Oracle® FLEXCUBE Investor Servicing

Upgrade Toolkit User Guide

Release 14.8.0.0.0
G32207-02
April 2025

Oracle FLEXCUBE Investor Servicing Upgrade Toolkit User Guide, Release 14.8.0.0.0

G32207-02

Contents

5 Annexure

Preface

Oracle FLEXCUBE Investor Servicing is a comprehensive mutual funds automation software from Oracle® Financial Servicing Software Ltd.©.

You can use the system to achieve optimum automation of all your mutual fund investor servicing processes, as it provides guidelines for specific tasks, descriptions of various features and processes, and general information.

This topic contains the following sub-topics:

- Purpose
- Audience
- Documentation Accessibility
- Critical Patches
- Diversity and Inclusion
- Conventions
- Screenshot Disclaimer
- Acronyms and Abbreviations

Purpose

This manual is designed to help acquaint you with the installation of **Oracle FLEXCUBE Investor Servicing** application.

Audience

This manual is intended for the following User/User Roles:

Table 1 Users and Roles

Users	Roles
Implementation team	Implementation of Oracle FLEXCUBE Investor Servicing
Presales team	Install Oracle FLEXCUBE Investor Servicing for demo purpose
Bank personnel	Who installs Oracle FLEXCUBE Investor Servicing

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 customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

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.

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.

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.

Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes.

Acronyms and Abbreviations

The list of the acronyms and abbreviations used are as follows:

Table 2 Acronyms and Abbreviations

Abbreviation	Description
FCIS	Oracle FLEXCUBE Investor Servicing
OEM	Oracle Enterprise Manager

Table 2 (Cont.) Acronyms and Abbreviations

Abbreviation	Description
EMS	Electronic Messaging Service
EJB	Enterprise Java Bean
MDB	Message Driven Beans

1

Upgrade and Conversion Approach

This topic gives an outline of the upgrade approach and all the activities involved in the entire upgrade process.

This topic contains the following sub-topics.

- Approach to import Data to Target Schema
  This topic gives an outline of the approach to import data to target schema.
- Upgrade Process Summary
  This topic gives a summary of the version upgrade process.

1.1 Approach to import Data to Target Schema

This topic gives an outline of the approach to import data to target schema.

Prepare the staging area only with the target version schema in this approach. This is the recommended approach.

Approach - Data Import to Target Schema

The steps involved in this method are as follows:

1. Prepare the staging area with the target version schema.
2. Insert the E-data (factory shipped static data) alone in the target schema using the consolidated insert scripts. You can do this using the **Oracle® FLEXCUBE Investor Servicing** Installer.

   The basic setup step should not be done through the installer.
3. Disable the **Not null** and **Check constraints** in the target schema.
4. Perform data import from the export dump of the production data.
5. Apply the conversion scripts for the columns with the **Not null** and **Check constraints** in the target schema to populate them with proper business data.
6. Enable the constraints in the target schema.

   The target schema will now act as the gold copy for the customer to resume the verification/ production activities.

- Advantages
  This topic explains the advantages of upgrade approach.
- Disadvantages
  This topic explains the disadvantages of upgrade approach.

1.1.1 Advantages

This topic explains the advantages of upgrade approach.

The advantages of this approach are as follows:

- Only one staging area prepared for the target version schema.

- The production p-data (transaction data) is imported without loss of data as constraints are disabled during the import.

1.1.2 Disadvantages

This topic explains the disadvantages of upgrade approach.

The disadvantage of this approach is as follows:

While enabling the constraints which are disabled earlier during the process, there might be a few columns that violate the constraints. You need to manually handle this.

1.2 Upgrade Process Summary

This topic gives a summary of the version upgrade process.

Upgrade Process Summary

1. Complete the mock upgrade activity.

   The mock upgrade provides a safe platform for the actual production environment upgrade.

2. Mock upgrade involves the following steps:

   Refer to the topic *Upgrade Database*.

   a. Identify the source database schema.

      The source system should be at the stage of Transaction Input just marked on migration date.

   b. Set up the target database as per the installation manuals.

   c. Exit the installer immediately after loading the static data.

   d. Retrofit the customization changes, if any, into the target database.

   e. Apply the DB object changes, source file changes and static data changes.

   f. Take a full dump of the source schema.

   g. Disable all the triggers and selective constraints in the target schema.

   h. Import the table data alone into the target database from the full dump of source schema.

   i. Enable all the triggers and constraints in the target schema.

3. Complete the Cut-over upgrade activity.

   During this, you will upgrade the production environment database. (Refer to the topic *Cutover Upgrade*.)

2

Mock Upgrade

This topic discusses the prerequisites and guides you through the process of upgrading **Oracle FLEXCUBE Investor Servicing** from a lower version to higher version.

The upgrade involves the following two activities:

- Mock upgrade activity
- Cutover upgrade activity

The mock upgrade activity provides a safe platform for the actual production environment upgrade. Once the mock upgrade is completed, you will have a ready target database which is termed as the **Gold Copy** for setting up the upgraded production environment.

- Prerequisites
  This topic discusses the prerequisites for upgrading **Oracle FLEXCUBE Investor Servicing** from a lower version to higher version.
- Mock Upgrade Activity
  This topic provides systematic instructions on Mock Upgrade activity.

2.1 Prerequisites

This topic discusses the prerequisites for upgrading **Oracle FLEXCUBE Investor Servicing** from a lower version to higher version.

Following are the prerequisites for the upgrade activity:

1. Prepare a copy of the production system covering all components of the product for mock upgrade.
2. Set up the Oracle DB parameters as per the **FCIS** recommendations for the destination schema.

   It is ideal to follow the steps of installation of a new version.
3. Identify and list out the installed components of the production system.
4. List out new components available in the new version that the customer proposes to use.
5. Identify and list out the details missed out in the new version. Internally discuss and suggest the actions proposed to address them.
6. Update the customer about the proposed plan and get the customer’s concurrence.
7. Following are the other interfacing teams to be involved in discussions for qualification:
   a. **For qualifying with the new version:** Address all changes required for qualification with the new version.
   b. **For qualifying with the existing interfacing system:** Identify and address the new interfacing requirements for the interfacing system to remain intact.
8. You need to understand the database upgrade strategy proposed below:
   a. Identify and document the migration steps that are planned.
   b. Set up the utilities for data comparison and data migration, if any, in the migration area.

   c. Identify and document the verification strategy.
   d. Prepare the staging area for both source schema and target schema (staging area for source schema is required only if the strategy followed mandates it; otherwise it is not required).
   e. Identify the conversion scripts to be applied post-upgrade.
9. Prepare a plan with timeline considering all changes required for a smooth upgrade.

2.2 Mock Upgrade Activity

This topic provides systematic instructions on Mock Upgrade activity.

Mock Upgrade Activity

The mock upgrade activity provides a safe platform for the actual production environment upgrade. You need to prepare a test area where the mock activity can be carried out.

During mock activity, you need to perform **User Acceptance Testing (UAT)** for the new functionalities that are added in the higher versions. While performing the actual migration, you need to take the maintenances and parameterizations done in UAT to the production environment.

The target database after the mock upgrade serves as a **Gold Copy** for you to set up the upgraded production environment.

You can truncate the p-Data tables from the **Gold Copy** and re-import from the production area. In the time between starting mock run activity and starting the actual production upgrade activity, if any of the static data is changed, then you need to handle such data manually.

Mock upgrade involves the following steps:

1. Upgrade Oracle database to the new version.
2. Identify temp soft changes to and take appropriate action to setup target version with temp soft changes (temp soft changes refer to the customization changes and bug fixed on the source version).
3. Set up target version of **Oracle FLEXCUBE Investor Servicing**.
4. Perform module specific changes if any.
5. Set up interfaces and adapters.
6. After upgrade, test the target version along with all interfaces and adapters.
7. Get sign off on the production environment upgrade.

- Apply Temp Soft Changes
  This topic gives the instructions to apply Temp Soft changes.
- Set up Target Schema
  This topic gives the instructions to set up Target Schema.

2.2.1 Apply Temp Soft Changes

This topic gives the instructions to apply Temp Soft changes.

Temp soft changes refer to the customization changes and bug fixes that are applied on the source version of the application used by the customer. You need to identify the temp soft changes that should be applied in the target version.

1. **Source File Changes:** You can use DIFF tools to compare the base version of the source application and the version used by the customer.
2. **Static Data Changes:** Use utility/scripts (refer topics Annexure) for identifying the differences in the factory shipped data.

2.2.2 Set up Target Schema

This topic gives the instructions to set up Target Schema.

You need to set up the target schema. For the purpose of illustration, let us consider a schema by name **DESTSCHEMA**.

You can use the **Oracle FLEXCUBE Investor Servicing** Installer to set up the target schema.

Follow the steps given below:

1. Create the target version database using the target version Installer.

   Refer to the *Installation manual* of the required version for details on setting up database.
2. Load the static data using installer. Refer to the installation manual of the required version for details on loading static data.
3. Exit the installer immediately after loading the static data.

   The basic setup step should not be done through the installer. At this point all data structures will be in place and static data tables will have the data populated as of the target version. But all schema objects like the source packages, triggers, procedures, functions, constraints, indexes, views, sequences, etc. would be available as of the base Kernel version.
4. If there are any customization changes that needs to retro-fitted in the target version schema, you may compile them now. You can also make the related static data changes. While doing the TEMPSOFT changes, you need to take care of the following:
   a. If the source version had an additional column with data, you need to manually move the same as the import of data from production has already been done.
   b. Apply the additional static data onto the upgraded schema.
5. You need to create a dummy schema in the same oracle instance as that of target schema.

   The dummy schema will have the same name as that of the source schema (from which dump was exported). Provide necessary grants for import/export. This is necessary to connect and import data later on from the dump.

3

Upgrade Database

This topic explains the activities involved in database upgrade.

This topic contains the following sub-topics:

- Import Data
  This topic provides the instructions to import data involved in database upgrade.
- Deploy Front-End Application
  This topic provides the systematic instructions to deploy Front-End application.
- Impact on Existing External System Interfaces
  This topic provides information on impact on existing External System Interfaces.
- Verify Data after Database Upgrade
  This topic provides a detailed instructions to verify data after database upgrade.

3.1 Import Data

This topic provides the instructions to import data involved in database upgrade.

Data Import Activities

The activities involved in database upgrade are given in the table below:

Table 3-1 Activity details in database upgrade

Activity No.	Activity Details	Source/ Destination	Dependency
1	For illustration purpose, consider that the name of the source schema used by the customer is **SOURCESCHEMA**. This contains the production data of the bank and the complete set of DB objects including tables, constraints, index sequences, source packages, triggers, etc., Disable the running Oracle jobs, if any. Create a full schema dump using the **expdp** utility in the **SOURCESCHEMA**. Name the export dump file as SITE_FULL_DUMP.DMP. The parameter file Export_Site_FULL_Dump.par can be used for this export (Refer *Annexure*).	Source	No Dependency. You can do this while the target schema is setup.
2	Configure the TNS in source and destination database to create DB link.	Common	

Table 3-1 (Cont.) Activity details in database upgrade

Activity No.	Activity Details	Source/ Destination	Dependency
3	Run the schema difference utility (Refer topic). This utility lists out the schema differences for the Tables and Columns. Run Create_DB_Link.sql in the destination schema. It will prompt for the site schema name, password and database name. Upon providing the details, MIG_DB database link will be created connecting source schema. In case creating a DB link to the production schema is disallowed, a staging area can be created and the DB link can be created to point to the same. Run TableDiff_Source_Dest.sql utility to identify the table difference between the **SOURCESCHEMA** and **DESTSCHEMA**. Copy the results to an Excel file. Run Existing_Table_Column_Diff.sql to identify the Table Column difference between the **SOURCESCHEMA** and **DESTSCHEMA**. Copy the spooled result to Excel file. This Excel file will act as a reference point of the schema differences between source DB and target DB. This file has the column level information and details like whether null values are allowed or not. For all the **not null** columns that are newly introduced in the target version, you need to handle the data import with special consideration because the import for these tables will fail if the records are present in the **SOURCESCHEMA** for the same. Based on the column differences, generate the scripts to disable the constraints for the new not null columns in the **DESTSCHEMA**, Along with this, generate the scripts to disable all the triggers. Use the stub Constraint_Trigger_Disable_Script.sql (Refer *Annexure*) to generate the following scripts: • ALTER_TRIGGER_DISABLE.sql - This sql contains the scripts to disable all the triggers. • ALTER_CONSTRAINTS_DISABLE.sql - This sql contains the script to disable only the not null, unique constraints and check constraints for a column without default value. Execute the above two scripts before importing the table data from site dump to the **DESTSCHEMA**.	Destination	Activity 1 and Activity 2

Table 3-1 (Cont.) Activity details in database upgrade

Activity No.	Activity Details	Source/ Destination	Dependency
4	Note that we have already created a dummy schema with the same name as the source schema to facilitate IMPDP command, which is used in the below command. Import the table data from the site dump using the par file given below: • Data pump import command: IMPDP source_schema_name/ pwd@target_instance PARFILE=<parameter file name with path> The parameter file Import_P-M_data.par can be used to import P Data, M-Data and P-M Data into the **DESTSCHEMA** (Refer *Annexure*). The parameter file Import_EM_data.par can be used to import the E-M data into the **DESTSCHEMA** (Refer *Annexure*). Refer the *import log* to ensure that all the table data is imported without any error. If there is any failure in the import, you need to analyse and handle it manually. While comparing the **SOURCESCHEMA** and **DESTSCHEMA** the stub Drop_Sequence_Script.sql generates the drop script for the common sequences. The drop script file name will be DROP_SEQUENCES.sql. Execute this script to drop the common sequences from **DESTSCHEMA**. After dropping the sequence, import the sequences from SITE_FULL_DUMP.DMP using the import par file Import_Sequence.par.	Destination	Activity 1, Activity 2 and Activity 3
5	Enable all the triggers and constraints. You need to manually handle the errors encountered while enabling the triggers and constraints.	Destination	Activity 4
6	The target database is now ready. Carry out the post- import activities provided in the next section. Carry out the post upgrade verification activities. You need to preserve the scripts applied while carrying out these activities to use them again if required.	Destination	Activity 5

Post Import Activities

Once the data import is completed, you need to perform the following post import activities:

- Recompile invalid objects

Issues in Data Import using IMPDP Utility

You may encounter any the following issues while importing data using IMPDP utility.

Table 3-2 Issues in Data Import using IMPDP Utility

Issue	Problem	Cause	Resolution
Import options not recognized.	Some of the import options may not be enabled in the server. One such example is the DATA_OPTIONS clause of the import, which is used in the E-M Data import par file.	Oracle parameter setup.	DBA needs to enable the same.
Data Import fails because of new indexes.	If the value for a column is null in the imported data which is going to be part of an index in the target then the import fails.	The existing column would have been added as part of a newly created unique index in the **DESTSCHEMA**. So, if the data for this column contains null values then the uniqueness is violated.	Disable the index, do the import, supply values to this column.
Data Import fails due to long columns.	If a varchar2 column was changed to long column in the higher versions, then the import fails.	IMPDP does not support importing varchar2 columns into long columns. It is given in Oracle documentation that long columns are deprecated and not recommended to create tables using long data type. Instead CLOB to be used.	As a workaround, instead of IMPDP utility, use the imp utility to import the tables affected by this issue.

3.2 Deploy Front-End Application

This topic provides the systematic instructions to deploy Front-End application.

For deploying the front-end application, follow the steps below:

1. Refer to the *installation manual* of the required version of the application.
2. Apply the temp soft changes, if any.
3. Ensure that the deployed EAR points to the upgraded database.

3.3 Impact on Existing External System Interfaces

This topic provides information on impact on existing External System Interfaces.

If the customer has any external interfaces maintained in the source application, you need to follow the steps below:

1. Communicate any format level changes (Interface files, Gateway XSDs) in existing interfaces to the external systems.
2. Communicate the changes in queues configuration, file locations etc to the external systems.

3. Communicate the changes in the tag names of the XSD files which are shared with other systems to the respective external system owners.

3.4 Verify Data after Database Upgrade

This topic provides a detailed instructions to verify data after database upgrade.

Verify Data after Database Upgrade

Once the database is upgraded, you need to do the following verifications:

- System wide data verification of reports and other check points
- Interface testing to check the connectivity
- Converted deals testing
- New deals testing
- New product maintenance testing
- Signoff

These verifications are explained in detail under the following headings.

System wide Data Verification

This verification includes the following steps.

Generic Checks
Generic check includes the following:

- Check the main parameter table **cstb_param** and **paramstbl** for the parameter values.
- Menu organization is as per the static factory shipped data and handled from the import.
- The bank may need to modify their user roles. You need to take care of the change of user roles for the new modules. A script will introduce all such roles into ALLROLES.

EOD and Performance Testing
This verification includes the following:

- Configure the following as part of FCJ configuration:
  - Install Oracle Web cache which is present in the Sizing document.
  - Change the Internet Explorer setting as per DBA server sizing document (Recommended IE settings).
  - Ensure that onsite changes are not done to introduce **no-cache** in the code.
  - Configure realistic user roles against the usual ALLROLES.
- Launch basic screens of high volume module sand process them onsite.
- Record and review the EOD and EOM timelines before and after upgrade to check if there are any major variances.
- Check if all the programmes maintained in the EOD window is run without being skipped.
- Check if all aspects of the EOD, i.e. module functionality and reports generation are covered.
- Test on a masked dump of the site if it is done offshore.

Interface Testing to Check Connectivity

As part of this verification, you need to perform the following activities:

- Test incoming and outgoing interfaces
- Check for all the channels that receive information from Oracle FCIS

Converted Deals Testing

You need to test the converted deals as follows:

- As part of the upgrade, the system will have new tables as well as new columns in the existing tables. Populate the additional fields and tables by testing the converted/migrated data.
- Perform basic life cycle testing for the converted contracts.
- Check the product maintenances and static maintenances for modifications.

New Deals Testing

You need to test the new deals as follows:

- Create new transactions on existing funds and observe the validation of default values.
- Test the basic life cycle of new deals.

New Product Maintenance Testing

Once the upgrade is completed, create a new product/fund in each module.

Sign Off

Get the customer sign off to go ahead with the upgrade of production environment.

Gold Copy

Gold Copy - DB Schema Setup
Once the above activities are completed, you can use the **DESTSCHEMA** as the **Gold Copy** to set up the database during production environment upgrade.

Gold Copy - Front-End Setup and Interface
Use the latest available executables to set up the various components for production upgrade. All interface related changes available in various files need to be deployed.

4

Cutover Upgrade

This topic provides the systematic instructions on cutover upgrade activities.

Cutover Upgrade Activity

The upgrade activities that you need to carry out during cutover are as follows:

- Activities in production environment
- Database upgrade in production environment
- Installation of other components

These activities are explained in detail as follows:

Activities in Production Environment

On the cut-over date, the following activities required to be done at the production environment level:

- Run Pre-EOD and bring the system to pre-EOD completed stage of migration date.
- Ensure that there are no unauthorized transactions in any module.
- Switch off the SWIFT and other interfaces.
- Bring down the application server with due notification.
- Bring down the Gateway server.

Database Upgrade in Production Environment

In the Gold Copy, truncate the transaction data tables and re-import the same from the latest production data dump. You can use the script files Truncate-pData.sql and Export-Source-PData.par (Refer topic Annexure).

Ideally the transaction data that has gone into the production data from the time of starting mock run to till time would be the data level change in the source. You need to handle this. However, during the mock run and verification activities the transaction data (p-Data) might have undergone changes. So you must truncate the p-Data tables using the scripts in the Gold Copy.

You need to repeat the database upgrade activity performed during the mock upgrade.

Refer to the topics *Upgrade Database* and *Post Import Activities*.

You need not do any static data comparison at this point. The implementation team takes care not to do any static data changes in the production environment.

Installation of other Components

The **Gold Copy** should be used for setup in production environment for all applicable components and various files.

5
Annexure

This topic provides the details of utility scripts used in the upgrade process.

Utility Scripts
The utility scripts are given in the following table:

Table 5-1 Utility Scripts

Script Name	Location	Remarks
Export_Site_FULL_Dump.par	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Param files to export the production schema at site. In this par file, replace the word **SOURCESCHEMA** with the actual schema name to be exported. Also change the **DIRECTORY**, '**DUMPFILE**, **LOGFILE** names as per the actual names used. This is applicable to all the par files supplied in the document.
Create_DB_Link.sql	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Script to create database link. Before creating the DB link, configure the TNS connection between the source and destination database.
TableDiff_Source_Dest.sql	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Scripts to list out the differences between the source schema and target schema.
Existing_Table_Column_Diff.sql	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Script to list out the differences in tables which are existing in both the schemas, but the columns are different.
Constraint_Trigger_Disable_Script.sql	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Script to disable constraints and triggers.
Constraint_Trigger_Enable_Script.sql	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Script to enable constraints and triggers.
Drop_Sequence_Script.sql	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Script to drop the sequence.

Table 5-1 (Cont.) Utility Scripts

Script Name	Location	Remarks
Import_P-M_data.par	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Param files to import static data.
Import_EM_data.par	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Param files to import non-static data.
Import_Sequence.par	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Par file to import sequence.
Truncate_PData.sql	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Scripts to truncate p-Data.
Export_Source_PData.par	/SOFT/TOOLS/ Upgradetoolkit/ Soft/ ImpExp/ Scripts	Param file to export required p_data tables.