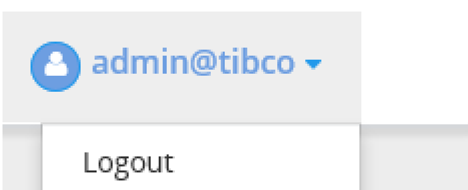
### Procedure

1. Select the value **Item Name** or **Party Name** from the dropdown box.
2. Enter the name of the:
    - item, if the selection is **Item Name**.
    - party if the selection is **Party Name**.
3. Click the search icon or press the `Enter` key.
   The results will be displayed on the page.

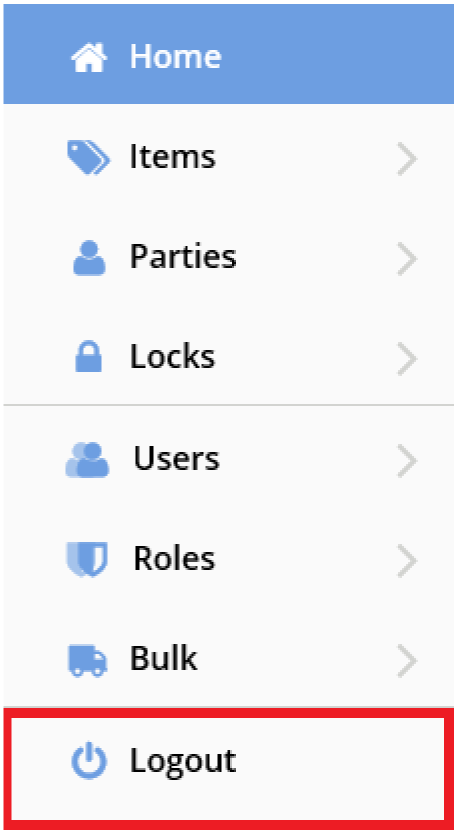
## Logging out of Fulfillment Subscriber Inventory

Logging out of Fulfillment Subscriber Inventory ends the session created to access the application. To log out of Inventory, perform the following steps:

***Fulfillment Subscriber Inventory Logout from Top Bar***



*Fulfillment Subscriber Inventory Logout from Menu*




**Procedure**

1. Click the profile name
2. Click **Logout**.

## Items

An Item represents any tangible or intangible product or service that has business context within any system that makes use of the Fulfillment Subscriber Inventory in the enterprise. This includes both OSS and BSS layers, as well as systems outside these domains. An item is identified with a unique identifier called an itemId. This identifier must be globally unique within the item domain and can be specified externally or generated by the Fulfillment Subscriber Inventory system as required.

Typically an item will represent any one of the following types of entities:

- Customer-Facing Service (CFS)
- Resource-Facing Service (RFS)
- Bundle
- Product
- Device
- Service
- Tariff

The mentioned list is not an exhaustive list of item types. Within the Fulfillment Subscriber Inventory system, an item is a generic object which has a type and subtype classification scheme. Users are free to define their own categories within this scheme, with the one limitation that each item must have a globally unique itemId.

Fulfillment Subscriber Inventory also provides the ability to manage a limited set of information that describes an item in the form of characteristics. These characteristics are name-value pairs which describe attributes of the item. Each characteristic is identified with a unique identifier called a characteristicId. This identifier must be unique within the scope of any given item instance and can be specified externally or generated by the Fulfillment Subscriber Inventory system as required.

The system also provides the ability to link an item to a list of orders that have interacted with the item. This could be considered an order audit log of changes to the item over its life cycle. Each order is identified with a unique identifier called an orderId. This identifier must be unique within the scope of any given item instance and must always be specified externally. This means multiple items can refer to the same order, but the same order may not appear on one item more than once.

Items may have relationships to other items, so the Fulfillment Subscriber Inventory system provides the capability of linking items together using relationships. These relationships do not enforce referential integrity and are keyed on a combination of the relationship type and the primary key of both the parent and child item entities, specifically the itemId. These values must be unique for any given item and must be specified externally.

There is also the capability of locking item objects within the Fulfillment Subscriber Inventory system. Each lock is identified with a unique identifier called a key. This identifier must be unique within the scope of any given item instance and must always be specified externally.

## Adding a New Item

To add a new item perform the following steps:

### Procedure

1. Click the **Items** menu.
2. Click the **+** button.
3. Provide the primary information, which includes:
    - **Item Type** – Some examples for the values are Enterprise, Customer, or Subscriber. This is a mandatory field.
    - **Item Name** – Enter a name for the item.
4. Provide the additional information, which includes:
    - **Item Status** – Some examples for the values are Active, Inactive, Suspended, or Pending Update. This is a mandatory field.
    - **Item Sub Type**
    - **Item ID** – The unique ID for the item. If a value is not provided Fulfillment Subscriber Inventory will generate the value.
    - **Item Ref**
    - **Product ID** – This is a mandatory field.
    - **Party ID** – This is a mandatory field.
    - **Product Version**
    - **Start Date** – This is a mandatory field.
    - **End Date**

5. Provide the characteristics information (this is optional), by clicking **+ Add** link in the Characteristics Info section. You can add multiple characteristics by clicking **+ Add** link. You can delete unwanted characteristics by clicking **x Remove** link. The characteristics information to be provided are as follows:

   - **Name** - This is a mandatory field.
   - **Value** - This is a mandatory field.
   - **ID**

6. Provide the orders information (this is optional), by clicking **+ Add** link in the Orders Info section. You can add multiple order information by clicking **+ Add** link. You can delete unwanted order information by clicking **x Remove** link. The order information to be provided are as follows:

   - **Order ID** – This is a mandatory field.
   - **Ref** – This is a mandatory field.
   - **Date** – This is a mandatory field.
   - **Line** – This is a mandatory field.
   - **Line action** – This is a mandatory field.
   - **Action mode**
   - **Plan item ID**
   - **Plan item action**
   - **Comments**

7. Provide the relationships information (this is optional), by clicking **+ Add** link in the Orders Info section. You can add multiple relationship information by clicking **+ Add** link. You can delete unwanted relationship information by clicking **x Remove** link. The relationship information to be provided are as follows:

   - **Child** – The target item to relate to. It has to exist in the system. This is a mandatory field.
   - **Forward type** – The name of the forward relationship `ProductComprisedOf`. This is a mandatory field.
   - **Reverse type** – The reverse name of the relationship `ProductContainedIn`. This is a mandatory field.

8. Click the **Submit** button.
   The item will be created and the control will be passed back to the Inventory Item page. The newly created Item will be displayed in the list of Items. To search for the newly created item see Searching an Item for more details.

## Searching an Item

You can search an Item using the common search bar. See Searching Items and Parties for more details. You can use the basic navigation buttons like Previous and Next to locate the desired item. You can also search for an item using the search feature. To search for an item from the Item menu perform the following steps:

### Procedure

1. Enter the search query with any or all of the parameters:

   - name
   - id

- type
- status

An example for a query format is as follows: `status=<Item Status>> AND name=<<Item Name>> AND type=<<Item Type>>`.

2. Click the **Search** button.
   The criteria satisfying the search parameters will be displayed.
3. Click **Save this Search?** Link to reuse the search query.
4. Enter a relevant name for the query and click the **Save** icon.
   The query will be saved for reuse.
5. Click **Home** menu.
   The saved query name will be displayed.
6. Click the query name.
   The query details will be displayed in the expanded view.
7. Click execute this Item query to execute the saved query again.
   The result or results will be displayed.

## Updating an Item

To update an existing Item, perform the following steps:

### Procedure

1. Click the **Items** menu.
2. Locate the desired item that needs to be updated using the **Previous** or **Next** button. You can also search the item using the Search feature. See Searching an Item for more details.
3. Click the item to be modified.
   The Update Item page opens.
4. Make the necessary modifications to the desired fields and click the **Update** button.
   The item will be updated and the control will be passed back to the Inventory Item page.

## Locking an Existing Item

Locks can only be added to an existing item. To add a lock to an existing item perform the following steps:

### Procedure

1. Click the **Items** menu.
2. Locate the desired item that needs to be locked using the **Previous** or **Next** button. You can also search the item using the Search feature. See Searching an Item for more details.
3. Click the item to be locked.
   The Update Item page opens.
4. Provide values to the following fields:
   - **Key** - Unique identifier of a lock on the item.
   - **Priority**
   - **Timeout** - The value entered should be in seconds.
5. Click the **Set** link.

6. Click the **Update** button.
   The item will be locked and the control will be passed back to the Inventory Item page.

### Deleting the Lock of an Item

To delete the lock of an item, perform the following steps:

**Procedure**

1. Click the **Items** menu.
2. Locate the desired item, whose lock needs to be deleted, using the **Previous** or **Next** button. You can also search the item using the Search feature. See Searching an Item for more details.
3. Click the item.
   The Update Item page opens.
4. Click the **Clear link** in the **List of locks** section.
5. Click the **Update** button.
   The lock will be deleted and the control is passed back to the Inventory Items page.

### Deleting an Existing Item

To delete an existing item, perform the following steps:

**Procedure**

1. Click the **Items** menu.
2. Locate the desired item that needs to be deleted using the **Previous** or **Next** button. You can also search the item using the **Search** feature. See Searching an Item for more details.
3. Click the item to be deleted.
   The Update Item page opens.
4. Click the **Delete** button.
   The item will be deleted and the control will be passed back to the Inventory Item page.

## Parties

Party represents any user that has business context within any system that makes use of the Inventory in the enterprise.

It includes both OSS and BSS layers, as well as systems outside the domains. A party is identified with a unique identifier called a partyId. This identifier must be globally unique within the party domain and can be specified externally or generated by the Inventory as required. Typically a party will be used to represent any one of the following types of entities:

- Customer
- Subscriber
- BAN
- BEN

The mentioned list is not an exhaustive list of party types. Within the Fulfillment Subscriber Inventory system, a party is a generic object which has a type and subtype classification scheme. Users are free to define their own categories within this scheme, and the only limitation is that each party must have a globally unique partyId.

Parties often exist in hierarchies, so the Fulfillment Subscriber Inventory system provides the capability of linking parties using parent child relationships. These relationships do not enforce referential integrity and are linked solely on the primary key of each party entity, specifically the partyId.

Fulfillment Subscriber Inventory also provides the ability to manage a limited set of information that describes a party in the form of characteristics. These characteristics are name-value pairs which describe attributes of the party. Each characteristic is identified with a unique identifier called a characteristicId. This identifier must be unique within the scope of any given party instance and can be specified externally or generated by the Inventory as required.

There is also the capability of locking party objects within the Inventory. Each lock is identified with a unique identifier called a key. This identifier must be unique within the scope of any given party instance and must always be specified externally.

Fulfillment Subscriber Inventory is not a CRM system. Hierarchies and characteristics should be used minimally to support the image information stored only when it is absolutely necessary.

## Adding a New Party

To add a new party perform the following steps:

### Procedure

1. Click the **Parties** menu.
2. Click the **+** button.
3. Provide the primary information, which includes
   - **Party Type** - The value can be either Enterprise, Customer, or Subscriber. This is a mandatory field.
   - **Party Status** - The value can be either Active, Inactive, Suspended, or Pending Update. This is a mandatory field.
4. Provide the additional information, which includes:
   - **Party Name**
   - **Party Sub Type**
   - **Party ID**
   - **Party Ref**
   - **Parent Party ID –** This may reference an existing partyID.
   - **Owned By –** If a value is not provided then the value will be set to the user ID that created the party.
5. Provide the characteristics information (this is optional), by clicking **+ Add** link in the **Characteristics Info** section. You can add multiple characteristics by clicking **+ Add** link. You can delete unwanted characteristics by clicking **x Remove** link. The characteristics information to be provided are as follows:
   - **Name** - This is a mandatory field.
   - **Value** - This is a mandatory field.
   - **ID**
6. Click the **Submit** button.
   The party will be created and the control will be passed back to the Inventory Party page. The newly created Party will be displayed in the list of parties. To search for party see Searching a Party for more details.

## Searching a Party

You can search a Party using the common search bar. You can use the basic navigation buttons like **Previous** and **Next** to locate the desired item. You can also search for a party using the search feature. To search for a party from the Parties menu perform the following steps:

### Procedure

1. Enter the search query with any or all of the parameters:

   - name
   - id
   - type
   - partyId
   - status

   An example for a query format is as follows: `status=<Party Status>> AND name=<<Party Name>> AND type=<<Party Type>>.`
2. Click the search button.
   The criteria satisfying the search parameters will be displayed.
3. Click **Save this Search?** link to reuse the search query.
4. Enter a relevant name for the query and click the Save icon.
   The query will be saved for reuse.
5. Click **Home** menu.
   The saved query name will be displayed.
6. Click the query name.
   The query details will be displayed in the expanded view.
7. Click execute this Party query to execute the saved query again.
   The result or results will be displayed.

## Updating a Party

To update a party, perform the following steps:

### Procedure

1. Click the **Parties** menu.
2. Locate the desired party that needs to be updated using the **Previous** or **Next** button. You can also search the item using the Search feature. See Searching a Party for more details.
3. Click the party to be modified.
   The Update Party page opens.
4. Make the necessary modifications to the desired fields and click the **Update** button.
   The party will be updated and the control will be passed back to the Inventory Party page.

## Locking a Party

Locks can only be added to an existing parties. To add a lock to an existing party perform the following steps:

### Procedure

1. Click the Parties menu.
2. Locate the desired party that needs to be locked using the **Previous** or **Next** button. You can also search the item using the Search feature. See Searching a Party for more details.
3. Click the party to be modified.
4. Provide values to the following fields:
    - **Key**
    - **Priority**
    - **Timeout**
5. Click the **Set** link.
6. Click the **Update** button.
   The party will be locked and the control will be passed back to the Inventory Party page.

## Deleting the Lock of a Party

To delete the lock of a party, perform the following steps:

### Procedure

1. Click **Parties** menu.
2. Click the Locked parties link to expand it.
3. Click the desired party that is locked.
   The Update Party page opens and you can view the lock details.
4. Click the **Clear** link in the **List of locks** section.
5. Click the **Update** button.
   The lock will be deleted and the control is passed back to the Inventory Parties page.

## Deleting a Party

To delete an existing party, perform the following steps:

### Procedure

1. Click the **Parties** menu.
2. Locate the desired party that needs to be deleted using the **Previous** or **Next** button. You can also search the party using the Search feature. See Searching a Party for more details.
3. Click the party to be deleted.
   The Update Party page opens.
4. Click the Delete button.
   The party will be deleted and the control will be passed back to the Inventory Party page.

# Locks

Party and Item entities has the capability for locking and lock management.

This is useful in a context where multiple concurrent processes are all working on the same object but it is desirably to control the order of access to the object in a defined sequence.

## Viewing Locked Items

To view locked items, perform the following steps:

#### Procedure

1. Click **Locks** menu.
2. Click the **Locked items** link to expand it.
3. Click the desired item that is locked.
   The Update Item page opens and you can view the lock details.

## Viewing Locked Parties

To view locked parties, perform the following steps:

#### Procedure

1. Click **Locks** menu.
2. Click the **Locked parties** link to expand it.
3. Click the desired party that is locked.
   The Update Party page opens and you can view the lock details.

# Roles

Roles are used to define what the users can perform, or the list of permissions the users are granted, in the Fulfillment Subscriber Inventory application.

## List of Permissions for a Role

A role can be assigned based on the following list of permissions:

| Permission Name | Description |
|---|---|
| Create parties | Can create parties |
| Read/query parties | Can read or query parties |
| Update parties | Can update parties |
| Delete parties | Can delete parties |
| Purge parties | Can purge parties |
| Create items | Can create items |
| Read/query items | Can read or query items |

| Permission Name | Description |
|---|---|
| Update items | Can update items |
| Delete items | Can delete items |
| Purge items | Can purge items |
| Start batches | - |
| Read batches | - |
| Complete batches | - |
| Update when current user is not lock owner | - |
| Super administrator | Super administrator has the permission to administer tenants on Fulfillment Subscriber Inventory. |
| User administrator | Can administer users and roles. |
| Bulk load manager | Can access bulk load feature. |

## Creating a New Role

To create a new role, perform the following steps:

#### Procedure

1. Click the **Roles** menu.
2. Click the **+** button.
3. Enter a name for the **Role ID**.
4. Select the desired list of permissions for the role. See List of Permissions for a Role for more details.
5. Click the **Submit** button.
   The new role will be created and will be displayed in the List of Roles page.

## Updating an Existing Role

To update an existing role, perform the following steps:

#### Procedure

1. Click the **Roles** menu.
2. Click the role to be modified from the list of roles.
3. Make the necessary modifications.
4. Click the **Update** button.
   The modifications will be applied to the role and the control will be passed back to the List of Roles page.

## Deleting an Existing Role

To delete an existing role you have to ensure that all users who have been assigned the role should be deleted. Perform the following steps to delete a role:

#### Procedure

1. Click the **Roles** menu.
2. Click the role to be deleted from the list of roles.
3. Click the **Delete** button.
   The role will be deleted and the control will be passed back to the List of Roles page.

# Users

You can add users with specific roles to administer how Fulfillment Subscriber Inventory is used by individual users.

Each user is assigned a role and each role has a list of permissions attached to it. See List of Permissions for a Role to know more about the permissions and see Creating a New Role to know how to add a new role.

## Creating a New User

To create a new user, perform the following steps:

#### Procedure

1. Click the **Users** menu.
2. Click the **+** button.
3. Provide values for the following fields in the Primary Information section:
   - **User ID**: The username for logging in.
   - **Password**: The password for logging in.
   - **Retype Password**: Re-enter the same value as the Password field.
4. Provide values for the following fields in the **Additional Info** section:
   - **Firstname**: First name of the user.
   - **Lastname**: Last Name of the user.
   - **Roles**: Select the role or roles the user can perform in the Fulfillment Subscriber Inventory system.
5. Click the **Submit** button.
   The new user will be created and will be displayed in the List of Users page.

## Updating an Existing User

To update an existing user, perform the following steps:

#### Procedure

1. Click the **Users** menu.
2. Click the user to be modified from the list of users.
3. Make the necessary modifications.

4. Click the **Update** button.

   The modifications will be applied to the user and the control will be passed back to the List of Users page.

## Deleting an Existing User

To delete an existing user perform the following steps:

**Procedure**

1. Click the **Users** menu.
2. Click the user to be deleted from the list of users.
3. Click the **Delete** button.
   The user will be deleted and the control will be passed back to the List of Users page.

# Bulk Load

Bulk Load refers to mass importing of data. You can use Bulk Load to migrate data from an existing system to Fulfillment Subscriber Inventory.

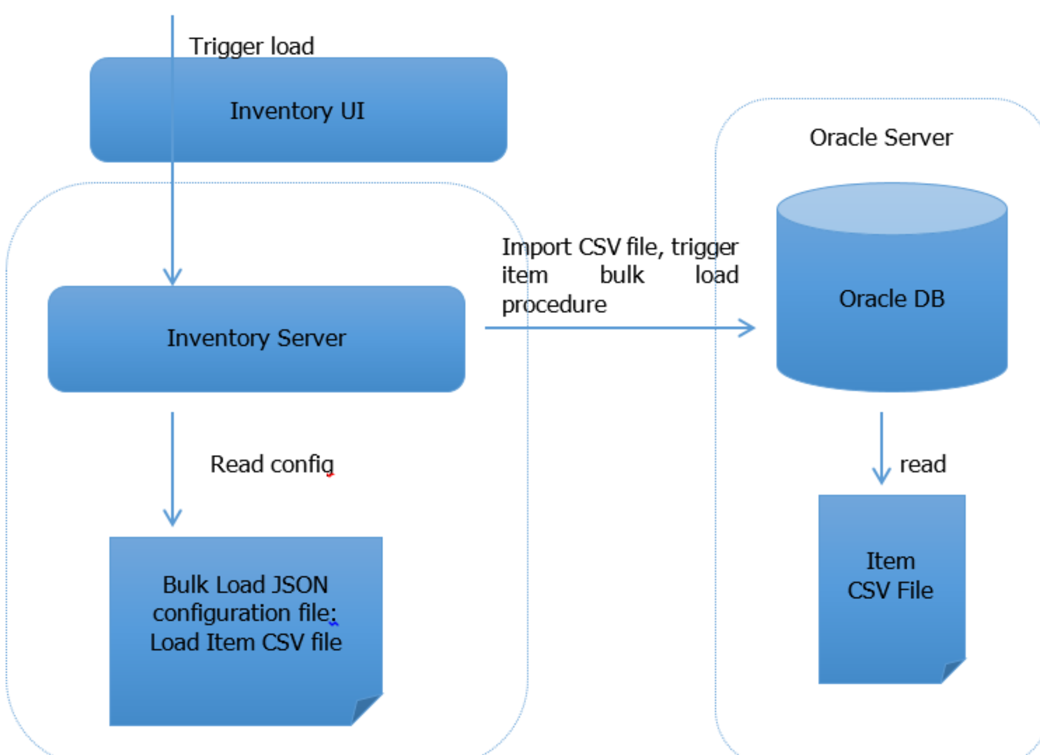
You will have to ensure that the data is in CSV format as the Oracle database reads the CSV files and treats it as a database table.

The Inventory system will automatically apply stored procedures on the data and creates internal items, parties, characteristics, relationships, or orders.

## Bulk Load Process

The following diagram describes the Bulk Load process:

*Bulk Load Flow Diagram*



Bulk Load should be performed on an empty database, as it can cause duplicates error, if the database contains IDs that are also present in the data loaded from CSV.

To bulk load items into the Fulfillment Subscriber Inventory system, you have to:

- Create a Bulk load JSON configuration file on the inventory server machine.
- Create an Item CSV File on the Oracle server machine that can be accessed from Oracle.
- Trigger a load from the Inventory user interface.

The Inventory Server will read the JSON configuration file. The JSON configuration file contains information about the CSV files that need to be loaded. The Inventory Server then triggers the read of CSV file by the Oracle server, and also triggers the conversion of the CSV file data into proper Inventory items. The Logs will be generated in the Oracle database and on the oracle server file system. These logs are accessible from the Inventory user interface bulk load pages.

## Oracle Bulk Load Setup

You must define a directory accessible by the Oracle database. You need to execute the following command:

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

You need to create the CSV file in the defined directory. The directory and the file should have read or write privileges for the Oracle process user.

## Bulk Load JSON Configuration

The Bulk Load JSON configuration drives the bulk load process. You get an indication for various load configurations, location of the CSV files, field names in the CSV files, and so on.

The configuration contains a main section defining the following parameters:

| | |
|---|---|
| oracleDirectoryName | Logical directory name for oracle. The name will be `FSI_BULK_DIR`. |
| timeStampFormat | Timestamp format that will be used by the CSV exported data. Timestamps follows the oracle notation. Sample format is "DD-MON-YYYY HH12.MI.SS PM". Example value: "01-JAN-2014 05.40.12 PM |
| commitSize | Frequency of commits when performing creation. |
| entities | List of entities to load. |

Each entity to load is a configuration object containing:

| | |
|---|---|
| type | The type of entity to load. Options are:<br>• `FSI_PARTY`<br>• `FSI_PARTY_CHARACTERISTIC`<br>• `FSI_ITEM`<br>• `FSI_ITEM_CHARACTERISTIC`<br>• `FSI_ITEM_RELATIONSHIP`<br>• `FSI_ITEM_ORDER`<br>• `FSI_ITEM_ORDER_COMMENT` |
| dataSourceName | The name of the CSV file relative to the inside of the `FSI_BULK_DIR` directory |
| dataSourceColumnList | The names of the columns in the CSV file. The possible values depend on the type of entity to load. Some columns are mandatory, some are optional. |

| Entity Type | Mandatory Columns | Optional Columns |
|---|---|---|
| FSI_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), 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)) |

| Entity Type | Mandatory Columns | Optional Columns |
|---|---|---|
| ITEM_CHARACTERISTIC | ID(VARCHAR2), ITEM_ID(VARCHAR2) (the item to relate to), CHARACTERISTIC_NAME(VARCHAR2), CHARACTERISTIC_VALUE(VARCHAR2) | |
| 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), | |
| ITEM_ORDER | ITEM_ID(VARCHAR2) (the item to relate to), ID(VARCHAR2), ORDER_REF(VARCHAR2), ORDER_DATE(TIMESTAMP(6)), LINE_NUMBER(VARCHAR2), LINE_ACTION(VARCHAR2) | LINE_ACTION_MODE(VARCHAR2), PLAN_ITEM_ID(VARCHAR2), PLAN_ITEM_ACTION(VARCHAR2) |
| ITEM_ORDER_COMMENTS | ORDER_ID, COMMENT_DETAIL | |

## Bulk Load Log Files

The log files for the bulk load will be created in the same directory where the bulk load files are placed.

There are four types of log files created for each data input file. The first three files are created by Oracle while creating the external table and are known as the log, bad, and discard files. The last file is created by the procedure which moves the records from the external table to the Oracle table.

| | |
|---|---|
| Log File | This file will be used by Oracle to log information about the process used to create the external table. The log file name will follow the format `Log_<EntityName>_<LoadId>.log`. |
| Bad File | This file contains records which cannot be loaded because of errors. The Bad file name will be of the format `Bad_<EntityName>_<LoadId>.bad` |
| Discard File | The file contains records that fail the condition in the LOAD WHEN clause of the statement used to create the external table. This file will only be created if any records are encountered that fail the LOAD WHEN condition. The Discard file name will be of the format `Discard_<EntityName>_<LoadId>.discard`. |

| | |
|---|---|
| Stored Procedure Logs | This file will be created by the procedures which will copy the records from the external able to the actual table. This file will display any errors encountered during the copying process. It will also display the current status of the load. The file name will follow the format `Log_SP_<EntityName>_<LoadId>.log`. |