

**Oracle Utilities Cloud Services**  
Data Upload Support Implementation Overview  
For 20C Releases  
**F35474-01**

December 2020

## Oracle Utilities Customer Cloud Services 20C Data Upload Support Implementation Overview

Copyright © 2017, 2020 Oracle and/or its affiliates. All rights reserved.

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.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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 fail-safe, 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.

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

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

---

---

# Contents

## Chapter 1

|  |            |
|--|------------|
| <b>Introduction</b> .....                          | <b>1-1</b> |
| Conversion Process Overview.....                   | 1-2        |
| Implementation Effort .....                        | 1-3        |
| What Is in the Newly Provisioned Environment?..... | 1-4        |

## Chapter 2

|  |            |
|--|------------|
| <b>Data Upload Support On Cloud</b> .....  | <b>2-1</b> |
| Overview .....                             | 2-2        |
| Provided by Cloud Service Foundation ..... | 2-3        |
| Provided by Applications .....             | 2-4        |
| Application Conversion Tool.....           | 2-4        |
| Application Accelerators.....              | 2-4        |

## Chapter 3

|  |            |
|--|------------|
| <b>Preparing for Conversion</b> .....          | <b>3-1</b> |
| Preparing Environment for Conversion.....      | 3-2        |
| Set Up Conversion Security.....                | 3-2        |
| Prepare Environment for Conversion .....       | 3-2        |
| Preparing Legacy Data Extract for Upload ..... | 3-4        |

## Chapter 4

|   |            |
|---|------------|
| <b>Data Upload Design</b> .....                         | <b>4-1</b> |
| Extract/Upload by Table or Maintenance Object .....     | 4-2        |
| CLOB Data in a Secondary File.....                      | 4-3        |
| Multiple Data Files for Single Table or MO Upload ..... | 4-3        |

## Chapter 5

|  |            |
|--|------------|
| <b>Data Upload Steps</b> .....                       | <b>5-1</b> |
| Upload Data into a Table or Maintenance Object ..... | 5-2        |
| Review Input Data File Spec .....                    | 5-2        |
| Create Input Data File(s).....                       | 5-2        |
| Switch Schema .....                                  | 5-2        |
| Cleanup Target Table .....                           | 5-3        |
| Upload Data .....                                    | 5-3        |
| Populate Key Table(s) .....                          | 5-3        |
| Data Upload Orchestration.....                       | 5-4        |
| Single Table Upload.....                             | 5-4        |
| Multiple Tables or MOs Upload.....                   | 5-4        |
| Full Conversion Chain per MO, Parallel Run.....      | 5-5        |
| Upload All + Subsequent Validate/Transform MOs.....  | 5-5        |

## Chapter 6

|  |            |
|--|------------|
| <b>Why, When, What, and How to Customize</b> ..... | <b>6-1</b> |
| Why Customize .....                                | 6-2        |

---

|   |     |
|---|-----|
| When to Customize.....                    | 6-2 |
| What to Customize.....                    | 6-2 |
| Control File .....                        | 6-2 |
| Additional Customization Items .....      | 6-3 |
| How to Customize .....                    | 6-3 |
| Tips and Important Mistakes to Avoid..... | 6-4 |
| Sample Artifacts and Data Files .....     | 6-6 |

# Chapter 1

---

## Introduction

This document contains data upload and implementation information relevant to the products included in Oracle Utilities Cloud Services. Most of the information is generic and applies to functionality that is available in each of the products as part of the Cloud Service Foundation. There are also some conversion tools that are documented with each specific product. (The specific products are referred to in this document as "products" or "applications".)

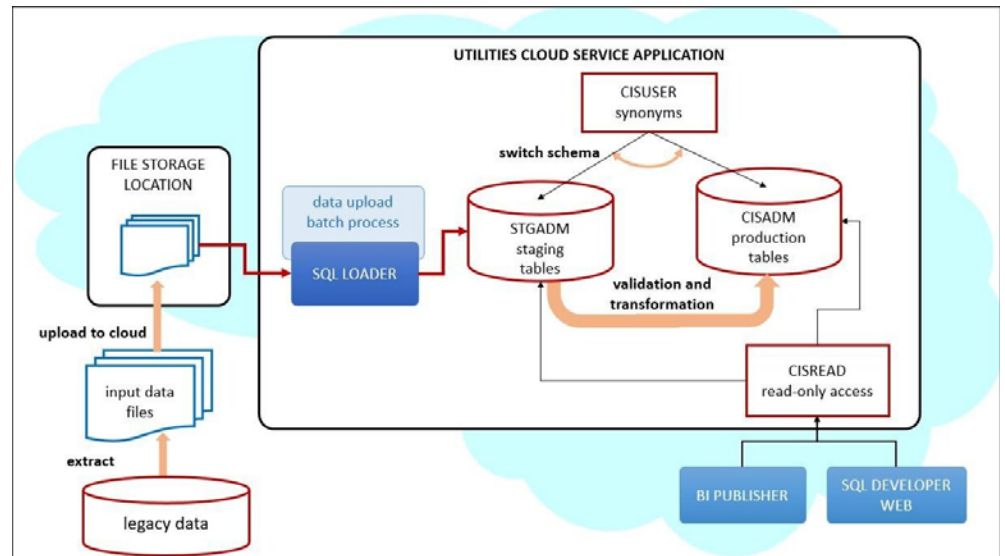
This chapter includes:

- [Conversion Process Overview](#)
- [Implementation Effort](#)
- [What Is in the Newly Provisioned Environment?](#)

# Conversion Process Overview

The goal of the Conversion Process is to migrate data from a legacy application into a target environment, and to begin running the application in the cloud. Due to cloud-related technical restrictions, legacy data cannot be uploaded directly into the software-as-a-service (SaaS) database.

Legacy data must be extracted into file(s) and compressed. The data files are uploaded to the cloud file storage location and then loaded into the target "staging" tables using Oracle SQL Loader. The data is validated, transformed, and finally inserted into "production" tables. Oracle Utilities cloud services include various tools supporting ad hoc SQL inquiries and reconciliation reports on both staging and production data.



---

# Implementation Effort

Implementers are expected to perform the following tasks for data conversion:

- Analyze the legacy data and decide what portion of it should be converted
- Map the legacy data to target Oracle Utilities Application Framework (OUAF) / Application data
- Develop legacy data extract process and produce input data files
- Adjust default data upload setup in OUAF / application, if needed
- Rehearse data upload and fine-tune configurations and/or legacy data extract, if needed
- Create reconciliation reports in BI Publisher
- Use uploaded data to try the subsequent conversion flow(s); bring the end-to-end conversion flow to perfection
- Execute the final conversion data upload run, a.k.a. cut-over
- Execute the application's data conversion processes.
- Disable conversion activities in the environment

# What Is in the Newly Provisioned Environment?

## **The production instance is available for conversion.**

Conversion activities do not co-exist well with the rest of the implementation. The massive data uploads, table truncation, and switching schema could disrupt business configurations development and testing. The production environment is the best candidate for conversion.

In the newly provisioned instance, the staging area in the database is created according to application specifications. The BI Publisher instance and SQL Developer Web / Oracle REST Data Services are connected to production and staging data.

## **The environment contains pre-configured conversion data upload setup.**

The default configurations are suitable for typical table volumes and common data formats. If your implementation does not include extremely large data volumes, special data formats, or other idiosyncratic requirements, the default setup can be used "as is".



# Chapter 2

---

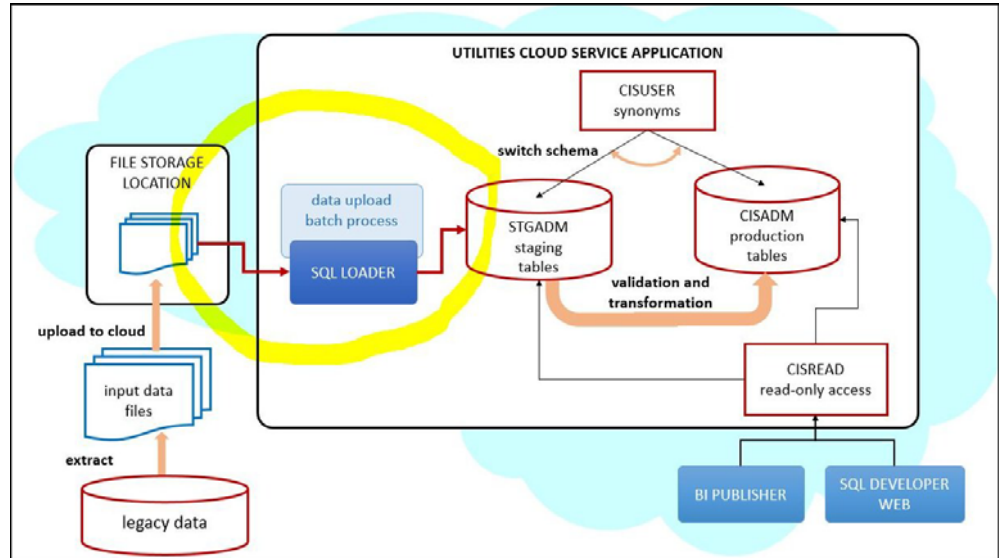
## Data Upload Support On Cloud

This chapter provides an overview of data upload support in Oracle Utilities cloud services, including:

- [Overview](#)
- [Provided by Cloud Service Foundation](#)
- [Provided by Applications](#)

# Overview

The highlighted portion of the flow shown below is supported by Oracle Utilities Cloud Service Foundation (CSF). The legacy data extract and the input file creation belong to the implementation. The business application provides conversion validation and transformation processes, as well as the definition(s) of the staging area.



---

# Provided by Cloud Service Foundation

Oracle Utilities Cloud Service Foundation (CSF) features metadata-driven configurable and customizable data upload with SQL Loader. It also provides support for basic database operations such as table clean-up (truncate), index enable/disable, and some others.

SQL Loader is an Oracle database utility that allows users to load data from external files into target DB tables. See Oracle DB SQL Loader Documentation for details.

The load of the Input Data File is performed according to the instructions recorded in a Control File. The Control File contains load options and parameters and also a list of data fields with formatting and parsing instructions.

For data upload on cloud, Control Files are pre-generated based on the metadata and conversion configurations and stored in the system.

Cloud Service Foundation allows users to generate control files, and it also provides a batch process that consumes the Input Data File(s), reads the pre-generated Control File and calls SQL Loader.

Cloud Service Foundation delivers the following:

- Batch processes. CSF batch controls are "generic", with no default value for the parameter that specifies target table or maintenance object (MO). These batch controls are used mostly for development and testing purposes. Applications are likely to supply "specific" batch controls for each target table or MO.
  - Batch Controls: Load Data into Table or Maintenance Object, Truncate Table or MO's Tables, Disable/Enable Indexes, Disable/Enable Triggers, Update Statistics, Populate Key Table, Cleanup Key Reference and XML resolution Tables, Generate Conversion Artifacts (bulk), and few others.
- Services accessible via online UI.
  - Switch Schema - executes the stored procedure that is re-directing the CISUSER synonyms between staging and production.
  - Generate Conversion Artifacts - creates input data file specifications and SQL Loader control files for specific converted objects. The artifacts are generated based on the metadata and according to the conversion data upload setup
- The instance of BI Publisher that is connected to the database with read-only access to the staging and production schema tables.
- The instance of SQL Developer Web that is connected to the database with read-only access to the staging and production schema tables.
- Predefined Configurations
  - Data delimiters and data format strings for date and date/time fields (Extendable Lookups)
  - Default Conversion Instructions (Conversion Task Types) for typical Table, MO and Key Table.
  - SQL Loader Control File fragments (Managed Content) for parallel and non-parallel load
  - Conversion Data Upload Master Configuration with default setup
  - User Groups and Template Users for conversion (suggested setup)

## Provided by Applications

Each application comes with its own Conversion Accelerator that includes admin and system data for suggested upload configurations. Applications also provide a set of processes and tools to validate and transform the uploaded "staging" legacy data into real production form.

### Application Conversion Tool

The Conversion Tool is usually comprised of processes, services, and configurations that support necessary legacy data validation and transformation.

The application connects to the `@application_user` schema in the database. This schema contains synonyms to the actual tables and views. Let's assume the legacy data has to be loaded into a table `TABLEXXX`. From the application perspective, the data upload process inserts the data into `@application_user.TABLEXXX`. The meaning of the synonym `@application_user.TABLEXXX` could be `@production_schema.TABLEXXX` or `@staging_schema.TABLEXXX`. Once the `switch_schema()` stored procedure is executed, the assumption is that the synonyms in `@application_user` schema are set to point to the "staging" tables, and the data upload may begin.

Possible approaches to the staging data upload into target tables are described below.

**Approach:** The legacy data is uploaded into a set of tables in the staging schema (STGADM).

Upon successful legacy data upload, the sequence of batch processes performs object-level validation and FK validation of the data in the staging tables, generates new keys and finally inserts the data into production tables.

**Approach:** The legacy data is extracted and loaded into Initial Sync table(s) in the production schema (CISADM). This approach is relevant under special circumstances such as extremely large data volume. Another reason for loading data directly into production could be a migration of existing Oracle Utilities application from on-premise to cloud, when the legacy data is actually a valid application data, conforming to target data formats and standards and there is no need for additional validations and key generation.

NOTE: the Switch Schema has to be performed anyway in order to set the internal system indicator to Staging and allow the data upload

A custom control file should be created in order to insert the data into production tables..

Upon successful legacy data upload, the data can be further processed using regular application processes.

### Application Accelerators

The accelerator usually contains suggested configurations for data upload support. It may include:

- Conversion Instructions for tables/maintenance objects with special data requirements (Conversion Task Types)
- Alternative Control File fragments and custom Control Files (Managed Content)
- Conversion Master Configuration

- Specific Batch Controls for each converted Table or MO, suggested batch job/ batch streams for suggested conversion activities orchestration
- Sample reconciliation reports
- Other system, admin, and/or configuration data

# Chapter 3

---

## Preparing for Conversion

This chapter describes how to prepare an environment and legacy data for conversion with Oracle Utilities cloud services, including:

- [Preparing Environment for Conversion](#)
- [Preparing Legacy Data Extract for Upload](#)

# Preparing Environment for Conversion

Preparing an environment for conversion involves the following:

- [Set Up Conversion Security](#)
- [Prepare Environment for Conversion](#)

## Set Up Conversion Security

Conversion activities comprise massive data manipulations and database operations such as disabling / enabling indexes, truncating tables, and other operations. Whoever works on the conversion project deals with the real client's data and may have access to sensitive customer information. Therefore it is important to determine implementer's roles and responsibilities in advance, and to provide the user with the appropriate authorization level.

Use the pre-configured user groups *Conversion Administration*, *Conversion Development*, and *Conversion Operations*, along with the corresponding Template Users K1CNVADM, K1CNVDEV and K1CNVOPR. Alternatively, design and define your own conversion user authorization setup.

## Prepare Environment for Conversion

- Enable conversion activities in the environment.
  - Run K1-CNVEN batch.
- Import the Conversion Data Upload Accelerator, if it was supplied by the application.
- Generate conversion artifacts.
  - To generate artifacts for all eligible tables and/or maintenance objects, submit a batch job for the K1-CNVAG batch control and use batch parameters to specify the scope for the generation: everything, Tables only or Maintenance Objects only.
    - As a result, new Conversion Task is generated for each Table and each Maintenance Object eligible for conversion. The artifacts are linked to the Conversion Tasks as attachments
  - To generate artifacts for an individual table or maintenance object, select **Admin**, select **Conversion Support**, and select **Generate Conversion Artifacts**. Choose "Table" or "Maintenance Object" and run the generator.
    - As a result, a new Conversion Task is generated for the selected table or maintenance object. The artifacts are linked to the Conversion Tasks as attachments.
- Query Conversion Tasks that were created for various Tables and Maintenance Objects and explore the generated artifacts:
  - **Input Data File Specifications.** This file contains the detailed field by field formatting instructions and other notes about the expected contents of the input data file. Use these instructions when preparing the legacy data extract.
  - **Control File.** This file is used by SQL Loader during the data upload

- **File List.** This file lists the name of the input files that has to be prepared for the data upload.
  - Multiple data files for a single object (Table or Maintenance Object) are expected if the data is being uploaded contains CLOB columns AND the upload is configured to load CLOB from secondary file.
- Switch schema to redirect the application to the staging data area.
  - Use the menu to navigate to the generator by selecting **Admin**, then selecting **Conversion Support**.
- Truncate tables in the staging data area to ensure that you will be uploading the data into clean empty tables.
- Disable indexes in the staging data area. This is required because SQL Loader is not capable of implicitly disabling partitioned indexes during the data upload
- Disable triggers in the staging data area.

The environment is now ready for the legacy data upload:

- Conversion is enabled
- SQL Loader Control Files have been generated, and
- Synonyms in the database schema point to the staging data area tables.

### Notes:

- Conversion activities are possible as long as conversion is enabled in the environment. Once the legacy data is successfully migrated, you should disable conversion by running the K1-CNVDS batch. By doing this you set an internal indicator that is queried by conversion-related processes, such as switch schema, data upload, table cleanup, and index/statistics update. These processes will only run when conversion is enabled.

**Important:** The Disable Conversion process should be executed ONLY ONCE right before the system is ready for go live. It is one-time event and is irreversible. Once disabled, conversion activities cannot be fully re-enabled as the assumption is that the re-enabling is happening while the application is running live in production. Enabling conversion after it has been disabled will result in the application running in the Incremental Conversion mode with its limitations.

- Switching the schema sets an internal flag that indicates whether the synonyms are pointing to "staging" or "production" area. The data upload is only allowed when the application is running in a "staging" mode.
- It is recommended to perform truncate operations at the maintenance object level as it will prevent leaving orphan records in the database. When truncating tables one by one always truncate child tables first.



# Preparing Legacy Data Extract for Upload

The legacy data mapping and extract will vary from one customer to another. The files created as a result of the extract process should conform to the specifications generated above. The resulting data extract files should be:

- Created according to the specifications
- Named according to the naming convention (see the online help and the specifications for more details)
- Optionally, the file might be compressed with gzip or zip (see the online help for details)

## Special Data Considerations:

Oracle Utilities Cloud Services provide support for Information Lifecycle Management (ILM) and Data Archiving.

All ILM-enabled objects contain the following fields:

- ILM Date (ILM\_DT)
- ILM Archive Switch (ILM\_ARCH\_SW).

The ILM and Data Archiving functionality is controlled by the combination of these two fields.

- The ILM Date field is used in conjunction with partitioning to group data by age.
- The ILM Archive Switch is set by a background process when a record meets the business rules specific to the record's Maintenance Object that indicates the record is eligible to be archived.

See **Information Lifecycle Management** in the application's *Administrative User Guide* for more information about how these fields are used.

When preparing the legacy data extract for a target table, perform the following steps:

- Access the Oracle Utilities application, search for a Conversion Instructions Conversion Task Type (see **Conversion Task Types** in the *Oracle Utilities Cloud Service Foundation Administrative User Guide*) for the target table or maintenance object and review the input data specifications. Determine if the field list contains the fields named ILM\_DT and ILM\_ARCH\_SW.
- In the data extract populate ILM\_ARCH\_SW field with value 'N'
- For the ILM\_DT field, the **ILM Date Fields** on page 3-5 table below lists the recommended column whose value should be used to populate the ILM\_DT for conversion data upload.
  - Locate your target table name in the list and determine how the ILM\_DT field should be populated
  - If the table is not listed, please contact Oracle Utilities support.

## ILM Date Fields

| Table Name         | ILM DT Initial Load                                    |
|--------------------|--|
| CI_TD_ENTRY        | CI_TD_ENTRY.CRE_DTTM                                   |
| F1_SYNC_REQ_IN     | F1_SYNC_REQ_IN.CRE_DTTM                                |
| F1_OUTMSG          | F1_OUTMSG.CRE_DTTM                                     |
| F1_SVC_TASK        | F1_SVC_TASK.CRE_DTTM                                   |
| F1_OBJ_REV         | F1_OBJ_REV.STATUS_UPD_DTTM                             |
| F1_BUS_FLG         | F1_BUS_FLG.CRE_DTTM                                    |
| F1_REMOTE_MSG      | F1_REMOTE_MSG.CRE_DTTM                                 |
| F1_STATS_SNPST     | F1_STATS_SNPST.CRE_DTTM                                |
| F1_ERASURE_SCHED   | F1_ERASURE_SCHED.STATUS_UPD_DTTM                       |
| F1_PROC_STORE      | F1_PROC_STORE.STATUS_UPD_DTTM                          |
| F1_GNRL_AUDIT      | F1_GNRL_AUDIT.CRE_DTTM                                 |
| D1_ACTIVITY        | D1_ACTIVITY.CRE_DTTM                                   |
| D1_COMM_IN         | D1_COMM_IN.CRE_DTTM                                    |
| D1_COMM_OUT        | D1_COMM_OUT.CRE_DTTM                                   |
| D1_DVC_EVT         | D1_DVC_EVT.CRE_DTTM                                    |
| D1_COMPL_EVT       | D1_COMPL_EVT.CRE_DTTM                                  |
| D1_INIT_MSRMT_DATA | D1_INIT_MSRMT_DATA.CRE_DTTM                            |
| D1_USAGE           | D1_USAGE.CRE_DTTM                                      |
| D1_USAGE_EXCP      | D1_USAGE_EXCP.CRE_DTTM                                 |
| D1_VEE_EXCP        | D1_VEE_EXCP.CRE_DTTM                                   |
| D1_ACTIVITY        | D1_ACTIVITY.CRE_DTTM                                   |
| CI_ADJ             | CI_ADJ.CRE_DT  |
| CI_APPR_REQ        | MIN(LOG_DTTM) on CI_APPR_REQ_LOG for given APPR_REQ_ID |
| CI_BILL            | CI_BILL.CRE_DTTM                                       |
| CI_BSEG            | CI_BSEG.CRE_DTTM                                       |
| CI_STM             | CI_STM.STM_DT  |
| C1_OFFCYC_BGEN     | C1_OFFCYC_BGEN.STATUS_UPD_DTTM                         |
| CI_BILL_CHG        | CI_BILL_CHG.START_DT                                   |
| CI_CASE            | MIN(LOG_DTTM) on CI_CASE_LOG table for given CASE_ID   |
| CI_FA              | CI_FA.CRE_DTTM   |

| <b>Table Name</b>    | <b>ILM DT Initial Load</b>                  |
|----------------------|---|
| CI_ENRL              | CI_ENRL.START_DT                            |
| CI_PAY_EVENT         | CI_PAY_EVENT.PAY_DT                         |
| CI_PAY               | CI_PAY_EVENT.PAY_DT                         |
| CI_MATCH_EVT         | CI_MATCH_EVT.CREATE_DT                      |
| C1_USAGE             | C1_USAGE.CRE_DTTM                           |
| C1_CUST_REL_REQ      | C1_CUST_REL_REQ.CRE_DTTM                    |
| CI_CC                | CI_CC.CC_DTTM or<br>CI_CC.LETTER_PRINT_DTTM |
| CI_MR                | CI_MR.READ_DTTM                             |
| C1_PA_RQST           | C1_PA_RQST.CRE_DTTM                         |
| C1_CS_RQST           | C1_CS_RQST.CRE_DTTM                         |
| C1_CS_REQ_ACCT       | C1_CS_REQ_ACCT.?                            |
| C1_CS_REQ_CONT       | C1_CS_REQ_CONT.CRE_DTTM                     |
| C1_CS_RQST_CONT_PROD | C1_CS_RQST_CONT_PROD.CRE_DTTM               |
| C1_CS_REQ_PER        | C1_CS_REQ_PER.CRE_DTTM                      |
| C1_CS_REQ_CVS_LOC    | C1_CS_REQ_CVS_LOC.CRE_DTTM                  |
| C1_CS_REQ_PREM       | C1_CS_REQ_PREM.CRE_DTTM                     |
| C1_MKTMSG_CHG        | C1_MKTMSG_CHG.MKT_CHG_DT                    |
| C1_MKTMSG_PAY        | C1_MKTMSG_PAY.MKT_PAY_DT                    |
| C1_MKTMSG_USG        | C1_MKTMSG_USG.MKT_USG_DT                    |

# Chapter 4

---

## Data Upload Design

There are several aspects implementation should consider when designing the legacy data extract processes and creating the Input Data Files. The data upload process is very flexible and configurable, and can be fine-tuned to address both application and client data specifics.

This chapter provides information about designing extract processes, including:

- [Extract/Upload by Table or Maintenance Object](#)
- [CLOB Data in a Secondary File](#)
- [Multiple Data Files for Single Table or MO Upload](#)

## Extract/Upload by Table or Maintenance Object

The SQL Loader allows users to insert data into one or multiple tables from a single input file. Choose the more convenient option, depending on the structure of the legacy data (source), data volumes, and extract technique:

- **Table-level.** Extract file contains data for the single table. The data is loaded into a table in the OUAF/ application database.
- **Maintenance Object-level.** Extract file contains data for the entire object. The data is loaded into a set of tables that represent the corresponding Maintenance Object in the OUAF/ application database.

Both options are supported in Cloud Service Foundation. Generate the artifacts and review the differences in the specifications.

The table below illustrates the difference between Table and Maintenance Object data file:

|                               | Table: CI_PER   | Maintenance Object PERSON:   |
|-------------------------------|---|--|
| <b>Target Object</b>          | Data file contains records for a single table.                      | Tables: <ul style="list-style-type: none"> <li>• CI_PER</li> <li>• CI_PER_NAME</li> <li>• CI_PER_ID ....etc</li> </ul> Data file contains records for multiple tables within Maintenance Object. Table name serves as "record type" qualifier.   |
| <b>Input Data File Layout</b> | 1234, IND, Doe,...<br>5678, IND, Moon,...<br>9063, BUS, ABC Corp,.. | CI_PER      1234, IND, Doe,...<br>CI_PER      5678, IND, Moon,...<br>CI_PER      9063, BUS, ABC Corp,..<br>CI_PER_ID    1234, SSN,72346781<br>CI_PER_ID    5678, SSN, 87635241<br>CI_PER_ID    9063, EIN, 09182835<br>CI_PER_ID    9063, TID, 82528555<br>CI_PER_NAME  1234, Doe, Mary<br>CI_PER_NAME  5678, Moon, Barry |

---

## CLOB Data in a Secondary File

CLOB data can be supplied as part of the record in the "main" data file or as a secondary file. Once again, the decision should be made based on the source data volumes, extract techniques, and the availability of the CLOB data in most records.

- If most of the records have CLOB column(s) populated, and/or the CLOB field often contains large amount of data, it may make sense to use a secondary file.
- Otherwise, if the CLOB column(s) are rarely populated and/or the CLOB field rarely contains large amount of data, you may choose to include the CLOB data in the record.

**Note:** If supplied as secondary file, the CLOB data file has to contain exactly as many records as the main file. This means that a line has to be added even for empty CLOB fields.

Both options are supported. The definition is controlled by the Conversion Instruction (Conversion Task Type).

## Multiple Data Files for Single Table or MO Upload

The Cloud Service Foundation data upload process supports the upload into single target (table or maintenance object) from multiple data files. For example, instead of extracting a large Payment table into a single *payment.csv* file, you can split the extract into *payment1.csv*, *payment2.csv*, *payment3.csv*, and so on.

It is recommended to keep the file size under 2 gigabytes. The number of files is unlimited. Naming conventions apply. See the online help for more details.

# Chapter 5

---

## Data Upload Steps

This chapter describes the steps involved in data upload, including:

- [Upload Data into a Table or Maintenance Object](#)
- [Data Upload Orchestration](#)

# Upload Data into a Table or Maintenance Object

The data upload stage may begin only after conversion artifacts have been generated.

## Review Input Data File Spec

- Retrieve the Conversion Task associated with the Table XXX
  - Navigate to **Admin**, then **Conversion Support**, then **Conversion Task Query** and select the "Table/Maintenance Object" **Query Option**.
  - Use search to populate either Table or Maintenance Object search criteria
  - From the search results, pick the latest entry.
- Load Conversion Task and locate a collection of Attachments.
- Find an attachment that represents *Input File Specification*.
- Click on the context menu to launch **Attachment** view the attachment contents.

## Create Input Data File(s)

The specification defines the expected input data record format. The data fields are listed in the order it expected to appear in each record. For each field, the specification contains the data type, size and format. The specification also describes:

- Data delimiter
- Enclosing characters (to enclose a single blank that will represent empty non-nullable field)
- Date and date time formats
- CLOB data delimiter
- Expected name(s) for the secondary data file(s)

Extract the legacy data into a file according to the specification.

Each line in the file should represent a row in the target table. In the maintenance object-level extract, each record represents a row in one of the maintenance object tables and the first 30 characters in each line contains the table name.

If CLOB data is to be provided as secondary file, create CLOB data files.

**Note:** the SQL Loader treats invalid secondary file differently than a missing secondary file:

- If the secondary file is missing, the process will report an error.
- If the CLOB data in the secondary file is invalid, the CLOB field in the target table will be initiated into NULL or blank.

The input data file might be supplied uncompressed or compressed. Supported compressed formats include gzip and zip (See online help for details).

## Switch Schema

Navigate to **Admin**, then **Conversion Support**, then **Switch Schema**, and select "Conversion" from the drop-down list and click **OK**.



## Cleanup Target Table

Run the K1-SCLTB batch process, specifying the target table or maintenance object as a parameter.

## Upload Data

Upload the input data file created above to the Object Storage location

Run the K1-CNVLD batch process, specifying the target table or maintenance object as a parameter. Detailed description of data upload parameters can be found in the online help.

## Populate Key Table(s)

According to OUAF DB design standards, a corresponding Key Table exists for each table with system-generated or sequential primary key. Under normal circumstances, the key tables are populated when an application creates a "main" record. In a conversion situation, where the data is inserted directly into the database, there are two possibilities to populate the Key Table:

- Create an input data file for the Key Table and upload it using the same batch K1-CNVLD.
- Populate the key Table programmatically, by running K1-CPKTB after successful "main" table or MO data upload. This batch can be used for both Table and MO-level upload.

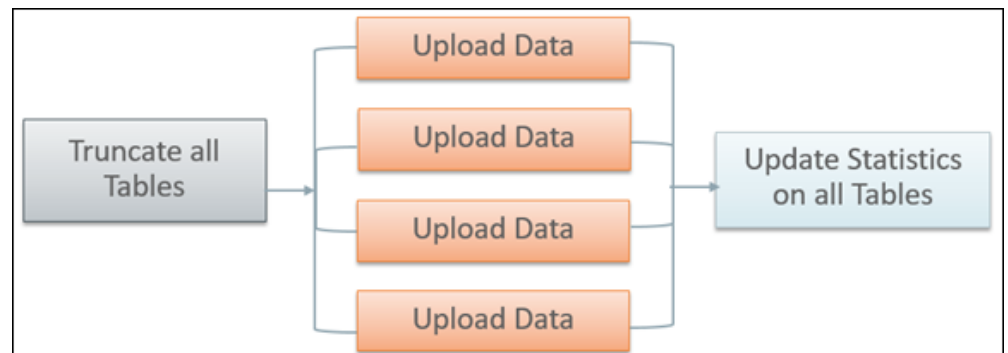
## Data Upload Orchestration

The SQL Loader is running in multiple threads and therefore it is not performing table truncation before loading (command APPEND). Hence, the target tables should be truncated prior to the load. For better performance the indexes have to be disabled before the load and re-enabled/statistics updated after the load. The batch jobs can be organized into various chain structures, as shown in the examples below.

### Single Table Upload



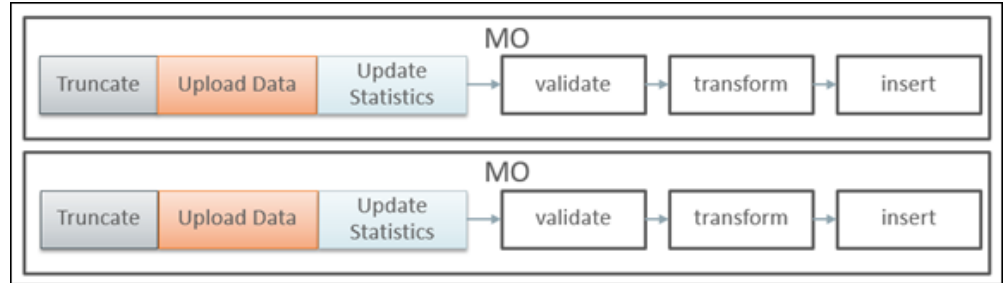
### Multiple Tables or MOs Upload



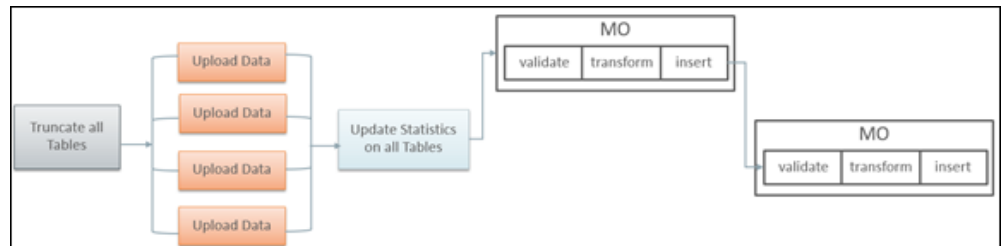
There are multiple strategies to orchestrate the entire conversion run and to build the optimal sequence of the conversion processes. Below are some of the many possibilities:

- upload all legacy data extract files simultaneously, then run the subsequent validation and transformation processes for the converted object in a certain order of precedence, to preserve referential integrity
- begin the upload of very large tables in advance, so all upload is finished simultaneously, then validate & transform
- include legacy data upload batch(es) in the batch job chain for the target object
- upload some of the data by maintenance object, some table by table
- process maintenance objects end-to-end simultaneously, if there are no inter-dependencies

## Full Conversion Chain per MO, Parallel Run



## Upload All + Subsequent Validate/Transform MOs



# Chapter 6

---

## Why, When, What, and How to Customize

Conversion-related configurations define the expected extract file layout and the SQL Loader run-time upload options and parameters. SQL Loader's Control Files are generated based on these configurations.

The Batch Job/Batch Job Chain setup defines the overall orchestration of the conversion process flows.

This chapter describes customizations to the data upload process including:

- [Why Customize](#)
- [When to Customize](#)
- [What to Customize](#)
- [How to Customize](#)
- [Tips and Important Mistakes to Avoid](#)
- [Sample Artifacts and Data Files](#)

## Why Customize

There are several reasons for customizing conversion configurations, including:

- fine-tuning data upload performance
- handling unusual data volumes
- marking additional table(s) as eligible for conversion
- reducing creation of unnecessary input files

## When to Customize

The layout of the legacy extract files should be finalized as soon as possible, to provide enough time for the extract process development.

The setup of the batch job chains is less critical at the beginning of the project. The initial suggested setup is likely to be included in the application Conversion Accelerator. Adjust the initial setup after you've performed the trial uploads of the actual data, assessed the performance and figured the optimal flows.

## What to Customize

### Control File

The majority of the customizations affect the contents of the generated Control File and the corresponding input data file specifications. The configurations are stored on the Conversion Task Types that represent Conversion Instructions.

- Customizing the Control File's load options and parameters may improve upload performance
- Fully customized Control File allows you to use alternative record parsing and other advanced SQL Loader configuration techniques.
- When CLOB data is supplied as Secondary Files, the system is expecting the input data files to exist and be named following the specific naming convention.
- For example, if the table has multiple CLOB fields, for every CLOB field that was not excluded from conversion, the system is expecting the secondary file's name to be suffixed with `_CLOB Field Name`. See the online help for more details.

**Data Delimiters and Enclosing Characters.** Examine the default Conversion Instructions (Conversion Task Type) setup. Either select another delimiter from the existing list or add new value to the Extended Lookup.

**CLOB as Secondary File?** The indicator is defined on Conversion Instructions (Conversion Task Type).

Applicable when CLOB is supplied as Secondary File:

**CLOB Columns Included in Conversion.** By default, the control file is generated as if all CLOB fields are part of the converted data. The legacy data does not necessarily

contain data for all CLOB fields, hence there is no reason to create empty files. The list of excluded CLOB columns is defined on Conversion Instructions (Conversion Task Type). Create new Conversion Instructions (Conversion Task Type) for the Table or MO with multiple CLOB fields and specify the exclusion list.

**Control File "Header" - Load Options.** A text stored as Managed Content. Contains the control file's fragment with options and load parameters. You can amend the options according to SQL Loader documentation. Examine the entries delivered with the product.

**Note:** The text contains several substitution parameters prefixed with %. The substitution happens at generation time or at run time. Preserve them while creating a custom Control File header.

If you wish to amend the load options and parameters only, create a new Managed Content entry. Modify default Conversion Instructions (Conversion Task Types) or create new ones and add Override Instructions to Conversion Master Configuration. Run Conversion Artifact Generator and create new customized Control File. See the online help for more details.

**Custom Control File.** A text stored as Managed Content and representing the entire Control File, including load options, parameters and the field list.

**Note:** Preserve substitution parameters (see the note above). The input data file specifications are not generated when the Custom Control File is used. Make sure that the fields in the input data files correlates to the field's list in the custom Control File.

## Additional Customization Items

**Table's Conversion Eligibility.** The table is considered eligible for conversion according to the indicator on the Metadata Table record. It is a system data and cannot be modified by the implementation. In order to make a non-converted Table eligible for Conversion, you should add an entry to the *Override Conversion Eligibility* list on the Conversion Master Configuration.

**Conversion Orchestration.** The suggested setup of the Batch Controls, Batch Jobs, and Chains is usually included in the application Conversion Accelerator. Adjust this setup by fine-tuning the number of threads, the chain structure(s) and other batch job parameters.

## How to Customize

Configurations can be amended on several levels:

- To modify the configuration globally, amend the default Conversion Instructions (Conversion Task Type) that is referenced on *Conversion Data Upload* Master Configuration
- To modify the option globally for all tables, amend the default Conversion Instruction for Table (Conversion Task Type) that is referenced on *Conversion Data Upload* Master Configuration

- To modify the option globally for all maintenance objects, amend the default Conversion Instruction for MO (Conversion Task Type) that is referenced on *Conversion Data Upload* Master Configuration
- To modify the option for a specific table(s) or maintenance objects, create new Conversion Instruction (Conversion Task Type) and add the Override Instruction for Table or MO on *Conversion Data Upload* Master Configuration
- To make a non-converted table eligible for conversion, add it to the Override Conversion Eligibility list on *Conversion Data Upload* Master Configuration

IMPORTANT! Regenerate Conversion Artifacts to apply the configuration changes. Download the updated input file specifications.

## Tips and Important Mistakes to Avoid

| Issue   | Details  |
|---|--|
| Run the process against the right target.   | <p>The data upload only runs if the environment is pointing to the STAGING schema.</p> <p>Navigate to Conversion Support ' Switch Schema. On the popup screen the current schema is displayed. Make sure the current schema is <b>Staging</b>.</p>   |
| Provide data files according to the specifications<br><br>Regenerate the artifacts after modifying the data upload configurations | <p>SQL Loader loads the data according to the Control File. Input Data File Specifications describe what is expected from the input data file:</p> <ul style="list-style-type: none"> <li>• Names of the data files</li> <li>• Data format for all fields</li> <li>• Data delimiters to be used in the input data file</li> </ul> <p>Every time the configuration has changed the artifacts must be regenerated in order to keep the configurations and the input data specifications in sync.</p>   |
| Provide input data files with CLOB data IF NECESSARY  | <p>Conversion Instruction defines whether CLOB data is provided as part of the main file or as a separate file. The system expects the data files to be provided according to this definition.</p> <p>Open the Input Data Specifications and read carefully. If the specification mentions that CLOB is to be provided as a secondary file, this is what Control File would inspect.</p> <p>If you wish to include CLOB data in the main file, verify that the Conversion Instruction is set correctly.</p> <p><b>If the configuration was modified you must regenerate the artifacts.</b></p> |

| Issue  | Details   |
|--|---|
| Avoid creating unnecessary data files for CLOB columns   | <p>By default the system expects the data to be provided for all target table columns.</p> <p>If the table contains multiple CLOB columns AND the CLOB data is provided as a secondary file, it means one input data file per column.</p> <p>To exclude unnecessary CLOB columns for a table or maintenance object, configure Conversion Instructions using the <i>K1-ConvArtMultClobMOTaskType</i> or <i>K1-ConvArtMultClobTblTaskType</i> business object and specify the Override Conversion Instruction on the Master Configuration</p> <p>Regenerate conversion artifacts and examine the input data specifications after changing the configuration</p> |
| Avoid truncating the entire staging data unintentionally | <p>The K1-SCLTB batch process allows you to truncate a specific table or maintenance object in the STAGING schema.</p> <p>The K1-CLNTB batch process allows you to truncate a specific table or maintenance object in the PRODUCTION schema.</p> <p>If submitted without input parameter specifying a table or maintenance object, these batches will process all tables eligible for conversion. This means that all your staging data will be wiped out at once.</p>  |
| Clean up duplicate PK values before the data upload      | <p>Indexes and constraints are disabled during data upload in order to boost performance.</p> <p>De-duplication during the data upload is not supported out-of-the-box</p> <ul style="list-style-type: none"> <li>• SQL Loader direct path upload doesn't perform duplicate check</li> <li>• No direct database access means no possibility to modify data via direct SQL after the upload</li> </ul> <p>Keep track of the legacy data that has been already uploaded</p> <p>If you re-upload the same data again, always clean up the target table(s).</p>   |



| Issue  | Details  |
|--|--|
| The business configurations and admin data has to be finalized and populated in Production prior for legacy data upload<br><br>Populate the legacy data extract with valid FK references to the admin/control data | <p><b>Once uploaded, the staging data cannot be "massaged"/modified thru direct SQL</b> (that because no database access is possible on cloud)</p> <p>Hence the overall conversion project steps are:</p> <ul style="list-style-type: none"> <li>• Design, test and complete business configurations. During this stage, multiple trial data uploads with dummy data could be performed</li> <li>• Populate admin data in Production</li> <li>• Create legacy data extract with valid admin data FK References</li> <li>• Upload data into staging tables</li> </ul> |
| Key Tables are not populated implicitly  | <p>The Key Tables in the staging schema tables are not populated automatically when the legacy data is uploaded into "main" tables.</p> <p>Upload the data into Key Tables separately or use the batch program provided by Cloud Service Foundation.</p>   |
| Override Conversion Eligibility is supported on Table level only   | <p>The conversion eligibility is overridden for individual tables. Override the eligibility for all the tables that belong to the maintenance object if you decided to convert the entire maintenance object.</p> <p><b>Note:</b> Overriding a table's conversion eligibility doesn't mean that the staging schema is automatically updated. It only means that the data upload processes will treat this table as a valid target table</p>  |

## Sample Artifacts and Data Files

To assist implementers with the conversion and data upload process, multiple sample artifacts and data files are available. The sample files are provided with your cloud service documentation. The samples illustrate various data upload scenarios for table- and MO-level upload. Within the master samples zip file, there are multiple zip archives, each of which contain the following:

- Control file, generated
- Input Data File Specification, generated
- Sample Data File, created according to the specification

The table below provides more details on each of the sample artifacts available.

| Target Object                                  | Sample Description   |
|--|--|
| Interval Data Set<br>(INT_DATA_SET)            | Regular maintenance object, CLOB field as a secondary file.<br><br>Configuration: Conversion Task Type K1-CNV-MO<br><br>Multiple data files (3)  |
| MO Customer Contact<br>(CUST_CONTACT)          | Regular maintenance object, CLOB fields in the main file.<br><br>Configuration: same as Conversion Task Type K1-CNV-TABLE, but the <i>CLOB as Secondary File</i> indicator set to false.   |
| Table Meter Read<br>(CI_MR)                    | Regular table, CLOB field as a secondary file.<br><br>Configuration: Conversion Task Type K1-CNV-TABLE.  |
| Table Adjustment<br>(CI_ADJ)                   | Regular table, CLOB field in the main file.<br><br>Configuration: same as Conversion Task Type K1-CNV-TABLE, but the <i>CLOB as Secondary File</i> indicator set to false.   |
| Table Initial Sync Request<br>(F1_SYNC_REQ_IN) | Table with Multiple CLOBs as secondary files.<br><br>Configuration:<br><br>For table with multiple CLOBs, the special Conversion Task Type was created based on the K1-ConvArtMultClobTblTaskType business object.<br><br>Override Control File (Managed Content) was created and used as a custom Control File.<br><br>Review the sample and note that there is a conditional input data selection. Only records with BO = W1-CompositeSyncReqGISAsset would be uploaded. A custom Control File is necessary if you have a requirement to manipulate the data during upload.<br><br>Input Data File Specification:<br><br>Since the Control File is fully custom, including the field list, the generated specification is describing expected file name(s) only. The data field formats, delimiters, sizes, and any other information related to the Input Data File layout should be determined based on the custom Control File. |

---

| <b>Target Object</b>   | <b>Sample Description</b>   |
|--|---|
| Tables Measurement and Measurement Component Identifier (D1_MSRMT, D1_MEASR_COMP_IDENTIFIER) | Regular tables, no CLOB data.<br>Configuration:<br>For D1_MEASR_COMP_IDENTIFIER - Conversion Task Type K1-CNV-TABLE<br><br>For D1_MSRMT - same as Conversion Task Type K1-CNV-TABLE, but Override Control File (Managed Content) was created and used as a custom Control File. |

---