Installation Guide
Oracle Banking Digital Experience
Patchset Release 22.2.6.0.0

Part No. F72987-01

April 2025

ORACLE

Installation Guide

April 2025

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax:+91 22 6718 3001

www.oracle.com/financialservices/

Table of Contents

1. Preface

1.1 Purpose

Welcome to the User Guide for Oracle Banking Digital Experience. This guide explains the operations that the user will follow while using the application.

1.2 Audience

This manual is intended for Customers and Partners who setup and use Oracle Banking Digital Experience.

1.3 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit, http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1.4 Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at Critical Patches, Security Alerts and Bulletins. All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by Oracle Software Security Assurance.

1.5 Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1.6 Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.

Italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1.7 Screenshot Disclaimer

The images of screens used in this user manual are for illustrative purpose only, to provide improved understanding of the functionality; actual screens that appear in the application may vary based on selected browser, theme, and mobile devices.

1.8 Acronyms and Abbreviations

The list of the acronyms and abbreviations that you are likely to find in the manual are as follows:

Abbreviation	Description
OBDX	Oracle Banking Digital Experience

2. Introduction

2.1 Purpose of the Document

The purpose of the OBDX Installation Manual is to provide a step by step overview on the installation process of the solution.

It includes:

- Reference to prerequisites software installation required for OBDX & OBDX installer
- Setup of OBDX with Oracle's own Core Banking and Origination Products along with Third-party HOST system.
- Running the installation in silent mode
- Advanced Configurations (Post installation)
- Installation Verification
- Multi-Entity Installation and configuration
- Best Practice
- Troubleshoot Overview

3. Prerequisites

OBDX pre-requisite software should be installed and available before proceeding.

For OBDX pre-requisite software setup refers document "**Oracle Banking Digital Experience Installer Pre-Requisite Setup Manual"** .

Installer Pre-requisite verification

Post installation of OBDX Installer prerequisite software's, verification can be done using below steps.

Note: Verification should be performed on Server where Oracle Weblogic is locally installed and by OS user (which is owner for Oracle Weblogic home directory) for non-root steps. The same user will be used to execute installer.

Oracle Instant client

Step 1: Login using root user.

Step 2 : Run below command to verify if Oracle Instant client is installed.

rpm -qa | grep oracle

```
[root@                    opt]# rpm -qa | grep oracle
oracle-cloud-agent-1.11.4-5207.el8.x86_64
oraclelinux-release-8.3-1.0.4.el8.x86_64
oraclelinux-release-el8-1.0-16.0.1.el8.x86_64
oracle-cloud-agent-1.10.0-4792.el8.x86_64
oraclelinux-developer-release-el8-1.0-6.el8.x86_64
oracle-epel-release-el8-1.0-3.el8.x86_64
oracle-logos-80.5-1.0.6.el8.x86_64
oracle-instantclient19.10-basic-19.10.0.0.0-1.x86_64
[root@                    opt]#
```

Note: Above package verification command is specific to Oracle Linux and RHEL distributions only. For other Linux distributions or OS please refer to OS specific package manager documentation.

Python:

Step 1: Execute python –V command

python3.11 -V

```
[devops@obdxwls ~]$
[devops@obdxwls ~]$ python3.11 -V
Python 3.11.9
[devops@obdxwls ~]$
```

Note: Ensure Python 3.11.0 supported version is installed. Above command should reflect the same.

cx_Oracle & Urwid:

Step 1: Execute python command

python

Note: Ensure Python 3.11.0 version should be available in PATH variable. Above execution should be done using Python 3.11.0.

Step 2: Import Urwid and check version

import urwid (Press Enter)

urwid.__version__

```
>>> import urwid
>>> print(urwid.__version__)
2.6.14
>>>
```

If version is displayed, then Urwid is installed and available for use.

Note: Ensure Urwid 26.14 supported version is installed. Above command should reflect the same.

Step 3: Similarly import cx_Oracle and check version

import cx_Oracle (Press Enter)

cx_Oracle.version

```
>>> import cx_Oracle
>>> print(cx_Oracle.__version__)
8.3.0
```

If version is displayed, then cx_Oracle is installed and available for use.

Note: Ensure cx_Oracle 8.3.0 supported version is installed. Above command should reflect the same.

Python 3.11 for Linux Operating System :

Step 1: Execute below commands to install the python 3.11.0

dnf groupinstall 'development tools'

dnf install bzip2-devel expat-devel gdbm-devel ncurses-devel openssl-devel readline-devel wget sqlite-devel tk-devel xz-devel zlib-devel libffi-devel

wget https://www.python.org/ftp/python/3.11.0/Python-3.11.0.tgz

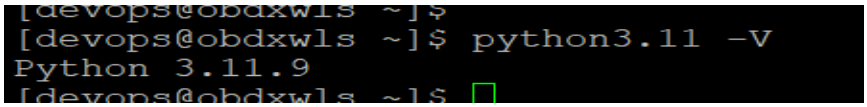
tar -xzvf Python-3.11.0.tgz

cd Python-3.11.0

./configure --enable-optimizations

make altinstall

python3.11 –version



Step2: Once above steps are executed successfully install the following required modules.

pip3.8 install --upgrade pip

pip3.8 install cx-Oracle==8.3

pip3.8 install urwid==2.6.14

4. Installation

Pre-Installation

- Install all the prerequisite software and packages mentioned above

Steps of Installation

- Download and extract the installer zip file (Base).
- Navigate to “<OBDX INSTALLER DIR>/core/config”
- Open the “installer.properties” file to maintain key configurations for BASE ENTITY (OBDX_BU)

```
##############################################################################################
# Installer Properties                                                                       #
#                                                                                            #
# All entries to be made immediately after the '=' and WITHOUT quotation marks. i.e. '' or ""  #
#                                                                                            #
##############################################################################################

#####################
#                   #
# Weblogic Details  #
#                   #
#####################

#Middleware home path. Example /home/obdxuser/Oracle/Middleware/Oracle_Home - where you have sub-directories like
wlserver,oracle_common etc.
MIDDLEWARE_HOME=/scratch/app/product/Oracle/Middleware/Oracle_Home

#JAVA home path. Example /home/obdxuser/jdk18 - where you have sub-directories like bin,jre,lib etc.
JAVA_HOME=/scratch/app/java

#Below two variables have been deprecated.###############

#DB_WITH_FLYWAY_EXECUTION YES OR NO. Example if we want to execution of db with cx_Oracle then value will be NO.
#DB_WITH_FLYWAY_EXECUTION=YES

#GRADLE home path. Example /home/obdxuser/flyway - where you have sub-directories like bin,jre,lib etc.
#FLYWAY_HOME=/scratch/obdx/flyway-7.9.2
###########################################################

#DB_EXECUTION_TYPE FLYWAY OR CX_ORACLE.
DB_EXECUTION_TYPE=FLYWAY

#DB_EXECUTION_TYPE_HOME home path of flyway - where you have sub-directories like bin,jre,lib etc.
DB_EXECUTION_TYPE_HOME=/scratch/obdx/flyway-7.9.2
```

IMPORTANT:

- Enter the values right after the **“=”**sign
- DO NOT change anything to the left of the **“=”**
- DO NOT change any of the flag values or pre-filled values (such as **WLS_JDBC_DIGX_NAME**, **WLS_JDBC_DIGX_JNDI**, Flag values etc) available in **“Factory Shipped”** section.
- Ensure there is no blank space after **“=”** sign, except specific flavor specific configuration.
- Throughout this document consider UBS as UBS core banking with OBPM as payments engine.

Only below parameters should be set in installer.properties file.

Component	Parameter	Description	Example
DB details (for OBDX schema)	**OBDX_DATABASE_HOST NAME**	Enter the hostname of the database server which would host the database schema for OBDX	abc.xyc.com
	OBDX_DATABASE_PORT	Enter the port number of the database listener	1521
	OBDX_DATABASE_SID	Enter the Oracle **Service Name** for database instance	obdxdb.in.oracle .com
	OBDX_DATABASE_SYS_U SER	Enter the username with 'sys' privileges	Sys
	POST_FIX	For OBDX schema name like "OBDX_DEV" POST FIX is 'DEV'. **SHOULD BE IN UPPERCASE ONLY.**	DEV
	OBDX_DBA_DIRECTORY_ NAME	Enter the directory name in which you want the OBDX schema tablespace datafile to be created. Enter Logical name (i.e. **DIRECTORY_NAME** column) from **DBA_DIRECTORIES** table NOT the physical path.	OBDX_DIR
	OBDX_AUDIT_DBA_DIREC TORY_NAME	Enter the directory name in which you want the OBDX AUDIT tablespace datafile to be created. Enter Logical name (i.e. **DIRECTORY_NAME** column) from **DBA_DIRECTORIES** table NOT the physical path.	OBDX_AUDIT_ DIR

Component	Parameter	Description	Example
EHMS DB details (to be configured only in-case of FLAVOR as UBS,FCORE &OBPM)	**EHMS_DATABASE_HOSTNAME**	Enter the hostname for EHMS database server	abc.xyz.com
	EHMS_DATABASE_PORT	Enter the port number of EHMS database listener	1521
	EHMS_SCHEMA_NAME	Enter the Complete OBDX-EXT (B1A1) HostInterfaceschema name you want installer to create as new schema. **SHOULD BE IN UPPERCASE ONLY.**	EHMS182SCHEMA
	EHMS_DBA_DIRECTORY_NAME	Enter the directory name in which you want the OBDX-EXT (B1A1) schema tablespace datafile to be created. Enter Logical name (i.e. **DIRECTORY_NAME** column) from **DBA_DIRECTORIES** table NOT the physical path.	OPATCH_LOG_DIR
	EHMS_DATABASE_SYS_USER	Enter the username with 'sys' privileges	Sys
	EHMS_DATABASE_SID	Enter the EHMS database Service Name	obdxehms.in.oracle.com
	EHMS_HOST_SCHEMA_NAME	Enter the EXISTING EHMS HOST schema name	OBDXUBS
	EHMS_CCY(to be configured for UBS and OBPM HOST only)	Enter the Country code for EHMS HOME Branch	GB
	EHMS_HB (to be configured for UBS and OBPM HOST only)	Enter the Branch code for code for EHMS HOME Branch	AT3
	EHMS_FCORE_FCUBS_SCHEMA_NAME (to be configured for FCORE HOST only)	FCORE-FCUBS schema name	FCRUBSHOST

Component	Parameter	Description	Example
Weblogic server details	**MIDDLEWARE_HOME**	Oracle Weblogic Server home path. Example /home/obdxuser/Oracle/Middleware/Oracle_Home - where you have sub-directories like wlserver,oracle_common etc.	/home/obdxuser/Oracle/Middleware/Oracle_Home
	JAVA_HOME	Path where JAVA (JDK) is installed	/home/obdxuser/jdk17.0.10
	DB_EXECUTION_TYPE_HOME	Path where FLYWAY is installed	/scratch/obdx/flyway-7.9.2
	DB_EXECUTION_TYPE	Database execution type	FLYWAY
	GRADLE_HOME	Path where GRADLE is installed	/home/obdxuser/gradle-7.9
	INSTALLATION_HOME	Path where OBDX is to be installed. All configuration files will be copied as a sub-directory “config” under this directory. **DO NOT KEEP INSTALLATION_HOME AS MiddlewareHome**.	/home/obdxuser/obdx
	WLS_DOMAIN_PATH	Path where OBDX Weblogic domain should be created. Users can now enter custom path as per their requirements.	/home/obdxuser/domains
	WLS_CLUSTER_NAME	Name of cluster; this cluster would have one single managed server.	obdx_cluster
	WLS_CLUSTER_NODE_HOSTNAME	Host name or IP address of managed server participating in the cluster. **Currently only single node is supported.**	abc.xyz.com
	WLS_ADMIN_SERVER_PORT	Weblogic AdminServer port. It is the port to access the administration console of the Weblogic server. Generally port 7001 is used as the AdminServer port. **Custom port are supported.**	7001
	WLS_ADMIN_SERVER_SSL_PORT	AdminServer SSL port. It is the port used to securely access (https) the administration console of the Weblogic server.	7002

Component	Parameter	Description	Example
	WLS_NODE_PORT	Node Manager Port. It is the port used by Node Manager to be configured for OBDX domain. Generally, 5556 is utilized as Node Manager Port. **Custom ports are supported.**	5556
	WLS_MS_SERVER_NAME	Managed server name. This will be the name of the managed server created in the cluster followed by indexes. eg- **If this is set as 'clip' managed servers would be clip1**.	Clip
	WLS_MS_SERVER_PORT	Managed Server Port. Managed server will utilize this port for hosting OBDX components and associated resources. **Custom ports are supported.**	9001
	WLS_DOMAIN_NAME	Enter Weblogic Domain name.	obdx_domain1
	WLS_DOMAIN_ADMIN_USER	Domain user ID. The user id will be used to access the Weblogic Administration console.	Weblogic
	WLS_NODE_TYPE	Weblogic Node Manager type	Plain/SSL
	WLS_MACHINE_NAME	Weblogic Node Manager machine name	obdx_machine
	APP_ROOT_DIR	Any empty directory path	/scratch/app/dir
	WLS_JMS_FILEUPLOAD_PS (to be configured for all OBDX supported HOST)	Set the paths for the persistent store of the FileUpload JMS modules. **DO NOT KEEP path as INSTALLATION_HOME or as sub directory inside INSTALLATION_HOME.**	/scratch/obdx/FileUpload
	WLS_JMS_AUDIT_PS (to be configured for all OBDX supported HOST)	Set the paths for the persistent store of the Audit JMS modules. **DO NOT KEEP path as INSTALLATION_HOME or as sub directory inside INSTALLATION_HOME.**	/scratch/obdx/Audit

Component	Parameter	Description	Example
	WLS_JMS_REPORT_PS (to be configured for all OBDX supported HOST)	Set the paths for the persistent store of the Reports JMS modules. **DO NOT KEEP path as INSTALLATION_HOME or as sub directory inside INSTALLATION_HOME.**	/scratch/obdx/Reports
	WLS_JMS_JPA_PS (to be configured for all OBDX supported HOST)	Set the paths for the persistent store of the JPA JMS modules. **DO NOT KEEP path as INSTALLATION_HOME or as sub directory inside INSTALLATION_HOME.**	/scratch/obdx/JPA
	WLS_JMS_EXTSYSRECEIVER_PS **(to be configured for all OBDX supported HOST)**	Set the paths for the persistent store of the ExtSystemReceiver JMS modules. **DO NOT KEEP path as INSTALLATION_HOME or as sub directory inside INSTALLATION_HOME.**	/scratch/obdx/Receiver
	WLS_JMS_EXTSYSSENDER_PS **(to be configured for all OBDX supported HOST)**	Set the paths for the persistent store of the ExtSystemSender JMS modules. **DO NOT KEEP path as INSTALLATION_HOME or as sub directory inside INSTALLATION_HOME.**	/scratch/obdx/Sender
OBDX Application Administrator user details	**OBDX_ADMIN_USERNAME**	Set username for OBDX application Admin user. **USERNAME IS CASE SENSITIVE. In-case of OUD as provider username should be the User ID mentioned during user creation steps mentioned in pre-requisite document (refer To create User and mapping it to the Group section)**	superadmin
	OBDX_ADMIN_EMAIL	Enter the Email ID for OBDX application admin user.	superadmin@oracle.com
	OBDX_ADMIN_CONTACT_NO	Enter the mobile number for OBDX application admin user. **COUNTRY CODE IS MUST.**	+911234567890

Note: Apart from above any other property values should not be modified

Ensure ORACLE_HOME, JAVA_HOME variable are set and their binaries are available in PATH variable before proceeding.

Login with OS user which was used to perform OBDX pre-requisite software installation (or has ownership on Oracle Weblogic home directory)

Ensure OBDX Installation home and filestore path's maintained in installer.properties exists and user running the installer has read-write permissions.

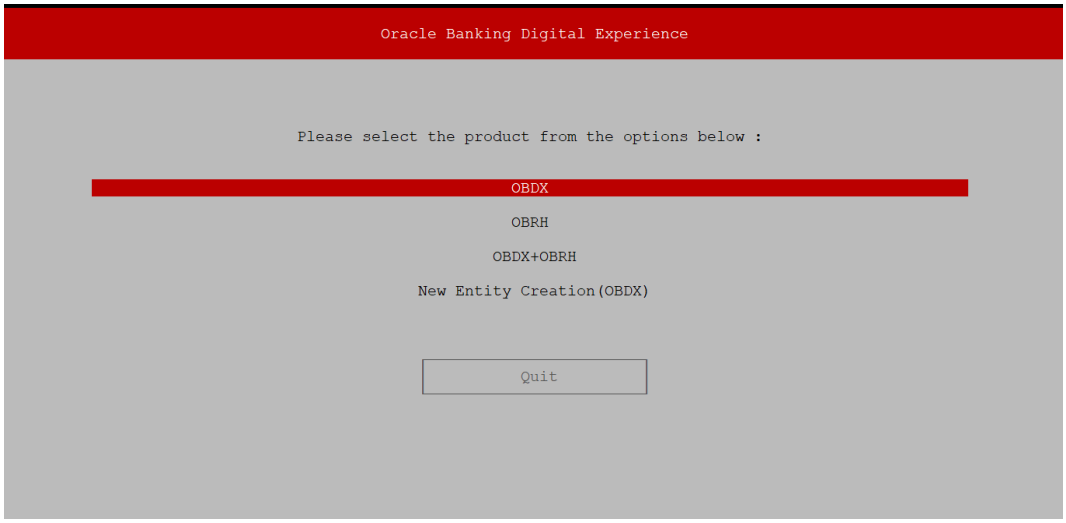
Installation Steps:

1. From your terminal navigate to <OBDX INSTALLER DIR>/

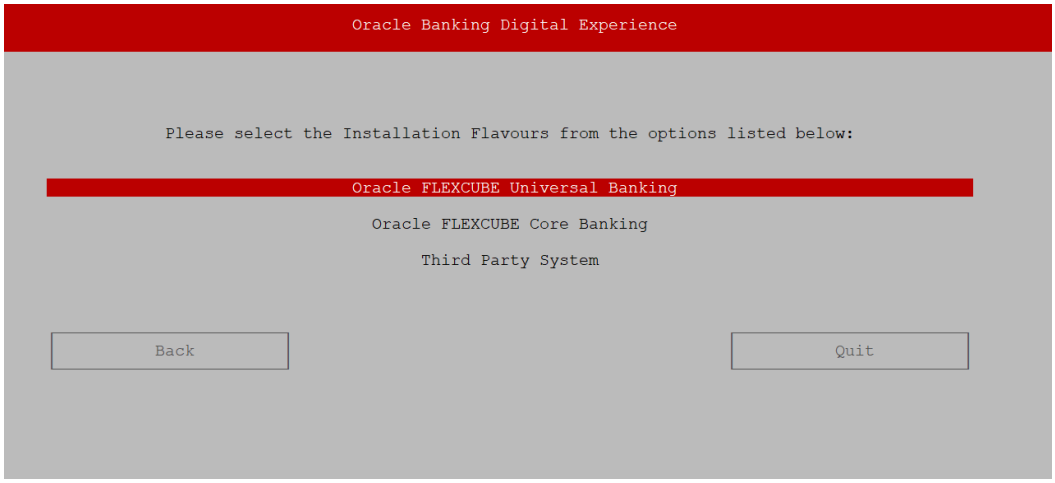
```
[devops@obdxwls OBDX_Installer]$ ls -lrt
total 365100
-rwxr-xr-x  1 devops devops      2107 Mar  3 04:52 setEnvUBS.sh
-rwxrwxrwx  1 devops devops     14476 Mar 17 05:08 runInstaller.py
drwxrwxrwx  3 devops devops        18 Mar 17 05:08 installables
-rwxrwxrwx  1 devops devops         0 Mar 17 05:08 __init__.py
drwxrwxrwx  6 devops devops        88 Mar 17 10:09 framework
-rw-r--r--  1 devops devops         0 Mar 17 10:13 digx-dashboard.log
-rw-r--r--  1 devops devops      4527 Mar 17 10:14 digx-entitlement-feed-data.log
-rw-r--r--  1 devops devops         0 Mar 17 10:14 digx-feed-data-task.log
-rw-r--r--  1 devops devops 373824180 Mar 20 05:35 obrh_adhoc_9_6_0.zip
drwxrwxrwx  5 devops devops      4096 Mar 20 08:54 obrh_adhoc
drwxr-x---  3 devops devops        18 Mar 20 09:04 obrh
drwxrwxrwx 14 devops devops      4096 Mar 24 09:48 ExecInstances
drwxrwxrwx  6 devops devops        79 Mar 24 09:52 core
[devops@obdxwls OBDX_Installer]$
[devops@obdxwls OBDX_Installer]$
[devops@obdxwls OBDX_Installer]$ python3.11 runInstaller.py
```

2. Enter the following command

python3.11 runInstaller.py

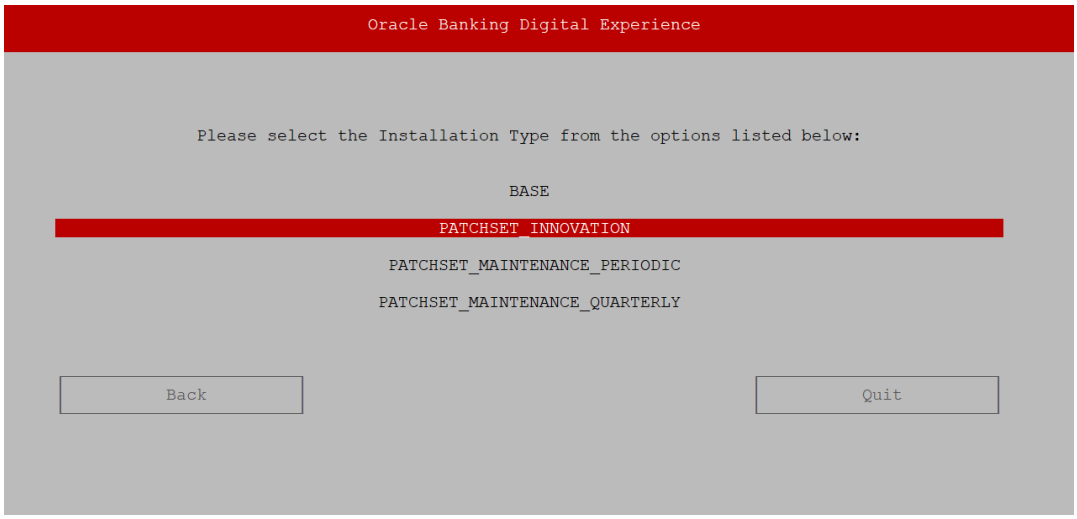


3. Select the appropriate product for Installation.

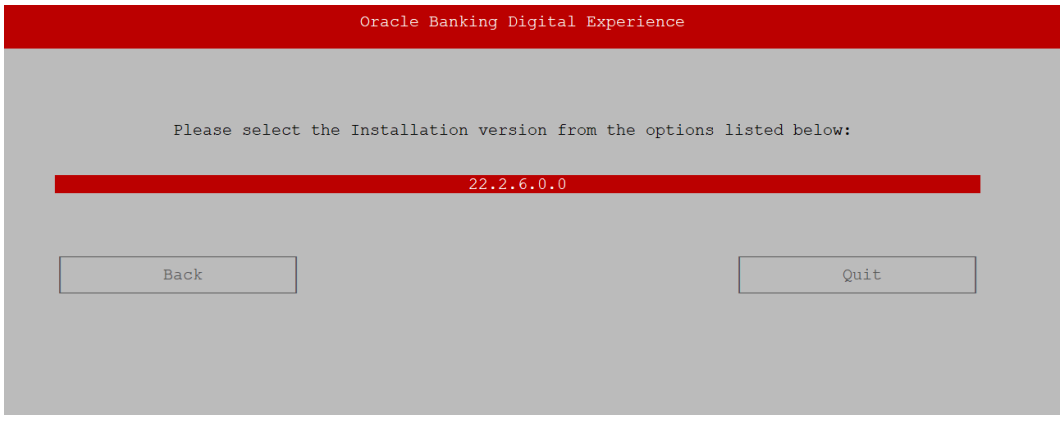


Oracle FLEXCUBE Universal Banking (OBDX with UBS)

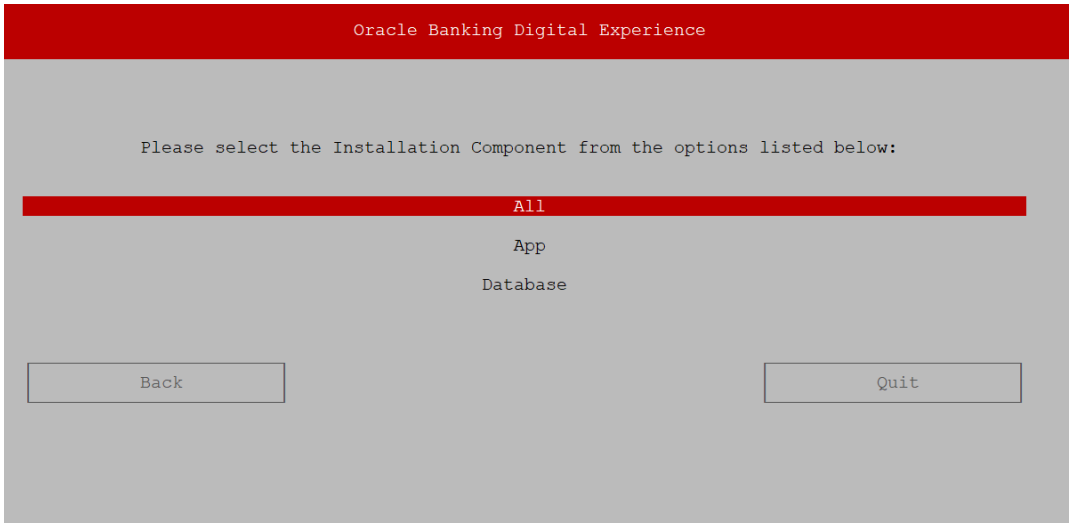
1. Select Installation type for the particular release



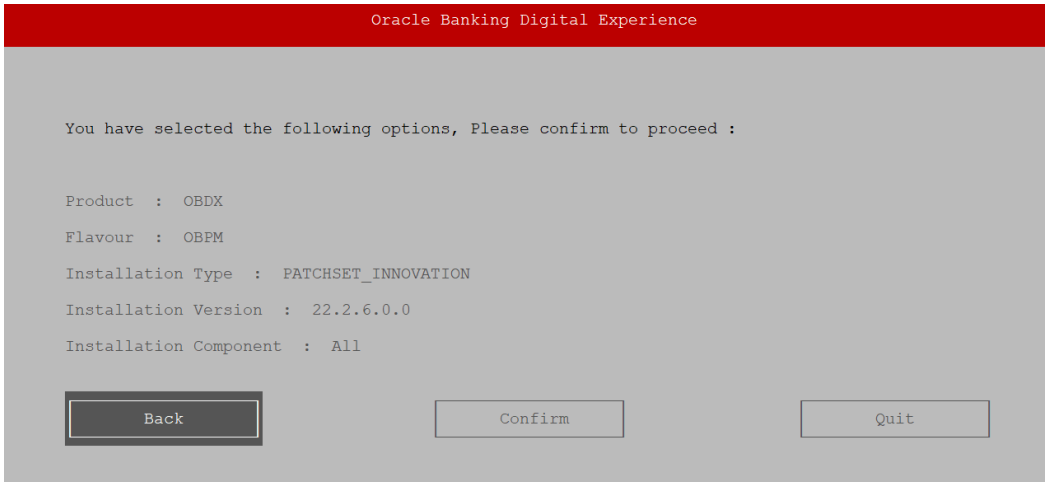
2. Select Installation version.



3. Select Installation component All (Database+App), App(Only application), Database(Only Database).



4. Click **Confirm** to proceed.
   OR
   Click **Back** to return to previous page.
   OR
   Click **Quit** to exit from installer run.



5. Post confirmation below screen will appear to take end-user input.

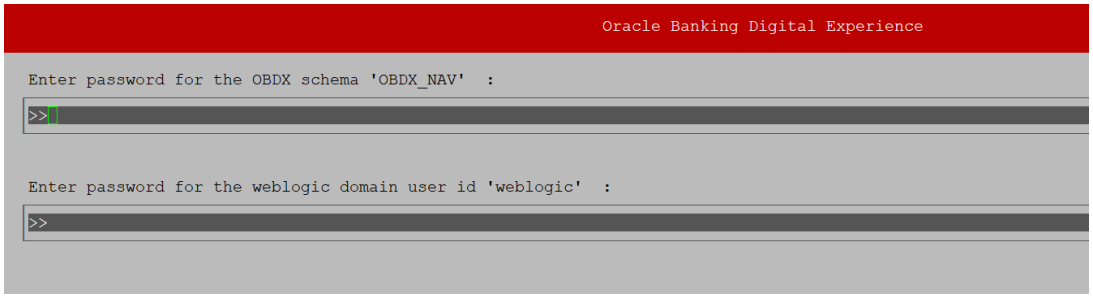
Enter below passwords:

- SYS privilege user password where OBDX schema would be created
- OBDX schema password
- Weblogic console administrator user password

- SYS privilege user password where UBS host schema exists
- Existing UBS HOST schema password
- New OBDX EHMS schema password
- Password for OBDX application administrative user (In-case of OUD as provider, password should be similar to one used while user creation in OUD (or User Password field))

Third Party System (OBDX with THP)

1. Post Third Party System selection, enter the required credentials details



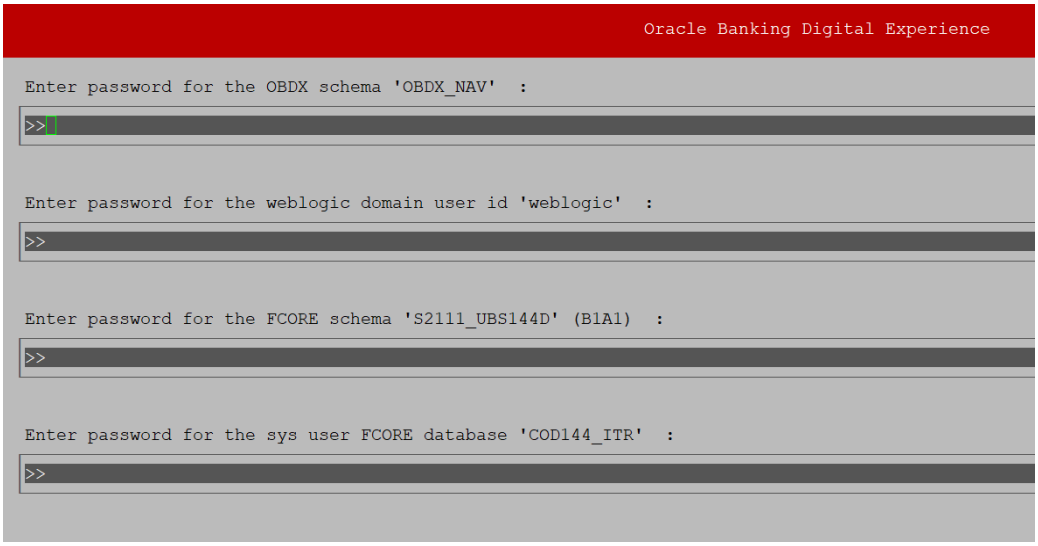
Enter below passwords:

- SYS privilege user password where OBDX schema would be created
- OBDX schema password
- Weblogic console administrator user password

  OBDX application admin user password

Oracle FLEXCUBE Core Banking (OBDX with FCORE)

1. Post Oracle FLEXCUBE Core Banking, enter the required credentials details



Enter below passwords:

- SYS privilege user password where OBDX schema would be created
- OBDX schema password

- Weblogic console administrator user password
- SYS privilege user password where FCORE host schema exists
- New OBDX EHMS schema password
- Password for OBDX application administrative user

Oracle FLEXCUBE Universal Banking with Oracle Banking Payments (OBDX with OBPM)

1. Enter the required credentials details

Enter below passwords:

- SYS privilege user password where OBDX schema would be created
- OBDX schema password
- Weblogic console administrator user password
- SYS privilege user password where OBPM host schema exists
- Existing OBPM HOST schema password
- New OBDX EHMS schema password
- Password for OBDX application administrative user

Installation Status

After selecting the mode and entering all required details, the status is displayed (as shown below) on the terminal to indicate the progress of the installation.

1. If DB_EXECUTION_TYPE set to **CX_ORACLE**

```
  >>>> STARTING OBDX PRODUCT INSTALLATION <<<<


 <<<<< Please check the logs file available at ExecInstances/09May1657/logs/app for any error  >>>>>>>


We are excuting the db with DB_WITH_FLYWAY_EXECUTION=NO
Starting OBDX Database Installation with OBPM FLAVOR
Database Path: /u02/app/oracle/oradata/OFCDB009_bomlcq/OFCDB009_BOM1CQ/B2169F489B0C1E32E053C305F40A9E33/datafile
Database Path: /u02/app/oracle/oradata/OFCDB009_bomlcq/OFCDB009_BOM1CQ/B2169F489B0C1E32E053C305F40A9E33/datafile
Creating Tablespace...
Tablespace Created
Creating User...
User Created
Creating Role...
Role Created
Executing Grants...
Execution of clip_master_script_main.sql started
Execution of clip_master_script_main.sql completed
Execution of clip_constraints_main.sql started
Execution of clip_constraints_main.sql completed
Execution of clip_seeds_executable_main.sql started
Execution of clip_seeds_executable_main.sql completed
SUCCESSFULLY installed OBDX database

Starting OBPM Database Installation...
Database Path: /scratch/app/oradata/ORA19C
Creating Tablespace...
Tablespace Created
Creating User...
User Created
Creating Role...
Roles Created
Executing Grants...
Executing OBPM  Grants...
Execution of table-scripts_main.sql started
Execution of table-scripts_main.sql completed
Execution of ubs_object_scripts_main.sql started
Execution of ubs_object_scripts_main.sql completed
Execution of obpm_object_scripts_main.sql started
Execution of obpm_object_scripts_main.sql completed
Execution of execute-seeds_main.sql started
```

2. If DB_EXECUTION_TYPE set to **FLYWAY**

```
[devops@obdxwls OBDX_Installer]$ python3.8 runInstaller.py
['BASE', 'OBPM', 'New', 'MODULE']

  >>>> STARTING OBDX PRODUCT INSTALLATION <<<<


 <<<<< Please check the logs file available at ExecInstances/09May1817/logs/app for any error  >>>>>>>


We are excuting the db with DB_WITH_FLYWAY_EXECUTION=YES
Starting OBDX Database Installation with OBPM FLAVOR
Database Path: /u02/app/oracle/oradata/OFCDB009_bomlcq/OFCDB009_BOM1CQ/B2169F489B0C1E32E053C305F40A9E33/datafile
Database Path: /u02/app/oracle/oradata/OFCDB009_bomlcq/OFCDB009_BOM1CQ/B2169F489B0C1E32E053C305F40A9E33/datafile
Creating Tablespace...
Tablespace Created
Creating User...
User Created
Creating Role...
Role Created
Executing Grants...
OBDX Scripts execution on progess...Please hold on it might take sometime
OBDX Scripts execution Successfully
SUCCESSFULLY installed OBDX database

Starting OBPM Database Installation...
Database Path: /scratch/app/oradata/ORA19C
Creating Tablespace...
Tablespace Created
Creating User...
User Created
Creating Role...
Roles Created
Executing Grants...
Executing OBPM  Grants...
OBPM Scripts execution on progess...Please hold on it might take sometime
```

```
Database Path: /u02/app/oracle/oradata/OFCDB009_bomlcq/OFCDB009_BOMLCQ/B2169F489B0C1E32E053C305F40A9E33/datafile
Database Path: /u02/app/oracle/oradata/OFCDB009_bomlcq/OFCDB009_BOMLCQ/B2169F489B0C1E32E053C305F40A9E33/datafile
Creating Tablespace...
Tablespace Created
Creating User...
User Created
Creating Role...
Role Created
Executing Grants...
Execution of clip_master_script_main.sql started
Execution of clip_master_script_main.sql completed
Execution of clip_constraints_main.sql started
Execution of clip_constraints_main.sql completed
Execution of clip_seeds_executable_main.sql started
Execution of clip_seeds_executable_main.sql completed
SUCCESSFULLY installed OBDX database

Starting OBPM Database Installation...
Database Path: /scratch/app/oradata/ORA19C
Creating Tablespace...
Tablespace Created
Creating User...
User Created
Creating Role...
Roles Created
Executing Grants...
Executing OBPM  Grants...
Execution of table-scripts_main.sql started
Execution of table-scripts_main.sql completed
Execution of ubs_object_scripts_main.sql started
Execution of ubs_object_scripts_main.sql completed
Execution of obpm_object_scripts_main.sql started
Execution of obpm_object_scripts_main.sql completed
Execution of execute-seeds_main.sql started
Execution of execute-seeds_main.sql completed
Execution of obpm-seeds_main.sql started
Execution of obpm-seeds_main.sql completed
SUCCESSFULLY installed OBPM database
Executed DIGX_FW_CONFIG_ALL_O.sql successfully
Executed DIGX_FW_ABOUT_OBPM.sql successfully
Executed DIGX_FW_CONFIG_VAR_B.sql successfully
Executed DIGX_FW_CONFIG_UBS_ALL_O.sql successfully
Policy seeding execution proccessing ...
```

3. When the installation completes, the below message is displayed.

```
Starting WEBLOGIC Setup and Configuration...
Weblogic Domain Created Successfully
Generating 2,048 bit DSA key pair and self-signed certificate (SHA256withDSA) with a validity of 9,999 days
        for: CN=Developer, OU=Department, O=Company, L=City, ST=State, C=CA
[Storing /scratch/app/domains/obdx_mod_domain/authserver.keystore]

Warning:
The JCEKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -src
keystore /scratch/app/domains/obdx_mod_domain/authserver.keystore -destkeystore /scratch/app/domains/obdx_mod_domain/authserver.keystore -deststoretype pkcs
2".
Starting Datasource Creation...
Datasource created Successfully
Starting JMS Creation...
JMS created Successfully
Starting Deployemt Creation...
Deployment created Successfully

Successfully Setup and Configured WEBLOGIC...

  >>>> OBDX PRODUCT INSTALLATION COMPLETED SUCCESSFULLY <<<<


[devops@obdxwls OBDX_Installer]$
```

5. Installation using Silent Mode

This chapter describes how to run the OBDX installer in silent mode.

What is silent-mode installation?

During installation in silent mode, the installation program reads the details for your configuration parameters (flavor; mode; passwords etc) from the environment variables (same session in which installer is executed) and installer.properties that you set before beginning the installation. The installation program does not display any configuration options during the installation process.

Below values to be exported before running installer in silent mode.

export PRODUCT=" "

export FLAVOUR=" "

export INSTALLER_VERSION=""

export Installation_Type=" "

export COMPONENT=""

export DB_SYS_PASSWORD=""

export SCHEMA_PASS=""

export DomainPassword=""

export DBAuthPassword=""

export EHMS_SCHEMA_PASS=""

export EHMS_HOST_SCHEMA_NAME_PASS=" "

export ENTITY_EHMS_DATABASE_HOSTNAME=" "

export ENTITY_EHMS_DATABASE_PORT=""

export ENTITY_EHMS_DATABASE_SID=" "

export ENTITY_EHMS_DATABASE_SYS_USER=" "

export ENTITY_EHMS_DATABASE_SYS_PASS=" "

export ENTITY_SCHEMA_NAME=””

export ENTITY_SCHEMA_PASS=" "

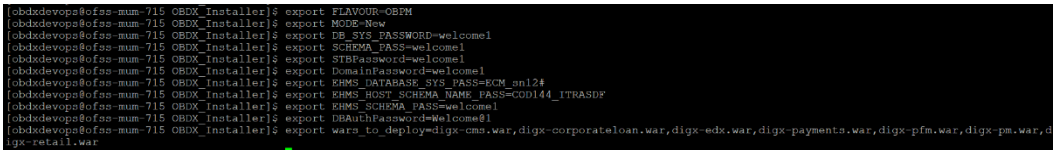
export ENTITY_EHMS_HOST_SCHEMA_NAME=" "

export ENTITY_EHMS_HOST_SCHEMA_NAME_PASS=" "

Steps for Silent-Mode Installation

- Download and extract the installer zip file (Base – non localization version).
- Navigate to “<OBDX INSTALLER DIR>/core/config”
- Open the “installer.properties” file to maintain key configurations for BASE ENTITY (OBDX_BU)

  **Refer to page 9 to 14 (step 4) for filling up installer.properties.
- Set the environment variables , as shown below



Below parameters should be set as environment variables, depending on the Host system the installer should be executed.

Host	Parameter	Description	Example
Environment variables to set for flavor:UBSFCORE	**FLAVOUR**	Flavour for installation **UBS** for Oracle FLEXCUBE Universal Banking 146.0.0.0 (OBDX with UBS) **FCORE** for Oracle FLEXCUBE Core Banking 11.8.0.0.0 (OBDX with FCORE)	**export FLAVOUR=OBPM** **or** **export FLAVOUR=FCORE**
	MODE	Mode of installation. ‘**New**’ in-case of a fresh installation of OBDX for the first run on server ‘**Clean**’ in-case of an existing OBDX installation that you want to overwrite OR in case of a previously failed installation or re-installation	**export MODE=New** **or** **export MODE=Clean**
	DB_SYS_PASSWORD	Sys user password of OBDX database (Existing)	**export DB_SYS_PASSWORD=obdx182sys**

Host	Parameter	Description	Example
	SCHEMA_PASS	Password for new schema on OBDX database	**export SCHEMA_PASS=obdx#182**
	DomainPassword	Password for Weblogic Administrator console	**export DomainPassword=wlsadmn**
	EHMS_DATABASE_SYS_PASS	Sys user password of EHMS HOST database (Existing)	**export EHMS_DATABASE_SYS_PASS=obdxehmssys**
	EHMS_HOST_SCHEMA_NAME_PASS **** Only required for UBS & OBPM Host. Ignore this parameter in-case of FCORE Host**	Password of existing EHMS HOST schema (Existing)	**export EHMS_HOST_SCHEMA_NAME_PASS =obdxehmshost**
	EHMS_SCHEMA_PASS	Password for new OBDX EHMS schema on EHMS HOST database	**export EHMS_SCHEMA_PASS= obdx182ehms**
	wars_to_deploy	Mention the optional wars to be deployed	**export wars_to_deploy=digx-cms.war,digx-corporateloan.war,digx-payments.war**
	DBAuthPassword	Password for new OBDX Administrator user of OBDX application (In-case of OUD as provider, password should similar to one used while user creation in OUD(or User Password field))	**export DBAuthPassword=obdxadmn**
	FLAVOUR	Flavour for installation **'OBDX'** for Third Party System 1.0 (OBDX with THP)	**export FLAVOUR=OBDX**
	Mode	Mode of installation.	

Host	Parameter	Description	Example
Environment variables to set for flavor: **OBDX (Third-party HOST)**		'**New**' in-case of a fresh installation of OBDX for the first run on server '**Clean**' in-case of an existing OBDX installation that you want to overwrite OR in case of a previously failed installation or re-installation	**export MODE=New** **or** **export MODE=Clean**
	DB_SYS_PASSWORD	Sys user password of OBDX database (Existing)	**export DB_SYS_PASSWORD=obdx182sys**
	SCHEMA_PASS	Password for new schema on OBDX database	**export SCHEMA_PASS=obdx#182**
	DomainPassword	Password for Weblogic Administrator console	**export DomainPassword=wlsadmn**
	wars_to_deploy	Mention the optional wars to deployed	**export wars_to_deploy=digx-cms.war,digx-corporateloan.war,digx-payments.war**
	DBAuthPassword	Password for new OBDX Administrator user of OBDX application (In-case of OUD as provider, password should similar to one used while user creation in OUD(or User Password field))	**export DBAuthPassword=obdxadmn**

Run the runInstaller.py file with '--silent ' argument along with '--base' option

```
[devops@obdxwls OBDX_Installer]$
[devops@obdxwls OBDX_Installer]$ python3.11 runInstaller.py --silent
```

Installation Status

The status is displayed on the terminal to indicate the progress of the installation.

If DB_EXECUTION_TYPE set to CX_ORACLE

```
  >>>> STARTING OBDX PRODUCT INSTALLATION <<<<


 <<<<< Please check the logs file available at ExecInstances/12May0626/logs/app for any error  >>>>>>>


We are excuting the db with DB_WITH_FLYWAY_EXECUTION=NO
Starting OBDX Database Installation with OBPM FLAVOR
Database Path: /u02/app/oracle/oradata/OFCDB009_bom1cq/OFCDB009_BOM1CQ/B2169F489B0C1E32E053C305F40A9E33/datafile
Database Path: /u02/app/oracle/oradata/OFCDB009_bom1cq/OFCDB009_BOM1CQ/B2169F489B0C1E32E053C305F40A9E33/datafile
Creating Tablespace...
Tablespace Created
Creating User...
User Created
Creating Role...
Role Created
Executing Grants...
Execution of clip_master_script_main.sql started
Execution of clip_master_script_main.sql completed
Execution of clip_constraints_main.sql started
Execution of clip_constraints_main.sql completed
Execution of clip_seeds_executable_main.sql started
Execution of clip_seeds_executable_main.sql completed
SUCCESSFULLY installed OBDX database

Starting OBPM Database Installation...
Database Path: /scratch/app/oradata/ORA19C
Creating Tablespace...
Tablespace Created
Creating User...
User Created
Creating Role...
Roles Created
Executing Grants...
Executing OBPM  Grants...
Execution of table-scripts_main.sql started
Execution of table-scripts_main.sql completed
Execution of ubs_object_scripts_main.sql started
Execution of ubs_object_scripts_main.sql completed
Execution of obpm_object_scripts_main.sql started
Execution of obpm_object_scripts_main.sql completed
Execution of execute-seeds_main.sql started
```

If DB_EXECUTION_TYPE set to **FLYWAY**

```
Password validated for sys
Password validated for sys
Password validated for COD144_ITR

  >>>> STARTING OBDX PRODUCT INSTALLATION <<<<


 <<<<< Please check the logs file available at ExecInstances/12May0721/logs/app for any error  >>>>>>>


We are excuting the db with DB_WITH_FLYWAY_EXECUTION=YES
Starting OBDX Database Installation with OBPM FLAVOR
Database Path: /u02/app/oracle/oradata/OFCDB009_bom1cq/OFCDB009_BOM1CQ/B2169F489B0C1E32E053C305F40A9E33/datafile
Database Path: /u02/app/oracle/oradata/OFCDB009_bom1cq/OFCDB009_BOM1CQ/B2169F489B0C1E32E053C305F40A9E33/datafile
Creating Tablespace...
Tablespace Created
Creating User...
User Created
Creating Role...
Role Created
Executing Grants...
OBDX Scripts execution on progess...Please hold on it might take sometime
OBDX Scripts execution Successfully
SUCCESSFULLY installed OBDX database

Starting OBPM Database Installation...
Database Path: /scratch/app/oradata/ORA19C
Creating Tablespace...
Tablespace Created
Creating User...
User Created
Creating Role...
Roles Created
Executing Grants...
Executing OBPM  Grants...
OBPM Scripts execution on progess...Please hold on it might take sometime
```

When the installation completes, the below message is displayed

```
Gradle Build Created Successfully
Starting Weblogic Domain Creation...

Starting WEBLOGIC Setup and Configuration...
Weblogic Domain Created Successfully
Generating 2,048 bit DSA key pair and self-signed certificate (SHA256withDSA) with a validity of 9,999 days
        for: CN=Developer, OU=Department, O=Company, L=City, ST=State, C=CA
[Storing /home/devops/domain/OBDX211TEST11/authserver.keystore]

Warning:
The JCEKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -src
keystore /home/devops/domain/OBDX211TEST11/authserver.keystore -destkeystore /home/devops/domain/OBDX211TEST11/authserver.keystore -deststoretype pkcs12".
Starting Datasource Creation...
Datasource created Successfully
Starting JMS Creation...
JMS created Successfully
Starting Deployemt Creation...
Deployment created Successfully

Successfully Setup and Configured WEBLOGIC...

  >>>> OBDX PRODUCT INSTALLATION COMPLETED SUCCESSFULLY <<<<


[devops@obdxwls OBDX_Installer]$
```

6. Steps to Create Credential Mapping

Credential Store Mapping

The OBDX system utilizes external integrations to facilitate seamless communication with various services. To establish these connections, credentials are required to authenticate and authorize access. These credentials are not hardcoded but rather initialized post-installation. They are subsequently encrypted and stored within the database, ensuring confidentiality and integrity. This subsequent section outlines the procedures and guidelines for configuring and managing these credentials within the OBDX environment.

To utilize the credential mapping functionality, retrieve the com.ofss.digx.CredentialsStore.jar file from the designated location:

OBDX_Installer/installables/OBDX/BASE/25.1.0.0.0/utils/tools

Running the Credential Mapping Application

Execute the application using the following command:

```
java -jar com.ofss.digx.CredentialsStore.jar <csv_file> <DataBaseCredentials> <DataSeedFlag>
```

Command Parameters:

1. <csv_file>

   Provide the path to your CSV file containing user credentials by replacing <csv_file> with the actual file location.

   CSV File Format Requirements

   The CSV file must adhere to the following structure:

   - Contain exactly three columns: type, username, and password
   - Include a header row with column names: type,username,password
   - Subsequent rows should contain individual credential entries, with each row representing a distinct set of credentials
   - Ensure that the value in the type column is unique for each credential entry

   Example CSV File

type	username	password
MERCHANT	OBDX	PASSWORD111

2. <DataBaseCredentials>

   Specify the <DataBaseCredentials> parameter as a comma-delimited string comprising the following components:

   - Database username

- Password
- JDBC URL (in the format jdbc:oracle:thin:@host:port/service_id)

The expected format for <DataBaseCredentials> is: username,password,jdbc_url.

Example: User,Password123,jdbc:oracle:thin:@host:port/service_id

Ensure accurate input of these values to establish a successful connection to the database.

3. <DataSeedFlag>

To control the seeding of data into the digx_fw_credentials table, set the <DataSeedFlag> parameter to 'Y' to populate the table with the generated credentials. Alternatively, specify 'N' to simply display the credentials without persisting them to the database.

Example command to run this

java -jar com.ofss.digx.CredentialsStore.jar data.csv
DB_USER,DB_PASSWORD,jdbc:oracle:thin:@//HOST:PORT/SERVICE_ID Y

Upon executing this utility, you will obtain an encrypted password, which can then be utilized in conjunction with other credentials. Subsequently, these credentials will be populated into the database.

Extensibility:

To leverage custom credentials inserted into the system, utilize the following code snippet:

```
ICredentialStore store =
CredentialStoreFactory.getCredentials(CredentialStoreKeys.CREDENTIAL_IPMLEMENTATION);

Credential credentials = store.getCredentials(<custom_type>);
```

Replace <custom_type> with the desired type associated with the custom credentials.

Import:

Import the jar implementation

```
"com.ofss.digx.infra:com.ofss.digx.infra.crypto.impl:$libs_digxVersion"
```

into your gradle project

To ensure proper configuration, verify that the entry in the digx_fw_config_all_b table has a prop_id of “credential_impl”, a category_id of “CredentialStore”, and a PROP_VALUE of “com.ofss.digx.infra.cred.DatabaseCredentialsStore”. Confirm that these values match exactly to guarantee correct functionality. If discrepancies are found, update the entry accordingly to reflect the specified values.

The AES key is no longer required to be explicitly inserted, as it is dynamically generated by the system when the utility is run and stored within the keystore located at DIGX_FW_KEYSTORE.

For any encryption operations that require the use of the AES key, utilize the SymmetricCryptographyProviderFactory class, which is available in the same JAR, instead of relying on the credential. This approach streamlines the encryption process and enhances overall security.

```
SymmetricCryptographyProviderFactory.getInstance().getLatestProvider().encrypt(data);

SymmetricCryptographyProviderFactory.getInstance().getLatestProvider().decrypt(data);
```

7. Installer Verification

Each execution creates a new directory as <DDMonthHHMM> under <OBDX INSTALLER DIR>/ExecInstances directory where installer execution logs as described below are stored.

Log Description	PATH
Summarized Installer Activity Log	<OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/obdx_installer.log
Summarized Database Logs	<OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/db/DB_installation.log
Detailed OBDX DB Logs per SQL file	<OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/db/OBDX/OBDX.log
Detailed EHMS schema Logs per SQL file (specific to EHMS host system only)	<OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/db/<EHMSHOST>/<EHMSHOST>.log <EHMSHOST> - values such as; FCORE; OBPM;
Weblogic Configuration Logs	<OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/app/app_debug.log <OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/app/domain.log <OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/app/datasource.log <OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/app/jms.log <OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/app/deployment.log
Detailed OBDX policy seeding logs	<OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/db/Entitlement.log <OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/db/Task.log <OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/db/Dashboard_seed.log Note: Check for SEVERE keyword; If found refer to Troubleshot section to re-run the policy
Policy seeding execution Log	<OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/db/seedPolicies.log

	Note: Should be empty if no errors during policy execution. In-case non-empty refer to Troubleshot section to re-run the policy

Check all the logs for any errors.

8. Installer Scope

OBDX Installer currently covers below activities:

Flavor: Third Party system (OBDX with THP)

Flavor	Activity	Detailed Activity List	New Installation	Reinstall
OBDX with THP	OBDX DB Setup	Create Tablespace	√	NA
		Create Schema and Role	√	√ (drop and re-create objects)
		Grants	√	√
		Load DB object (DDL's and DML's)	√	√
		Compile Schema	√	√
		Policy Seeding	√	√
	Weblogic Setup and Configuration	Create and Configure AdminServer, Machine, Managed Server and Cluster	√	√
		Configure NodeManager	√	√
		Configure JDBC	√	√
		JMS servers, Persistent stores and JMS Modules	√	√
		Application Deployment	√	√
		JTA	√	√
		Enable Production Mode	√	√
		Start AdminServer and NodeManager	√	√
	OBDX Configuration	Copy config files into OBDX Installation Home	√	√ (Delete old and copy new from installer zip)

Flavor: Oracle FLEXCUBE Core Banking (OBDX with FCORE)

Flavor	Activity	Detailed Activity List	New Installation	Reinstall
OBDX with FCORE	OBDX DB Setup	Create Tablespace	√	NA
		Create Schema and Role	√	√ (drop and re-create objects)
		Grants	√	√
		Load DB object (DDL's and DML's)	√	√
		Compile Schema	√	√
		Policy Seeding	√	√
	EHMS DB Setup	Create Tablespace	√	NA
		Create Schema and Role	√	√ (drop and re-create objects)
		Grants	√	√
		Load DB object (DDL's and DML's)	√	√
		Compile Schema	√	√
	Weblogic Setup and Configuration	Create and Configure AdminServer, Machine, Managed Server and Cluster	√	√
		Configure NodeManager	√	√
		Configure JDBC	√	√
		JMS servers, Persistent stores and JMS Modules	√	√
		Application Deployment	√	√
		JTA	√	√
		Enable Production Mode	√	√

Flavor	Activity	Detailed Activity List	New Installation	Reinstall
		Start AdminServer and NodeManager	√	√
	OBDX Configuration	Copy config files into OBDX Installation Home	√	√ (Delete old and copy new from installer zip)

Flavor: Oracle FLEXCUBE Universal Banking with Oracle Banking Payments (OBDX with OBPM)

Flavor	Activity	Detailed Activity List	New Installation	Reinstall
OBDX with OBPM (14.6.0.0.0 version)	OBDX DB Setup	Create Tablespace	√	NA
		Create Schema and Role	√	√ (drop and re-create objects)
		Grants	√	√
		Load DB object (DDL's and DML's)	√	√
		Execute OBPM HOST specific scripts	√	√
		Compile Schema	√	√
		Policy Seeding	√	√
	EHMS DB Setup	Create Tablespace	√	NA
		Create Schema and Role	√	√ (drop and re-create objects)
		Grants	√	√
		Load DB object (DDL's and DML's)	√	√
		Compile Schema	√	√
	Weblogic Setup and Configuration	Create and Configure AdminServer, Machine, Managed Server and Cluster	√	√
		Configure NodeManager	√	√

Flavor	Activity	Detailed Activity List	New Installation	Reinstall
		Configure JDBC	√	√
		JMS servers, Persistent stores and JMS Modules	√	√
		Application Deployment	√	√
		JTA	√	√
		Enable Production Mode	√	√
		Start AdminServer and NodeManager	√	√
	OBDX Configuration	Copy config files into OBDX Installation Home	√	√ (Delete old and copy new from installer zip)

9. Post Installation Steps

For more information, refer the **Oracle Banking Digital Experience Connector Credential Store Guide.pdf**

Functionality / Module	OutBound Connection Pool Name
VAM	ra/DIGXConnectorOBVAM

Fileupload with UBS

Refer below document for File upload configuration with UBS

- **Oracle Banking Digital Experience File Upload Report Configuration**

Origination with OBO

Refer below document (section 5 and 6) for enabling Origination with OBO

- **Oracle Banking Digital Experience OBO Mid-Office and Third Party Setup and Configuration Guide**

Trade Finance (LC and BG) with OBTFPM

Refer below document for enabling 'Letter Of Credit' issuance and 'Bank Guarantee' issuance with Oracle Banking Trade Finance Management.

Oracle Banking Mid-Office Product Setup and Configuration Guide

OHS

OHS server needs to be configured for all FLAVOR's as a mandatory activity.

To configure OHS server follow steps mentioned in below document before proceeding further.

- **Oracle Banking Digital Experience OHS User Interface Configuration**

Feedback module:

In order to enable Scale (Rating) icons please refer the section **Creating Procedure** of **Oracle Banking Digital Experience Content Upload Guide** user manual.

9.1 Enabling Kafka in OBDX

9.1.1 Overview

OBDX now supports Apache Kafka as a messaging system in addition to JMS. Kafka provides high throughput, scalability, and fault tolerance, making it an excellent choice for event-driven architectures. OBDX will work with either JMS or Kafka but not both simultaneously. This section explains how to enable Kafka.

9.1.2 Enable Kafka

- JMS is the default messaging system in OBDX.
- **Enable Kafka only if you want to switch the existing message system to Kafka** by executing the following configuration queries in the database. If not enabled, the system will continue using JMS.

Configuration Queries:

UPDATE DIGX_FW_CONFIG_ALL_B SET prop_value = 'KAFKA' WHERE prop_id = 'MESSAGE_BROKER_TYPE';

UPDATE DIGX_FW_CONFIG_ALL_B SET prop_value = '{HOSTIP}:{HOSTPORT}' WHERE prop_id = 'bootstrap.servers';

Replace {HOSTIP}:{HOSTPORT} with the IP address and port of the Kafka broker running the Kafka service.

Note: Ensure that Kafka is **properly installed** before making this change.

If you want to migrate the existing messaging system to Kafka, you need to implement Kafka consumer and producer equivalents for all the customized JMS queues and topics. Also, If there are any new customized Kafka topics to be created, the corresponding producer and consumer implementations must be developed. Refer to **Section: Messaging System Integration for OBDX** in the document **"Oracle Banking Digital Experience Extensibility Guide"** for the detailed steps and guidelines in implementing Kafka and JMS.

9.1.3 OBDX Pre-defined External Kafka Topic Configurations

OBDX listens to the following external Topics of OBVAM(Oracle Banking Virtual Account Management), OBLM(Oracle Banking Liquidity Management) and OBO(Oracle Banking Origination).

- lm.accountnotification
- vam.virtualaccountnotification
- vam.accountstatusnotification
- vac.accountfacilitynotification
- externalSystemAlertMessage

For these Topics, the following properties need to be updated in the table **DIGX_FW_CONFIG_ALL_B**. Details should be obtained from the respective HOST system. If the HOST does not support the SSL property, it should be removed.

PROP_ID	CATEGORY_ID	PROP_VALUE
<topic_name>@ssl.truststore.password	KAFKA_CONFIG	<truststore_password>
<topic_name>@sasl.jaas.config	KAFKA_CONFIG	org.apache.kafka.common.security.scram.ScramLoginModule required username="<jaas_username>" password="<jaas_password>";
<topic_name>@bootstrap.servers	KAFKA_CONFIG	<host_ip>:<host_port>
<topic_name>@ssl.truststore.location	KAFKA_CONFIG	<truststore_location>
<topic_name>@security.protocol	KAFKA_CONFIG	<security_protocol>
<topic_name>@sasl.mechanism	KAFKA_CONFIG	<sasl_mechanism>

10. OBDX Logging Configuration

Logging Level Configuration with SLF4J & Logback in Weblogic

Logging at package and class levels can be externalized/customized by maintaining a common logback file outside the application for all the wars. This file will be configured as a server start argument.

1. Use the attached sample reference file and copy it to any physical path. (For example, /scratch/obdx/domains/obdx_domain/logbackOverride.xml)
   Sample code :

logbackOverride.xml

```
<configuration scan="true" scanPeriod="10 minutes">


   <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">

      <!-- encoders are assigned the type

         ch.qos.logback.classic.encoder.PatternLayoutEncoder by default -->

      <encoder>

         <pattern>%date{dd MMM yyyy;HH:mm:ss.SSS} [%thread] %X{ecid} %-5level %logger{100}[%X{FILE_IDENTIFIER} %X{FILE_REF_ID}] - %msg%n</pattern>

      </encoder>

   </appender>

         <!--

         <logger name="com.ofss.digx.app.sms.service.user.login" level="info"/>

   <logger name="com.ofss.digx.app.sms.service.user.User" level="debug"/>

   -->

   <root level="ERROR">

      <appender-ref ref="STDOUT" />

   </root>


</configuration>
```

2. Configure the same above path in server start arguments as follows.
   -Dlogback.configurationFile=/scratch/obdx/domains/obdx_domain/logbackOverride.xml

Enable package and class level logging :

If you want to change the logging level of a particular class or a package, you can do so by adding the following snippet in the external logback file and taking managed server restart. (Refer to the sample file)

i. To configure package logging level:
<logger name="com.ofss.digx.app.sms.service.user.login" level="info"/>

ii. To configure class logging level :
<logger name="com.ofss.digx.app.sms.service.user.User" level="debug"/>

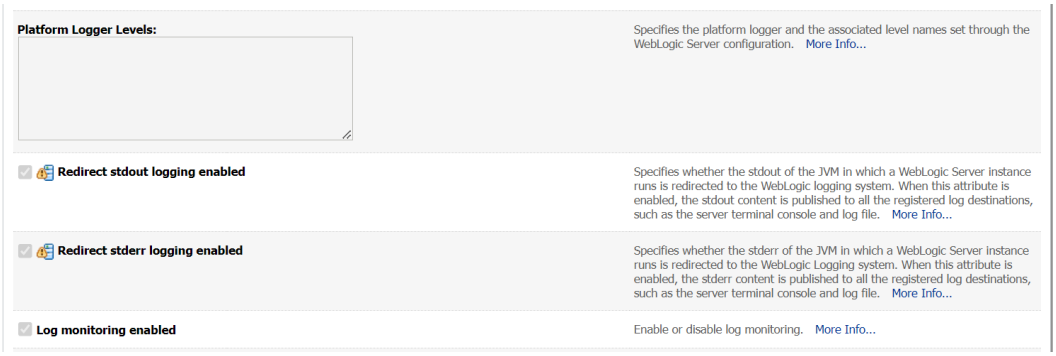
Note: In order to get the changes reflected without server restart, you can add a "scan" attribute to the <configuration> element in the external logback file. By default, the configuration file will be scanned for changes once every minute. To configure your desired scan period, add the attribute "scanPeriod" with value in milliseconds, seconds, minutes, or hours.
For example,
<configuration scan="true" scanPeriod="2 minutes">
This will scan for the configuration file every 2 minutes for any changes.

Redirecting stdout and stderr logs into a log file :

To redirect standard out and error logs to a log file, please follow the below steps.

Login to Weblogic console → Take Lock & Edit session → Go to Servers inside Environment menu →

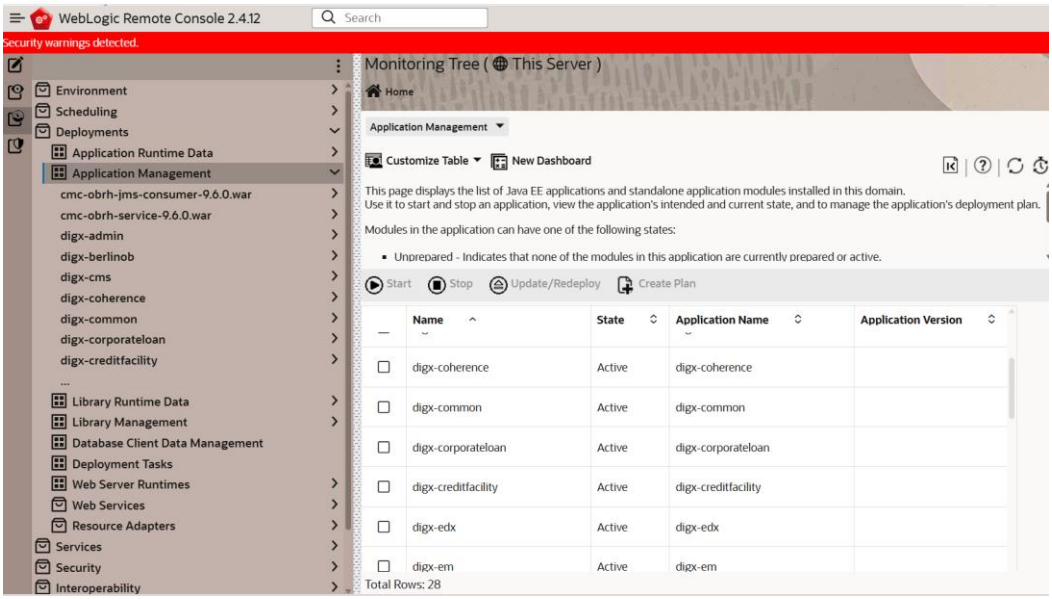
Click on the managed server → Go to Logging tab → Advanced → Check the boxes “Redirect stdout logging enabled” and “Redirect stderr logging enabled” as shown below.

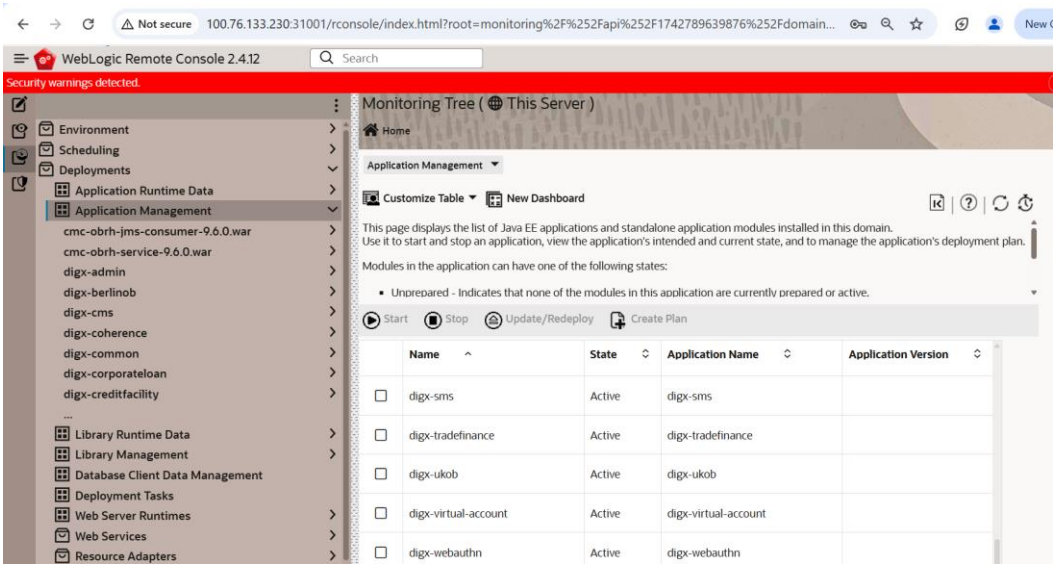


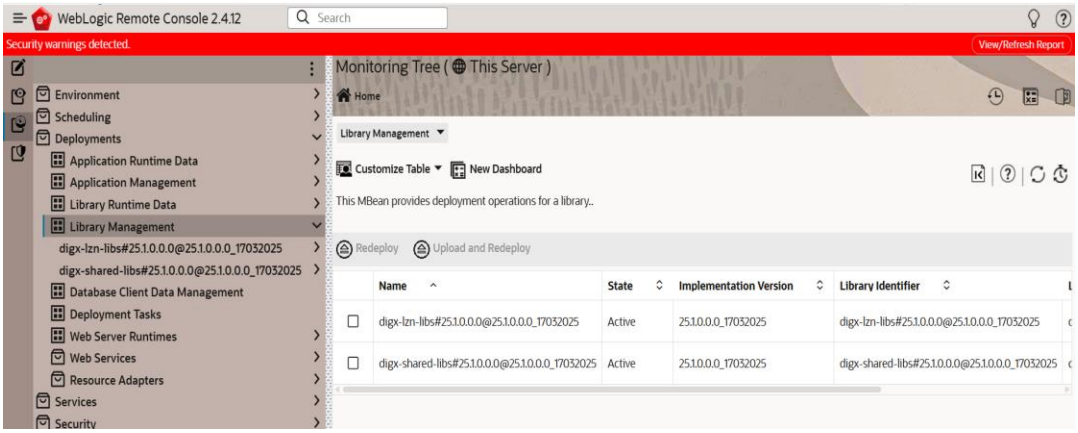
11. OBDX Product Verification

Start managed server and verify all deployed applications are in Active state (as shown below).

Domain wise deployment wars status



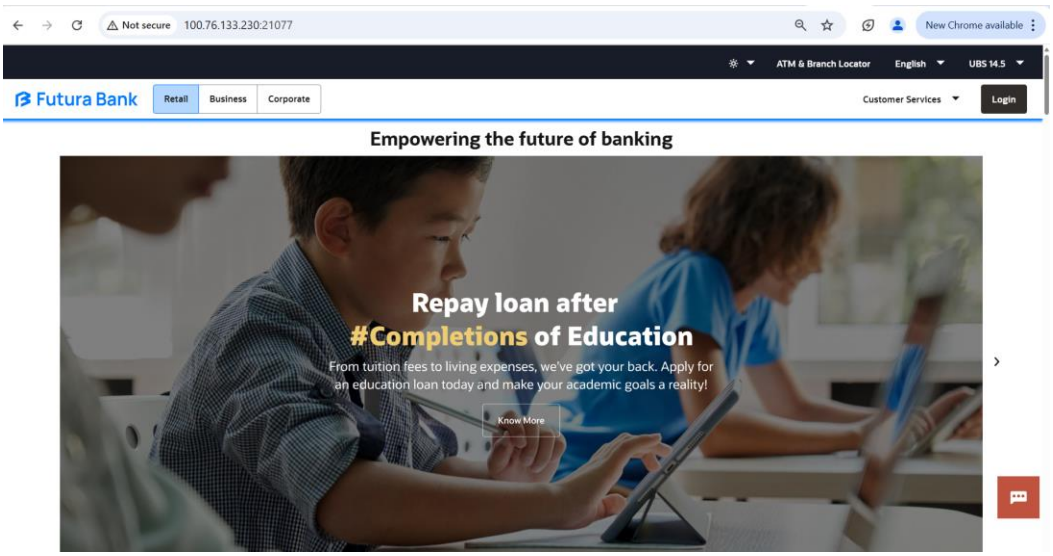


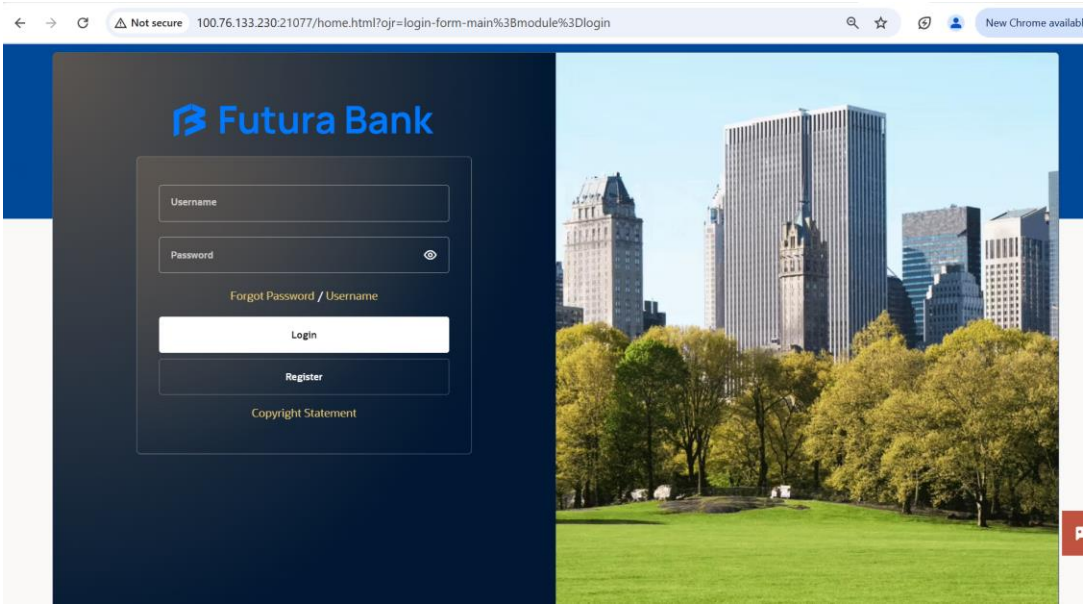


To verify the installation, launch below URL

http://<OHS server ip or hostname>:<OHS port>

Check if the page loads successfully.





Day1 Configuration

Universal Banking Solution (OBDX with UBS)

Refer below document (Section 3. System Configuration) for Day1 configuration required for integration with UBS

Oracle Banking Digital Experience System Configuration

Once day1 is completed, application is available for end-user transactions.

Note: Post Day1 restart of Managed server is mandatory

Third Party System (OBDX with THP)

Refer below document (Section 5. System Configuration – Host System as Third Party) for Day1 configuration required for integration with Third-party System

Oracle Banking Digital Experience System Configuration

Once day1 is completed, application is available for end-user transactions.

Note: Post Day1 restart of Managed server is mandatory

Chat Bot Configuration:

Refer below document for Chat Bot configuration.

Oracle Banking Digital Experience Chatbot Configuration

Mobile Application Builder:

Refer below documents for Mobile Applications build and setup.

Oracle Banking Digital Experience Mobile Application Builder-Android

Oracle Banking Digital Experience Mobile Application Builder-iOS

Mid Office Configuration:

Refer below document for Mid Office Configurations i.e. Trade Finance, Corporate Lending.

Oracle Banking Mid-Office Product Setup and Configuration Guide.

Account Uniqueness Configuration:

Some core banking systems support same account number in multiple branches within the entity. OBDX has support for such core banking systems. However, the configuration is not enabled by default. In case the Bank has core banking system which supports and provides same account numbers across multiple branches, the following scripts should be executed per entity for enabling the support.

```
Insert into DIGX_FW_CONFIG_ALL_O (PROP_ID, PREFERENCE_NAME, PROP_VALUE,
DETERMINANT_VALUE, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATED_DATE)
values
('obdx.host.account.uniqueness','ExtSystemsConfig','BRANCH','<ENTITY_ID>','ofssuser',sysd
ate,'ofssuser',sysdate);
```

```
Insert into DIGX_FW_CONFIG_ALL_O (PROP_ID, PREFERENCE_NAME, PROP_VALUE,
DETERMINANT_VALUE, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATED_DATE)
values
('obdx.host.accountbranch.delimiter','ExtSystemsConfig','@~','<ENTITY_ID>','ofssuser',sys
date,'ofssuser',sysdate);
```

Note: Ensure that <ENTITY_ID> has been replaced with correct Entity ID for the corresponding entity.

12. OBDX Product Security

Refer below document for OBDX product security configuration:

Oracle Banking Digital Experience Security Guide

13. OBDX Product – Best Practice

13.1 Tablespace for AUDIT INDEX

The index's used by AUDIT table should be moved into new tablespace from current AUDIT tablespace.

Follow below steps

- Create a new tablespace
- Give quota to OBDX schema

  alter user <OBDX_SCHEMA> quota unlimited on <OBDX_AUDIT_INDEX_TABLESPACE>;

- Drop and create below index by mapping the newly created tablespace
  - OBDX_Installer\installables\db\OBDX\ddl\oracle\audit\IDX_DIGX_AL_API_AUDIT_LOGGING.sql
  - OBDX_Installer\installables\db\OBDX\ddl\oracle\audit\IDX_DIGX_AL_API_AUDIT_LOG_HIST.sql
  - OBDX_Installer\installables\db\OBDX\ddl\oracle\audit\IDX_DIGX_AL_AUDIT_LOGGING.sql
  - OBDX_Installer\installables\db\OBDX\ddl\oracle\audit\IDX_DIGX_AL_AUDIT_LOGGING_1.sql
  - OBDX_Installer\installables\db\OBDX\ddl\oracle\audit\IDX_DIGX_AL_AUDIT_LOGGING_2.sql
  - OBDX_Installer\installables\db\OBDX\ddl\oracle\audit\IDX_DIGX_AL_AUDIT_LOGGING_3.sql
  - OBDX_Installer\installables\db\OBDX\ddl\oracle\audit\IDX_DIGX_AL_AUDIT_LOGGING_4.sql
  - OBDX_Installer\installables\db\OBDX\ddl\oracle\audit\IDX_DIGX_AL_AUDIT_LOGGING_DETAILS.sql

14. JPA and OBDX multi-cluster

In a multi-cluster environment, below JPA related changes should be implemented

- Go to Weblogic server
- Open config\META-INF\persistence.xml
- Append below configuration for all data-source

  <property name="eclipselink.cache.coordination.jms.host" value="t3://<WEBLOGIC-HOST-NAME OR IP>:<MANAGED-SERVER-PORT>/" />

  Replace with respective hostname or IP and Port no (this should be the managed server port number which hosts the JPA queues in the cluster)

Key pointers;

- Multi-cluster here refer's to :
  - Single cluster with multiple nodes (2 or more physical servers hosting the OBDX product)
  - 2 or more Weblogic cluster's
- Ensure these (persistence.xml) changes are available to all Managed server by maintaining appropriate classpath

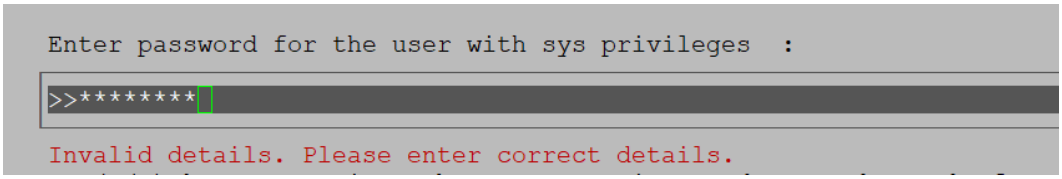
15. Troubleshoot Overview

This section describes how to troubleshoot OBDX setup.

Invalid database password

This topic contains troubleshooting information if you receive an error when attempting to connect to the database server.

If you get the following error:



Try one of the following:

- Verify that the database is running.
- Check Network connectivity between Weblogic Server and Database server.
- Check the database configuration in installer.properties file
- Verify that the entered password is correct.

cx_oracle module

This topic contains troubleshooting information about problems with cx_Oracle python module.

If you get the following error:

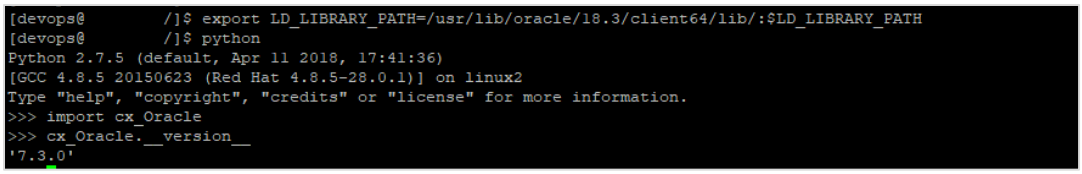
Execute the below command:

export LD_LIBRARY_PATH=/usr/lib/oracle/19.10/client64/lib:$LD_LIBRARY_PATH

python

import cx_Orace

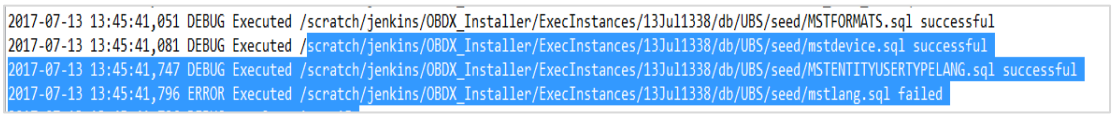
cx_Oracle.__version__



Failed Database Scripts

This topic contains troubleshooting information in case of database script failures.

If you get the following error in DB_installation.log:

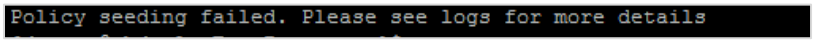


Check the detailed log of the failed SQL file at <OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM> /logs/db folder.

Failure of Policy Seeding

This topic contains troubleshooting information if policy seeding fails during installation.

If you get the following error:



Try one of the following:

Check if Entitlement.log is created on following path <OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM>/logs/db/ and contains any SEVERE errors for Entitlement policy seeding.

Check if Task.log is created on following path <OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM>/logs/db/ and contains any SEVERE errors for Task policy seeding.

Check if Dashboard_seed.log is created on following path <OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM>/logs/db/ and contains any SEVERE errors for Dashboard policy seeding.

- Check the seedPolicies.log in <OBDX INSTALLER DIR>/ExecInstances/<DDMonthHHMM>/logs/db/ directory if it contains any runtime errors generated during execution of the policies Seeding in OBDX schema

Fix the problem by following below steps:

- Login to OBDX installer server
- Over-write the policies files (Day0Policy.csv; Entitlement.csv; Resources.csv and Task.csv) from OBDX Product zip into <OBDX INSTALLER DIR>/installables/policies directory
- Browse to <OBDX INSTALLER DIR>\ installables\policies
- Edit Entitlement_log4j.properties , Task_log4j.properties & Dashboard_seed_log4j.properties . Replace <logs_path> with directory where policy seeding logs will be generated

```
############################################################

# default file output is in user's home directory.
#java.util.logging.FileHandler.pattern = %h/java%u.log
java.util.logging.FileHandler.pattern = <logs_path>/Task.log
java.util.logging.FileHandler.limit = 50000
java.util.logging.FileHandler.count = 1
#java.util.logging.FileHandler.formatter = java.util.logging.XMLFormatter
java.util.logging.FileHandler.formatter = java.util.logging.SimpleFormatter
java.util.logging.SimpleFormatter.format= [%1$tc] %4$s: %2$s - %5$s %6$s%n


# Limit the message that are printed on the console to INFO and above.
java.util.logging.ConsoleHandler.level = OFF
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter
```

```
############################################################

# default file output is in user's home directory.
#java.util.logging.FileHandler.pattern = %h/java%u.log
java.util.logging.FileHandler.pattern = /scratch/Task.log
java.util.logging.FileHandler.limit = 50000
java.util.logging.FileHandler.count = 1
#java.util.logging.FileHandler.formatter = java.util.logging.XMLFormatter
java.util.logging.FileHandler.formatter = java.util.logging.SimpleFormatter
java.util.logging.SimpleFormatter.format= [%1$tc] %4$s: %2$s - %5$s %6$s%n
```

- Run below command manually if “SEVERE” error logs are found in Task.log

java -jar -Djava.util.logging.config.file='<logs.properties>’ com.ofss.digx.utils.feed.data.task.jar "Task.csv" "oracle.jdbc.OracleDriver,<OBDX Schema name>,<OBDX Schema password>,jdbc:oracle:thin:@<OBDX DB hostname or IP>:<OBDX DB listener port>/<OBDX Service Name>"

Example:

java -jar -Djava.util.logging.config.file='Task_log4j.properties' com.ofss.digx.utils.feed.data.task.jar 'Task.csv' "oracle.jdbc.OracleDriver,OBDX_THP181,Welcome#1,jdbc:oracle:thin:@ xx.xx.xx.xx:1521/OBDX"

- Run below command manually if "SEVERE" error logs are found in Entitlement.log

java -jar -Djava.util.logging.config.file='<logs.properties>' com.ofss.digx.utils.entitlement.feed.data.jar 'Resources.csv,Entitlement.csv,Day0Policy.csv' 'KERNEL' "oracle.jdbc.OracleDriver,<OBDX Schema name>,<OBDX Schema password>,jdbc:oracle:thin:@<OBDX DB hostname or IP>:<OBDX DB listener port>/<OBDX Service Name>"

Example.:

java -jar -Djava.util.logging.config.file='Entitlement_log4j.properties' com.ofss.digx.utils.entitlement.feed.data.jar 'Resources.csv,Entitlement.csv,Day0Policy.csv' 'KERNEL' "oracle.jdbc.OracleDriver,OBDX_THP201,Welcome#1,jdbc:oracle:thin:@ xx.xx.xx.xx:1521/OBDX"

Note: Remove the space between multiple csv's if there is any.

- Run below command manually if "SEVERE" error logs are found in Dashboard_seed.log

java -jar -Djava.util.logging.config.file='<logs.properties>' com.ofss.digx.utils.dashboard.jar '<path>/ dashboard_json' "oracle.jdbc.OracleDriver,<OBDX Schema name>,<OBDX Schema password>,jdbc:oracle:thin:@<OBDX DB hostname or IP>:<OBDX DB listener port>/<OBDX Service Name>"

Example.:

java -jar -Djava.util.logging.config.file= Dashboard_seed_log4j.properties' com.ofss.digx.utils.dashboard.jar '/installables/policies/dashboard_json' "oracle.jdbc.OracleDriver,OBDX_THP201,Welcome#1,jdbc:oracle:thin:@xx.xx.xx.xx:1521/OBDX"

- Post successfully execution, restart Managed server.