



TIBCO® Order Management

Installation and Configuration

Version 6.1.0 | October 2024

Contents

Contents	2
About this Product	5
Installation Overview	6
Operating System and Database Requirements	6
Required Products	7
Required Third-Party Jars	7
Required TIBCO Products	9
TIBCO Environment	10
Installation Scenario	10
Installation Environment Variables	11
Deployment Guidelines for Optimal CPU Usage	13
Installation and Deployment	15
Post-Installation Tasks	16
Task 1: Copying Dependencies	17
Task 2: Creating the Database	17
Creating a PostgreSQL Database for the Admin User	18
Creating a PostgreSQL Database for Catalog User	19
Creating a PostgreSQL Database for the Order User	21
Creating a PostgreSQL Database for the Archival Service	22
Creating a PostgreSQL Database for Jeopardy Service User	23
Creating a PostgreSQL User Database	25
Creating a PostgreSQL Database for the Broker Service User	26
Creating an Oracle Database for the Admin User	27
Creating an Oracle Database for Catalog User	29

Creating an Oracle Database for the Order User	30
Creating an Oracle Database for the Archival Service User	31
Creating an Oracle Database for Jeopardy Service User	33
Creating an Oracle User Database	34
Creating an Oracle Database for the Broker Service User	36
Task 3: Creating or Upgrading the TIBCO Enterprise Message Service Channel	37
Task 4: Configuring and Starting the Authorization Service	38
Task 5: Registering Tenant	42
Task 6: Creating Mandatory Users	42
Task 7: Configuring and Starting the Configurator Service	43
Task 8: Configuring and Starting Configurator UI	46
Task 9: Uploading Metadata file, App Properties, Config Files through Configurator UI	48
Task 10: Configuring minimum requirements through Configurator UI	53
Task 11: Starting or Restarting the Services	71
Deployment of TIBCO® Order Management On Kubernetes	73
Order Management Resource Allocation	77
Resource allocation for Order Management services on a cloud platform	77
Resource allocation of JVM for Order Management services on a non-cloud platform (bare metal box or VM)	78
Installing Helm Chart	79
Migration	81
Before you begin the migration	93
Migrating Orders from Previous Versions to TIBCO OM 6.1.0	97
Offline Migration	98
Online Migration	102
Online-Offline Migration	103
TIBCO Documentation and Support Services	105

Legal and Third-Party Notices	107
--	------------

About this Product

TIBCO® Order Management is an elastic, catalog-driven order management system for digital service providers. It accepts orders from any customer engagement system and orchestrates the tasks required for fulfilling the orders.

TIBCO® Order Management is the next generation of TIBCO® Fulfillment Order Management and partially replaces the old product. To better align TIBCO® Fulfillment Order Management with market demand, the product's capabilities have been reorganized into two new products: TIBCO® Order Management and TIBCO® Offer and Price Engine.

Installation Overview

TIBCO Order Management provides advanced order provisioning and fulfillment features.

This chapter provides information about the system requirements in terms of operating systems and disk space, the setup for TIBCO Order Management installation, variables and properties required, and important files necessary to perform the post-install log analysis.

The following are links to detailed information for prerequisites, pre-install requirements and settings, and the install and post-install steps.

Pre-install

To ensure that you have a good experience installing TIBCO Order Management, it is always a good practice to check whether your computer is ready for installation. The pre-install section gives you the prerequisites to install TIBCO Order Management.

1. [Operating System and Database Requirements](#): This section provides you with information about the platforms that TIBCO Order Management supports.
2. [Required Products](#): TIBCO Order Management requires some software components to be installed. For a complete list of versions and platforms supported, see the `TIB_om_6.1.0_readme.txt` file. Install and configure them in the mentioned order.

Install

- [Installation and Deployment Options](#): This chapter provides you with information about the flow of the installation and the post-install steps you have to perform.

Post-install

[Post-Installation Tasks](#): After you complete the installation of TIBCO Order Management, you must complete the post-installation tasks. The post-installation tasks list the configuration and deployment steps of the components that you have just installed.

Operating System and Database Requirements

TIBCO Order Management supports the following platforms and databases:

Operating System

Red Hat Enterprise Linux Server 8.x 64-bit on x86-64

Debian 12 64-bit on x86-64

Partitioned Database

PostgreSQL 13.x

Oracle 19c

**Note:** For version number details, refer to the product Readme document.

Required Products

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

Required Third-Party Jars

The following table lists the required third-party jars:

Jars	Download link
antlr-2.7.7	https://repo1.maven.org/maven2/antlr/antlr/2.7.7/antlr-2.7.7.jar
dom4j-2.1.3	dom4j-2.1.3=https://repo1.maven.org/maven2/org/dom4j/dom4j/2.1.3/dom4j-2.1.3.jar
jms-2.0	Copy from <EMS_HOME>/lib directory

Jars	Download link
tibjms	Copy from <EMS_HOME>/lib directory
javassist-3.23.1-GA	https://repo1.maven.org/maven2/org/javassist/javassist/3.23.1-GA/javassist-3.23.1-GA.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-jpa-2.1-api-1.0.2.Final	https://repo1.maven.org/maven2/org/hibernate/javax/persistence/hibernate-jpa-2.1-api-1.0.2.Final/hibernate-jpa-2.1-api-1.0.2.Final.jar
javax.persistence-api-2.2.jar	https://repo1.maven.org/maven2/javax/persistence/javax.persistence-api/2.2/javax.persistence-api-2.2.jar
javax.activation-api-1.2.0.jar	https://repo1.maven.org/maven2/javax/activation/javax.activation-api/1.2.0/javax.activation-api-1.2.0.jar
ojdbc11	https://repo1.maven.org/maven2/com/oracle/database/jdbc/ojdbc11/21.1.0.0/ojdbc11-21.1.0.0.jar
jboss-logging-annotations-1.2.0.Beta1	https://repo1.maven.org/maven2/org/jboss/logging/jboss-logging-annotations/1.2.0.Beta1/jboss-logging-annotations-1.2.0.Beta1.jar
jboss-transaction-api_1.2_spec-1.1.1.Final	https://repo1.maven.org/maven2/org/jboss/spec/javax/transaction/jboss-transaction-api_1.2_spec/1.1.1.Final/jboss-transaction-api_1.2_spec-1.1.1.Final.jar
jboss-logging-3.4.3.Final	https://repo1.maven.org/maven2/org/jboss/logging/jboss-logging/3.4.3.Final/jboss-logging-3.4.3.Final.jar

Jars	Download link
javax.servlet-api-4.0.1	https://repo1.maven.org/maven2/javax/servlet/javax.servlet-api/4.0.1
jandex-2.4.2.Final	https://repo1.maven.org/maven2/org/jboss/jandex/2.4.2.Final/jandex-2.4.2.Final.jar

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.

Note: For product version details, you can also see the TIBCO Order Management readme file.

Required TIBCO Products

The TIBCO Order Management requires some TIBCO products.

Required TIBCO Products for TIBCO Order Management

The following table lists the required TIBCO products for TIBCO Order Management:

Required TIBCO Products for TIBCO Order Management

Product & Version	Purpose	For more information, see:
TIBCO Enterprise Message Service™ 8.7.x	Standards-based messaging software that can serve as the backbone of a Service-oriented architecture by providing Java Message Service (JMS)-compliant communications across a wide range of platforms and application technologies.	<i>TIBCO Enterprise Message Service™ Installation</i>

Product & Version	Purpose	For more information, see:
-------------------	---------	----------------------------

Note:

- TIBCO Enterprise Message Service (TIBCO EMS) is a prerequisite but is not a part of the TIBCO Order Management product license.
- It is a good practice that you configure TIBCO Enterprise Message Service for fault tolerance. For more information, see *TIBCO Enterprise Message Service™ Appliance Installation and Reference* . If you do not configure fault tolerance, you might lose order information in the event of a server failure.

TIBCO Environment

The TIBCO environment is a directory where one or more TIBCO products are installed. It is also called installation directory or TIBCO_HOME because it corresponds to the environment variable \$TIBCO_HOME.

- For root users, the default installation directory is `/opt/tibco`.
- For non-root users, the default installation directory is `/myhome/tibco`, where *myhome* is the home directory of the user.

You can choose any other directory. For performance reasons, it is a good practice to install the product on a local disk (as opposed to a network-mounted or Network File System partition).

Installation Scenario

TIBCO Order Management is made of several subsystems or components. Each of those components has a specific responsibility. Here is a brief list of the major components:

- Order Management System UI

- Configurator UI
- Orchestrator
- Automated Order Plan Development
- Authorization service
- Catalog service
- Archival service
- Order Management Migration service
- Data service
- tmf-om-adapter service
- Configurator service
- EncryptPWDUtility
- Jeopardy
- Broker service

For a complete list of all the components and their detailed role and relationships, see *TIBCO Order Management Concepts and Architecture* and *TIBCO Order Management User Guide*.

Installation Environment Variables

Recommended Environment Variables Setup

It is a good practice to set the following environment variables where TIBCO Order Management is installed:

Environment Variable	Value
export <i>OM_HOME</i>	<i>\$TIBCO_HOME/om/6.1</i>
export <i>JAVA_HOME</i>	<i>\$TIBCO_HOME/om/6.1/jdk-17.0.9</i>

Environment Variable	Value
CLASSPATH	\$EMS_LIB_PATH/*:\$CLASSPATH
EMS_LIB_PATH	\$EMS_HOME/lib
export <i>EMS_HOME</i>	\$TIBCO_HOME/ems/8.7
export <i>PATH</i>	\$EMS_HOME/bin:\$PATH
export <i>LD_LIBRARY_PATH</i>	\$JAVA_HOME/lib:\$LD_LIBRARY_PATH
<i>EXTERNAL_LIB_PATH</i>	\$OM_HOME/externalLib

Deployment Guidelines for Optimal CPU Usage

This topic provides deployment strategies for achieving optimal CPU usage across various service configurations. It includes recommendations for scenarios where services are deployed on a single instance and when the Enterprise Messaging Service (EMS) is hosted on a separate instance.

Deployment Configurations

- [All Services on the Same Instance](#)
- [EMS on a different Instance](#)

All services on the same instance

When deploying all services (EMS, Config, Broker, Orchestrator, Aopd, PC, Catalog) on the same instance, follow these guidelines:

- Orchestrator (`internalProcessorSize`): Allocate resources equivalent to 1x the number of processors.
- Aopd (`planGenerationReceiverCount`): Allocate resources equivalent to 1/4x the number of processors.

For optimal performance, configure the instance with 36 vCPUs and 60 GB RAM.

Example: Deploy one OMS instance and three OMS services on a single instance.

EMS on a different instance

For setups where EMS is hosted on a different instance from other services, use the following configuration:

- Orchestrator (`internalProcessorSize`): Allocate resources equivalent to 2x the number of processors.
- AOPD (`planGenerationReceiverCount`): Allocate resources equivalent to 1/2x the number of processors.

For optimal performance, configure the instance with 16 vCPUs and 30 GB RAM.

Example: Deploy three OMS instances distributed across two instances.

Installation and Deployment

Download the TIBCO Order Management 6.1.0 build from [TIBCO eDelivery](#). The following project artifacts are downloaded in the OM_HOME folder.

- TIB_om_6.1.0_debian.zip or TIB_om_6.1.0_rhel.zip - extract to the OM_HOME folder (as per your required OS Debian or Red Hat Enterprise Linux)

The following components are present in the TIB_om_6.1.0_debian.zip or TIB_om_6.1.0_rhel.zip file.

- db - contains Oracle, PostgreSQL, and Microsoft SQL Server database scripts.
- docker - contains files and artifacts to create and run Docker containers for all Order Management microservices.
- ems - contains files related to the creation or deletion of queues or topics.
- externalLib - here you need to copy all the [external jars](#) required for all microservices. You also need to copy the tibjms.jar and jms-2.0.jar files for TIBCO Enterprise Message Service™ and ojdbc11.jar for Oracle.
- helm - contains all files required to deploy Order Management services on the Kubernetes cluster.
- Kubernetes - contains Kubernetes scripts for all Order Management services.
- roles - contains all the configuration files for all the microservices.
- samples - sample models, catalog-client, process component services, RestTestHarnessBW6.zip, web service requests, and response files.
- schemas - schemas and http/jms WSDL
- seed-data - contains all the application properties files, configuration files, and metadata files for all the microservices.
- jdk-17.0.9_linux-x64_bin.deb or jdk-17.0.9_linux-x64_bin.tar.gz



Note: For a complete list of versions and supported platforms, see the TIB_om_6.1.0_readme.txt file.

Post-Installation Tasks

This section gives an overview of the steps you have to perform in the given order on successful installation of TIBCO Order Management.

The post-installation tasks can be carried out manually or through automation.

- To configure the installed components manually, complete the following steps:
 1. [Task 1: Copying Dependencies](#)
 2. [Task 2: Creating the Database](#)
 3. [Task 3: Creating or Upgrading the TIBCO Enterprise Message Service Channel](#)
 4. [Task 4: Configuring and Starting the Authorization Service](#)
 5. [Task 5: Registering Tenant](#)
 6. [Task 6: Creating Mandatory Users](#)
 7. [Task 7: Configuring and Starting the Configurator Service](#)
 8. [Task 8: Configuring and Starting Configurator UI](#)
 9. [Task 9: Uploading Metadata file, App Properties, Config Files through Configurator UI](#)
 10. [Task 10: Configuring minimum requirements through Configurator UI](#)
 11. [Task 11: Starting or Restarting the Services](#)
- To configure the installed components through automation, see [Automating postinstallation configurations](#).



Note: Use the HTML documentation to copy and paste code snippets into the XML files.

Task 1: Copying Dependencies

Order Management System does not ship with all the required 3rd party dependencies. A consolidated `copyLib.sh` script is provided under `$OM_HOME/roles` for all the components.

Procedure

1. Copy the required [external jars](#) to the `$OM_HOME/externalLib` directory.
2. Copy the `tibjms.jar` and `jms-2.0.jar` files for TIBCO Enterprise Message Service™ and `ojdbc11.jar` file for Oracle to the `$OM_HOME/externalLib` directory.
3. Run the `$OM_HOME/roles/copyLib.sh` script.

Task 2: Creating the Database

If this is your first time installing TIBCO Order Management, then create the database by running the provided scripts.



Note: To run the SQL scripts, you must have appropriate permission. EMS and database Hotfix scripts are never cumulative. You have to run the upgrade scripts for every Hotfix.

TIBCO Order Management supports the PostgreSQL and Oracle databases. Depending on the database you use, choose the following tasks.

- PostgreSQL database
 - [Creating a PostgreSQL Database for the Admin User](#)
 - [Creating a PostgreSQL Database for Catalog User](#)
 - [Creating a PostgreSQL database for Order User](#)
 - [Creating a PostgreSQL Database for the Archival Service User](#)
 - [Creating a PostgreSQL Database for Jeopardy Service User](#)
 - [Creating a PostgreSQL User Database](#)
 - [Creating a PostgreSQL Database for the Broker Service User](#)
- Oracle database

- [Creating an Oracle Database for the Admin User](#)
- [Creating an Oracle Database for Catalog User](#)
- [Creating an Oracle Database for Order User](#)
- [Creating an Oracle Database for the Archival Service User](#)
- [Creating an Oracle Database for Jeopardy Service User](#)
- [Creating an Oracle User Database](#)
- [Creating an Oracle Database for the Broker Service User](#)



Note: Only when the identityProviderType=postgres/oracle, the Oracle user database or PostgreSQL user database are needed.

Creating a PostgreSQL Database for the Admin User

Procedure

1. Open the `$OM_HOME/db/dbscripts/postgreSQL/admin/bin/postgres_admin_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
PG_HOME	PostgreSQL database home
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

Property	Update the Value for
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
PG_ADMIN_TABLESPACE_LOCATION	admin tablespace location
PG_ADMIN_TABLESPACE	admin tablespace
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if executing only DDL/DML.
BASE_INSTALLATION_SCRIPTS	database_ddl.sql, proc_app_properties_change.sql, proc_get_timestamp.sql

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/postgreSQL/admin/bin/db-setup.sh` script.

Creating a PostgreSQL Database for Catalog User

Procedure

1. Open the `$OM_HOME/db/dbscripts/postgreSQL/catalog/bin/postgres_catalog_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
PG_HOME	PostgreSQL database home

Property	Update the Value for
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_CATALOG_USER	New Catalog user
PG_CATALOG_PASSWORD	Password for the Catalog user
PG_CATALOG_DATABASE	Catalog database name
PG_CATALOG_SCHEMA	Catalog schema
PG_CATALOG_TABLESPACE	Catalog tablespace
PG_CATALOG_TABLESPACE_LOCATION	Catalog tablespace location
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if executing only DDL/DML.
BASE_INSTALLATION_SCRIPTS	catalog_ds_ddl.sql

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/postgreSQL/catalog/bin/db-setup.sh` script.

Creating a PostgreSQL Database for the Order User

Procedure

1. Open the `$OM_HOME/db/dbscripts/postgreSQL/order/bin/postgres_order_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
PG_HOME	PostgreSQL database home
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_DATABASES	PostgreSQL superuser database
PG_ORDER_USER	New Order user
PG_ORDER_PASSWORD	Password for the Order user
PG_ORDER_DATABASE	Order database name
PG_ORDER_SCHEMA	Order schema
PG_ORDER_TABLESPACE	Order tablespace
PG_ORDER_TABLESPACE_LOCATION	Order tablespace location
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.

Property	Update the Value for
DEFAULT_TENANT	TIBCO
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if running only DDL/DML.
BASE_INSTALLATION_SCRIPTS	order_ds_ddl.sql,oms_seed_orderlock.sql,order_ds_dml.sql

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/postgreSQL/order/bin/db-setup.sh` script.

Creating a PostgreSQL Database for the Archival Service

Procedure

1. Open the `$OM_HOME/db/dbscripts/postgreSQL/archival/bin/postgres_archival_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
PG_HOME	PostgreSQL database home
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

Property	Update the Value for
PG_ARCHIVAL_USER	New Archival user
PG_ARCHIVAL_PASSWORD	Password for the Archival user
PG_ARCHIVAL_DATABASE	Archival database name
PG_ARCHIVAL_SCHEMA	Archival schema
PG_ARCHIVAL_TABLESPACE	Archival tablespace
PG_ARCHIVAL_TABLESPACE_LOCATION	Archival tablespace location
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if you run only DDL/DML.
BASE_INSTALLATION_SCRIPTS	archival_database_ddl.sql

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/postgreSQL/archival/bin/db-setup.sh` script.

Creating a PostgreSQL Database for Jeopardy Service User

Procedure

1. Open the `$OM_HOME/db/dbscripts/postgreSQL/jeopardy/bin/postgres_jeopardy_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
PG_HOME	PostgreSQL database home
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_JEOPARDY_USER	New Jeopardy user
PG_JEOPARDY_PASSWORD	Password for the Jeopardy user
PG_JEOPARDY_DATABASE	Jeopardy database name
PG_JEOPARDY_SCHEMA	Jeopardy schema
PG_JEOPARDY_TABLESPACE	Jeopardy tablespace
PG_JEOPARDY_TABLESPACE_LOCATION	Jeopardy tablespace location
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
DEFAULT_TENANT	TIBCO
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if you run only DDL/DML.
BASE_INSTALLATION_SCRIPTS	jeopardy_database_ddl.sql,jeopardy_lock.sql,jeopardy_database_dml.sql

2. Save and close the file.

3. Run the `$OM_HOME/db/dbscripts/postgreSQL/jeopardy/bin/db-setup.sh` script.

Creating a PostgreSQL User Database

i Note: Only when the `identityProviderType=POSTGRES`, the PostgreSQL user database is needed.

Procedure

1. Open the `$OM_HOME/db/dbscripts/postgreSQL/user/bin/postgres_user_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
PG_HOME	PostgreSQL database home
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_USER	New database user
PG_PASSWORD	Database user password
PG_USER_DATABASE	User database
PG_USER_SCHEMA	User schema
PG_USER_TABLESPACE_LOCATION	User tablespace location

Property	Update the Value for
PG_USER_TABLESPACE	User tablespace
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if you run only DDL/DML.

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/postgreSQL/user/bin/db-setup.sh` script.

Creating a PostgreSQL Database for the Broker Service User

Procedure

1. Open the `$OM_HOME/db/dbscripts/postgreSQL/broker/bin/postgres_broker_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
PG_HOME	PostgreSQL database home
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_	PostgreSQL superuser database

Property	Update the Value for
DATABASE	
PG_BROKER_USER	New Broker user
PG_BROKER_PASSWORD	Password for the Broker user
PG_BROKER_DATABASE	Broker database name
PG_BROKER_SCHEMA	Broker schema
PG_BROKER_TABLESPACE_LOCATION	Broker tablespace location
PG_BROKER_TABLESPACE	Broker tablespace
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
DEFAULT_TENANT	TIBCO
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if running only DDL/DML.
BASE_INSTALLATION_SCRIPTS	database_ddl.sql

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/postgreSQL/broker/bin/db-setup.sh` script.

Creating an Oracle Database for the Admin User

Procedure

1. Open the `$OM_HOME/db/dbscripts/oracle/admin/oracle_admin_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
ORCL_HOST	Oracle database host
ORCL_PORT	Oracle port
ORCL_USERNAME	Oracle superuser name
ORCL_PASSWORD	Oracle superuser password
ORCL_SERVICENAME	Service name of the database
ORCL_ADMIN_USER	New admin user
ORCL_ADMIN_PASSWORD	Password for the admin user
ORCL_ADMIN_TABLESPACE	Admin tablespace
ORCL_TABLESPACE_SIZE	Size of the tablespace
ORCL_MINSIZE	Minimum tablespace size
ORCL_MAXSIZE	Maximum tablespace size
ORCL_DATAFILE_PATH	data file path of Oracle
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if executing only DDL/DML.
BASE_INSTALLATION_SCRIPTS	database_ddl.sql,proc_app_properties_change.sql:true,proc_get_timestamp.sql:true,blobConvertorFunction.sql:true

2. Save and close the file.

3. Run the `$OM_HOME/db/dbscripts/oracle/admin/db-setup.sh` script.

Creating an Oracle Database for Catalog User

Procedure

1. Open the `$OM_HOME/db/dbscripts/oracle/catalog/oracle_catalog_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
ORCL_HOST	Oracle database host
ORCL_PORT	Oracle port
ORCL_USERNAME	Oracle superuser name
ORCL_PASSWORD	Oracle superuser password
ORCL_SERVICENAME	Service name of the database
ORCL_CATALOG_USER	New Catalog user
ORCL_CATALOG_PASSWORD	Password for the Catalog user
ORCL_CATALOG_TABLESPACE	Catalog tablespace
ORCL_TABLESPACE_SIZE	Size of the tablespace
ORCL_MINSIZE	Minimum tablespace size
ORCL_MAXSIZE	Maximum tablespace size
ORCL_DATAFILE_PATH	data file path of Oracle

Property	Update the Value for
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if executing only DDL/DML.
BASE_INSTALLATION_SCRIPTS	OMS_DDL_CATALOG.sql

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/oracle/catalog/db-setup.sh` script.

Creating an Oracle Database for the Order User

Procedure

1. Open `$OM_HOME/db/dbscripts/oracle/order/oracle_order_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
ORCL_HOST	Oracle database host
ORCL_PORT	Oracle port
ORCL_USERNAME	Oracle superuser name
ORCL_PASSWORD	Oracle superuser password
ORCL_SERVICENAME	Service name of the database
ORCL_ORDER_USER	New Order user
ORCL_ORDER_	Password for the Order user

Property	Update the Value for
PASSWORD	
ORCL_ORDER_TABLESPACE	Order tablespace
ORCL_TABLESPACE_SIZE	Size of the tablespace
ORCL_MINSIZE	Minimum tablespace size
ORCL_MAXSIZE	Maximum tablespace size
ORCL_DATAFILE_PATH	Data file path of Oracle
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
DEFAULT_TENANT	TIBCO
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if you run only DDL/DML.
BASE_INSTALLATION_SCRIPTS	OMS_DDL.sql,OMS_Seed_OrderLock.sql:true, order_ds_dml.sql:true,blobConvertorFunction.sql:true, insertPartitionDateProcedure.sql:true, OMS_Create_Future_Partitions_SQL_Weekly.sql:true, OMS_Future_Partitions_Weekly_JOB.sql:true

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/oracle/order/db-setup.sh` script.

Creating an Oracle Database for the Archival Service User

Procedure

1. Open the `$OM_HOME/db/dbscripts/oracle/archival/oracle_archival_`

db.properties file in a suitable editor and update the following values:

Property	Update the Value for
ORCL_HOST	Oracle database host
ORCL_PORT	Oracle port
ORCL_USERNAME	Oracle superuser name
ORCL_PASSWORD	Oracle superuser password
ORCL_SERVICENAME	Service name of the database
ORCL_ARCHIVAL_USER	New Archival user
ORCL_ARCHIVAL_PASSWORD	Password for the Archival user
ORCL_ARCHIVAL_TABLESPACE	Archival tablespace
ORCL_TABLESPACE_SIZE	Size of the tablespace
ORCL_MINSIZE	Minimum tablespace size
ORCL_MAXSIZE	Maximum tablespace size
ORCL_DATAFILE_PATH	Data file path of Oracle
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if executing only DDL/DML.
BASE_INSTALLATION_SCRIPTS	archival_database_ddl.sql,blobConvertorFunction.sql:true

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/oracle/archival/db-setup.sh` script.

Creating an Oracle Database for Jeopardy Service User

Create an Oracle database for the Jeopardy by running the provided scripts.

Procedure

1. Open the `$OM_HOME/db/dbscripts/oracle/jeopardy/oracle_jeopardy_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
ORCL_HOST	Oracle database host
ORCL_PORT	Oracle port
ORCL_USERNAME	Oracle superuser name
ORCL_PASSWORD	Oracle superuser password
ORCL_SERVICENAME	Service name of the database
ORCL_JEOPARDY_USER	New Jeopardy user
ORCL_JEOPARDY_PASSWORD	Password for the Jeopardy user
ORCL_JEOPARDY_TABLESPACE	Jeopardy tablespace
ORCL_TABLESPACE_SIZE	Size of the tablespace

Property	Update the Value for
ORCL_MINSIZE	Minimum tablespace size
ORCL_MAXSIZE	Maximum tablespace size
ORCL_DATAFILE_PATH	data file path of Oracle
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
DEFAULT_TENANT	TIBCO
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if executing only DDL/DML.
BASE_INSTALLATION_SCRIPTS	jeopardy_database_ddl.sql,OMS_Seed_JeopardyLock.sql:true, jeopardy_database_dml.sql:true

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/oracle/jeopardy/db-setup.sh` script.

Creating an Oracle User Database

i Note: Only when the `identityProviderType=ORACLE`, the Oracle user database is needed.

Procedure

1. Open the `$OM_HOME/db/dbscripts/oracle/user/oracle_user_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
ORCL_HOST	Oracle database host
ORCL_PORT	Oracle port
ORCL_USERNAME	Oracle superuser name
ORCL_PASSWORD	Oracle superuser password
ORCL_SERVICENAME	Service name of the database
ORCL_USER	New database user
ORCL_USER_PASSWORD	Password for the database user
ORCL_USER_TABLESPACE	User tablespace
ORCL_USER_TABLESPACE_SIZE	Size of the tablespace
ORCL_MINSIZE	Minimum tablespace size
ORCL_MAXSIZE	Maximum tablespace size
ORCL_DATAFILE_PATH	Data file path of Oracle
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if executing only DDL/DML.

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/oracle/user/db-setup.sh` script.

Creating an Oracle Database for the Broker Service User

Create an Oracle database for the Broker service user by running the provided scripts.

Procedure

1. Open the `$OM_HOME/db/dbscripts/oracle/broker/oracle_broker_db.properties` file in a suitable editor and update the following values:

Property	Update the Value for
ORCL_HOST	Oracle database host
ORCL_PORT	Oracle port
ORCL_USERNAME	Oracle superuser name
ORCL_PASSWORD	Oracle superuser password
ORCL_SERVICENAME	Service name of the database
ORCL_BROKER_USER	New Broker user
ORCL_BROKER_PASSWORD	Password for the Broker user
ORCL_BROKER_TABLESPACE	Broker tablespace
ORCL_TABLESPACE_SIZE	Size of the tablespace
ORCL_MINSIZE	Minimum tablespace size
ORCL_MAXSIZE	Maximum tablespace size
ORCL_DATAFILE_PATH	Data file path of Oracle

Property	Update the Value for
IS_CLOUD_PLATFORM	The default value is false. Set it to true if creating a database on a cloud platform.
DEFAULT_TENANT	TIBCO
EXECUTE_DDL_DML_ONLY	The default value is false. Set it to true if executing only DDL/DML.
BASE_INSTALLATION_SCRIPTS	database_ddl.sql

2. Save and close the file.
3. Run the `$OM_HOME/db/dbscripts/oracle/broker/db-setup.sh` script.

Task 3: Creating or Upgrading the TIBCO Enterprise Message Service Channel

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

1. To create the EMS channel, go to `$EMS_HOME/bin` and run the following command:

```
$ tibemsadmin -server "tcp://localhost:7222" -user admin -password "admin" -ignore -script $OM_HOME/ems/CreateEMSChannel.txt
```

2. To enable the TIBCO EMS dynamic queue creation ability, create the queue with name ">", and execute the following command:

```
create queue >
```

Task 4: Configuring and Starting the Authorization Service

Procedure

1. Before starting the Authorization service, configure the application properties from the following table in the \$OM_HOME/roles/authorization-service/standalone/config/application.properties file. To update the properties, you can refer to the sample file present under the \$OM_HOME/samples/authorization-services directory for Oracle and PostgreSQL as per your requirement. The default values are set for PostgreSQL.

In the case of OIDC EXTERNAL identityProviderType, update the following properties in the \$OM_HOME/seed-data/app-properties/ConfigValues_Common.json file for mapping tenant, username, and user roles specific to your organization.

```
{
  "propName": "tenantIdMapping",
  "propDescription": "key in the token claims that refers to
tenantId",
  "propValue": "TENANTID",
  "valueType": "string",
  "isTenantProperty": "false"
},
{
  "propName": "userNameMapping",
  "propDescription": "key in the token claims that refers to
userName",
  "propValue": "user_name",
  "valueType": "string",
  "isTenantProperty": "false"
},
{
  "propName": "userRoleMapping",
  "propDescription": "key in the token claims that refers to
userRole",
  "propValue": "authorities",
  "valueType": "string",
  "isTenantProperty": "false"
}
```

Category	Element	Default Value
General	server.port	9091
	default.tenant.id	TIBCO
	auth.superuser.appld	auth
	auth.superuser.appKey	ENC(P2yXphz4OVM=)
	allowedUserRoles	ROLE_ADMIN, ROLE_USER
Actuator Endpoints Properties	management.endpoints.web.exposure.include	Health, ready, loggers
Authentication Token Generation Configuration	authorization.access.token.validity	43200
	authorization.refresh.token.validity	2592000
Allowed Cross	allowedCorsOrigins	http://localhost:9091,http://localhost:9090,http://localhost:9092,

Note: It is a good practice to change this default value and set your own key in an encrypted value. Refer to the 'Encrypt Password Utility' section in the *TIBCO® Order Management Administration* guide.

Note: In the case of EXTERNAL identityProviderType (OIDC), add the UserRoles specific to your organization in this property value.

Category	Element	Default Value
Origin Resources		http://localhost:9094,http://localhost:9099,http://localhost:9095,http://localhost:9102,http://localhost:9100,http://localhost:9093,http://localhost:9089,http://localhost:9104,http://localhost:8090,http://localhost:8093,http://localhost:8090
Relational Database Connection Properties	datasourceDriverClassName	org.postgresql.Driver
	adminDsUrl	jdbc:postgresql://localhost:5432/admindb?currentSchema=adminschema
	adminDsUsername	Adminuser
	adminDsPassword	ENC(O4UrXXgTEmyecFyHLo+lvw==)
	hibernateDialect	org.hibernate.dialect.PostgreSQLDialect
	hibernateDsDefaults	False
	adminHibernateShowSql	False
	adminDsInitialSize	10
	adminDsMaxWait	30000
	adminDsMaxActive	100
	adminDsMaxIdle	100
	adminDsMinIdle	10

Category	Element	Default Value
	datasourceValidationQuery	SELECT 1
	adminDsTestOnBorrow	True
	adminDsValidationInterval	5000
	adminDsTestWhileIdle	true
	adminDsTimeBetweenEvictionRunsMillis	5000
	adminDsMinEvictableIdleTimeMillis	5000
	adminDsNumTestsPerEvictionRun	5
	adminDsDefaultAutoCommit	false
	adminDsRollbackOnReturn	false
	adminDsCommitOnReturn	false
Host Header Filter Properties	hostHeaderSecurityFlag	false
	trustedDomain	localhost
Get total	getTotalDsConnection	false

Category	Element	Default Value
	number of db connection in management/health API	

2. Start the authorization service by running the `start.sh` script from the `$OM_HOME/roles/authorization-service/standalone/bin` directory.

Task 5: Registering Tenant

Procedure

1. Register the tenant by using the tenant registration API. For more details, see the "Tenant Registration APIs" section in the *TIBCO® Order Web Services Guide*.

Task 6: Creating Mandatory Users

Procedure

1. Create an Admin user with the default tenant Id and user roles as 'ROLE_ADMIN'. See the "Create User" section in the *TIBCO® Order Management Administration* guide.

i Note: You can set the `apikey` value in the `ConfigValues_Common.json` file. This value is used for inter-service communications.

Task 7: Configuring and Starting the Configurator Service

Procedure

1. Before starting the Configurator service, configure the application properties from the following table in the \$OM_HOME/roles/configurator/standalone/config/application.properties file. The default values are set for PostgreSQL.

Category	Element	Default Value
General	server.port	9090
	security.key	ENC(nSa0k6lmjPPN8ZA5SO6BpQ==)
	PluggableCache	Relational
	configuratorAccessRoles	ROLE_ADMIN
Relational Database Connection Properties	adminDsUrl	jdbc:postgresql://localhost:5432/admindbll?currentSchema=adminschema_v2
	adminDsUsername	Adminuserll
	adminDsPassword	ENC(O4UrXXgTEmyecFyHLo+lvw==)
	adminDsHibernateDefaults	false
	datasourceDriverClassName	org.postgresql.Driver
	hibernateDialect	org.hibernate.dialect.PostgreSQLDialect

Category	Element	Default Value
	adminDsHibernateShowSql	False

Category	Element	Default Value
	adminDsInitialSize=	10
	adminDsMaxWait	30000
	adminDsMaxActive	100
	adminDsMaxIdle	100
	adminDsMinIdle	10
	datasourceValidationQuery	SELECT 1
	adminDsTestOnBorrow	True
	adminDsValidationInterval	5000
	adminDsTestWhileIdle	true
	adminDsTimeBetweenEvictionRunsMillis	5000
	adminDsMinEvictableIdleTimeMillis	5000
	adminDsNumTestsPerEvictionRun	5
	adminDsDefaultAutoCommit	false
	adminDsRollbackOnReturn	false
	adminDsCommitOnReturn	false

Category	Element	Default Value
Actuator Endpoints Properties	management.endpoints.web.exposure.include	Health, refresh, loggers
	management.endpoint.health.show-details	ALWAYS
Notification Properties	notificationChannel	app_properties_events
	listenIntervalInMillis	1000
	purgeNotificationIntervalInMinutes	15
	purgeNotificationStartDelayInMinutes	15
	purgeNotificationOffsetInMinutes	5
Host Header Filter Properties	hostHeaderSecurityFlag	false
	trustedDomain	localhost

2. Start the configurator service by running the `start.sh` script from `$OM_HOME/roles/configurator/standalone/bin` location.

Task 8: Configuring and Starting Configurator UI

Procedure

1. Before starting the Configurator UI, configure the application properties from the following table in the `$OM_HOME/roles/configurator-ui/standalone/config/application.properties` file.

Category	Element	Default Value
General	server.port	9104
	configuratorServiceUrl	http://localhost:9090
	configuratorServiceRetryCount	5
	configuratorServiceRetryDuration	5
	configuratorTrustStoreAbsolutePath	C:/Users/cacert
	configuratorTrustStorePassword	tibco123
	configuratorTrustStoreType	jks
	authorizationServiceTokenEndPoint	http://localhost:9091
	authServiceApiKey	auth
	authServiceApild	auth
	authServiceRetryDuration	3
	authServiceRetryCount	5
	authServiceTrustStoreAbsolutePath	cacerts
	authServiceTrustStorePassword	changeit
	authServiceTrustStoreType	pkcs12

Category	Element	Default Value
Actuator Properties	management.endpoints.web.exposure.include	Health
User name mapping	userNameMapping	user_name,upn

2. Start the configurator UI service by running the `start.sh` script from `$OM_HOME/roles/configurator-ui/standalone/bin` location.

Task 9: Uploading Metadata file, App Properties, Config Files through Configurator UI

Once you start the Configurator UI, you can upload the metadata file, application properties, and configuration files through the UI. These files are present in the `$OM_HOME/seed-data` directory.

Metadata: It defines the required application properties and configuration files for each application.

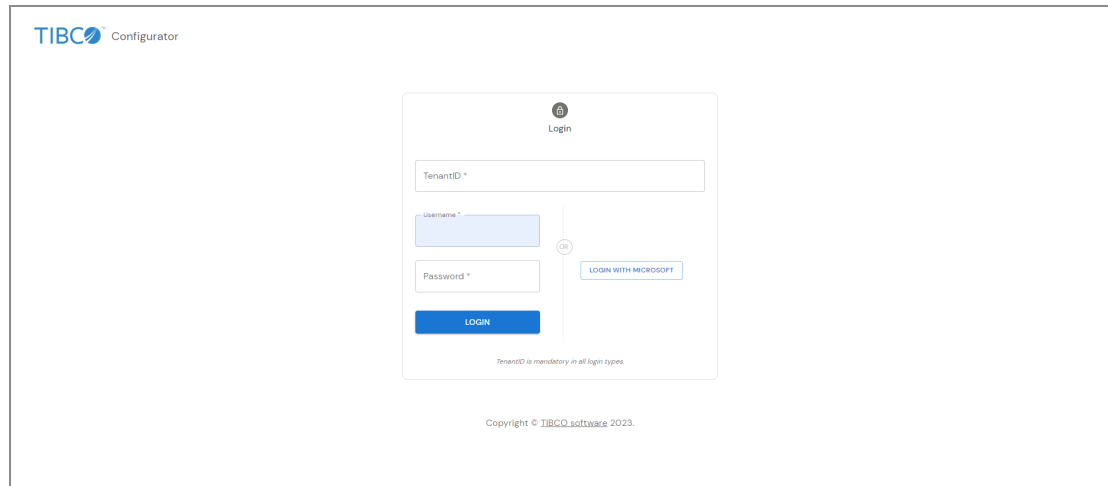
App properties: These are application properties in JSON format.

Config files: These are Logback configuration files in XML format.

Logging in to Configurator UI

Procedure

1. To access the Login page, visit `http://<host>:<port number>`, where:
 - `host` is the computer where you have started Configurator UI service
 - `port` is the port number of the machine where Configurator UI listens for the requests. (**Default:** 9104)



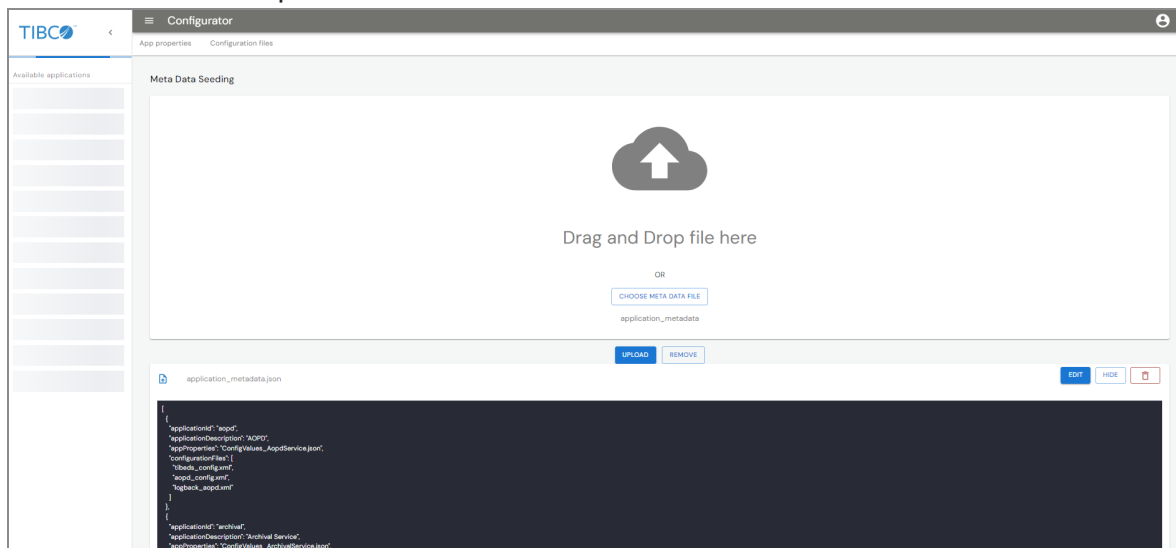
2. Enter the credentials in the **TenantID**, **Username**, and **Password** fields and then click **LOGIN**.

The UI dashboard opens.

Upload Metadata file

Procedure

1. Click **SEEDING** to upload the metadata file.



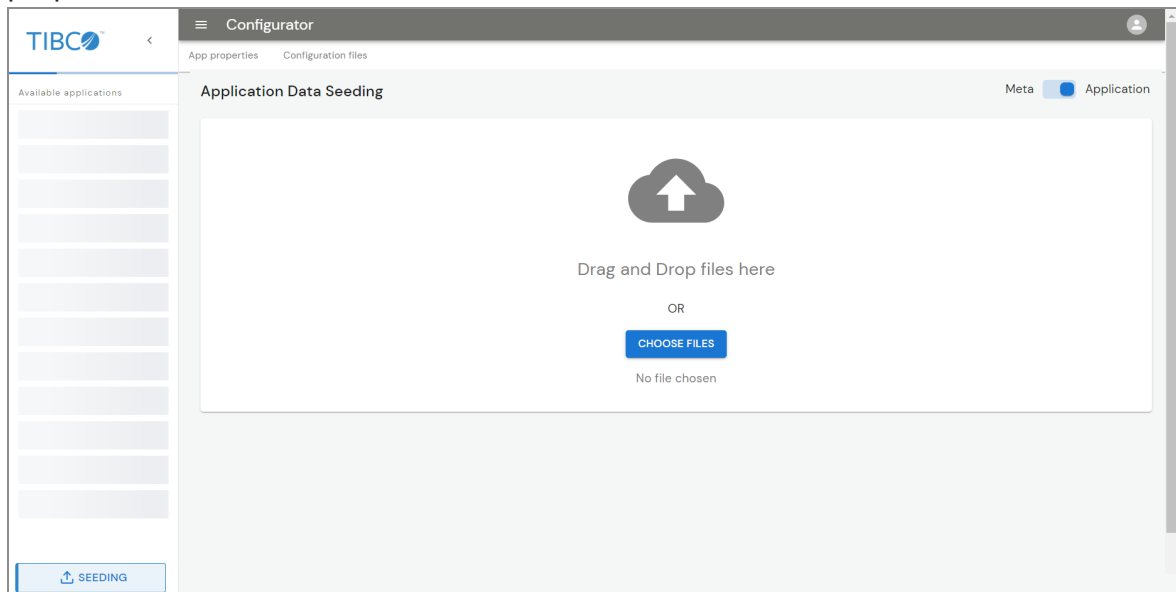
2. Use the drag-and-drop function or click **CHOOSE META DATA FILE** to select the `application_metadata.json` file from the `$OM_HOME/seed-data` directory.


3. Finally, click **UPLOAD** to upload the file.

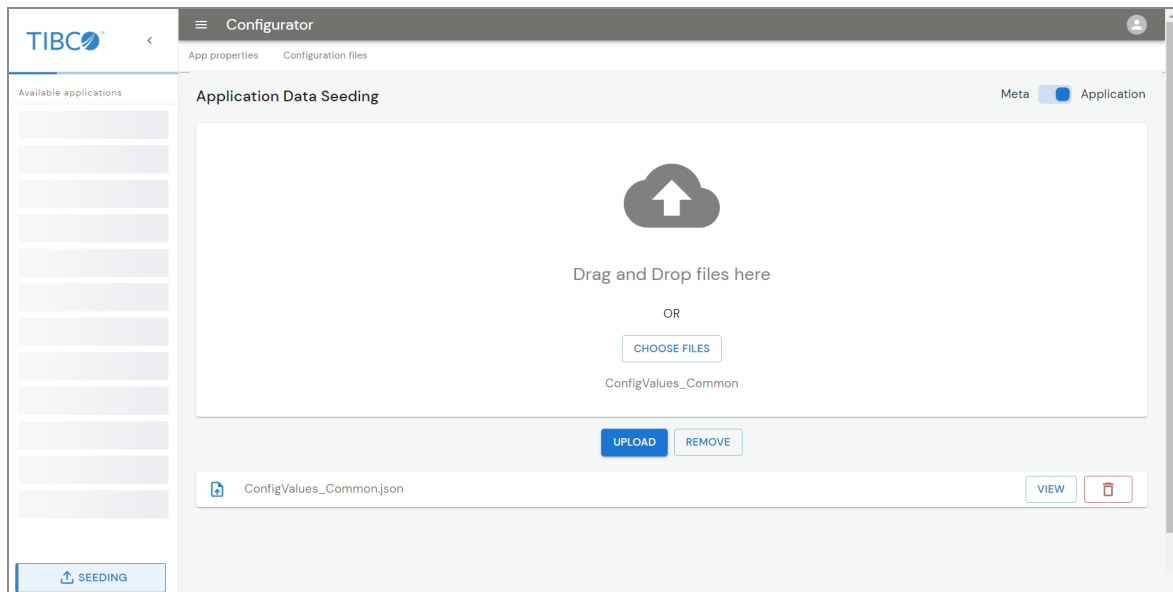
Upload Application Properties

Procedure

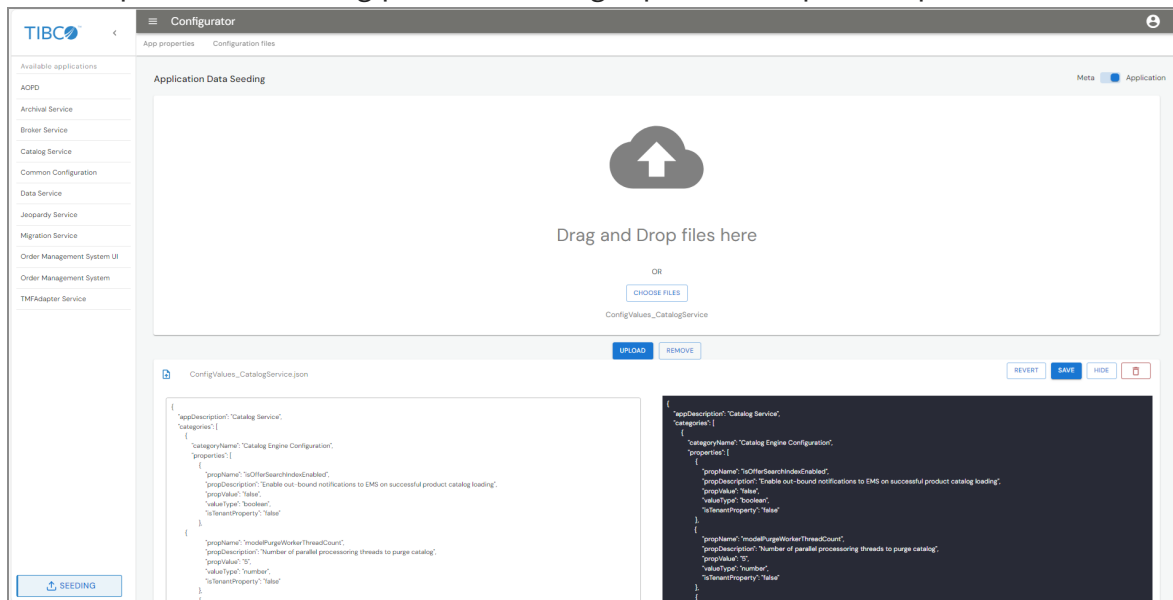
1. Toggle the switch to select **Application** and click **SEEDING** to upload the app properties files.



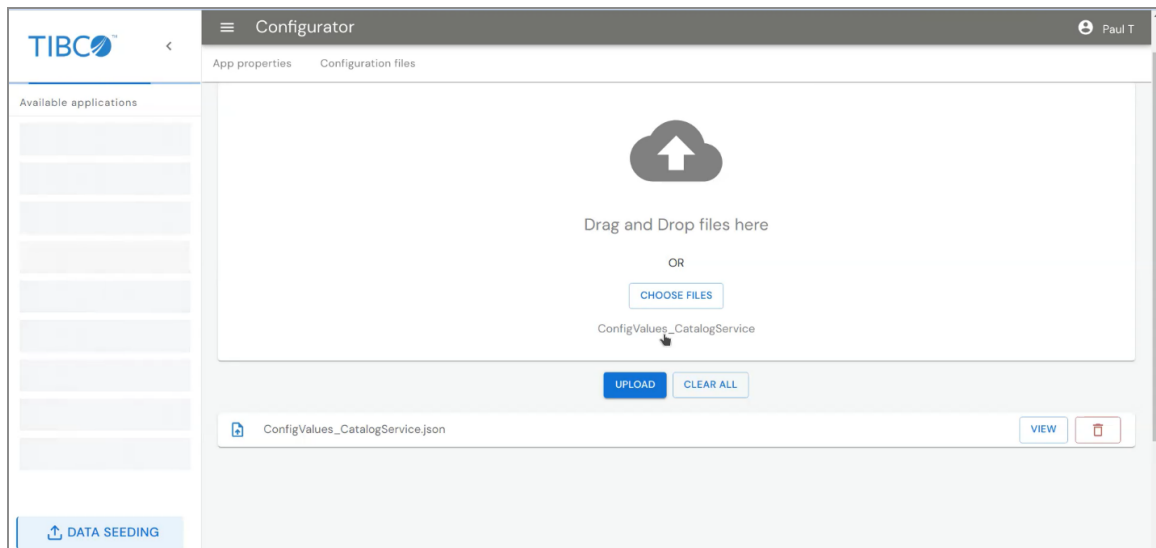
2. Use the drag-and-drop function or click **CHOOSE FILES** to select the app properties files from the `$OM_HOME/seed-data/app-properties` directory. Once you have uploaded a file, the file is shown below the **CHOOSE FILES** button. A pop-up message is added that shows at the top-right corner to let the users know that the files have been uploaded and can now be removed from this page by clicking the  bin button on the right side of the file name.



3. To view an application property file, click **VIEW**. The file opens in text editor mode. The left pane is the editing pane and the right pane is the preview pane.



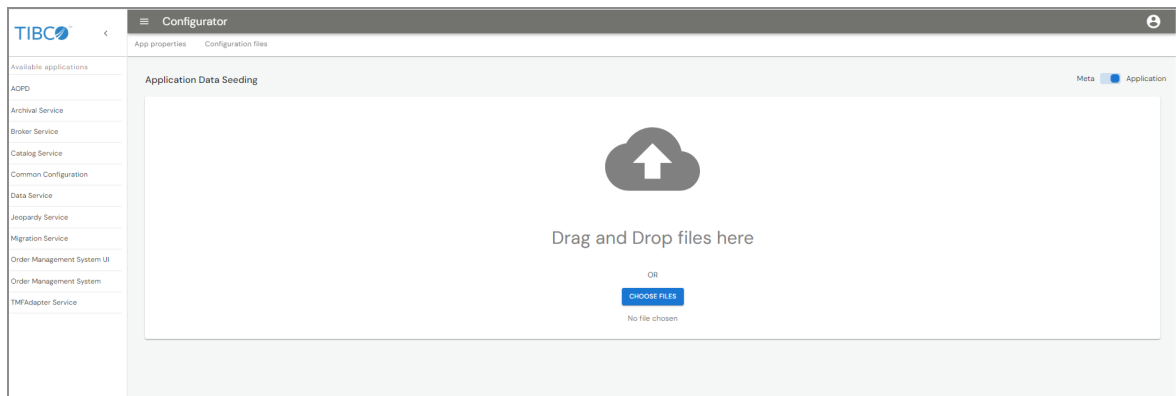
4. To discard or save the changes, click **REVERT** or **SAVE**.
5. To hide or delete the files, click **HIDE** or **DELETE**.
6. Finally, click **UPLOAD** to upload the files. You can also use the **UPLOAD ALL** or **CLEAR ALL** option to upload or remove all the files at once.




Upload Configuration Files

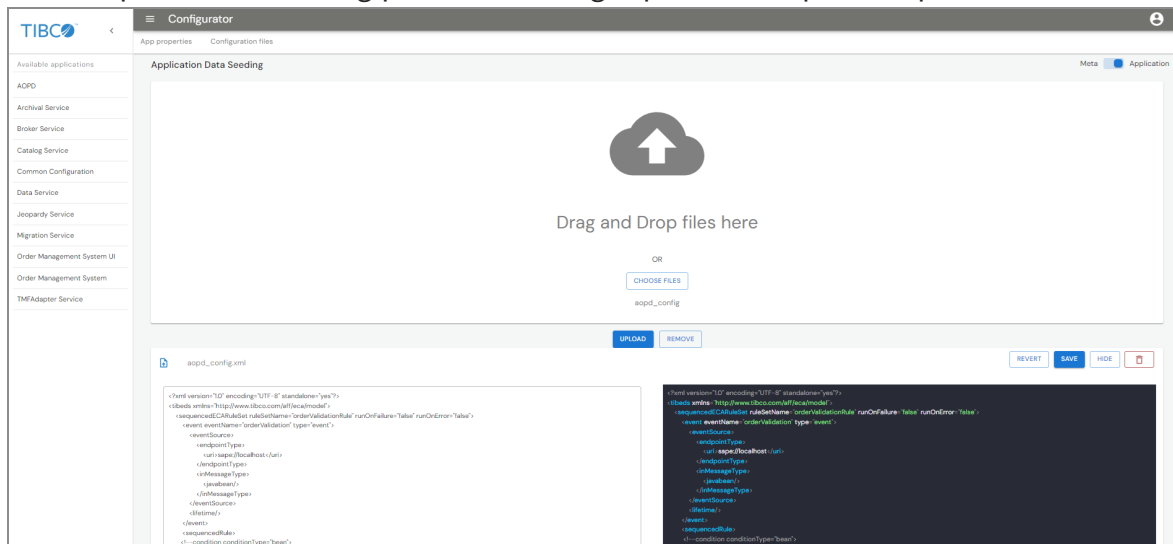
Procedure

1. To add a configuration file from the `$OM_HOME/seed-data/config-files` directory, click **DATA SEEDING**.



2. Use the drag-and-drop function or click **CHOOSE FILES** to select the app properties files from the `$OM_HOME/seed-data/config-files` directory. Once you have uploaded a file, the file is shown below the **CHOOSE FILES** button. A pop-up message is added that shows at the top-right corner to let the users know that the files have been uploaded and can now be removed from this page by clicking the  bin button on the right side of the file name.
3. To view an application property file, click **VIEW**. The file opens in text editor mode.

The left pane is the editing pane and the right pane is the preview pane.



4. To discard or save the changes, click **REVERT** or **SAVE**.
5. To hide or delete the files, click **HIDE** or **DELETE**.
6. Click **UPLOAD ALL** to upload all the config files.



Note: On the **Configuration files** tab, you can verify that all the config files are present or not and accordingly you can add the missing files.

Task 10: Configuring minimum requirements through Configurator UI

Once you upload the metadata file, application properties, and configuration files, you have to configure the minimum required properties for each service through Configurator UI to make them ready to start.

Common Configuration

Procedure

1. On the Configurator UI, navigate to **Common Configuration > App properties** and

select the **Authorization Server Configuration Properties Used for Swagger UI** category.

2. Update the property value as per the following table:

Property name	Value
authorizationServiceTokenEndPoint	IP address or the DNS name of the auth server.

Catalog Service

Procedure

1. On the Configurator UI, navigate to **Catalog Service > App properties**.
 - a. Select the **EMS Configurations for Online Catalog Publishing from PSC** category and update the properties values as per the following table:

Property name	Value
emsServerURL	IP address or the DNS name where EMS is running
emsServerUsername	Username of the EMS server
emsServerPassword	Password of the EMS server

- b. Select the **Catalog Relational Data Source Configuration** category and update the properties values as per the following table:

Property name	Value
catalogDsUrl	Data source URL (Oracle or PostgreSQL)
catalogDsUsername	Username of the catalog data source
catalogDsPassword	Password of the catalog data source
datasourceDriverClassName	Class name of the data source driver (Oracle or PostgreSQL)
datasourceValidationQuery	SQL query that is used to validate connections

- c. Select the **Generic Relational Data Source Configuration** category and update the properties value as per the following table:

Property name	Value
databaseType	Type of database (Oracle or PostgreSQL)
hibernateDialect	Hibernate

Property name	Value
	dialect (Oracle or PostgreSQL)

- d. Select the **Catalog Engine Configuration** category and update the properties value as per the following table:

Property name	Value
allowedCorsOrigins	IP address or the DNS name where OMSUI is running

AOPD

Procedure

1. On the Configurator UI, navigate to **AOPD > App properties**.
 - a. Select the **EMS Configurations** category and update the properties values as per the following table:

Property name	Value
emsServerURL	IP address or the DNS name where EMS is running
emsServerUsername	Username of the

Property name	Value
	EMS server
emsServerPassword	Password of the EMS server

- b. Select the **Generic Relational Data Source Configuration** category and update the properties value as per the following table:

Property name	Value
databaseType	Type of database (Oracle or PostgreSQL)
hibernateDialect	Hibernate dialect (Oracle or PostgreSQL)

- c. Select the **Relational Data Source Configuration** category and update the properties values as per the following table:

Property name	Value
catalogDsUrl	Data source URL (Oracle or PostgreSQL)
catalogDsUsername	Username of the catalog data source

Property name	Value
catalogDsPassword	Password of the catalog data source
datasourceDriverClassName	Class name of the data source driver (Oracle or PostgreSQL)
datasourceValidationQuery	SQL query that is used to validate connections

Order Management System

Procedure

1. On the Configurator UI, navigate to **Order Management System > App properties**.
 - a. Select the **Messaging Configuration** category and update the properties values as per the following table:

Property name	Value
emsServerURL	IP address or the DNS name where EMS is running
emsServerUsername	Username of the EMS server

Property name	Value
emsServerPassword	Password of the EMS server

- b. Select the **Catalog Relational Data Source Configuration** category and update the properties values as per the following table:

Property name	Value
catalogDsUrl	Data source URL (Oracle or PostgreSQL)
catalogDsUsername	Username of the catalog data source
catalogDsPassword	Password of the catalog data source

- c. Select the **Order Relational Data Source Configuration** category and update the properties values as per the following table:

Property name	Value
orderDsUrl	Data source URL (Oracle or PostgreSQL)
orderDsUsername	Username of the order data source

Property name	Value
orderDsPassword	Password of the order data source

- d. Select the **Generic Relational Data Source Configuration** category and update the properties value as per the following table:

Property name	Value
databaseType	Type of database (Oracle or PostgreSQL)
datasourceDriverClassName	Class name of the data source driver (Oracle or PostgreSQL)
datasourceValidationQuery	SQL query that is used to validate connections
hibernateDialect	Hibernate dialect (Oracle or PostgreSQL)

- e. Select the **Archival Service Configurations** category and update the properties values as per the following table:

Property name	Value
archivalGetOrderDetailsURL	IP address or the DNS name where Archival is running

- f. Select the **Orchestrator Functional Configuration** category and update the properties values as per the following table:

Property name	Value
allowedCorsOrigins	IP address or the DNS name where OMSUI is running

- g. Select the **Broker Service Configurations** category and update the properties values as per the following table:

Property name	Value
brokerServiceBaseUrl	IP address or the DNS name where the Broker service is running

Archival Service

Procedure

1. On the Configurator UI, navigate to **Archival Service > App properties**.
 - a. Select the **EMS Configurations for Orchestrator - Archival Communication** category and update the properties values as per the following table:

Property name	Value
emsServerURL	IP address or the DNS name where EMS is running
emsServerUsername	Username of the EMS server
emsServerPassword	Password of the EMS server

- b. Select the **Relational Database Configurations** category and update the properties values as per the following table:

Property name	Value
archivalDsUrl	Data source URL (Oracle or PostgreSQL)
archivalDsUsername	Username of the archival data source

Property name	Value
archivalDsPassword	Password of the archival data source
datasourceDriverClassName	Class name of the data source driver (Oracle or PostgreSQL)
datasourceValidationQuery	SQL query that is used to validate connections
hibernateDialect	Hibernate dialect (Oracle or PostgreSQL)

- c. Select the **Orchestrator Configuration** category and update the properties values as per the following table:

Property name	Value
orchestratorBaseUrl	IP address or the DNS name where Orchestrator is running

- d. Select the **Archival Engine Configuration** category and update the properties values as per the following table:

Property name	Value
allowedCorsOrigins	IP address or the DNS name where OMSUI is running

Order Management System UI

Procedure

1. On the Configurator UI, navigate to **Order Management System UI > App properties**.
 - a. Select the **Messaging Configurations** category and update the properties values as per the following table:

Property name	Value
emsServerURL	IP address or the DNS name where EMS is running
emsServerUsername	Username of the EMS server
emsServerPassword	Password of the EMS server

- b. Select the **OMS UI Engine Configuration** category and update the properties values as per the following table:

Property name	Value
archivalServiceUrl	IP address or the DNS name where Archival service is running
catalogServiceUrl	IP address or the DNS name where Catalog service is running
jeopardyServiceUrl	IP address or the DNS name where Jeopardy service is running
orchestratorServiceUrl	IP address or the DNS name where Orchestrator service is running
brokerServiceBaseUrl	IP address or the DNS name where the Broker service is running
aopdBaseUrl	IP address or the DNS name

Property name	Value
	where the AOPD service is running

Data Service

Procedure

1. On the Configurator UI, navigate to **Data Service > App properties**.
 - a. Select the **EMS Configurations** category and update the properties values as per the following table:

Property name	Value
emsServerURL	IP address or the DNS name where EMS is running
emsServerUsername	Username of the EMS server
emsServerPassword	Password of the EMS server

- b. Select the **Order Data Source Configuration** category and update the properties values as per the following table:

Property name	Value
orderDsUrl	Data source URL (Oracle or PostgreSQL)
orderDsUsername	Username of the order data source
orderDsPassword	Password of the order data source
datasourceDriverClassName	Class name of the data source driver (Oracle or PostgreSQL)
datasourceValidationQuery	SQL query that is used to validate connections

- c. Select the **Generic Relational Data Source Configuration** category and update the properties value as per the following table:

Property name	Value
databaseType	Type of database (Oracle or PostgreSQL)
hibernateDialect	Hibernate

Property name	Value
	dialect (Oracle or PostgreSQL)

Jeopardy Service

Procedure

1. On the Configurator UI, navigate to **Jeopardy Service > App properties**.
 - a. Select the **Catalog Service Configuration** category and update the properties values as per the following table:

Property name	Value
catalogServiceBaseUrl	IP address or the DNS name where Catalog service is running

- b. Select the **Messaging Configurations** category and update the properties values as per the following table:

Property name	Value
emsServerURL	IP address or the DNS name where EMS is running
emsServerUsername	Username of

Property name	Value
	the EMS server
emsServerPassword	Password of the EMS server

- c. Select the **Jeopardy Data Source Configuration** category and update the properties values as per the following table:

Property name	Value
jeopardyDsUrl	Data source URL (Oracle or PostgreSQL)
jeopardyDsUsername	Username of the order data source
jeopardyDsPassword	Password of the order data source
hibernateDialect	Hibernate dialect (Oracle or PostgreSQL)
datasourceDriverClassName	Class name of the data source driver (Oracle or PostgreSQL)

Property name	Value
datasourceValidationQuery	SQL query that is used to validate connections

- d. Select the **Jeopardy General Configuration** category and update the properties values as per the following table:

Property name	Value
allowedCorsOrigins	IP address or the DNS name where OMSUI is running

- e. Select the **Orchestrator Service Configuration** category and update the properties values as per the following table:

Property name	Value
orchestratorBaseUrl	IP address or the DNS name where Orchestrator is running

Broker Service

Procedure

1. On the Configurator UI, navigate to **Broker Service > App properties**.
 - a. Select the **Catalog Client Configuration** category and update the properties

values as per the following table:

Property name	Value
catalogServiceBaseUrl	IP address or the DNS name where the Catalog service is running

- b. Select the **Broker service data source properties** category and update the properties values as per the following table:

Property name	Value
brokerDsUsername	Username of the Broker data source
brokerDsPassword	Password of the Broker data source
brokerDsUrl	Data source URL (Oracle or PostgreSQL)

Task 11: Starting or Restarting the Services

See [Order Management Resource Allocation](#), if you want to use resource allocation for Order Management services.



Note: You can choose to provide 'Xms Xmx' or 'MinRAMPercentage MaxRAMPercentage' parameters based on your requirements.

Procedure

1. To restart the configurator service, navigate to the \$OM_HOME/roles/configurator/standalone/bin directory and run the ./start.sh script.
2. To restart the configurator-ui service, navigate to the \$OM_HOME/roles/configurator-ui/standalone/bin directory and run the ./start.sh script.
3. To start the other services, navigate to the \$OM_HOME/roles/<servicename>/standalone/config and update the application.properties file for configuratorServiceUrl, the value is the IP or DNS name of the configurator service.
4. Navigate to the \$OM_HOME/roles/<servicename>/standalone/bin/ directory and run the ./start.sh script.



Note: You must start the Authorization and configurator service first and then all the other services.

Deployment of TIBCO® Order Management On Kubernetes

You can containerize the TIBCO® Order Management application in Docker and then deploy it on the Kubernetes cluster. To containerize the application, you must build and run the Docker images by using the bundled Dockerfile. The deployment uses Kubernetes concepts such as, deployment, pod, kubectl, Kubernetes service. For more details, see [Kubernetes Documentation](#).

Building Docker Images

Preparation for the deployment starts with building Docker images using the Dockerfile. The `$OM_HOME/docker` directory contains subdirectories with the ready-to-use Dockerfile and other scripts required to build the images. Detailed instructions on how to use the Dockerfile and build the image are provided in the respective Dockerfile and the Readme located at the same location. See the "Docker" section in the *TIBCO® Order Management Administration* guide for more details.

Deployment on Kubernetes

After creating the Docker files, you must push them in the Docker registry so that they can be accessed by Kubernetes deployment. Kubernetes scripts are available at the `$OM_HOME/kubernetes` directory.

Prerequisites

- Copying dependency: See [Task 1: Copying Dependencies](#) for more details.
- Creating the database: See [Task 2: Creating the Database](#) for more details.
- TIBCO EMS service channel creation: See [Task 3: Creating or Upgrading the TIBCO Enterprise Message Service Channel](#) for more details.
- Configuring the installation: See [Task 10: Configuring minimum requirements through Configurator UI](#) for more details.

Example

The following example includes a snippet of the `kubernetes-deploy-run-catalog-service.yml` file:

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: catalog-service
  labels:
    app: catalog-service
spec:
  replicas: 1
  selector:
    matchLabels:
      app: catalog-service
  template:
    metadata:
      name: catalog-service
      labels:
        app: catalog-service
    spec:
      containers:
        - name: catalog-service
          image: tibco/catalog-service:6.1.0
          ports:
            - containerPort: 9092
              readinessProbe:
                failureThreshold: 3
                httpGet:
                  path: /management/health/readiness
                  port: 9092
                  scheme: HTTP
                periodSeconds: 300
                successThreshold: 1
                timeoutSeconds: 3
              livenessProbe:
                failureThreshold: 3
                httpGet:
                  path: /management/health/liveness
                  port: 9092
                  scheme: HTTP
                periodSeconds: 300
                successThreshold: 1
                timeoutSeconds: 3
          env:
            - name: min_ram_percentage
              value: "25.0"
            - name: max_ram_percentage

```

```

        value: "100.0"
        resources:
            limits:
                cpu: '1'
                memory: 4Gi
            requests:
                cpu: '0.1'
                memory: 1Gi
---
apiVersion: v1
kind: Service
metadata:
  name: catalog-service-service
spec:
  selector:
    app: catalog-service
  ports:
    - port: 9092
      targetPort: 9092
      nodePort: 30004
  type: LoadBalancer

```

Procedure

1. Run the following command on the worker node (in master-worker Kubernetes cluster setup) to create TIBCO® Order Management Catalog deployment and service:

```
CMD>kubectl apply -f D:/Folder_Path/kubernetes-deploy-run-catalog-service.yml
```

The deployment.apps/catalog-service and service/catalog-service folders are created.

2. To check the container status, run the following commands:

```
- kubectl get all
- kubectl get pods
```



Note: The Running status indicates that a container is running without any issues.

3. To check logs inside the container, run the following command: CMD>kubectl exec -it Podname -- bash

Example: kubectl exec -it catalog-service-755fb5f69c-jch2s -- bash

When the catalog pod is up and running, users can access the Swagger-UI by accessing endpoints exposed for the service.

Example: `http://EXTERNAL-IP:9092/swagger-ui.html`

4. To delete deployment and service, run the following command:

```
CMD> kubectl delete -f D:/Folder_Path /kubernetes-deploy-run-catalog-service.yml
```

The deployment.apps "catalog-service" and service "catalog-service" folders are deleted.



Note: Similar steps are applicable for all services.

Order Management Resource Allocation

Resource allocation for Order Management services on a cloud platform

1. Docker containers

Modify the .env file present in the \$OM_HOME/docker directory for MinRAMPercentage and MaxRAMPercentage usage for the following values:

```
min_ram_percentage=25.0
max_ram_percentage=100.0
```

2. Kubernetes

Kubernetes users need to modify the individual Kubernetes script files as required.

Example:

For the configurator service, modify the kubernetes-deploy-run-configurator.yml file:

```
env:
- name: min_ram_percentage
  value: "25.0"
- name: max_ram_percentage
  value: "100.0"
```

3. Helm Charts

Helm Charts users must change MinRAMPercentage and MaxRAMPercentage based on their requirements in the values.yaml file:

```
min_ram_percentage: 25.0
max_ram_percentage: 100.0
```

Resource allocation of JVM for Order Management services on a non-cloud platform (bare metal box or VM)

When any Order Management service is started from a bare-metal Linux machine, the user must set the JVM parameters for each micro-service that starts for Order Management as per the following value:

```
./start.sh -XX:MinRAMPercentage=25.0 -XX:MaxRAMPercentage=100.0
```

i Note: The values for the `cpu/memory` resources change as per the load on a particular micro-service. If the user chooses any type of out-of-the-box caching offered along with Order Management, by default, the values for the `cpu/memory` resources must be increased as required.

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. For TIBCO Order Management, Helm chart supports on the following cloud platforms:

- Azure Kubernetes Service (AKS)
- Amazon Elastic Kubernetes Service (EKS)
- Google Kubernetes Engine (GKE)

Before you begin

1. Helm Client version 3.5.0 to 3.9.x must be installed on your Kubernetes cluster.
2. EMS server must be deployed on the same Kubernetes cluster (in the same subnet) with all the required queues, topics, and bridges.
3. Create database users by running the scripts present under the `$OM_HOME/db/dbscripts` directory. Ensure that the database is in the same subnet for the cloud instance that you use.
4. Create Docker images as follows:
 - a. Ensure that the `<third-party libraries>link` is present in the virtual machine that you are working on and then run the `$OM_HOME/roles/copyLib.sh` script.
 - b. Run the `$OM_HOME/docker/copy-required-files.sh` script.
 - c. Create Docker images for all the services. For supporting commands, see the `$OM_HOME/docker/Readme.txt` file.
 - d. Push the Docker images to the required Docker registry.

Procedure

1. Log in to the Kubernetes cluster.
2. Copy the `/om_services` folder from the `$OM_HOME/helm` directory to the Kubernetes cluster.
3. Set the environment variables and Docker image names for all the required services

in the `values.yaml` file that is present at the `/om_services` folder.

4. Run the helm chart from the location where the copied directories are present.

Example: `helm installs om ./om_services`

i Note: The `values.yaml` file contains the required properties for starting authorization service, configurator service, and configurator UI services. Create required users from the authorization service and upload required `app_properties`, `metadata`, and `config` files as per components from the configurator service. Then you can deploy all services. For more details, refer to the `README.md` file from the helm directory.

5. Modify and configure the `values.yaml` file to deploy required Order Management services by selecting components to be deployed as `true`.
Mostly changes are related to database, EMS, and intercommunication of microservices.
6. Choose your desired ingress controller by adding the value for `ingressClassName` field in the `values.yaml` file. The default supported one is `nginx`. You can configure all the services for ingress control in the `om_ingress.yaml` file present inside the `templates` directory.
7. Run the helm chart again.

Result

A Helm chart is deployed with all the services present in the chart. See the `$OM_HOME/samples/helm/values.yaml` file for reference.

i Note: If you want to enable SSL for Order Management services in helm chart, see 'Configuring SSL for TIBCO® Order Management' section in the *TIBCO® Order Management Administration*.

Migration

You can migrate orders to TIBCO Order Management 6.1.0 (OM 6.1.0) from the following previous versions:

- TIBCO® Fulfillment Order Management 4.0.2 HF-14 (FOM 4.0.2 HF-14)
- TIBCO Order Management - Long Running 5.0.1 HF-6 (OM-LR 5.0.1 HF-6)
- TIBCO Order Management 5.1.0 HF-8 (OM 5.1.0 HF-8)
- TIBCO Order Management 6.0.0 HF-3 (OM 6.0.0 HF-3)

Database Details

To migrate the orders from the previous versions, the `migrated_orders` table needs to be present in the previous versions order database. This table is generated after running the migration scripts in the following locations:

- For PostgreSQL, `<OM_HOME>/db/dbscripts/migrationTo610/postgreSQL`
- For Oracle, `<OM_HOME>/db/dbscripts/migrationTo610/oracle`

i Note: If you migrate the database from any previous version to the 6.1.0 version, you have to republish the catalog.

Configure the application properties file

Before starting, the Migration service gets the necessary configuration properties from the Configurator component. Some properties, found in the `<OM_HOME>/roles/om-migration/standalone/config/application.properties` file, must be available for the server to start smoothly.

Property Name	Property Value	Description
server.port	9100	Port at which

Property Name	Property Value	Description
		the Migration service starts
configuratorServiceUrl	http://localhost:9090	Configurator base URL to connect and download configuration properties and configuration files
configuratorServiceRetryCount	5	Number of retries to perform in case of request failure to Configurator
configuratorServiceRetryDuration	5	Delay between each retry, in seconds
configuratorTrustStoreAbsolutePath	C:/Users/cacert	Trust store file path if the Configurator is on HTTPS
configuratorTrustStorePassword	tibco123	Trust store password if the Configurator is on HTTPS
configuratorTrustStoreType	jks	Trust store type if the Configurator is on HTTPS

Property Name	Property Value	Description
security.key	ENC (nSa0k6lmjPPN8ZA5SO6BpQ==)	Security key to securely connect to Configurator. The default value is encrypted of 1t1s@asy

Configure the configuration values for migration file

The Configurator must have specific configuration properties. These properties, found in the <OM_HOME>/seed-data/app-properties/ConfigValues_Migration.JSON file, must be uploaded to the configuration before the server starts.

Archival Database Configurations for 6.1.0 Application

The Migration service moves orders to an Archival database using the specified database configurations.

Property Name	Property Value	Description
archivalDatasourceDriverClassName	org.postgresql.Driver	Archival Data Source Driver Name
archivalDatasourceValidationQuery	SELECT 1	SQL query that is used to validate connections
archivalDsInitialSize	10	Number of connections that are established

Property Name	Property Value	Description
		when the connection pool is started
archivalDsMaxActive	100	Maximum number of active connections that can be allocated from this pool at the same time
archivalDsMaxIdle	100	Maximum number of connections that must be kept in the idle pool
archivalDsMaxWait	10000	Maximum number of milliseconds that the pool waits when there are no available connections
archivalDsMinIdle	100	Minimum number of connections that must

Property Name	Property Value	Description
		be kept in the idle pool
archivalDsPassword	ENC(0yIDLwbMh07VvOWs4tFwDA==)	Data Source Password
archivalDsTestOnBorrow	true	Enable connection validation before being borrowed from the pool
archivalDsUrl	jdbc:postgresql://localhost:5432/archivaldbll?currentSchema=archivalschema	Data Source URL
archivalDsUsername	archivaluserll	Data Source Username
archivalDsValidationInterval	5000	Data source validation interval in milliseconds
archivalHibernateDialect	org.hibernate.dialect.PostgreSQLDialect	Hibernate Dialect
archivalHibernateDsDefaults	false	Enable Hibernate Metadata Defaults
archivalHibernateShowSql	false	Hibernate Show SQL

Order Data Source Configuration

The Migration service migrates orders from FOM 4.0.2 HF-14, OM-LR 5.0.1 HF-6, OM 5.1.0 HF-8, and OM 6.0.0 HF-3 using the following database configurations.

Property Name	Property Value	Description
omsDsDriverClassName	oracle.jdbc.driver.OracleDriver	Data source driver class name
omsDsInitialSize	10	Number of connections that are established when the connection pool is started
omsDsMaxActive	100	Maximum number of active connections that can be allocated from this pool at the same time
omsDsMaxIdle	100	Maximum number of connections that must be kept in the idle pool
omsDsMaxWait	30000	Maximum number of milliseconds

Property Name	Property Value	Description
		that the pool waits when there are no available connections
omsDsMinIdle	100	Minimum number of connections that must be kept in the idle pool
omsDsPassword	ENC(iuGvcNQz8iS3QS1kSVGuwA==)	Data source password
omsDsTestOnBorrow	true	Enable connection validation before being borrowed from the pool
omsDsUrl	jdbc:oracle:thin:@//localhost:1521/orcl.apac.tibco.com	Data source URL
omsDsUsername	tenantpdb	Data source Username
omsDsValidationInterval	5000	Data source validation interval in milliseconds

Property Name	Property Value	Description
omsDsValidationQuery	select 1 from dual	SQL query that is used to validate connections
omsHibernateDialect	org.hibernate.dialect.Oracle10gDialect	Hibernate dialect
omsHibernateDsDefaults	false	Enable hibernate metadata defaults
omsHibernateShowSql	false	Hibernate show SQL

Admin Data Source Configuration for Legacy Application

The Migration service migrates orders from either FOM 4.0.2 HF-14 or OM-LR 5.0.1 HF-6. The required configuration for the admin database are given in the following table.

Property Name	Property Value	Description
adminDatabaseDriver	oracle.jdbc.driver.OracleDriver	Data source driver class name
adminDatabaseUrl	jdbc:oracle:thin:@//localhost:1521/orcl.apac.tibco.com	Data source URL
adminDatabaseUsername	Admin Data Source Username	adminpdb

Property Name	Property Value	Description
adminDatabasePassword	ENC(GakyNNVqkpgAGrpT+ex7lg==)	Data source password

Messaging Configuration

The Migration service maintains a connection to the EMS server to process orders using the following configurations.

Property Name	Property Value	Description
emsServerPassword	ENC(T9aNk07NMU=)	EMS server password
emsServerURL	tibjmsnaming://localhost:7222	EMS server URL
emsServerUsername	admin	EMS server username
omsOrdersSequencerSubmitOrder	tibco.aff.oms.ordersSequencer.submitOrder	Submit order sequencer queue
omsOrdersSequencerSubmitOrderReceiveTimeout	1000	Submit order sequencer queue receive TimeOut in milliseconds
securityProtocol		For SSL configuratio

Property Name	Property Value	Description
		n, the value must be ssl
sslEnableVerifyHost		For SSL enable, the verify host value must be false
jndiConnectionFactory	GenericConnectionFactory	JNDI connection factory

Migration Functional Configuration

Functional configurations for the migration service are given in the following table.

Property Name	Property Value	Description
com.tibco.archival.order.udf.searchkeys	*	Frequently searched Order UDF names in ordersByCriteria API
com.tibco.archival.orderline.udf.searchkeys	*	Frequently searched Order Line UDF names in ordersByCriteria API
com.tibco.fom.orch.enableMilestoneReleaseDuringActivation	true	Enable Milestone Release during activation

Property Name	Property Value	Description
com.tibco.fom.orch.enableOrderSequencing	false	Enable the migration of sequenced orders from 5.0.* Application
com.tibco.offlineMigration.migrateOrderInFinalState	false	Enable migration of orders in final state
com.tibco.oms.legacy.tenantId	TIBCO	5.0.* Application Tenant ID to consider for order migration
enableAuditTrail	true	Enable migration of audit trail
enableLegacyMigration	true	Enable migration from legacy (LR) version

Order Data Source Configuration for 6.1.0 Application

Database configurations for the 6.1.0 application are as follows.

Property Name	Property Value	Description
orderDatasourceDriverClass Name	org.postgresql.Driver	Data source driver class name
orderDatasourceValidationQuery	SELECT 1	SQL query that is used to validate connections
orderDsInitializeSize	10	Number of connections

Property Name	Property Value	Description
		that are established when the connection pool is started
orderDsMaxActive	100	Maximum number of active connections that can be allocated from this pool at the same time
orderDsMaxIdle	100	Maximum number of connections that must be kept in the idle pool
orderDsMaxWait	10000	Maximum number of milliseconds that the pool waits when there are no available connections
orderDsMinIdle	100	Minimum number of connections that must be kept in the idle pool
orderDsPassword	Data Source Password	ENC (A9fTymcpAOSVjnpb6jkvj A==)
orderDsTestOnBorrow	false	Enable connection validation before being borrowed from the pool
orderDsUrl	jdbc:postgresql://localhost:5432/ orderdbll?currentSchema=orders chemall	Data source URL

Property Name	Property Value	Description
orderDsUsername	orderdsuserll	Data source Username
orderDsValidationInterval	5000	Data source validation interval in milliseconds
orderHibernateDialect	org.hibernate.dialect.PostgreSQLDialect	Hibernate dialect
orderHibernateDsDefaults	false	Enable hibernate metadata defaults
orderHibernateShowSql	false	Hibernate show SQL

Before you begin the migration

The messages, corresponding to the in-process orders in the previous versions, are not allowed to remain in the pending state on the queues mentioned in the following section. These messages must be processed before upgrading. However, there are a couple of queues on which messages can remain pending.

i Note: When you try to migrate the OPD, Feasibility, OPDERROR, or PQF orders, it gives an error in the log as the plan is not created. You must process all the pending messages from these queues before migration. Therefore the migrations for such orders are directly rejected and only the valid orders are processed.

i Note: In OM 6.1.0, the plan item is not completed if its intermediate milestone is in pending state. You must notify milestone before migrating an order.

i Note: Audit trails are migrated as is without any transformations.

i Note: Complete all the amendments from the previous versions before migrating the orders.

- Stop the northbound system (for example, Siebel CRM), which sends the order request messages to TIBCO Order Management for fulfillment. Doing this ensures that there are no new order messages coming on the **tibco.aff.oms.ordersService** queue. All the existing messages must be processed by the OMS server component.
- Stop the southbound process component systems, which are integrated with Order Management for processing various requests for plan items, such as execute request, suspend request, activate request, and milestone release request. Doing this ensures that there are no messages pending on the following queues:
 - **tibco.aff.orchestrator.planItem.execute.reply**
 - **tibco.aff.orchestrator.planItem.suspend.reply**
 - **tibco.aff.orchestrator.planItem.milestone.notify.request**
 - **tibco.aff.orchestrator.planItem.suspend.request**

All the existing messages must be processed by the OMS server component in the older version of the product from where you are migrating.

Ensure that there are no new messages coming from process components on the following queues associated with the JMS-based data access interfaces, which are used to get the order data and get or set the plan/plan item data:

- **tibco.aff.tds.order.read.request**
- **tibco.aff.tds.plan.request**
- **tibco.aff.tds.plan.read.request**

All the existing messages must be processed by the OMS server in the older version of the product from where you are migrating.

- Ensure that there are no messages pending on the following queues related to the various types of order requests submitted to the Orchestrator:
 - **tibco.aff.orchestrator.order.suspend**
 - **tibco.aff.orchestrator.order.activate**
 - **tibco.aff.orchestrator.order.withdraw**
- If the order feasibility check is enabled in the Orchestrator configuration, ensure that there are no messages pending on the following queues:
 - **tibco.aff.orchestrator.provider.order.feasibility.request**
 - **tibco.aff.orchestrator.provider.order.feasibility.reply**

The external feasibility provider component must process all the request messages and the Orchestrator must process all the reply messages.

- Ensure that there are no messages pending on any of the following queues that are used for the integration between Orchestrator and the AOPD components for execution plan generation:
 - **tibco.aff.orchestrator.provider.order.opd.request**
 - **tibco.aff.orchestrator.provider.order.opd.reply**
- If order prequalification handling is enabled in the Orchestrator configuration, ensure that there are no messages pending on the following queues:
 - **tibco.aff.orchestrator.provider.order.prequal.failed.request**
 - **tibco.aff.orchestrator.provider.order.prequal.failed.reply**

The external prequalification failed request handler must process all the request messages and the Orchestrator must process all the reply messages.

- Ensure that there are no messages pending on any of the following queues that are used for integration between Orchestrator and the external plan item error handler component for processing the failed plan item requests:
 - **tibco.aff.orchestrator.provider.planItem.failed.request**
 - **tibco.aff.orchestrator.provider.planItem.failed.reply**

Considering the pending messages on all the earlier mentioned queues are processed by the respective components of the previous versions.

Messages can be pending on the following queues. These are the outbound queues for the Orchestrator to send various requests for plan items to the process components. The messages on these queues are processed once the process component systems are started after the upgrade.

- **tibco.aff.orchestrator.planItem.execute.request**
- **tibco.aff.orchestrator.planItem.suspend.request**
- **tibco.aff.orchestrator.planItem.activate.request**
- **tibco.aff.orchestrator.planItem.milestone.release.request**

If the requestReply header property is set to false in the GetOrder, GetPlan, GetPlanItem, SetPlan, or SetPlanItem data access requests, there can be pending messages on the following queues:

- `tibco.aff.tds.order.reply`
- `tibco.aff.tds.plan.reply`

If the `requestReply` header property is set to `true`, messages can be pending on the queues passed as `replyTo` destinations in the requests. These are also the outbound queues for the OMS server and the pending messages on these queues are also processed once the process component systems are started after the upgrade.

After ensuring that no further processing is going on in any of the servers of the older version of the product from where you are migrating, they can be shut down. Also, shutdown all the external components, such as feasibility provider, pre-qualification failed request handler, external OPD, and plan item error handler component.

Backing up the EMS Messages

You must back up the pending EMS messages when you are migrating from any of the versions.

Installing TIBCO Order Management 6.1.0

Download the TIBCO Order Management 6.1.0 build from [TIBCO eDelivery](#) and extract the `TIB_om_6.1.0_debian.zip` or `TIB_om_6.1.0_rhel.zip` file to the `OM_HOME` folder. For more information, see "Installation and Deployment" of the *TIBCO® Order Management Installation and Configuration*.



Note:

- You cannot migrate orders that are in OPD status. To migrate such orders, ensure that the orders have passed the OPD state.
- When the same order ID is present in both Order Management - Long Running 5.0.1 HF-6/Fulfillment Order Management 4.0.2 HF-14 and Order Management 5.1.0 HF-8/OM 6.0.0 HF-3, the order of the latter gets replaced during migration.
- Catalog is not migrated. You need to reload models on Order Management 6.1.0

All the above steps are mandatory for both online and offline migration. Once you have completed all the above mentioned steps, you can start the following services from Order Management 6.1.0:

- Orchestrator
- AOPD (Automated Order Plan Development)
- Archival
- DataService
- OMS UI
- Migration service
- Catalog

Migrating Orders from Previous Versions to TIBCO OM 6.1.0

For migration, you can use any of the following procedures:

- [Offline Migration](#)
- [Online Migration](#)
- [Online-Offline Migration](#)

i Note: Migration of orders from multi-tenant databases is supported without any configuration or restarting the services.

i Note: Migration for Jeopardy and Order Messages is not supported.

i Note: When you migrate orders from OM 5.1.0 HF-8 or OM 6.0.0 HF-3 to OM 6.1.0, you must set `enableLegacyMigration` to false from the Migration service.

i Note: Migration is not supported for Microsoft SQL server.

Final Order

When the `migrateOrderInFinalState` flag is true:

- Final regular orders are converted from the previous versions order database and stored solely within the OM 6.1.0 Archival database.
- For final amend orders, data is converted from the previous versions order database and stored in the OM 6.1.0 Archival database. Additionally, data are from the `order_amendment` table is retrieved, converted, and saved into the corresponding Archival amendment tables.

Offline Migration

This process is called offline migration because before you start using the Orchestrator from Order Management 6.1.0, you need to complete the migration. This is an on-demand process. You can choose which orders to migrate.

Procedure

1. On the **Configurator UI**, navigate to **Migration service > App properties**.
 - a. Select the **Admin Data Source Configuration for legacy Application** category and provide admin database property details of the previous version from where the migration is done. For more information, see [Admin Data Source Configuration for Legacy Application](#).
 - b. Select the **Order Data Source Configuration** category and provide the default tenant database property details of the previous version from where the migration is done. For more information, see [Order Data Source Configuration](#).
 - c. Select the **Messaging Configuration** category and provide the property details of the previous version from where the migration is done. For more information, see [Messaging Configuration](#).



Note: The current EMS and the previous (source) version must be the same.

- d. Select the **Archival Database Configurations for 6.1 Application** category and provide property details from the current database. For more information, see

[Archival Database Configurations for 6.1.0 Application](#) .

- e. Select the **Order Data Source Configuration for 6.1 Application** category and provide property details from the current database. For more information, see [Order Data Source Configuration for 6.1.0 Application](#).
 - f. (Optional) Select a **Migration Functional Configuration** category and set the value of the `com.tibco.offlineMigration.migrateOrderInFinalState` flag as true to migrate the final state (CANCELED, COMPLETE) orders. For more information, see [Migration Functional Configuration](#).
2. Navigate to the `$OM_HOME/roles/om-migration/standalone/config` directory and update the `application.properties` file for `configuratorServiceUrl` (value is the IP or DNS name of the Configurator service).
 3. To start the Migration service, navigate to the `$OM_HOME/roles/om-migration/standalone/bin` directory and run the `./start.sh` script.
 4. To access the Swagger REST API, enter the following endpoint in a browser:
`http://localhost:9100`.
 5. Run the request to migrate orders from the following endpoints:
 - FOM 4.0.2 HF-14 or OM-LR 5.0.1, `/v1/lr/migration/order`
 - OM 5.1.0 HF-8, `/v1/migration/orders`
 - OM 6.0.0 HF-3, `/v1/migration/order`

You can enter various criteria, such as `orderId`, `orderRef`, and `dateRange`.

Sample

```
{
  "orderId": "string",
  "orderRef": "string",
  "dateRange": {
    "startDate": "2024-09-12T15:11:50.366Z",
    "endDate": "2024-09-12T15:11:50.366Z"
  },
  "status": [
    "string"
  ],
  "headerUdfs": [
    {
```

```

        "name": "string",
        "value": "string"
      }
    ],
    "orderLineUdfs": [
      {
        "name": "string",
        "value": "string"
      }
    ]
  }
}

```

Request Schema

Parameter	Description
orderId	Order ID to be migrated
orderRef	Order reference of Order to be migrated
dateRange	Date range in between which all orders are to be migrated
status	Status of the order to be migrated
headerUdfs	UDF headers of order to be migrated
orderLineUdfs	Order line UDFs of order to be migrated

This API handles migration of orders from FOM 4.0.2 HF-14 or OM-LR 5.0.1 HF-6 to OM 6.1.0

Request validations and corresponding error responses

- If all search criteria are empty

```

{
  "status": "BAD_REQUEST",
  "message": "All search criteria are either null or empty"
}

```

```
}
```

- Invalid search criteria

```
{
  "status": "NOT_FOUND",
  "message": "No order that matches the given criteria is
available to migrate."
}
```

- If the request is invalid

```
Error: response status is 400
```

Response Body

```
{
  "migratedOrderIds": [
    "string"
  ]
}
```

Response Schema

Parameter	Description
migratedOrderIds	Order ids which are migrated

Responses

HTTP Status	Description
200	Orders are migrated
208	Reported orders are already migrated
400	Bad request

HTTP Status	Description
401	Unauthorized request
404	Not found
500	Internal server error

6. Start the other services. For more information, see "Task 10: Starting or Restarting the Services" of the *TIBCO® Order Management Installation and Configuration* guide.
7. Start the southbound system so that the Orchestrator service can start processing these messages.
8. Once you observe that the existing messages for the in-flight orders have been fulfilled by the Orchestrator, start the new incoming orders flow from the northbound system.

Result

The in-progress orders are migrated to the Order database and Archival database and the final state orders are migrated to the Archival database. If Order Sequencing is enabled (`com.tibco.fom.orch.enableOrderSequencing` is set as true), the in-progress sequenced orders are migrated one by one to the Order database and Archival database with Order Status as Blocked.

- Based on the request, it migrates orders to OM 6.1.0 database in bulk or single order.
- Based on the `enableAuditTrail` flag value, it migrates audit trail data.
- Based on the `migrateOrderInFinalState` flag value, it migrates orders in their final state.

Online Migration

This process is called online migration, because the in-flight orders from the source version can be processed in 6.1.0 version directly. You have to stop the services from the source version of the product to start the online migration process.

Online migration is performance intensive as the in-flight orders are migrated on the fly. For each in coming message processing from southbound, there is a call to the Migration service from Orchestrator.

Procedure

1. On the **Configurator UI**, navigate to **Order Management System > App properties**.
2. Select the **Migration Service Configurations** category and set the migrationURL property as follows:
 - For FOM 4.0.2 HF-14 or OM-LR 5.0.1,
http://<host>:9100/v1/lr/migration/order.
 - For OM 5.1.0 HF-8, http://<host>:9100/v1/lr/migration/orders.
 - For OM 6.0.0 HF-3, http://<host>:9100/v1/lr/migration/order.
3. Once you configure these properties, restart the Orchestrator service.
4. Start the southbound system so that the Orchestrator service can start processing these messages.
5. Once you observe that the existing messages for the in-flight orders have been fulfilled by Orchestrator, start the new incoming orders flow from the northbound system.

Online-Offline Migration

Both the online and offline Migration services have their advantages and disadvantages. To overcome that, it is a best practice to use them together as per your requirements. For the in-flight orders, online migration is best suited. For orders that stay pending for a long duration, offline migration is more useful.

Troubleshooting for orders migrated from FOM 4.0.2 HF-14 or OM-LR 5.0.1 HF-6

When you migrate orders from FOM 4.0.2 HF-14 or OM-LR 5.0.1 HF-6 to OM 6.1.0, the orders remain in EXECUTION even though pending messages are processed in OM 6.1.0.

For the workaround, you can perform the following steps:

1. **Ensure Broker service awareness:** Confirm that the Broker service recognizes the new Orchestrator instance, OM 6.1.0, and that it is properly configured to communicate with the new instance.
2. **Register the FOM 4.0.2 HF-14 or OM-LR 5.0.1 HF-6 Instance:** Register the FOM

4.0.2 HF-14 or OM-LR 5.0.1 HF-6 member id (originator) with the new Broker service. This instance does not communicate with the Broker service and eventually becomes inactive.

3. **Monitor Inbound Queue:** Once the FOM 4.0.2 HF-14 or OM-LR 5.0.1 HF-6 instance is inactive, the Broker service monitors the Orchestrator inbound queue for replies that include the originator header, using the member ID as the identifier.
4. **Proceed with Migration:** When waiting for the FOM 4.0.2 HF-14 or OM-LR 5.0.1 HF-6 instance to become inactive, continue migrating the orders.
5. **Start the New Orchestrator:** After the FOM 4.0.2 HF-14 or OM-LR 5.0.1 HF-6 instance is inactive, start the new Orchestrator and Process Component (PC) instance in OM 6.1.0 to ensure that message processing resumes properly.

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 [Product Documentation website](#), mainly in HTML and PDF formats.

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

Product-Specific Documentation

The documentation for this product is available on the [TIBCO® Order Management Product Documentation](#) page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our [product Support website](#).
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the [product Support website](#). If you do not have a username, 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 CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

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

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

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

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. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

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.

Cloud SG 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 a specific version of Cloud SG software on a specific operating system platform.

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

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

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 "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.cloud.com/legal>.

Copyright © 2010-2024. Cloud Software Group, Inc. All Rights Reserved.