

Oracle Utilities Data Upload Support on the Cloud

Implementation Overview

Release 18.2

F13945-01

Last Updated: February 6, 2019

Oracle Utilities Data Upload Support on the Cloud

F13945-01

Copyright © 2019, 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 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 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 Xeon 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, Opteron, the AMD logo, and the AMD Opteron 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.

Documentation Accessibility

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

Access to Oracle Support

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

Contents

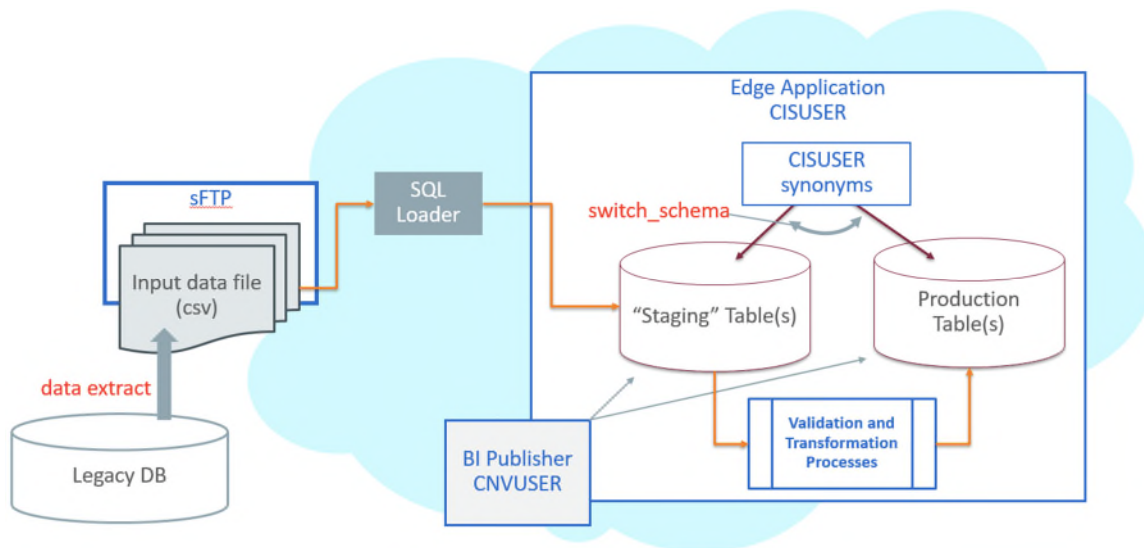
- Introduction..... 1
- Conversion Process Overview..... 1
- Data Upload Support on Cloud..... 1
 - Provided by Cloud Service Foundation (CSF)..... 2
 - Provided by Applications 3
 - Application Conversion Tool..... 3
 - Application Accelerators..... 5
- Implementation Effort..... 5
- What Is in the Newly Provisioned Environment? 6
- Preparing Environment for Conversion..... 6
- Preparing Legacy Data Extract for Upload 7
- Data Upload Design 7
 - Extract/Upload by Table or MO 8
 - CLOB Data in a Secondary File 8
 - Multiple Data Files for Single Table or MO Upload 8
- Data Upload Steps 8
 - Upload Data into a Table or Maintenance Object 9
 - Review Input Data File Spec 9
 - Create Compressed Input Data File 9
 - Switch Schema..... 9
 - Cleanup Target Table..... 10
 - Upload Data..... 10
 - Populate Key Table(s) 10
- Data Upload Orchestration..... 10
- Why, When, What, and How to Customize..... 12
 - Why..... 12
 - When..... 12
 - What..... 12
 - Control File 12
 - Additional Customization Items 13
 - How..... 14
- Sample Artifacts and Data Files 14

Introduction

This document contains data upload and implementation information relevant to the following Oracle Utilities cloud services: Customer Cloud Service, Meter Solution Cloud Service, Work and Asset Cloud Service, and Operational Device Cloud Service. 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 17.2. 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”.)

Conversion Process Overview

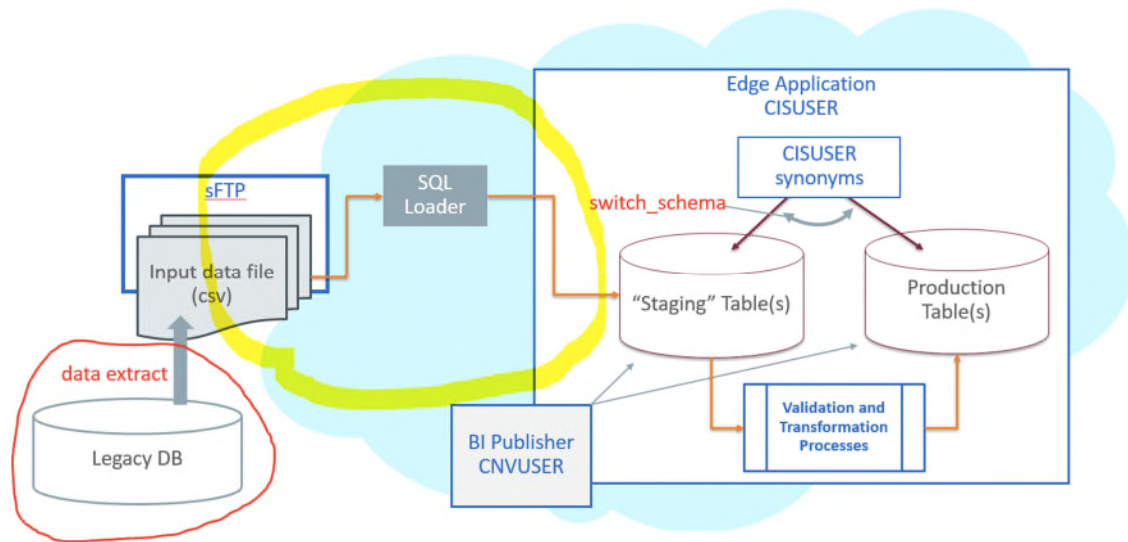
The goal of the Conversion Process is to migrate data from a legacy application into a target environment, and to begin running the application in the cloud. Due to cloud-related technical restrictions, legacy data cannot be uploaded directly into the software-as-a-service (SaaS) database. First, the legacy data must be extracted into file(s) and compressed. Second, the files must be uploaded to the cloud and then loaded into the target "staging" tables using Oracle SQL Loader. The data is validated, transformed, and finally inserted into "production" tables. Oracle BI Publisher may be used for ad-hoc reconciliation reports.



Data Upload Support on Cloud

The highlighted portion of the flow shown below is supported by Cloud Service Foundation (CSF). The legacy data extract and input file creation belong to the

implementation. Each application also “owns” conversion validation and transformation processes, as well as the definition(s) of the staging area.



Provided by Cloud Service Foundation (CSF)

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. Control Files are pre-generated by the system based on the metadata and conversion configurations. CSF allows users to generate control files, and it also provides a batch process that consumes the Input Data File(s), pulls the Control File from the internal server location, and calls SQL Loader.

CSF 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, Update Statistics, Populate Key Table* 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 all converted objects. The artifacts are generated based on the metadata and according to the conversion data upload setup
- **Conversion directory structure sFTP site.** Used for legacy extract file upload and SQL Loader output and reference files download
- **The instance of BI Publisher** that is connected to the database with read-only access to the combined staging and production data.
- **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 staging schema design and the product's 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. Each application also "owns" the definition of the staging data area and implements the *switch_schema()* stored procedure. The approach may differ from product to product.

An application may define "staging" as a separate database schema containing its own tables and views, to which the legacy data would be uploaded. Alternatively, a subset of tables in production could be used as a "staging" area, meaning the legacy data would be uploaded directly into production tables.

CSF is running against the [@application_user](#) schema. 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 CSF 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](#). It depends on the Conversion Tool design in the application. Once the *switch_schema()* stored procedure is executed, the assumption is that the synonyms in the [@application_user](#) schema are set to point to the "staging" tables, and the data upload may begin.

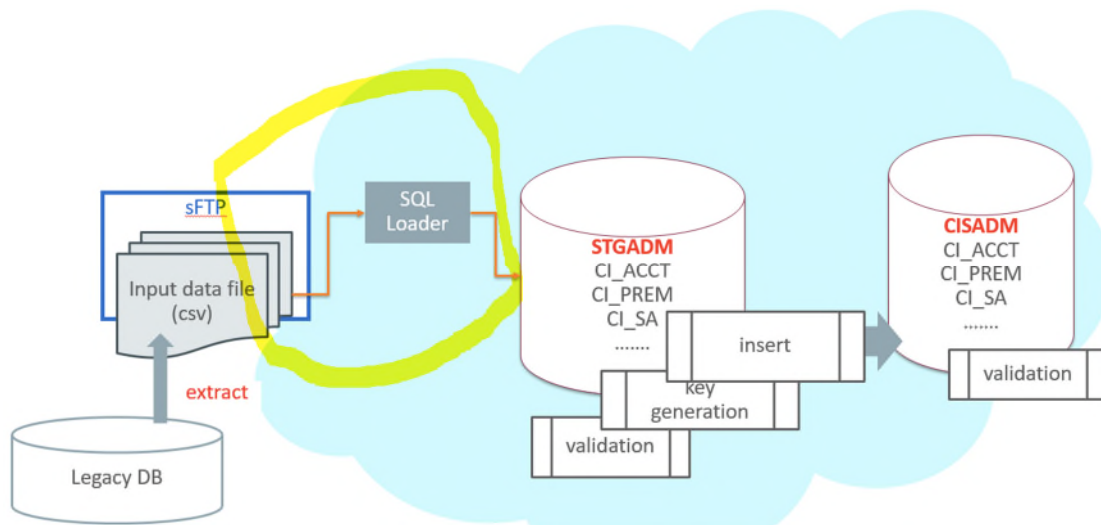
Two examples of how different application design the staging schema are described below.

OUCCS

The legacy data is uploaded into a set of tables in the staging schema (STGADM). The *switch_schema()* creates synonyms, read-only views, and tables in staging (STGADM) and production (CISADM) schemas.

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.

The highlighted portion of the flow is facilitated by CSF.

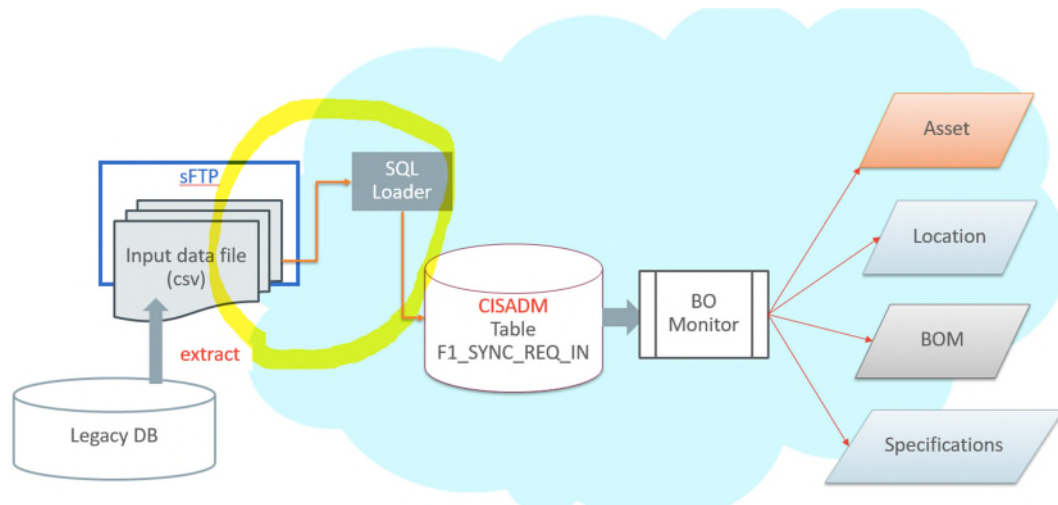


OUWACS

The legacy data is extracted and loaded into Initial Sync table(s) in production. There is no STGADM schema in the database. The *switch_schema()* does not manipulate synonyms, and they always point to the production tables.

Upon successful legacy data upload, the BO Monitor batches are processing the Initial Sync Requests and create master and transactional data.

The highlighted portion of the flow is facilitated by CSF.



Application Accelerators

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

- Conversion Instructions for unusual tables/MOs (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 easier conversion activities orchestration
- Sample reconciliation reports
- Other system, admin, and/or configuration 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 and map it 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 is connected to the read-only superposition of 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".

Preparing 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.**
 - Use the menu to navigate to Admin → Conversion Support and run the generator with option *Generate All? = Y*

- Download the **generated input data specifications** by running the batch K1-CNVUS.
- Fetch the specifications from the sFTP site.
- **Switch Schema** to redirect CSF to the staging data area.
 - Use the menu to navigate to the generator through Admin → Conversion Support.

The environment is now ready for the legacy data upload: conversion is **enabled**, the SQL Loader **Control Files are generated**, and the **synonyms** in the database schema are **pointing 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.
- Switching the schema does not affect conversion disabled/enabled indicator. It sets another internal flag that indicates whether the synonyms are pointing to "staging" or "production" area. The data upload is only allowed when the synonyms are pointing to "staging" (in other words, CSF is running in a "staging" mode).

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 Online Help and the specifications for more details)
- Compressed with gzip (see Online Help for details)

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.

Extract/Upload by Table or MO

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 OUAG/ application database.
- **MO-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 OUAF/ application database.

Both options are supported in CSF. Generate the artifacts and review the differences in the specifications.

CLOB Data in a Secondary File

The 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 often 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 CSF Data Upload process supports the upload into single target (Table or MO) 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 Online Help for more details.

Data Upload Steps

Upload Data into a Table or Maintenance Object

This process assumes that conversion artifacts have been generated, and the schema is switched to Production.

Review Input Data File Spec

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

NOTE: Alternatively, locate the appropriate specification on the sFTP site.

Create Compressed Input Data File

The specification defines the expected input file 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 MO-level extract, each record represents a row in one of the MO tables and the first 30 characters in each line contains the Table name.

Compress the file using gzip (See Online Help for details).

Switch Schema

Navigate to **Admin** → **Conversion Support** → **Switch Schema**, select **Conversion** from the dropdown and click OK.

Cleanup Target Table

Run batch K1-SCLTB, specifying the target Table or MO as a parameter.

Upload Data

Upload the input data file created above to the sFTP site.

Run batch K1-CNVLD, specifying the target Table or MO as a parameter.

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 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.

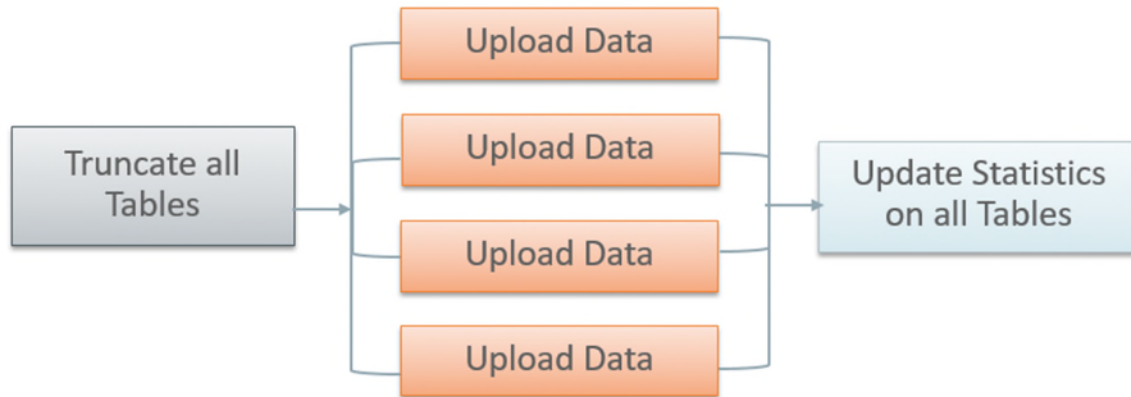
Data Upload Orchestration

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



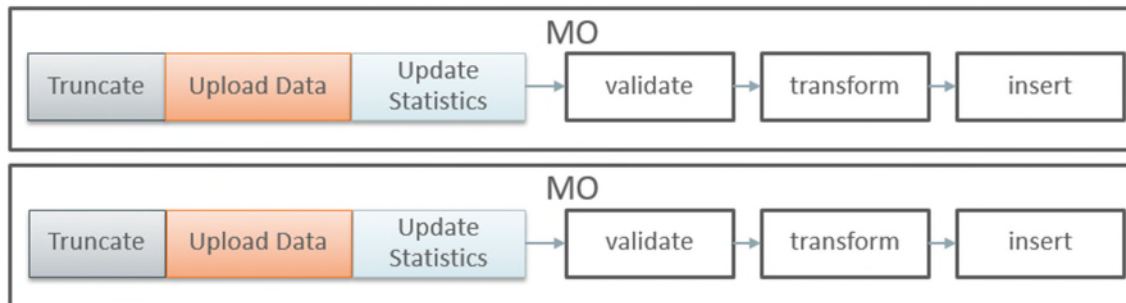
Multiple Tables or MOs Upload



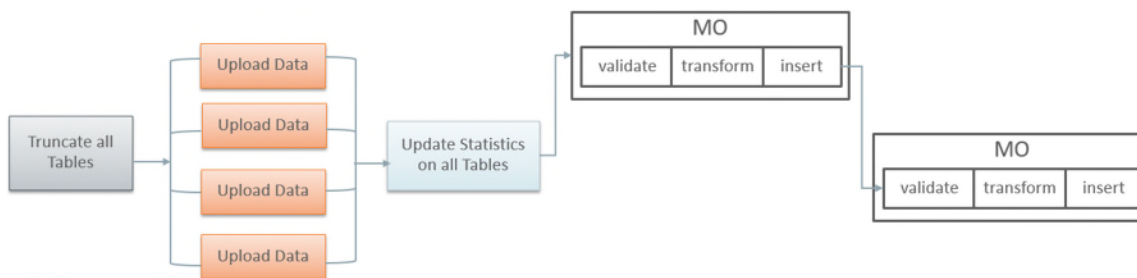
There are multiple strategies to orchestrate the entire conversion run and to build the optimal sequence of the conversion processes. Below are some of the many possibilities:

- 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 MO, some table by table
- process MO-s end-to-end simultaneously, if there are no inter-dependencies

Full Conversion Chain per MO, Parallel Run



Upload All + Subsequent Validate/Transform MOs



Why, When, What, and How to Customize

Conversion-related configurations define the expected extract file layout, 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.

Why

There are several reasons for customizing conversion configurations.

- fine-tune data upload performance
- handle unusual data volumes
- mark additional table(s) for conversion
- reduce creation of unnecessary input files

When

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

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 *_. See 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 is not necessarily contains the 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 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

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 MO's, 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 MO's, create new Conversion Instruction (Conversion Task Type) and add the Override Instruction for Table or MO on *Conversion Data Upload Master Configuration*

Note: Always regenerate Conversion Artifacts after making any configuration changes. Download the updated input file specifications.

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 or are available from Oracle Utilities Customer Success. 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
MO Customer Contact (CUST_CONTACT)	<p>Regular MO, CLOB fields in the main file.</p> <p>Configuration: same as Conversion Task Type K1-CNV-TABLE, but the <i>CLOB as Secondary File</i> indicator set to <i>false</i>.</p>
Table Adjustment (CI_ADJ)	<p>Regular table, CLOB field in the main file.</p> <p>Configuration: same as Conversion Task Type K1-CNV-TABLE, but the <i>CLOB as Secondary File</i> indicator set to <i>false</i>.</p>
Table Initial Sync Request (F1_SYNC_REQ_IN)	<p>Table with Multiple CLOBs as secondary files.</p> <p>Configuration:</p> <p>For table with multiple CLOBs, the special Conversion Task Type was created based on the BO K1-ConvArtMultClobTbITaskType.</p> <p>Override Control File (Managed Content) was created and used as a custom Control File.</p> <p>Review the sample and note that there is a conditional input data selection. Only records with BO = W1-CompositeSyncReqGISAsset would be uploaded. Custom Control File is justified if you have a requirement to manipulate the data during upload.</p> <p>Input Data File Specification:</p> <p>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.</p>

Target Object	Sample Description
<p>Tables Measurement and Measurement Component Identifier (D1_MSRMT, D1_MEASR_COMP_IDENTIFIER)</p>	<p>Regular tables, no CLOB data.</p> <p>Configuration:</p> <p>For D1_MEASR_COMP_IDENTIFIER - Conversion Task Type K1-CNV-TABLE</p> <p>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.</p>