

Upgrade Toolkit User Guide
Oracle FLEXCUBE Investor Servicing
Release 14.6.0.0.0
Part No. F56386-01
May 2022





Upgrade Toolkit User Guide
May 2022

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/

Copyright © [2007], [2022], Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Table of Contents

| | |
|--|------------|
| 1. PREFACE | 1-1 |
| 1.1 INTRODUCTION..... | 1-1 |
| 1.2 INTENDED AUDIENCE | 1-1 |
| 1.3 DOCUMENTATION ACCESSIBILITY | 1-1 |
| 1.4 SCOPE..... | 1-1 |
| 1.5 ORGANIZATION | 1-2 |
| 1.6 RELATED INFORMATION SOURCES..... | 1-2 |
| 2. UPGRADE AND CONVERSION APPROACH | 2-1 |
| 2.1 APPROACH - DATA IMPORT TO TARGET SCHEMA..... | 2-1 |
| 2.1.1 <i>Advantages</i> | 2-1 |
| 2.1.2 <i>Disadvantages</i> | 2-1 |
| 2.2 UPGRADE PROCESS SUMMARY | 2-2 |
| 3. MOCK UPGRADE | 3-1 |
| 3.1 PREREQUISITES | 3-1 |
| 3.2 MOCK UPGRADE ACTIVITY | 3-2 |
| 3.2.1 <i>Applying Temp Soft Changes</i> | 3-2 |
| 3.2.2 <i>Setting up Target Schema</i> | 3-2 |
| 3.3 UPGRADING DATABASE..... | 3-3 |
| 3.3.1 <i>Data Import Activities</i> | 3-3 |
| 3.3.2 <i>Deploying Front End Application</i> | 3-6 |
| 3.3.3 <i>Impact on Existing External System Interfaces</i> | 3-6 |
| 3.3.4 <i>Verifying Data after Database Upgrade</i> | 3-7 |
| 4. CUT-OVER UPGRADE ACTIVITIES | 4-1 |
| 4.1 INTRODUCTION..... | 4-1 |
| 4.2 ACTIVITIES IN PRODUCTION ENVIRONMENT | 4-1 |
| 4.3 DATABASE UPGRADE IN PRODUCTION ENVIRONMENT..... | 4-1 |
| 4.4 INSTALLATION OF OTHER COMPONENTS | 4-1 |
| 5. ANNEXURE | 5-1 |
| 5.1 UTILITY SCRIPTS | 5-1 |

1.1 Introduction

Customers who use the lower versions of Oracle FLEXCUBE Investor Servicing may need to upgrade to the latest version as the support to the older versions will phase out.

This document guides you through the standard strategy for the upgrade activity. In this document, you can find necessary information required to carry out the upgrade activity from any lower version of Oracle FLEXCUBE Investor Servicing to the latest version.

1.2 Intended Audience

This document is intended for the following audience:

- Implementation team
- Partners

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>.

1.4 Scope

This document covers the upgrade strategy/methodology for migration from lower version of Oracle FLEXCUBE Investor Servicing (12.0.1) to the latest version (12.4).

Note

Upgrade from other older versions of Oracle FLEXCUBE Investor Servicing are not in the scope, as the target versions may not have all the features of the source version.

1.5 Organization

This manual is organized into the following chapters:

| Chapter | Description |
|------------------|--|
| Chapter 1 | <i>Preface</i> gives information on the intended audience. It also lists the various chapters covered in this User Manual. |
| Chapter 2 | <i>Upgrade and Conversion Approach</i> gives an outline of the upgrade and conversion approaches. This also provides a summary of all the activities involved in the entire upgrade process. |
| Chapter 3 | <i>Mock Upgrade</i> discusses the prerequisites and guides you through the process of mock upgrade of Oracle FLEXCUBE Investor Servicing from a lower version to higher version. |
| Chapter 4 | <i>Cut-over Upgrade Activities</i> explains the activities that you need to carry out during cut-over. |
| Chapter 5 | <i>Annexure</i> provides the details of utility scripts and conversion scripts used in the upgrade process. |

1.6 Related Information Sources

For more information, refer to the following documents:

- Oracle FLEXCUBE Installation Manual
- Oracle FLEXCUBE User Manuals

2. Upgrade and Conversion Approach

This chapter gives an outline of the upgrade approach. This also provides a summary of all the activities involved in the entire upgrade process.

2.1 Approach - Data Import to Target Schema

In this approach, you will prepare the staging area only with the target version schema. This is the recommended approach.

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.

2.1.1 Advantages

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

2.1.2 Disadvantages

The disadvantages of this approach are 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.

2.2 Upgrade Process Summary

A summary of the version upgrade process is provided below:

1. Complete the mock upgrade activity. The mock upgrade provides a safe platform for the actual production environment upgrade. Mock upgrade involves the following steps (*Refer 'Upgrading Database' section in this document*):
 - Identify the source database schema. The source system should be at the stage of Transaction Input just marked on migration date.
 - Setup the target database as per the installation manuals. Exit the installer immediately after loading the static data.
 - Retrofit the customization changes, if any, into the target database. Apply the DB object changes, source file changes and static data changes. Take a full dump of the source schema.
 - In the target schema, disable all the triggers and selective constraints.
 - Import the table data alone into the target database from the full dump of source schema.
 - In the target schema, enable all the triggers and constraints.
2. Complete the cut-over upgrade activity. During this, you will upgrade the production environment database. (*Refer 'Cut-over Upgrade Activities' section in this document.*)

3. Mock Upgrade

This chapter 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
- Cut-over 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.

3.1 Prerequisites

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:
 - For qualifying with the new version: Address all changes required for qualification with the new version.
 - 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:
 - Identify and document the migration steps that are planned.
 - Set up the utilities for data comparison and data migration, if any, in the migration area.
 - Identify and document the verification strategy.
 - 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).
 - Identify the conversion scripts to be applied post-upgrade.
9. Prepare a plan with timeline considering all changes required for a smooth upgrade.

3.2 **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:

- Upgrade Oracle database to the new version
- 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)
- Setup target version of Oracle FLEXCUBE Investor Servicing
- Perform module specific changes if any
- Set up interfaces and adapters
- After upgrade, test the target version along with all interfaces and adapters
- Get sign off on the production environment upgrade

3.2.1 **Applying 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.

Source File Changes

You can use DIFF tools to compare the base version of the source application and the version used by the customer.

Static Data Changes

For identifying the differences in the factory shipped data, use utility/scripts mentioned in the Annexure section in this document.

3.2.2 **Setting 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.

4. 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.
5. 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 TEMP_SOFT changes, you need to take care of the following:
 - 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.
 - Apply the additional static data onto the upgraded schema.
6. 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.

3.3 Upgrading Database

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

3.3.1 Data Import Activities

| Activity No. | Activity Details | Source/ Destination | Dependency |
|--------------|--|---------------------|--|
| 1 | <ul style="list-style-type: none"> • For illustration purpose, consider that the name of the source schema used by the customer is 'SOURCE_SCHEMA'. 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 SOURCE_SCHEMA. Name the export dump file as 'SITE_FULL_DUMP.DMP'. The parameter file 'Export_Site_FULL_Dump.par' can be used for this export (<i>Refer 'Annexure' section in this document</i>). | Source | No Dependency. You can do this while the target schema is setup. |
| 2 | <ul style="list-style-type: none"> • Configure the TNS in source and destination database to create DB link. | Common | |

| | | | |
|---|---|-------------|---------------------------|
| 3 | <ul style="list-style-type: none"> • Run the schema difference utility (<i>Refer 'Annexure' section in this document</i>). 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 SOURCESHEMA and DESTSCHEMA. Copy the results to an Excel file. • Run 'Existing_Table_Column_Diff.sql' to identify the Table Column difference between the SOURCESHEMA 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 SOURCESCHMA 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' (<i>Refer 'Annexure' section in this document</i>) to generate the following scripts. <ul style="list-style-type: none"> ➤ 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 |
|---|---|-------------|---------------------------|

| | | | |
|---|--|-------------|---------------------------------------|
| 4 | <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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 (<i>Refer 'Annexure' section in this document</i>). The parameter file 'Import_EM_data.par' can be used to import the E-M data into the DESTSCHEMA (<i>Refer 'Annexure' section in this document</i>). 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 | <ul style="list-style-type: none"> Enable all the triggers and constraints. You need to manually handle the errors encountered while enabling the triggers and constraints. | Destination | Activity 4 |
| 6 | <ul style="list-style-type: none"> 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 | Destination | Activity 5 |

3.3.1.1 Post Import Activities

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

- Recompile invalid objects

3.3.1.2 Issues in Data Import using IMPDP Utility

You may encounter any the following issues while importing data 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. |

3.3.2 Deploying 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.

3.3.3 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

3.3.4 Verifying 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.

3.3.4.1 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 supported browser setting as per DBA server sizing document
 - 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

3.3.4.2 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

3.3.4.3 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.

3.3.4.4 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.

3.3.4.5 New Product Maintenance Testing

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

3.3.4.6 Signoff

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

3.3.4.7 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. Cut-over Upgrade Activities

4.1 Introduction

The upgrade activities that you need to carry out during cut-over are as follows.

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

4.2 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

4.3 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 'Annexure' section in this document*).

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 'Upgrading Database' and 'Post Import Activities' sections in this document.

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.

4.4 Installation of Other Components

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

5. Annexure

5.1 Utility Scripts

The utility scripts are given in the following table:

| 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 'SOURCE SCHEMA' with the actual schema name to be exported. Also change the 'DIRECTORY', 'DUMPFIL', '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. |

| Script Name | Location | Remarks |
|--------------------------|---|---|
| Drop_Sequence_Script.sql | /SOFT/TOOLS/ Upgradetoolkit/ Soft/ImpExp/ Scripts | Script to drop the sequence. |
| 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. |