Oracle Utilities Cloud Services

Implementation Guide

For 22B Releases

F58592-02

August 2022

Oracle Utilities Cloud Services 22B Implementation Guide

Copyright © 2017, 2022 Oracle and/or its affiliates.

Contents

Chapter 1

Part One

Implementation Guidelines

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Part Two

Data Conversion and Migration

Chapter 8

Chapter 9

Chapter 10

Chapter 11

Chapter 12

Chapter 13

Chapter 14

Part Three

File-Based Integration

Chapter 15

Chapter 16

Chapter 17

Part Four

Oracle REST Data Services

Chapter 18

Chapter 19

Part Five

Product-Specific Integrations

Chapter 20

Part Six

Web Services

Chapter 21

Chapter 1

Cloud Services Implementation Guide

Welcome to the Oracle Utilities Cloud Services Implementation Guide. This guide provides information about implementation of Oracle Utilities cloud services, including:

- Oracle Utilities Billing Cloud Service
- Oracle Utilities Customer Care and Billing Cloud Service
- Oracle Utilities Customer Cloud Service
- Oracle Utilities Generation Asset Manager Cloud Service
- Oracle Utilities Meter Solution Cloud Service
- Oracle Utilities Operational Device Cloud Service
- Oracle Utilities Rate Cloud Service
- Oracle Utilities Work and Asset Cloud Service

This document includes:

- Part One: Implementation Guidelines
- Part Two: Data Conversion and Migration
- Part Three: File-Based Integration
- Part Four: Oracle REST Data Services
- Part Five: Product-Specific Integrations
- Part Six: Web Services in Oracle Utilities Cloud Services

Part One

Implementation Guidelines

This section describes global implementation guidelines that apply to all Oracle Utilities cloud services running on Oracle Cloud Infrastructure (OCI), which includes

- Oracle Utilities Billing Cloud Service
- Oracle Utilities Customer Care and Billing Cloud Service
- Oracle Utilities Customer Cloud Service
- Oracle Utilities Generation Asset Manager Cloud Service
- Oracle Utilities Meter Solution Cloud Service
- Oracle Utilities Operational Device Cloud Service
- Oracle Utilities Rate Cloud Service
- Oracle Utilities Work and Asset Cloud Service

Note that these cloud services are all based on the Oracle Utilities Application Framework (OUAF), which supports many different configuration and extension methods, almost all of which are available for use in the cloud. This section provides recommendations for many aspects of set-up and operation of the services. Note that it assumes familiarity with OUAF concepts and tools.

In a nutshell, the top cloud service implementation rules to be aware of are the following:

- Use Groovy code in Scripts (not Java)
- Use existing data structures to extend the base model - such as Characteristics and the Fact table
- Use plug-in driven batch can be used in many scenarios for data fixes - this will ensure proper data validation

The guidelines in this are intended to help implementers to configure and run their cloud services efficiently.

This section include the following chapters:

- Chapter 7: Data Access and Analytics

Chapter 2

Post-Provisioning Setup

When a customer of one of the Oracle Utilities Cloud Services receives notification that their cloud service was provisioned, there are a number of tasks that have to be performed before they can start with normal implementation activities.

This chapter provides implementation guidelines related to post-provisioning setup, including:

- Initial Identity Management Setup
- Initial Object Storage Setup
- Initial Cloud Service Setup
- Language Pack Setup (Optional)

Initial Identity Management Setup

Initial set up related to Identity Management includes the following:

- Create a password for the cloud administrator and log in into each of the provisioned environments.
- Create a password for a special pre-defined user for the Process Automation Tool (user: K1IPROCESS).
- Define the initial set of user groups and assign appropriate access rights to each.
- Create Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) Identity Domains user mapping configuration in each of the cloud service environments (optional).

Refer to **Identity and Access Management with Identity Domains** in the *Oracle Utilities Cloud Services Administration Guide* for more information.

Initial Object Storage Setup

Initial set up related to Object Storage includes the following:

- Create a password for the cloud administrator in the Oracle Cloud Infrastructure (OCI) account (that includes object storage) and login into the OCI console.
- Create the default structure in object storage for the cloud service:
  - Create default Users, Groups and Policies.
  - Create default Compartments and Buckets.
- Configure the cloud service connections to object storage:
  - Create key rings in each cloud service environment.
  - Generate API keys in each cloud service environment.
  - Create File Storage extendable lookup values in each environment for each object storage compartment (with necessary details for object storage).
  - Register the API keys in the appropriate OCI users for each of the cloud service environments.

Refer to **Object Storage Setup** in the *Oracle Utilities Cloud Service Administration Guide* for more information.

Initial Cloud Service Setup

Initial set up for general cloud service use includes the following:

- Perform the process automation tool setup in each of the provisioned environments (see details below).
- Setup the security definitions for process automation in each of the environments. This is done by executing the Process Automation Security Setup BPA script and providing the following input:
  - Login ID of the Process Automation User: K1IPROCESS
  - Password of that user in IAM Identity Domains.
- Perform a Flush-All on the cloud service environments.

Refer to **Process Automation Tool** in the *Oracle Utilities Cloud Service Foundation Administrative User Guide* for more information.

Process Automation Tool Setup (for a new cloud service)

In order to set up the Process Automation Tool for all of your provisioned environments (for example, Dev, Test, Prod), run the following batch job in each environment:

- **Batch Code**: K1-IPAIS
- **User ID**: SYS_INT
- **Parameters**:
  - **SCRIPT01**: K1InvokePAS
  - **SCRIPT01_DATA**: INIT,<Internal Service Code>,<Current Env Code>,file-storage://OS-SHARED/CMA-Files,file-storage://OS-SHARED/CMA-Files,<Env List>

    where:

    - <Internal Service Code> = CCB/CCS/MSC/WAC (Please refer to the **Short Code** in the Process Automation Product extendable lookup for more details)
    - <Current Env Code> = DEV/DEV01..10/TEST/TEST01..10/PROD
    - <Env List> = environment codes separated by comma without spaces, for example: DEV,TEST,PROD

For example, when the job runs in the PROD environment for Customer Cloud Service (CCS) the parameters would be as follows:

```
SCRIPT01_DATA: INIT,CCS,PROD,file-storage://OS-SHARED/CMA-
Files,file-storage://OS-SHARED/CMA-Files,DEV,TEST,PROD
```

In order to finalize the process automation tools setup, run the process automation security setup in each environment. Refer to **Process Automation Tool** in the *Oracle Utilities Cloud Service Foundation Administration User Guide* for more information.

Users must flush their browser cache after this security set up.

Process Automation Tool Setup (after adding new environments to your cloud service)

If you add new environments to your cloud service, you must run a similar batch job in each environment.

For NEW environments, run the same job as after the initial provisioning (see above.)

For EXISTING environments (that existed before the addition of the new environments), run the following:

- **Batch Code**: K1-IPAIS
- **User ID**: SYS_INT
- **Parameters**:
  - **SCRIPT01**: K1InvokePAS
  - **SCRIPT01_DATA**: ADD,,<Current Env Code>,,,<Added Env List - for the current environment>

    where:

    - <Current Env Code> = DEV/DEV01..10/TEST/TEST01..10/ PROD
    - <Added Env List> = environment codes separated by comma without spaces, for example: DEV,TEST,PROD

For example, if you added DEV01 and TEST01 environments, when the job runs in the PROD environment the parameters should be as follows:

```
SCRIPT01_DATA: ADD,,PROD,,,DEV01,TEST01
```

In order to finalize the process automation tools setup, run the process automation security setup in each environment. Refer to **Process Automation Tool** in the *Oracle Utilities Cloud Service Foundation Administration User Guide* for more information.

Users must flush their browser cache after this security set up.

Language Pack Setup (Optional)

English (with the locale en-US) is provided as the default language by the system and all system metadata is delivered with English descriptions and labels. The system provides support for defining other languages and supports multiple languages in a single environment.

System users can use the system in their preferred language, as long as a translation into that language has been provided. A user sees the system in the language defined on their user record. If enabled, users can use the **Switch Language** zone to switch to another supported language real time.

Available Languages with Cloud Services

The following table lists languages available with Oracle Utilities cloud services.

Product	Release	User Interface / Online Help	Language
Oracle Utilities Customer Cloud Service	22B	User Interface	Arabic and Latin American Spanish
Oracle Utilities Customer Cloud Service	22B	Online Help	Arabic
Oracle Utilities Customer Cloud Service	22A	User Interface	Arabic and Latin American Spanish
Oracle Utilities Customer Cloud Service	22A	Online Help	Arabic
Oracle Utilities Customer Cloud Service	21C	User Interface	Arabic and Latin American Spanish
Oracle Utilities Customer Cloud Service	21C	Online Help	Arabic
Oracle Utilities Customer Cloud Service	21B+	User Interface	Latin American Spanish

Setup Instructions

Use the following procedure to set up a language an Oracle Utilities cloud service.

1. Define the Language Code for the language to be added and indicate that it is enabled. For details on this procedure, see **Defining Languages** in the *Administrative User Guide*.

   Note: Please use 'ARA' or 'ESA' as the Language Code for Arabic and for Latin American Spanish, respectively, when defining language codes for those languages.

2. Confirm that the desired language code is listed in the Available Languages with Cloud Services section above.

3. Run the "F1-LANG" batch control.

   - This process copies descriptions of all language-enabled tables from an existing translation (e.g., English). The copied values act as placeholders while the strings are translated into the new language. It is necessary to do this as a first step in order to create records using the new language code created in the previous step.

   - The batch process also updates the new language rows with the translated metadata descriptions from the language pack, if installed.

     Note: The language pack updates all language entries for base owned system data. If your implementation updates base owned labels and descriptions after applying a language pack, they will be overwritten the next time an updated language pack is applied. Note that most user facing labels and messages support defining an Override Label or Override Description. This information is not updated by the base product and should be utilized if your implementation desires a specific label or description.

Chapter 3

Security and Access

This chapter provides implementation guidelines related to security and access, including:

- Identity Management
- Server Access

Identity Management

Use of Identity Domains in OCI Identity and Access Management

In Oracle Cloud Infrastructure, cloud services are provisioned using Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) Identity Domains to manage user creation, application access, passwords, etc. This service at the 'Oracle Apps' tier is included with the Oracle Utilities cloud service subscription. See Identity and Access Management with Identity Domains for more information on IAM.

By default Oracle Cloud Infrastructure Identity and Access Management allow access to the application front-end from any IP address. There are capabilities in IAM Identity Domains to add sign-on policies that allow or deny IP addresses through the use of allowlists (though some features may require higher tier licensing).

User Provisioning with Identity and Access Management

Application users are added through Oracle Cloud Infrastructure Identity and Access Management (IAM) which are used to manage the user lifecycle (i.e. you can disable a user, or reset a user's password in IAM). The access rights of the user within the application are controlled using the settings on the cloud service User record. Identity and Access Management uses Application Roles and Groups: a user must be linked to the Application Roles that they need access to. This linking can also be 'indirect' by linking a new user to a Group which has access. Creation of cloud service User records is done 'just-in-time' - upon the first login to the application, after authentication via Identity and Access Management, a call is made to verify access to the application, and using the returned information including the user's IAM Groups, a template user in the cloud service can be found and used as the 'copy from' source.

Instructions: The security administrator should create an initial User record with full access to the cloud service (including administration functionality). This user should be used to configure "Template Users" and mappings to or IAM Groups. See **Identity and Access Management with Identity Domains** in the *Oracle Utilities Cloud Services Administration Guide* for more information. Note that the Cloud Service Foundation also provides several Template Users that have necessary access for process automation.

Server Access

While server access is restricted exclusively to members of the Oracle Cloud Infrastructure and Oracle Utilities Development Operations (DevOps) teams, logs are available to users. See the Server Logs - Online and Batch section in Chapter 7: Operational Guidelines for more information.

Chapter 4

Configuration Tools

This chapter describes specific implementation guidelines related to use of Oracle Utilities Application Framework Configuration Tools, including:

- Customization Tools Summary
- Algorithm Types and Algorithms
- Application Environments
- Creating Batch Processes

Customization Tools Summary

While most of each cloud service application's customization options are supported, some are not and others may be limited in certain areas. The table below outlines configuration options and their availability when implementing Oracle Utilities cloud services.

Category	Option	Supported?	Comment
Business Entities	Add custom business objects for product maintenance objects.	Yes	Assuming the Maintenance Object supports Business Object functionality.
	Extend a product business object's structure and rules.	Yes	See the Algorithm Types and Algorithms section for more information.
	Add custom maintenance objects.	No	Creation of new tables is not supported. See the Database Access section for more information. Use of the SDK tool to generate Java artifacts is not supported. See the Algorithm Types and Algorithms section for more information.
User Interface	Add a custom portal.	Yes	Restricted to a single tab page in 20C and previous versions. Additional tab pages are supported as of release 21A.
	Extend a product portal with custom zones.	Yes	
	Extend a product multi-query search with custom query options.	Yes	
	Customize a product menu. This includes adding new custom menu lines, hiding and reordering lines.	Yes	
	Add custom indexes to support custom queries	No	See the Database Access section for more information.
Batch processes	Add a custom batch process.	Yes	The program cannot be written in Java. See the Creating Batch Processes section for more information. The process may only access designated locations in Object Storage. See the File Access - Cloud Object Storage section for more information.

Category	Option	Supported?	Comment
Web Services	Add custom inbound and outbound web services.	Yes	Use Outbound Messaging and Inbound Web Services. The services cannot rely on XSL transformations to occur in the cloud. See the File Access - Cloud Object Storage section for more information.
Reports	Add stored procedures to support custom reports.	No	See the Database Access section for more information.
	Run the high volume extract reports like Bill Print via Analytics Publisher.	No	See the *Oracle Utilities Cloud Services Frequently Asked Questions Guide* for more information.
	Load and extract data from external data sources (such as xml data models, and so on), and/or write to Oracle Cloud Object Storage.	No	Analytics Publisher can only access the primary database that is provisioned with the associated cloud service (such as CCS, MSCS, WACS, BCS, and so on).

Algorithm Types and Algorithms

New algorithm types and algorithms can be created during implementation using Scripts. Custom Java-based algorithm types are NOT permitted.

Write custom algorithm types using either Groovy or Oracle Utilities Application Framework's XML-based scripting. Refer to **Defining Algorithms, Plug-In Scripts, and Using Groovy within Scripts** in the *Administrative User Guide* or online help for information about creating algorithms using Groovy.

Key Guidelines of Groovy scripting are:

- Review the third party groovy allowlist (available within the application)
- Be careful with goto statements - it is easy to create endless loops
- Review SQL Function Allowlist (Refer to F1-SQLFunctionWhiteList Managed Content)
- Update/Delete SQL Statements are not allowed
- Explain Plan of the query(s) needs to be examined for all SQLs written in custom code.

When crafting custom SQL queries, you must consider performance. Run explain on all of your SQLs using rule hint before delivering code range scans and nested loops only. Plans with 'table access' are not acceptable. Use the SQL Developer Web toolset to check SQL.

Application Environments

Each cloud service by default comes with three environments designated as Development, Test, and Production. Test and Production are sized as full-sized environments based on the billable metric of the subscription, while Development is a smaller environment.

Customers can request additional non-production environments through the initial sales order or in a subsequent order for an additional subscription. When asking for more environments, the names of the base and additional environment are predefined and cannot be changed.

Environment Names and Codes

Cloud service environments have an environment code and name. The environment code is used to identify the environment and enable migration processes (such as configuration migrations) between the environments. The environment code is also used at installation/provisioning time. The table below lists all the possible environments that can be provisioned for each cloud service.

Environment Code	Name	Type	Default	Additional
DEV	Development	Development	Yes	No
TEST	Test	Test	Yes	No
PROD	Production	Production	Yes	No
DEV01..DEV10	Development 1 .. 10	Development	No	Yes
TEST01..TEST10	Test 1 .. 10	Test	No	Yes

Application / Environment Access and URL Tokens

When environment provisioning is complete, customers / implementers will receive a list of links to the various product environments included with their subscription. There are cases in which cloud service applications need to access other cloud service or on-premises applications or other environments, such as:

- Data/Configuration Migration
- Data Conversion
- Redirecting a user to another application as part of a business process transaction that is integrated across products
- Invoking web services of a different application (another cloud service or SOA for existing SOA-supported cloud integrations)
- Invoking a web service of the same application in a different environment. This type of communication is used to help automate inter-environment processes like configuration migrations. See the Code and Configuration Migration section in Chapter 7: Operational Guidelines for more information.

The following URL Tokens are available for direct navigation or web service calls for each cloud service.

Token	Description
EXT_PUB	Prefix token that should be used to reference external addresses in Message Senders. For example, to reference paymentcorp.payusa.com endpoint URL you would need to use the following notation: @EXT_PUB@paymentcorp.payusa.com. Note: The target URL must be on the allowlist.
CCS_WS CCS_ONLINE	Customer Cloud Service address for web service calls. Customer Cloud Service address for online access.
MSCS_WS MSCS_ONLINE	Meter Solution Cloud Service address for web service calls. Meter Solution Cloud Service address for online access.
WACS_WS WACS_ONLINE	Work and Asset Cloud Service address for web service calls. Work and Asset Cloud Service address for online access.
AICS_ONLINE	Analytics Insights Cloud Service (aka DataRaker) address for online access.
INT_WS	Invoke SOA web services (for integration via SOA)
BI_PUBLISHER_ADMIN	Analytics Publisher address for web services calls.
BI_PUBLISHER_ADMIN_INT	Used with Reporting Options with Analytics Publisher as "Reporting Server From App Server" for Batch reporting.
BIP_DEF_DIR	Used as soft parameter value ("Output Directory") to Analytics Publisher extract algorithm type.
BIP_DEF_PATH	Used as soft parameter value ("Report Absoluted Path") to Analytics Publisher extract algorithm type.
DEV_WS, DEV01_WS-DEV10_WS TEST_WS, TEST01_WS-TEST10_WS PROD_WS	Web service addresses for all possible environments. For example, DEV_WS can point to Customer Cloud Service in the DEV domain while PROD_WS will point to the same application but in the PROD domain.

Usage Examples:

- The outbound message sender on the Customer Cloud Service configured for invoking a Meter Solution Cloud Service Inbound Web Service service named "ABC" (in production) will be have the URL definition of @MSCS_WS@abc.
- The value of @MSCS_WS@ will be different in each environment so that the same token can be used in all environments, and the runtime value translation will be based on the environment invoking the call.
- Internal-facing tokens such as DEV_WS can be used for inter-domain communications, such as the automation of configuration migration between product domains. These tokens are used by the Process Automation Tool within Cloud Service Foundation.

- External facing addresses will use the @EXT_PUB@ prefix, for example: @EXT_PUB@paymentcorp.payusa.com/api/int01/addPayment
- The port is not required in the URL definition when using EXT_PUB as all outbound calls from cloud services are sent via Https and port 443 is implied.

Creating Batch Processes

Custom batch jobs can be written using the plug-in driven batch job functionality supported by the Oracle Utilities Application Framework. There are three broad categories of batch jobs that may be implemented using plug-in driven batch.

- **Ad-hoc Processing**. This covers any batch job that should select records in the system and perform some type of logic for each record.
    - The system provides a Select Records plug-in for retrieving the records in the system based on criteria. This plug-in requires the selection SQL (properly tuned) to be defined as a parameter. Logic in the plug-in script may be used to set filter criteria if needed. The plug-in script may be written using XPath scripting.
    - The system also provides a Process Record plug-in where each record may be reviewed and some appropriate action may be performed. This plug-in may be written in either XPath or Groovy scripting.
    - See also the Data Fix with Plug-in Driven Batch section in Chapter 7: Operational Guidelines for more information.
- **Extract a Batch of Records**. This covers any batch job that produces an extract of records. The same plug-in spots described for Ad-hoc processing are applicable here. The Select Records plug-in is used for selecting the records eligible for extraction, for example from as staging table. The Process Record plug-in us responsible for returning the data to be written to the extract for each record. This plug-in may be written in either XPath or Groovy scripting. However, because the output of the plug-in is one or more schema objects to include in the extract, XPath scripting may be better suited.
- **Upload Records from a File**. This covers any batch job that needs to read a file and create records in the system based on the content. The system provides a File Upload plug-in spot. This plug-in is responsible for calling appropriate APIs to read the content of the file and store the data in appropriate tables, for example a staging table. This type of plug-in must use Groovy as the APIs are not accessible using the XPath scripting language.

Chapter 5

Data Conversion Guidelines

This chapter provides general guidelines related to data conversion.

Data conversion refers to the migration of data from a client's legacy system (on-premise or cloud) to the application database(s) within Oracle Utilities cloud services. Since no direct access is permitted to the application database, data conversion support is provided to facilitate SQL Loader-based data upload via Cloud Service Foundation tools. Staging tables cleanup is also supported.

Refer to Part Two: Data Conversion and Migration in this guide for more detailed information about data conversion and migration.

Data Conversion Approach

The implementation project is expected to extract the legacy data into flat files and upload these files to a specific location on the cloud, and then to run a sequence of batch processes that moves the data into corresponding tables in the special Staging database schema. The subsequent processing of the staging data, along with its insertion into production tables, is specific for each cloud service. Refer to Data Conversion and Migration for more information about specific data conversion and migration scenarios. Sample upload files are also available from the Oracle Utilities Documentation library for the relevant cloud service.

It is recommended to start with small set of data covering most of the unique / critical scenarios, perform the object / FK validations and try to resolve the data quality conversion issues by comprehensive testing (both online and batches). Gradually increase the data volume to avoid running full scale of converted data early resulting in application errors (invalid data will often lead to a lot of 'noisy errors' that can be avoided with this approach).

In the data extract populate the ILM_ARCH_SW as follows:

- Set ILM_ARCH_SW with a value of “Y” for high-volume tables. In specific, the ILM_ARCH_SW field MUST be set to “Y” for the following tables used with Oracle Utilities Customer Cloud Service and Oracle Utilities Meter Solution Cloud Service:
  - D1_DVC_EVT (Device Event)
  - D1_INIT_MSRMT_DATA (Initial Measurement Data)
  - D1_USAGE (Usage Transaction)

- Set ILM_ARCH_SW with a value of “N” for all other tables

Data Conversion Tips

The following high-level tips are important for data conversion efforts:

1: Data Upload Indexes and Constraints

Data conversion is performed by processing legacy data extract files using SQL Loader. During the data upload, the indexes and constraints are disabled and duplicate keys are not validated. See Oracle SQL Loader Documentation for details.

Please ensure that you cleanse the data extract file and remove duplicates prior to the upload.

2: Key Tables in Staging Area

The Key tables in the staging area are not populated automatically. The Key Table data has to be created with the corresponding Environment ID and then uploaded as a separate extract. Refer to Data Conversion and Migration for more information.

3: CLOB Data Upload with Secondary Files

The CLOB data upload with secondary files is not supported when there are multiple CLOB columns in the table. Configure the conversion task type to include CLOB data in the main extract, amend Conversion Master Configuration, and regenerate Conversion Artifacts. Refer to Data Conversion and Migration for more information.

Chapter 6

Integration Guidelines

This chapter provides guidelines related to integration with Oracle Utilities cloud services including:

- Integration Methods
- Integration Middleware
- Allowlisting

Integration Methods

The primary integration methods supported with cloud services are (a) inbound and outbound files, and (b) inbound and outbound web services. Other protocols and methods (JMS, SQL Net, etc.) are not currently supported.

Besides standard Oracle Utilities Application Framework integration modules, no additional extract, transform, and load (ETL) capabilities or middleware are provided with cloud service offerings. Oracle Cloud middleware solutions—such as SOA Cloud Service (available via Platform-as-a-Service) or Integration Cloud Service —need to be licensed to address advanced integration requirements such as complex ETL, orchestration, etc. Alternatively an on-premise middleware solution could be used.

Integration Method: File-Based

Inbound File Processing: Files are uploaded to Object Storage and processed via scheduled batch jobs. Implementation-specific file parsing and processing logic can be introduced using browser-based Oracle Utilities Application Framework tools. Refer to **Uploading Records in the Plug-in Driven Background Processes** section in the *Administrative User Guide* or the online help for more information. Please also review the File Access - Cloud Object Storage section in Chapter 7: Data Access and Analytics.

Outbound File Processing: File-based extracts can be generated and made available for download and further processing. Implementation-specific file processing and generation logic can be introduced using browser-based Oracle Utilities Application Framework tools. Refer to **Processing System Records in the Plug-in Driven Background Processes** section in the *Administrative User Guide* or the online help for more information.

Large-volume data conversion and loading is supported. See Chapter 5: Data Conversion Guidelines for more information.

Integration Method: Web Services

Web services are supported through Inbound Web Services (IWS) and Outbound Messages. All inbound and outbound web services communication must be HTTPS. Refer to **Inbound Web Services** and **Outbound Messages** in the *Administrative User Guide* or the online help for more information.

Note that in order to call Inbound Web Services, you must provide a user/password for authentication and authorization (the user must be defined in Identity and Access Management Identity Domains with the 'AppWebServices' Application Role and as an application User). Inbound Web Services support both SOAP and REST. Outbound Messages may only reference public IP addresses, and those addresses must be on an 'allowlist' (which can be provided to Cloud Operations via a service request ticket).

For integrations that involve outbound synchronization to other systems driven by online activity, real-time synchronous outbound messages are not recommended. Rather use the business object batch monitor processing on a frequent basis to process queued messages. This involves using the deferred monitor batch set on the PENDING state of the Sync Request so that message processing occurs asynchronously.

SSL certificates must be created using certification authority. Self-signed SSL certificates are not supported. Also reference the How to access SOAP and REST Services in Oracle

Utilities Enterprise Cloud Services document on My Oracle Support (Document ID 2564697.1).

Upload and attachment of implementation-specific xsl files to process xml payloads is supported through the **Managed Content** portal for relevant product or cloud service. Refer to **Maintaining Managed Content** in the *Administrative User Guide* or the online help for more information.

Integration Middleware

While file-based integration does not require middleware, often real-time integration benefits from the use of a middleware platform to facilitate message delivery, error handling, and data transformation. With Oracle Utilities cloud services, there are several different middleware options which may be useful, and in some cases prebuilt integrations are available to integrate Oracle applications. This section describes several middleware options (note: these are not included with your cloud service subscription).

Integration Middleware: Oracle Integration Cloud (OIC)

Oracle Integration Cloud Service (OIC) is an integration platform offered as Software-as-a-Service - it provides a modern web-based user interface to set up integration connection points, and uses application catalogs of available services provides data mapping capabilities and statistics on message flows. The Oracle Integration Cloud suite includes additional analytics and other tools. Oracle Utilities uses OIC as an integration platform to link with other Oracle applications such as Oracle Field Service, Fusion ERP, and others. OIC can also handle file-based integrations. As a cloud offering Oracle supports upgrades to the service, but note that Disaster Recovery is not currently part of the standard offering.

Integration Middleware: Oracle SOA Suite on Marketplace - Platform-as-a-Service

This option uses the Oracle SOA Suite on Marketplace hosted as a Platform-as-a -Service (PaaS) - thus allowing for full control and development capability. As a PaaS, the customer is responsible for managing the software, updates, etc.

Integration Middleware: Oracle SOA Suite On-Premises

This option uses the Oracle SOA Suite hosted on premises- thus allowing for full control and development capability. The customer is responsible for managing the software, updates, etc.

Allowlisting

Allowlisting is required to specify allowable access destinations on the public internet.

IP Allowlisting

IP Allowlists enable customers to control how data flows into or out of their SaaS environments.

Outbound Traffic

Outbound traffic is controlled via allowlist of IP addresses. Only HTTPS traffic is allowed to port 443.

Configuring IP Allowlists

To configure IP allowlists, customers must log a service request and follow the steps outlined in the **Cloud Operations** section of the *Oracle Utilities Cloud Services Administration Guide* to provide configuration details.

Chapter 7

Data Access and Analytics

This chapter provides guidelines related to data access and analytics, including:

- Analytics Publisher
- Database Access
- Reports and Queries
- File Access - Cloud Object Storage

Analytics Publisher

Oracle Utilities Analytics Visualization is included in the cloud service subscription, available via a separate URL for each environment.

Analytics Publisher is available and included in the service as a reporting/query tool.

> **Note**: Analytics Publisher and Oracle Utilities Analytics Visualization extract and display data from the primary database only.

SQL Developer Web is also available and included in the service for querying the database (see **SQL Developer Web** in the *Oracle Utilities Cloud Services Implementation Guide* for more information).

Data extraction is supported via the Generalized Data Extract and Specialized Data Extract functionality, and/or DataConnect (CCS & MSCS only) may be a starting point for extraction of data for a BI/reporting tool such as Cognos.

Analytics Publisher can only access the primary database that is provisioned with the associated cloud service (such as CCS, MSCS, WACS, BCS, and so on). External data sources (such as xml data models, and so on) are currently NOT supported.

Database Access

No direct access is permitted to the application database either through Toad, SQL Developer, or command line utilities. This also means that you **cannot** create new tables or related data including new Maintenance Objects, custom audit tables, and database links.

Query access is supported both in Analytics Publisher and SQL Developer Web (see **SQL Developer Web** in the *Oracle Utilities Cloud Services Implementation Guide*), which are available as part of the cloud deployment. For more information about Analytics Publisher, see https://www.oracle.com/middleware/technologies/analytics-publisher.html.

Analytics Publisher deployment includes a JDBC data source configured with credentials that allow access to read-only synonyms in the production schema.

Note that in some cases it may be feasible to create a custom zone in the application to provide online display to view data.

Reports and Queries

Reports and queries can be run using Analytics Publisher, which is included with the cloud service deployment. Analytics Publisher deployment includes a JDBC data source configured with credentials that allow access to read-only synonyms in the production schema. Note that there are several output formats, and we have found that PDF performs best for larger reports.

In order to configure reports within cloud services via Analytics Publisher, implementations are required to configure the **Reporting Options** and relevant reporting Algorithms (refer to Application / Environment Access and URL Tokens for parameter values for base algorithms types).

Reporting options with default configuration includes the following:

- Reporting Server from App Server = @BI_PUBLISHER_ADMIN_INT@
- Reporting Engine User ID = <use the user name having Analytics Publisher role assigned>
- Reporting Engine Password = <use the password for above mentioned user>
- Reporting folder = ccs <default reporting folder in Analytics Publisher but implementations can change it, if needed.>
- Reporting Server from Browser= @BI_PUBLISHER_ADMIN@

Oracle Utilities cloud services also offer the option of using SQL Developer Web and Oracle Utilities Analytics Visualization for reports and queries.

File Access - Cloud Object Storage

All inbound and outbound file-based data is staged in Oracle Cloud Object Storage (a separate service that the customer must license).

Cloud Object Storage involves creation of a set of compartments and buckets (each compartment can have many buckets, and have child compartments), and the compartments are represented within the application as values for the File Storage Configuration (F1-FileStorage) extendable lookup. If the customer creates a new compartment, a new value needs to be specified in the extendable lookup, with OCID references to the user/tenancy/compartment. Once that is set up, the application can reference particular buckets as needed.

The format for Object Storage paths is as follows:

```
file-storage://<Extendable Lookup value for Compartment>/<bucket>
- example: file-storage://OS-SHARED/CMA-Files
```

Typically there are a few places where a 'path' to an Object Storage bucket can be specified, such as on batch job parameters, and some Master Configurations. Refer to the **Object Storage Setup** in the *Oracle Utilities Cloud Services Administration Guide* for more information. Refer to the Object Storage documentation for more information about Oracle Cloud Object Storage.

Uploading and Downloading File To and From Object Storage

There are two main options of exchanging files between Oracle Cloud Object Storage and the outside world.

- The Oracle Infrastructure Console User Interface which allows authorized users to upload or download files to and from object storage buckets.
- The object storage APIs which allow other applications to interact with object storage and exchange files as well as other actions.

When customers or implementers need to upload or download files in bulk, the option of the Oracle Infrastructure Console User Interface can be cumbersome.

There are Oracle and 3rd party tools that customers can install (for example, the Oracle Storage Gateway) that offer integration solutions for file exchange with object storage.

The Oracle Integration Cloud (OIC) offers an Object Storage 'adapter' which allows OIC to move files/objects in or out of Object Storage. This uses the OIC REST Adapter.

Any other client that can make REST calls could also interact with Object Storage.

Part Two

Data Conversion and Migration

This section provides data conversion, migration, 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 section includes:

Chapter 8

Overview

The chapters in this section provide guidelines for migrating and/or converting data between application environments, including moving data from existing applications into Oracle Utilities Customer Cloud Service or Oracle Utilities Meter Solution Cloud Service. Existing applications can include legacy applications as well as on-premises implementations of Oracle Utilities applications such as Oracle Utilities Customer Care and Billing and Oracle Utilities Meter Data Management.

See **Conversion** in the *Application Framework Administrative User Guide* for information about the general conversion process.

This overview chapter includes:

- Data Conversion and Migration Overview
- Terms and Definitions
- Database Tables
- Additional Information

Data Conversion and Migration Overview

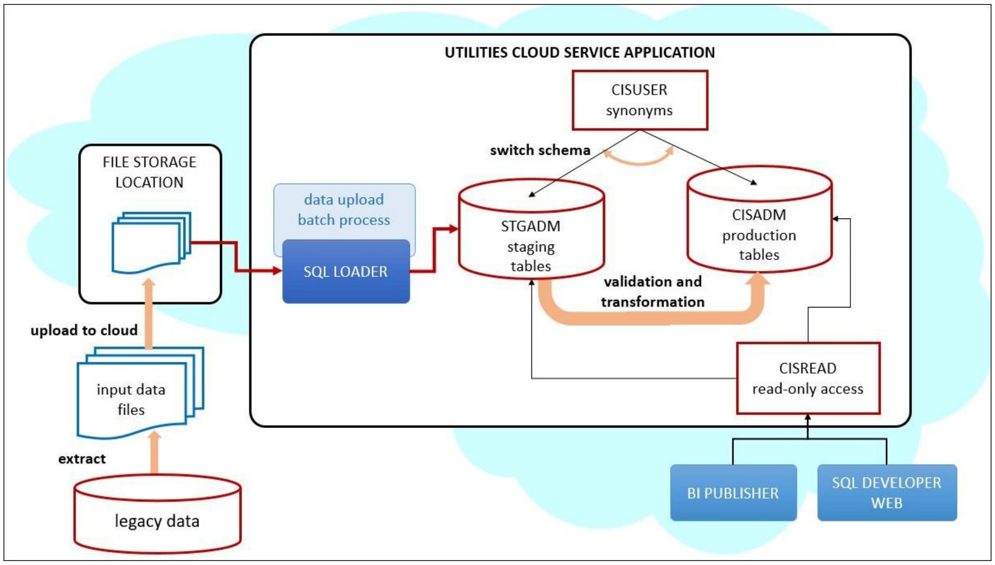
This section provides an overview of data conversion and migration, including:

- Conversion Process Overview
- Implementation Effort
- What Is in the Newly Provisioned Environment?
- Data Conversion and Migration on Cloud

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.

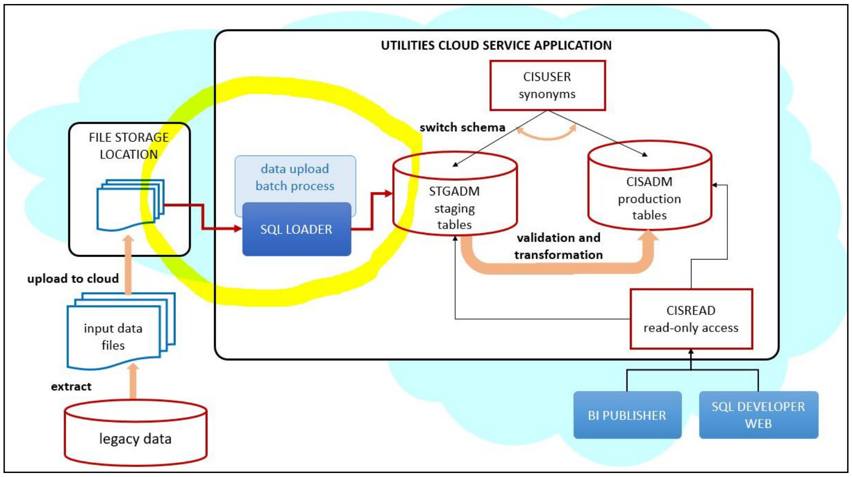
The default configurations are suitable for typical table volumes and common data formats. If your implementation does not include extremely large data volumes, special data formats, or other idiosyncratic requirements, the default setup can be used "as is".

Data Conversion and Migration on Cloud

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

- Provided by Cloud Service Foundation
- Provided by Applications

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 requiremenets (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

Terms and Definitions

The processes described in this section use the following terms:

Term	Definition
Synchronize	Data is created through a data synchronization process. For example, Usage Subscriptions used by Meter Data Management functionality are synchronized from Service Agreements used by Customer Care and Billing functionality. See Data Synchronization in the Application Framework Administrative User Guide for information about the data synchronization process.
Upload	Data is uploaded directly into the database from a legacy application.
Bind	Direct relationships between records, such as between devices and assets, are created via batch processing. See Binding MDM Device ID and ODM Asset ID for more information.
Import	Data is imported from an existing Oracle Utilities application such as Oracle Utilities Customer Care and Billing and Oracle Utilities Meter Data Management.
Legacy CIS	A legacy customer information system from which data will be migrated/converted.
Legacy MDM	A legacy meter data management application from which data will be migrated/converted.
CCB (CC&B)	Oracle Utilities Customer Care and Billing
MDM	Oracle Utilities Meter Data Management
ODM	Oracle Utilities Operational Device Management

Database Tables

The processes described in this section reference a number of different types of database tables. These are described below.

Production Tables: Tables used by application when running in a production environment. Examples of production tables include CI_SP (Service Point - CCB), D1_SP (Service Point - MDM), W1_ASSET (Asset). Production tables are accessible through the Table portal in the Customer Cloud Service application.

Staging Tables: Tables used to facilitate import and migration into the product database. Staging tables are not accessible through the Customer Cloud Service application user interface.

Key Tables: Tables used to facilitate generation of keys. Examples of key tables include CI_SP_K (Service Point Key), D1_SP_K (Service Point - MDM), W1_ASSET_K (Asset Key). Key tables are accessible through the Table portal in the Customer Cloud Service application.

Scope and Assumptions

The processes described in this section are based on the following scope and assumptions:

Legacy data has been loaded to the staging tables. See **Data Conversion Support for Cloud Implementations** in the *Oracle Utilities Cloud Service Foundation Administrative User Guide* for information about loading data into staging tables.

The process described in this document covers Meter Data Management conversion delivered for converting legacy master data such as Contacts Devices, Device Configurations, Measuring Components, Service Points, Install Events, and Usage Subscriptions.

The process described in this document includes Operational Device Management conversion pertinent to deployments of Oracle Utilities Customer Cloud Service that include customer, meter and operational device (asset) functionality.

Additional Information

Refer to the following documentation for additional information about data conversion and migration with Oracle Utilities cloud services:

- *Oracle Utilities Cloud Service Foundation Administrative User Guide*

These documents can be found in the **Supporting Cloud Service Guides** section on the Oracle Utilities Customer Cloud Service or Oracle Utilities Meter Solution Cloud Service documentation website.

Chapter 9

Data Conversion and Migration Scenarios

This chapter outlines the ways in which specific types of data are migrated/converted in a number of specific application configurations, including:

- Legacy Customer Information System to C2MO
- Legacy Customer Information System to C2M
- Legacy Meter Data Management to MDM/ODM
- Customer Care and Billing to C2M
- Customer Care and Billing to C2MO
- Customer Care & Billing and Meter Data Management to C2M
- Customer Care & Billing and Meter Data Management and Operational Device Management to C2MO
- Customer Care & Billing and Meter Data Management to C2MO
- Meter Data Management to MDM/ODM

The following abbreviations are used in the scenarios described in this chapter.

Abbreviation	Description
C2MO	Customer Cloud Service including Customer Care and Billing, Meter Data Management, and Operational Device Management functionality.
C2M	Customer Cloud Service including Customer Care and Billing and Meter Data Management functionality.
MDM/ODM	Meter Solution Cloud Service including Meter Data Management and Operational Device Management functionality.

Notes:

- The tables following layout the data in each of the functional areas (Customer Care and Billing, Meter Data Management, and Operational Device Management) as appropriate.
- Data of corresponding types are aligned in the same row in each table. For example, Person data in CCB corresponds to Contact data in MDM.

Legacy Customer Information System to C2MO

Customer Care and Billing Data		**Meter Data Management Data**		**Operational Device Data**	
Person	Upload from Legacy CIS	Contact	Synchronize from CCB		
Account	Upload from Legacy CIS				
Service Agreement (SA)	Upload from Legacy CIS	Usage Subscription (US)	Synchronize from CCB		
Premise	Upload from Legacy CIS				
Service Point (SP)	Upload from Legacy CIS	Service Point (SP)	Synchronize from CCB	Node	Synchronize from MDM
		Device	Upload from Legacy CIS	Asset	Upload from Legacy CIS
			Bind Device IDs		Bind Device IDs
		Device Configuration	Upload from Legacy CIS		
		Measuring Component	Upload from Legacy CIS		
		Install Event	Upload from Legacy CIS Has Foreign Keys to Service Point that do not exist in staging because they were synchronized.	Asset Node	Synchronize from MDM

Legacy Customer Information System to C2M

Customer Care and Billing Data		**Meter Data Management Data**	
Person	Upload from Legacy CIS	Contact	Synchronize from CCB
Account	Upload from Legacy CIS		
Service Agreement (SA)	Upload from Legacy CIS	Usage Subscription (US)	Synchronize from CCB
Premise	Upload from Legacy CIS		
Service Point (SP)	Upload from Legacy CIS	Service Point (SP)	Synchronize from CCB
		Device	Upload from Legacy CIS
		Device Configuration	Upload from Legacy CIS
		Measuring Component	Upload from Legacy CIS
		Install Event	Upload from Legacy CIS

Legacy Meter Data Management to MDM/ODM

Meter Data Management Data		**Operational Device Management Data**	
Contact	Upload from Legacy MDM		
Usage Subscription (US)	Upload from Legacy MDM		
Service Point (SP)	Upload from Legacy MDM	Node	Synchronize from MDM
Device	Upload from Legacy MDM Bind Device IDs	Asset	Upload from Legacy MDM Bind Device IDs
Install Event	Upload from Legacy MDM	Asset Node	Synchronize from MDM

Customer Care and Billing to C2M

Customer Care and Billing Data		**Meter Data Management Data**	
Person	Import from CCB	Contact	Synchronize from CCB
Account	Import from CCB		
Service Agreement (SA)	Import from CCB	Usage Subscription (US)	Synchronize from CCB
Premise	Import from CCB		
Service Point (SP)	Import from CCB	Service Point (SP)	Synchronize from CCB
		Device	Upload from CCB
		Device Configuration	Upload from CCB
		Measuring Component	Upload from CCB
		Install Event	Upload from CCB

Customer Care and Billing to C2MO

Customer Care and Billing Data		Meter Data Management Data		Operational Device Management Data	
Person	Import from CCB	Contact	Synchronize from CCB		
Account	Import from CCB				
Service Agreement (SA)	Import from CCB	Usage Subscription (US)	Synchronize from CCB		
Premise	Import from CCB				
Service Point (SP)	Import from CCB	Service Point (SP)	Synchronize from CCB	Node	Synchronize from MDM
		Device	Upload from CCB Bind Device IDs	Asset	Upload from CCB Bind Device IDs
		Device	Upload from CCB		
		Measuring Component	Upload from CCB		
		Install Event	Upload from CCB	Asset Node	Synchronize from MDM

Customer Care & Billing and Meter Data Management to C2M

Customer Care and Billing Data		Meter Data Management Data	
Person	Import from CCB	Contact	Import from MDM
Account	Import from CCB		
Service Agreement (SA)	Import from CCB	Usage Subscription (US)	Import from MDM
Premise	Import from CCB		
Service Point (SP)	Import from CCB	Service Point (SP)	Import from MDM
		Device	Import from MDM
		Device Configuration	Import from MDM
		Measuring Component	Import from MDM
		Install Event	Import from MDM

Customer Care & Billing and Meter Data Management and Operational Device Management to C2MO

Customer Care and Billing Data		**Meter Data Management Data**		**Operational Device Management Data**	
Person	Import from CCB	Contact	Import from MDM		
Account	Import from CCB				
Service Agreement (SA)	Import from CCB	Usage Subscription (US)	Import from MDM		
Premise	Import from CCB				
Service Point (SP)	Import from CCB	Service Point (SP)	Import from MDM	Node	Import from ODM
		Device	Import from MDM	Asset	Import from ODM
		Device Configuration	Import from MDM		
		Measuring Component	Import from MDM		
		Install Event	Import from MDM	Asset Node	Import from ODM

Customer Care & Billing and Meter Data Management to C2MO

Customer Care and Billing Data		**Meter Data Management Data**		**Operational Device Management Data**	
Person	Import from CCB	Contact	Import from MDM		
Account	Import from CCB				
Service Agreement (SA)	Import from CCB	Usage Subscription (US)	Import from MDM		
Premise	Import from CCB				
Service Point (SP)	Import from CCB	Service Point (SP)	Import from MDM	Node	Synchronize from MDM
		Device	Import from MDM	Asset	Upload from MDM
		Device Configuration	Import from MDM		
		Measuring Component	Import from MDM		
		Install Event	Import from MDM	Asset Node	Synchronize from MDM

Meter Data Management to MDM/ODM

Meter Data Management Data		**Operational Device Management Data**	
Contact	Import from MDM		
Usage Subscription (US)	Import from MDM		
Service Point (SP)	Import from MDM	Node	Synchronize from MDM
Device	Import from MDM	Asset	Upload from Legacy MDM
Device Configuration	Import from MDM		
Measuring Component	Import from MDM		
Install Event	Import from MDM	Asset Node	Synchronize from MDM

Chapter 10

Data Conversion and Migration Design

There are several aspects implementation should consider when designing the legacy data extract processes and creating the Input Data Files. The data conversion 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:

Target Object	Table: CI_PER Data file contains records for a single table.	Maintenance Object PERSON: Tables: CI_PER CI_PER_NAME CI_PER_ID ….etc Data file contains records for multiple tables within Maintenance Object. Table name serves as “record type” qualifier.	
Input Data File Layout	1234, IND, Doe,… 5678, IND, Moon,... 9063, BUS, ABC Corp,..	CI_PER CI_PER CI_PER CI_PER _ID CI_PER _ID CI_PER _ID CI_PER _ID CI_PER _NAME CI_PER _NAME	1234, IND, Doe,… 5678, IND, Moon,... 9063, BUS, ABC Corp,.. 1234, SSN,72346781 5678, SSN, 87635241 9063, EIN, 09182835 9063, TID, 82528555 1234, Doe, Mary 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 11

Data Conversion and Migration Processes

This chapter provides specifics regarding the processes used when migrating/converting customer data, meter data, and measurement data. This includes:

- Customer Data Migration
    - Legacy Customer Information System Upgrading to Oracle Utilities Cloud Service
    - Customer Care and Billing Upgrading to Oracle Utilities Cloud Service
    - Integrated Customer Care & Billing and Meter Data Management Upgrading to Oracle Utilities Cloud Service
    - Integrated Customer Care & Billing and Meter Data Management Upgrading to Oracle Utilities Cloud Service
    - Integrated Customer Care & Billing, Meter Data Management and Operational Device Management Upgrading to Oracle Utilities Cloud Service
    - Meter Data Management Install Event Migration
- Meter Data Migration
    - Legacy Meter Data Management Upgrading to Oracle Utilities Meter Solution Cloud Service
    - Meter Data Management Upgrading to Oracle Utilities Meter Solution Cloud Service
    - Integrated Meter Data Management and Operational Device Management Upgrading to Oracle Utilities Meter Solution Cloud Service
    - Meter Conversion
    - Binding Meter Data Management Device IDs and Operational Device Management Asset IDs
    - Asset Conversion
- Measurement Data Migration
    - Required Configuration for Measurement Upload

Customer Data Migration

This section outlines the steps for migrating customer data to an Oracle Utilities Customer Cloud Service implementation that includes Customer Care and Billing, Meter Data Management, and Operational Device Management functionality.

Legacy Customer Information System Upgrading to Oracle Utilities Cloud Service

In this scenario a new customer using a legacy CIS application is upgrading to Oracle Utilities Customer Cloud Service. This outline assumes that the legacy data has been loaded into staging tables.

Step	Description	Remarks
1	Convert legacy person, account, premise, service point and service agreement to CC&B person, account, premise, service point and service agreement.	
2	Convert legacy meter, meter configuration, register to MDM device, device configuration and measuring component.	See Meter Conversion.
3	Synchronize CC&B person, service point and service agreement to MDM contact, service point and usage subscription.	
4	Copy production MDM SP to staging MDM SP	See Meter Data Management Install Event Migration.
5	Build the staging MDM SP key mapping table.	
6	Convert legacy install event to MDM install event.	
7	Convert legacy meter to ODM asset.	The following steps only apply if ODM functionality is included.
8	Bind the MDM device to ODM asset.	See Binding Meter Data Management Device IDs and Operational Device Management Asset IDs.
9	Bind the ODM asset to MDM device.	See Binding Meter Data Management Device IDs and Operational Device Management Asset IDs.
10	Synchronize MDM service point to ODM service point.	
11	Synchronize MDM install event to ODM asset disposition	

Customer Care and Billing Upgrading to Oracle Utilities Cloud Service

In this scenario an existing Customer Care and Billing customer is upgrading to Oracle Utilities Customer Cloud Service. This outline assumes that the CC&B meter data has been loaded into the MDM and ODM meter and asset staging tables.

Step	Description	Remarks
1	Export CC&B database to single instance database.	
2	Convert CC&B meter, meter configuration and register to MDM device, device configuration and measuring component.	See Meter Conversion.
3	Synchronize CC&B person, service point and service agreement to MDM contact, service point and usage subscription.	
4	Build the staging CC&B SP key mapping table.	See Meter Data Management Install Event Migration.
5	Copy production MDM SP to staging MDM SP.	
6	Build the staging MDM SP key mapping table.	
7	Convert CC&B install events to MDM install events.	
8	Convert CC&B meters to ODM assets.	The following steps only apply if ODM functionality is included.
9	Bind the MDM devices to ODM assets.	See Binding Meter Data Management Device IDs and Operational Device Management Asset IDs.
10	Bind the ODM assets to MDM devices.	See Binding Meter Data Management Device IDs and Operational Device Management Asset IDs.
11	Synchronize MDM service points to ODM service points.	
12	Synchronize MDM install events to ODM asset dispositions.	

Integrated Customer Care & Billing and Meter Data Management Upgrading to Oracle Utilities Cloud Service

In this scenario an existing customer using integrated CC&B and MDM applications is upgrading to Oracle Utilities Customer Cloud Service. This outline assumes that the MDM meter data has been loaded into the ODM staging tables.

Step	Description	Remarks
1	Export CC&B database to database.	
2	Export MDM database to database.	
3	Convert MDM meters to ODM assets.	The following steps only apply if ODM functionality is included.
4	Synchronize MDM service points to ODM service points.	
5	Synchronize MDM install events to ODM asset dispositions.	
6	Create the MDM device identifiers from ODM asset identifiers.	

Integrated Customer Care & Billing and Meter Data Management Upgrading to Oracle Utilities Cloud Service

In this scenario an existing customer using integrated CC&B and MDM applications is upgrading to Oracle Utilities Customer Cloud Service. This outline assumes that the MDM meter data has been loaded into the ODM staging tables.

Step	Description	Remarks
1	Export CC&B database to database.	
2	Export MDM database to database.	
3	Convert MDM meters to ODM assets.	The following steps only apply if ODM functionality is included.
4	Synchronize MDM service points to ODM service points.	
5	Synchronize MDM install events to ODM asset dispositions.	
6	Create the MDM device identifiers from ODM asset identifiers.	

Integrated Customer Care & Billing, Meter Data Management and Operational Device Management Upgrading to Oracle Utilities Cloud Service

In this scenario an existing customer using integrated CC&B, MDM and ODM applications is upgrading to Oracle Utilities Customer Cloud Service.

Step	Description	Remarks
1	Export CC&B database to database.	
2	Export MDM database to database.	

Step	Description	Remarks
3	Export ODM database to database.	

Meter Data Management Install Event Migration

This section describes the issue when converting Meter Data Management install events into Oracle Utilities Customer Cloud Service using master data synchronization and other batch processes.

Install Event Conversion from Legacy CIS application

When migrating legacy data from a CIS application to Oracle Utilities Customer Cloud Service implementation that includes Customer Care and Billing, Meter Data Management, and Operational Device Management functionality, the approach is to stage the customer related legacy data to corresponding CC&B staging tables.

After customer-related data such as persons, service points, and service agreements are successfully converted in the CC&B production tables, the initial load batch jobs are submitted to synchronize this customer-related data to MDM as contacts, service points, and usage subscriptions.

Similarly, after meter-related data such as devices, device configurations, measuring components, and install events are loaded to corresponding MDM staging tables, conversion batch jobs should be submitted.

The conversion process works well for devices, device configurations, and measuring components but not for install events.

This is due to the fact that MDM service point production table is populated through a synchronization process which results in an empty staging MDM SP table (D1_SP) and MDM SP legacy and production key mapping table (DK_SP), which are used in the foreign key resolution when inserting records to production MDM SP table.

To fix this issue, the staging MDM SP table (D1_SP) and MDM SP legacy and production key mapping table (DK_SP) must be populated with data from production MDM SP and SP identifier tables.

The following batch jobs are required to be run to populate these tables before running the MDM install event conversion process.

Batch Control	**Description**	**Remarks**
X1CPSTSP	Copy To Staging D1_SP	Populates the staging MDM SP be copying the data from production SP with the legacy SP ID as the primary key
X1PODSPK	Copy To Staging DK_SP	Populates the MDM SP legacy and production key mapping table with the old key as legacy SP ID and with the new key as production SP ID

Install Event Conversion from Customer Care and Billing

When migrating legacy data from CC&B to Oracle Utilities Customer Cloud Service implementation that includes Customer Care and Billing, Meter Data Management, and Operational Device Management functionality, the approach is to re-use the existing customer-related data by porting these to the database using export/import or data pump database functions.

Customer-related data such as persons, service points, and service agreements are synchronized to MDM contacts, service points, and usage subscriptions through initial load batch jobs.

CC&B meter-related data such as meters, meter configurations, registers, and install events are loaded to corresponding MDM staging tables (devices, device configurations, measuring components, and install events) and conversion batch jobs should be submitted.

The same issue will be encountered regarding the install events conversion due to the empty staging MDM SP table (D1_ SP) and MDM SP legacy and production key mapping table (DK_SP), used in the foreign key resolution when inserting records to production MDM SP table. In addition, the job that populates the MDM SP legacy and production key mapping table (DK_SP) will not work because it is dependent on the CC&B SP legacy and production key mapping table (CK_SP).

To fix this issue, the CC&B SP legacy and production key mapping table (CK_SP) will have to be populated with data from production CC&B SP.

The following batch jobs are required to be run to populate these tables before running the MDM install event conversion process.

Batch Control	Description	Remarks
X1POCSPK	Copy To Staging CK_SP	Populates the CC&B legacy and production key mapping table old and new keys with the production CC&B SP ID
X1CPSTSP	Copy To Staging D1_SP	Populates the staging MDM SP be copying the data from production SP with the legacy SP ID as the primary key
X1PODSPK	Copy To Staging DK_SP	Populates the MDM SP legacy and production key mapping table with the old key as legacy SP ID and with the new key as production SP ID

Meter Data Migration

This section outlines the steps for migrating customer data to an Oracle Utilities Meter Solution Cloud Service implementation that includes Meter Data Management, and Operational Device Management functionality.

Legacy Meter Data Management Upgrading to Oracle Utilities Meter Solution Cloud Service

In this scenario a new customer using a legacy MDM application is upgrading to Oracle Utilities Meter Solution Cloud Service. This outline assumes that the legacy data has been loaded into the staging tables.

Step	Description	Remarks
1	Convert legacy contacts, service points, and usage subscriptions to MDM contacts, service points, and usage subscriptions.	See Meter Conversion.
2	Convert legacy devices, device configurations, measuring components to MDM devices, device configurations, and measuring components.	See Meter Conversion.
3	Convert legacy install events to MDM install events.	See Meter Conversion.
4	Convert legacy meters to ODM assets.	The following steps only apply if ODM functionality is included.
5	Bind the MDM devices to ODM assets.	See Binding Meter Data Management Device IDs and Operational Device Management Asset IDs.
6	Bind the ODM assets to MDM devices.	See Binding Meter Data Management Device IDs and Operational Device Management Asset IDs.
7	Synchronize MDM service points to ODM service points.	
8	Synchronize MDM install events to ODM asset dispositions.	

Meter Data Management Upgrading to Oracle Utilities Meter Solution Cloud Service

In this scenario an existing customer using the MDM application is upgrading to Oracle Utilities Meter Solution Cloud Service. This outline assumes that the MDM meter has been loaded into the ODM asset staging tables.

Step	Description	Remarks
1	Export MDM database to the database.	
2	Convert MDM meters to ODM assets.	The following steps only apply if ODM functionality is included.
3	Create MDM device identifiers from ODM asset identifiers.	
4	Synchronize MDM service point to ODM service point.	
5	Synchronize MDM install event to ODM asset disposition.	

Integrated Meter Data Management and Operational Device Management Upgrading to Oracle Utilities Meter Solution Cloud Service

In this scenario an existing customer using integrated MDM and ODM applications is upgrading to Oracle Utilities Meter Solution Cloud Service.

Step	Description	Remarks
1	Export MDM database to database.	
2	Export ODM database to database.	

Meter Conversion

The table below shows the logical sequence in running the meter conversion jobs implementing the framework conversion tools. This outline assumes that the legacy data has been loaded into the staging tables

	Contact	Device	Device Configuration	Measuring Component	Service Point	Usage Subscription	Install Event
Insertion	Insert D1_ CONTACT (D1CNT00I)	Insert D1_ DVC (D1DVC00I)	Insert D1_ DVC_CFG (D1DC000I)	Insert D1_ MEASR_ COMP (D1MC000I)	Insert D1_SP (D1SP000I)	Insert D1_US (D1US000I)	Insert D1_ INSTALL_EV T (D1IE000I)
Key Generation	Generate Keys for D1_ CONTACT (D1CNT00K)	Generate Keys for D1_DVC (D1DVC00K)	Generate Keys for D1_DVC_ CFG (D1DC000K)	Generate Keys for D1_ MEASR_ COMP (D1MC000K)	Generate Keys for D1_SP (D1SP000K)	Generate Keys for D1_US (D1US000K)	Generate Keys for D1_ INSTALL_EV T (D1IE000K)

	Contact	Device	Device Configuration	Measuring Component	Service Point	Usage Subscription	Install Event
Legacy Data Validation	Validate D1-CONTACT (D1CNT00V)	Validate D1-DEVICE (D1DVC00V)	Validate D1-DVCCONFIG (D1DC000V)	Validate D1-MEASRCOMP (D1MC000V)	Validate D1-SP (D1SP000V)	Validate D1-US (D1US000V)	Validate D1-INSTLEVT (D1IE000V)
Production Data Validation	Validate D1-CONTACT (D1CNT00V)	Validate D1-DEVICE (D1DVC00V)	Validate D1-DVCCONFIG (D1DC000V)	Validate D1-MEASRCOMP (D1MC000V)	Validate D1-SP (D1SP000V)	Validate D1-US (D1US000V)	Validate D1-INSTLEVT (D1IE000V)
XML Resolution	Resolve XML for D1-CONTACT (D1CNT00R)	Resolve XML for D1-DEVICE (D1DVC00R)	Resolve XML for D1-DVCCONFIG (D1DC000R)	Resolve XML for D1-MEASRCOMP (D1MC000R)	Resolve XML for D1-SP (D1SP000R)	Resolve XML for D1-US (D1US000R)	Resolve XML for D1-INSTLEVT (D1IE000R)
	Insert D1_CONTACT_IDENTIFIER (D1CNTIDI)	Insert D1_DVC_IDENTIFIER (D1DVCIDI)	Insert D1_DVC_CFG_CHAR (D1DCCHRI)	Insert D1_MEASR_COMP_IDENTIFIER (D1MCIDTI)	Insert D1_SP_IDENTIFIER (D1SPIDTI)	Insert D1_US_IDENTIFIER (D1USIDTI)	Insert D1_INSTALL_EVT_CHAR (D1IECHRI)
	Insert D1_CONTACT_CHAR (D1CNTCHI)	Insert D1_DVC_CHAR (D1DVCCHI)		Insert D1_MEASR_COMP_CHAR (D1MCCHRI)	Insert D1_SP_CHAR (D1SPCHRI)	Insert D1_US_CHAR (D1USCHRI)	
	Insert D1_CONTACT_NAME (D1CNTNMI)			Insert D1_MEASR_COMP_REL (D1MCRELI)	Insert D1_SP_CONTACT (D1SPCNTI)	Insert D1_US_CONTACT (D1USCNTI)	
	Insert D1_CONTACT_PHONE (D1CNTPHI)				Insert D1_SP_EQPMNT (D1SPEQPI)	Insert D1_US_SP (D1USSP0I)	
						Insert D1_US_SP_CHAR (D1USSPCI)	

Binding Meter Data Management Device IDs and Operational Device Management Asset IDs

This section describes the batch jobs used to populate the device identifier and asset identifier tables to bind device and asset IDs.

Device and Asset are Loaded with Converted Data

This section applies to data migration scenarios where both device and asset tables are loaded with legacy data from either a CIS application or existing CC&B application upgrading to Oracle Utilities Customer To Meter.

The following batch jobs must be run to populate the both Device Identifier and Asset Identifier tables to bind both converted device and asset IDs.

Batch Control	Description	Remarks
X1-LIDAD	Populate Asset Identifier with Device from Legacy Device	
X1-LIDDA	Populate Device Identifier with Asset from Legacy Device	

Asset Loaded with Converted Data

This section applies to the following data migration scenario:

- Existing separate instances of CC&B and MDM applications upgrading to Oracle Utilities Customer Cloud Service
- Existing MDM application upgrading to Oracle Utilities Meter Solution Cloud Service

The following batch job must be run to populate the Device Identifier table with an entry that binds the device ID with the converted asset ID.

Batch Control	Description	Remarks
X1-IDATD	Populate Device Identifier from Asset	

Asset Conversion

Conversion Sequence

	Node	Asset
Legacy Data Validation	Validate W1-NODE (W1NOD00V)	Entity Validation for W1-ASSET (W1AST00V)
Key Generation	Generate Keys for W1_NODE (W1NOD00K)	Generate Keys for W1_ASSET (W1AST00K)
XML Resolution	Resolve XML for W1-NODE (W1NOD00R)	Resolve XML for W1_ASSET (W1AST00R)
Insertion	Insert W1_NODE (W1NOD00I)	Insert W1_ASSET (W1AST00I)
	Insert W1_NODE_CHAR (W1NODCHI)	Insert W1_ASSET_IDENTIFIER (W1ASTIDI)
	Insert W1_NODE_IDENTIFIER (W1NODIDI)	Insert W1_ASSET_NODE (W1ASTNDI)
		Insert W1_ASSET_CHAR (W1ASTCHI)
Production Data Validation	Validate W1-NODE (W1NOD00V)	Entity Validation for W1-ASSET (W1AST00V)

Measurement Data Migration

Migrating data from the Measurement table (D1_MSRMT) requires a special treatment because of the extremely large volume of data being migrated.

For cloud service implementations, the recommendation is to directly upload the legacy measurements into the production Measurement table using the Cloud Service Foundation tool. This is because the data conversion process will be much faster compares to staging the legacy data and running conversion jobs.

Implementations may opt to use the Oracle Utilities Application Framework conversion tool to stage measurement and then mass insert the staged data into production measurement table.

Caveats with this approach:

- It will be a slower process compared to directly loading the data.
- Insertion batch controls are not provided with MDM, however, implementation should be able to create custom batch controls based on templates provided with the OUAF.

On-premises implementations can also directly upload the legacy measurements to the production measurement table. The only difference is that the Cloud Service Foundation tool is not available, so the implementation will have to use SQL loader. Alternatively, they may opt for measurement staging approach using the OUAF conversion tool.

SQL Loader offers various performance improvement measures and supports default field values, date time caching, multiple record types (delimited, fixed length, binary)

The product supports the ability to create 100% custom control file for specific table.

Required Configuration for Measurement Upload

The following sections outline configuration tasks required for upload of measurement data.

Download Control File

The file for legacy measurement contains the legacy register or channel (measuring components) that need to be resolved before mass loading the data into the production database.

This process uses a specific control file to optimize the upload performance. This control file will be responsible for resolving or deriving the production measuring component key for the corresponding legacy key upon insertion to the production table.

This control file is called D1_MSRMT_CTL.ctl, and can be found in the *Data Upload Sample Data Files* zip file available in the **Supporting Cloud Service Guides** section on the Oracle Utilities Customer Cloud Service or Oracle Utilities Meter Solution Cloud Service documentation website.

Create Managed Content for the Control File

1. Select **Admin**, then **System**, then **Managed Content**, then **Add**.
2. Enter a code for the control file in the **Managed Content** field.

3. Select "XML" from the **Managed Content Type** drop-down list.
4. Enter a name for the control file in the **Description** field.
5. Click the **Schema** tab, and paste the D1_MSRMT_CTL.ctl control file text into the **Editor** area.
6. Click **Save**.

Create the Conversion Task Type for the Measurement Table (D1_MSRMT)

1. Select **Admin**, then **Conversion Support**, then **Conversion Task Type**, then **Add**.
2. Select "Conversion Instructions - General" from the **Service Task Type** drop-down list and click OK.
3. Enter a code and **Description** for the conversion task type.
4. Select "Conversion Artifacts - Table" from the **Related Transaction BO** drop-down list.
5. Select the managed content created from the previous step from the **Override Control File** drop-down list in the **Conversion Artifacts Instructions** section.
6. Click **Save**.

Refer to the *Oracle Utilities Cloud Service Foundation Administrative User Guide* for more information about Conversion Task Types.

Setup the Conversion Data Upload Master Configuration

1. Select **Admin**, then **General**, then **Master Configuration**.
2. Select and broadcast 'Conversion Data Upload Configuration' from the list
3. Click **Edit**.
4. Add an entry in the **Override Instructions - Table** section for the Measurement table (D1_MSRMT) and specify the Conversion Task Type created from the previous step.
5. Click **Save**.

Refer to the *Oracle Utilities Cloud Service Foundation Administrative User Guide* for more information about Conversion Data Upload Configuration master configuration.

Generate Conversion Artifacts for the Measurement Table (D1_MSRMT)

1. Select **Admin**, then **Conversion Support**, then **Generate Conversion Artifacts**.
2. Enter or search for D1_MSRMT in the **Table** field.
3. Click **Generate**.
4. A warning appears, click **OK** and then click **Continue** when prompted.

Refer to the *Oracle Utilities Cloud Service Foundation Administrative User Guide* for more information about using the Conversion Artifact Generator.

Run Conversion Batch Processing for the Measurement Table (D1_MSRMT)

1. Upload the file containing legacy data to an Object Storage location. This location should be defined as a value for the File Storage Configuration (F1-FileStorage) extendable lookup.
2. Run the Conversion - Load Data using SQL Loader batch process (K1-CNVLD) using the following parameters:
    - **Input File Storage**: The Object Storage location (defined as a value for the File Storage Configuration (F1- FileStorage) extendable lookup) that contains the file to be uploaded.
    - **Table**: D1_MSRMT
    - Other parameters as appropriate.

Refer to the *Oracle Utilities Cloud Service Foundation Administrative User Guide* for more information about running conversion batch controls with files in Object Storage locations.

Chapter 12

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 the ILM_ARCH_SW field as follows:
    - Set the field with a value of "Y" for high-volume tables. In specific, the ILM_ARCH_SW field MUST be set to "Y" for the following tables used with Oracle Utilities Customer Cloud Service and Oracle Utilities Meter Solution Cloud Service:
        - D1_DVC_EVT (Device Event)
        - D1_INIT_MSRMT_DATA (Initial Measurement Data)
        - D1_USAGE (Usage Transaction)
    - Set the field with a value of "N" for all other tables
- For the ILM_DT field, the ILM Date Fields 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_SNPSHT	F1_STATS_SNPSHT.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

Table Name	ILM DT Initial Load
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 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 13

Data Conversion and Migration Steps

This chapter describes the steps involved in data conversion and migration, 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)

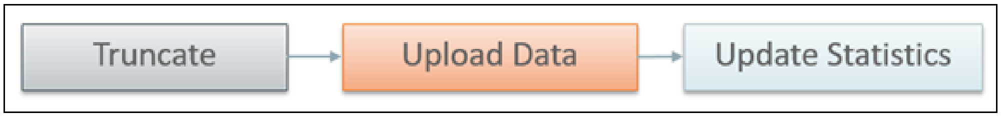
According to OUAF DB design standards, a corresponding Key Table exists for each table with system-generated or sequential primary key. Under normal circumstances, the key tables are populated when an application creates a "main" record. In a conversion situation, where the data is inserted directly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

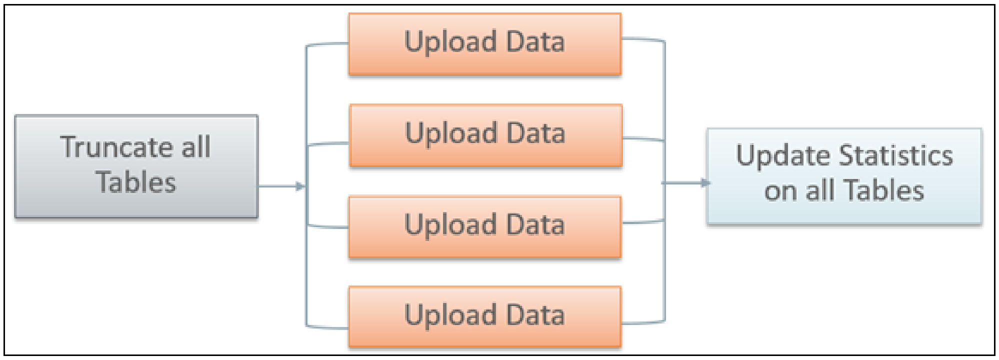
Data Upload Orchestration

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



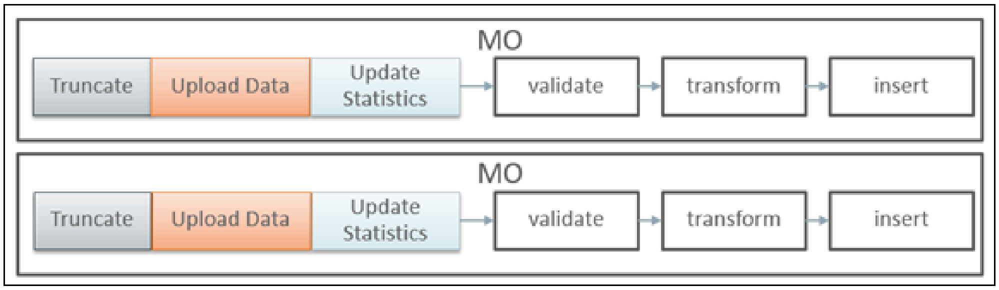
Multiple Tables or MOs Upload



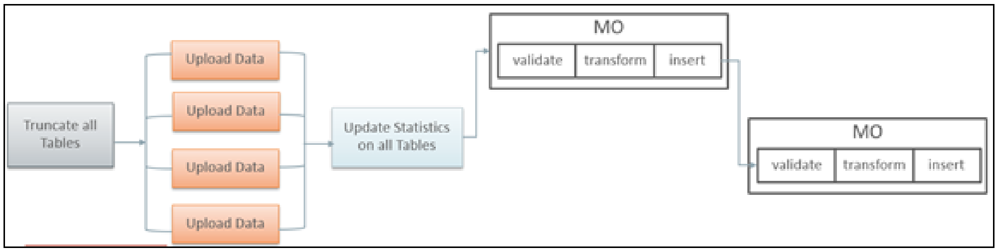
There are multiple strategies to orchestrate the entire conversion run and to build the optimal sequence of the conversion processes. Below are some of the many possibilities:

- 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 14

Customizing Data Conversion and Migration

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 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 maintnance 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.	The data upload only runs if the environment is pointing to the STAGING schema. Navigate to Conversion Support ' Switch Schema. On the popup screen the current schema is displayed. Make sure the current schema is **Staging**.
Provide data files according to the specifications Regenerate the artifacts after modifying the data upload configurations	SQL Loader loads the data according to the Control File. Input Data File Specifications describe what is expected from the input data file: • Names of the data files • Data format for all fields • Data delimiters to be used in the input data file Every time the configuration has changed the artifacts must be regenerated in order to keep the configurations and the input data specifications in sync.
Provide input data files with CLOB data IF NECCESSARY	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. 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. If you wish to include CLOB data in the main file, verify that the Conversion Instruction is set correctly. **If the configuration was modified you must regenerate the artifacts.**

Issue	Details
Avoid creating unnecessary data files for CLOB columns	By default the system expects the data to be provided for all target table columns. 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. To exclude unnecessary CLOB columns for a table or maintenance object, configure Conversion Instructions using the *K1-ConvArtMultClobMOTaskType* or *K1-ConvArtMultClobTblTaskType* business object and specify the Override Conversion Instruction on the Master Configuration Regenerate conversion artifacts and examine the input data specifications after changing the configuration
Avoid truncating the entire staging data unintentionally	The K1-SCLTB batch process allows you to truncate a specific table or maintenance object in the STAGING schema. The K1-CLNTB batch process allows you to truncate a specific table or maintenance object in the PRODUCTION schema. 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.
Clean up duplicate PK values before the data upload	Indexes and constraints are disabled during data upload in order to boost performance. De-duplication during the data upload is not supported out-of-the-box • SQL Loader direct path upload doesn't perform duplicate check • No direct database access means no possibility to modify data via direct SQL after the upload Keep track of the legacy data that has been already uploaded If you re-upload the same data again, always clean up the target table(s).

Issue	Details
The business configurations and admin data has to be finalized and populated in Production prior for legacy data upload Populate the legacy data extract with valid FK references to the admin/control data	**Once uploaded, the staging data cannot be "massaged"/modified thru direct SQL** (that because no database access is possible on cloud) Hence the overall conversion project steps are: • Design, test and complete business configurations. During this stage, multiple trial data uploads with dummy data could be performed • Populate admin data in Production • Create legacy data extract with valid admin data FK References • Upload data into staging tables
Key Tables are not populated implicitly	The Key Tables in the staging schema tables are not populated automatically when the legacy data is uploaded into "main" tables. Upload the data into Key Tables separately or use the batch program provided by Cloud Service Foundation.
Override Conversion Eligibility is supported on Table level only	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. **Note**: 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 threat this table as a valid target table
Loading Data Directly into the Production (CISADM) Schema	The following configuration steps are required to load data directly into the Production (CISADM) schema tables: • Create a custom Control File for the target table. • Generate the control file with default conversion instructions and copy the contents. • Modify the INTO... clause to add 'CISADM.' in front of a target table name • Create a new Managed Content entry. Copy the entire control file text and save. • Create a new Conversion Task Type for the target Table • Specify the new Managed Content as an Override Control File. • In the Data Upload Support Master Configuration, create an Override Table Instruction entry. Specify the target table and the new Conversion Task Type. • Generate Conversion Artifacts for the table.

Issue	Details
Loading Very Large Data Volumes	Avoid SQL-based conditions in the control file when loading very large data volumes. The default values and SQL-based conditions will cause SQL Loader to switch to the conventional path load which performs row-by-row inserts. The best results are achieved with a direct path load. More threads doesn't necessarily mean better performance. The optimal overall data load performance is achieved when the threads (and their corresponding SQL Loader processes) are targeting different partitions. Additional guidelines: • Partitioning by month is required for best performance • Load multiple months in parallel for best performance & scalability • Start with ONE thread per month • Increment number of threads per month. If performance does not increase, try a smaller increment or stay with your last best. For example: loaded 12 months data with 48 threads, 4 threads/month • Large data files are preferable • Many small files have the overhead of spinning up new SQL*Loader process for each file • Set longer SQL timeout on the data upload batch process • Disable indexes before loading • Rebuild indexes after direct path load • Reduce or stop the activities in the environment when performing the massive data upload

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 (INT_DATA_SET)	Regular maintenance object, CLOB field as a secondary file. Configuration: Conversion Task Type K1-CNV-MO Multiple data files (3)
MO Customer Contact (CUST_CONTACT)	Regular maintenance object, CLOB fields in the main file. Configuration: same as Conversion Task Type K1-CNV-TABLE, but the *CLOB as Secondary File* indicator set to false.
Table Meter Read (CI_MR)	Regular table, CLOB field as a secondary file. Configuration: Conversion Task Type K1-CNV-TABLE.
Table Adjustment (CI_ADJ)	Regular table, CLOB field in the main file. Configuration: same as Conversion Task Type K1-CNV-TABLE, but the *CLOB as Secondary File* indicator set to false.
Table Initial Sync Request (F1_SYNC_REQ_IN)	Table with Multiple CLOBs as secondary files. **Configuration**: For table with multiple CLOBs, the special Conversion Task Type was created based on the K1-ConvArtMultClobTblTaskType business object. Override Control File (Managed Content) was created and used as a custom Control File. 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. **Input Data File Specification**: 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.

Part Three

File-Based Integration

The next three chapters describe how implementations can integrate and exchange information from Oracle Utilities Cloud Services to other applications and vice versa through file based integration. File upload and download processes are used by some implementation for data connect, payment upload, letters extract, financial extracts, other business processes.

Oracle Utilities Cloud Services can exchange data files from one application to another by:

- Directly accessing Oracle Cloud Object Storage
- Integrating with Oracle Integration Cloud.

  For more information about this approach, refer to the Oracle Integration Cloud documentation at https://docs.oracle.com/en/cloud/paas/integration-cloud/

These chapters provides information on how Oracle Utilities Cloud Services, specifically Oracle Utilities Customer Cloud Service (CCS), access data files from Oracle Cloud Object Storage, including:

- Chapter 15: Object Storage Connection Management
- Chapter 16: File Export Sample Implementation
- Chapter 17: File Import Sample Implementation

Chapter 15

Object Storage Connection Management

This chapter outlines how to manage connections between Oracle Utilities cloud services and Oracle Object Storage, including:

- Oracle Object Storage Setup
- Oracle Utilities Cloud Service Configuration for Object Storage Connection
- Register API Key to Oracle Cloud Object Storage

Oracle Object Storage Setup

Before initiating a file transfer from an Oracle Utilities cloud service to Oracle Cloud Object storage or vice versa, you must first make sure that the basic administration tasks in Oracle Cloud Infrastructure related to Object Storage have been completed properly, and that the compartments and buckets where the import and export files are stored have been setup.

For more information on Oracle Cloud Object Storage setup for Oracle Utilities Cloud Services, including Oracle Utilities Customer Cloud Service (CCS), please see the *Oracle Utilities Cloud Services Object Storage Setup Guide*.

Oracle Utilities Cloud Service Configuration for Object Storage Connection

Authentication and connection between the Oracle Utilities cloud service and Object Storage enables batch processes to import and export files from and to Object Storage locations. Setting up this authentication requires the following in your Oracle Utilities cloud service:

- Creating API Keys
- Creating An Object Storage Connection

Refer to the *Oracle Utilities Cloud Services Object Storage Setup Guide* for details concerning setting up Keys and Key Rings, an object storage connection configuration, and registering the API key.

> **Note**: You can use the same API Keys and Object Storage Connection setup for both import and export process.

Creating API Keys

Create a key ring for the cloud service environment. The key ring should be active and should have a set of private/public encryption key pairs. This key ring will be included in the Object Storage Connection Configuration.

Creating Key Rings and Pairs

Authentication between the Oracle Utilities cloud service and Oracle Object Storage requires an API signature key. See **API Key Management** in the *Oracle Utilities Cloud Services Object Storage Setup* Guide for more information.

API key rings and key pairs are maintained in the **Key Ring** portal in the cloud service. This portal contains the following zones:

- **Key Ring**: Displays basic information about the key ring
- **Key Pairs**: Displays a list of key pairs for the current key ring

Key rings are defined by the following:

- **Key Ring**: A unique code for the key ring
- **Key Ring Class**: Signature (default)

- **Status**: The current status of the key ring.

  Note: Key pairs can only be generated for Active key rings. Once a key ring has been deactivated, you can no longer create key pairs for that ring.

- **Description**: A name for the key ring (this will be referenced in the File Location extendable lookup, see below)

Once the key ring is created, you need to generate and activate the key pair. Click **Generate Key** to generate a key pair for the key ring.

Key Pairs are defined by the following:

- **Sequence**: The sequence of the key pair (the order in which the key pair was created)
- **Creation Date/Time**: The date/time when the key pair was created
- **Key Status**: The current status of the key pair. Key pairs are inactive when first created.
- **Public Key**: Click **View** to open a dialog box containing the public key.
- **Action**: Click **Activate** to activate an inactive key pair.

Click **Activate** in the **Actions** column in the **Key Pairs** zone. A dialog box opens displaying the following message: "Warning(s): Activating a key assumes that you have already registered the public key with the appropriate third parties. Press **Cancel** to abort." Click **OK** to activate the key. The **Key Status** column will change to "Active".

Note: Be sure to register the API Key with Object Storage by copying the public key to Oracle Identification and Access Management. To copy the public key, click **View** in the **Actions** column in the **Key Pairs** zone, and select and copy the text in the **View Public Key** dialog box. Refer to Register API Key to Oracle Cloud Object Storage for more information.

Creating An Object Storage Connection

Create an object storage connection via the File Storage Configuration extendable lookup (F1-FileStorage). This defines the Object Storage location where the files will be stored.

Creating File Storage Extendable Lookup Values

Apart from the authentication, the cloud service also needs information about the Object Storage locations to be used. Object Storage locations are defined by values in the File Storage Configuration (F1-FileStorage) extendable lookup. These file storage configurations will be referenced by the batch processes that will import or export records.

Values for the File Storage Configuration extendable lookup are defined by the following:

- **Value**: A unique code for the extendable lookup value. This value will be referenced as a batch control parameter value.
- **Description**: A description of the extendable lookup value
- **Status**: The current status of the value. Select "Active".
- **File Storage Details**: This section defines details for the object storage location, including:

- **File Adapter**: The type of file adapter for the location. Select “Oracle Cloud Object Storage”.
- **User**: The user Oracle Cloud ID (ODIC) for the object storage location
- **Tenancy**: The tenancy Oracle Cloud ID (ODIC) for the object storage location
- **Compartment**: The compartment Oracle Cloud ID (ODIC) for the object storage location
- **Namespace**: The namespace for the object storage location
- **Key Ring**: The Key Ring you created earlier
- **Region**: The region of the object storage tenancy for the connection (Values for this field are defined in the F1_REGION_FLG lookup.

Refer to **External File Storage** in the *Oracle Utilities Application Framework Administrative User Guide* and the *Oracle Utilities Cloud Services Object Storage Setup Guide* for more information.

Register API Key to Oracle Cloud Object Storage

Once the key ring and key pair have been created in the Oracle Utilities cloud service, copy the public key from the key pair and add the public key to the **User API Key** in Oracle Identification and Access Management (IAM). See the **User API Keys** section in the **Security and Access Management** section of the **Managing Object Storage** chapter in the *Oracle Cloud Infrastructure Services* documentation.

Chapter 16

File Export Sample Implementation

This chapter provides an example of implementing a file export process, which includes:

- Creating a File Export Batch Process
- Configuring the Export Process

Creating a File Export Batch Process

Oracle Utilities cloud services provide a way to extract system information into a file using a file-export batch process. This process can be configured to export and store extracted files in an Object storage location.

Oracle Utilities cloud services provide a sample Batch Control (F1-PDBEX) which supports file export functionality and which can be used as 'template' to implement custom file export functionality as needed by specific implementations. Along with this out of the box batch control, cloud services also provide related objects (such as scripts.) that also can be used as templates while creating custom export processes.

Please note that the main data extraction logic lies within the script as described below. Customer can look at and copy the F1-GenProcEx script to implement their own data extraction logic.

This guide will follow a "bottom up" approach regarding the creation and configuration of cloud service data and objects to facilitate the file export process. The first step is to create a data area and script using that data area. Other objects for the file export will be created next.

For this sample implementation, we will export Premise information from Customer Cloud Service to Object Storage. This involves creating two algorithms (one for file export and one for selecting records to export) and a batch control.

To create a File Export Algorithm:

1. Create a data area that defines the schema for holding premise information.
2. Create a Plug-in script with the *Batch Control - Process Record* Algorithm Entity. This script may be based on the F1-GenProcEx script and modified as needed. Make sure to use the data area created above to hold premise information.

   The script will be used by the algorithm that will perform the file export.
3. Create an Algorithm Type similar to F1-GENPROCEX, based on the plug-in script created in the last step.
4. Create an algorithm based on the algorithm type.

To create a record selection algorithm:

1. Duplicate the F1-GENPROCSR algorithm to create a custom algorithm that will be used for selecting records.
2. Update the algorithm’s parameters.
   - Use the **SQL** parameter name to define the new query for retrieving records from the cloud service database.
   - Keep the **Batch Strategy** parameter as *THDS*.
   - Define the key field on the query under the **Key Field** parameter.

To create a batch control:

1. Duplicate the F1-PDBEX batch control to create a new batch control. Navigate to **Algorithms** tab on the batch control and replace the existing algorithms with the file export and record selection algorithms created above.
2. This newly created batch control contains parameters for file storage that must be modified as described in the following section.

Configuring the Export Process

This section describes the setup needed to enable export processing, including establishing communication between the Oracle Utilities cloud service and Object Storage, configuring batch parameters, and testing the export process.

- Setting Up Communication Between Cloud Service and Object Storage
- Configuring File Export Batch Parameters
- Testing the Export Process

Setting Up Communication Between Cloud Service and Object Storage

The export process requires authentication and communication between the Oracle Utilities cloud service and Object Storage. See Chapter 15: Object Storage Connection Management for more information about setting up this communication and authentication.

Configuring File Export Batch Parameters

The next step is to configure the following parameters of the file export batch control created earlier:

- **File Name**: the name of the exported file
- **File Path**: the path to file location in Object Storage. The format for the path is as follows:

  file-storage://<File Location>/<Bucket>

  where:

  - **<File-Location>**: The File Storage Configuration extendable lookup value defined for that file. This will include the compartment identification.
  - **<Bucket>**: The object storage bucket in the compartment that is defined as part of the File Storage Configuration extendable lookup value.

  For example, the "File-Export" bucket in a compartment that is referenced in the "AB-Export" File Storage Configuration extendable lookup value can be referenced as:

  "file-storage://AB-Export/File-Export"

- **Other Parameters**: The batch control supports other optional parameters, including:

  - **XML Root Name**: used to declare the name of root-node in generated xml (when exporting files in xml format)
  - **File Delimiter**: used to define the delimiter used in delimited files

  Refer to the batch control for information about other parameters.

Testing the Export Process

The last step is to test the export process.

To test the export process:

1. Run the file export batch process.
2. Navigate to the **Batch Run Tree** portal to verify the job has run successfully.
3. Navigate to targeted bucket inside Object Storage and verify the exported file.

Chapter 17

File Import Sample Implementation

Oracle Utilities cloud services include a batch control that can be used as a template for creating batch controls to import data from a file to the application. This template batch control is called Plug-in Driven File Upload Template (F1-PDUPL).

To use this template, an implementation can duplicate the F1-PDUPL batch control and provide the required algorithm for the "File Upload" system event. The algorithm associated with the batch control is responsible for using provided APIs to read the content of the file and store the data in appropriate table(s) such as a staging table or the FACT table. (we will use the FACT table in this sample).

The plug-in scripts written to implement this type of algorithm must use the Groovy script engine version as the APIs are not accessible using the XPath scripting language. The sample plug-in scripts provided illustrate using the various available APIs to upload a flat file, xml file or a delimited file. Implementation can write their own plug-in scripts to handle their specific file upload needs.

The following steps summarize how to implement a new file import background process:

- Identifying Upload File Content Data
- Uploading File to Oracle Cloud Object Storage
- Creating a File Import Batch Process
- Configuring the Import Process

For more information on how to use Plug-in Driven background processing for import and upload, refer to the following section in the *Oracle Utilities Application Framework Administrative User Guide*:

Background Processes

Understanding Background Processes

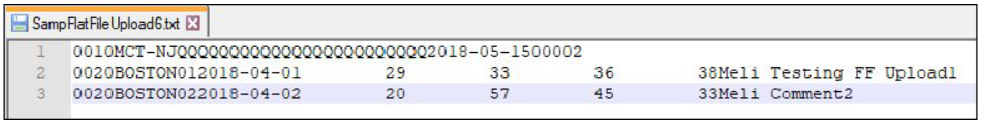
Plug-in Driven Background Processes

Uploading Records

Identifying Upload File Content Data

An important step in creating an import process is to define the format of the data files you plan to import, identify the different values in the file and map the data to fields in one or more appropriate tables in the application.

For example, the SampFlatFileUpload6.txt file has the following content:



```
0010MCT-NJQQQQQQQQQQQQQQQQQQQQQQQQ2018-05-1500002
0020BOSTON012018-04-01        29        33        36        38Meli Testing FF Upload1
0020BOSTON022018-04-02        20        57        45        33Meli Comment2
```

This sample data contains 'degree day' data, including a header record and two data records.

The header record contains the following components (the length of the field is shown in parenthesis):

- Record Type (4) (Value: 0010)
- Source (30) (Value: MCT-NJ)
- Date Transmitted (10) (Value: 2018-05-15)
- Number of Records (5) (Value 00002)

Individual data records have the following components:

- Record Type (4) (Value: 0020)
- Area (8) (Value: BOSTON01, BOSTON02)
- Degree Date (10) (Value: 2018-04-01, 2018-04-02)
- Degree Day (10) (Value: 29, 20)
- Minimum Temperature (10) (Value: 33, 57)
- Average Temperature (10) (Value: 36, 45)
- Maximum Temperature (10) (Value: 38, 33)
- Comments (254) (Value: Meli Testing FF Upload..., Meli Comment2)

Uploading File to Oracle Cloud Object Storage

Once you have a sample file in the correct format, you need to upload the file to the location in Object Storage from where you plan to upload and import your data.

Creating a File Import Batch Process

Creating a file import process involves creating the following components in the Oracle Utilities cloud service:

- Plug-In Script
- Algorithm Type and Algorithm
- File Upload Batch Control

Plug-In Script

The first step is to create a plug-in script that will process the data in the upload file. This plug-in script should be created for the "Batch Control - File Upload" **Algorithm Entity**.

You can use sample plug-in scripts provided or create a new plug-in script with logic required for reading the record and identifying each record detail to properly create the insert statements for storing the data in the appropriate application tables.

The sample plug-in scripts provided illustrate how to call the supplied APIs for processing different types of source data including fixed position, comma delimited, and XML formats.

Sample Plug-in Scripts for Algorithm Entity: Batch Control - File Upload

Plug-In Script	Description
F1UplSmplFlt	Sample Flat File Upload Script
F1UplSmplDlm	Sample Delimited File Upload
F1UplSmplXML	Sample XML File Upload

Note: The F1UplSmplFlt sample script is designed to work with the sample data above.

Algorithm Type and Algorithm

The next step is to create an Algorithm Type and Algorithm that use the plug-in script you created above. In this sample file import implementation, the source data uses the fixed position format, so the Algorithm Type and Algorithm should use the F1UplSmplFlt script.

To create a new algorithm:

- Create an Algorithm Type using the F1UplSmplFlt plug-in script.
- Create a corresponding Algorithm for the Algorithm Type created above.

File Upload Batch Control

The last part of the cloud service configuration is to create a batch control that will use the new algorithm to import the data. You can create a new batch control by duplicating a template batch control and reference the new algorithm. The base product includes the Plug-in Driven File Upload (F1-PDUPL) batch control which can be used as a template.

To create a new batch control:

1. Duplicate the F1-PDUPL sample batch control to create your own batch control.
2. In the **Algorithms** tab, define the algorithm created above for the "File Upload" **System Event**.
3. Define default parameters for the batch control, if required.

Configuring the Import Process

This section describes the setup needed to enable file import processing, including establishing communication between the Oracle Utilities cloud service and Object Storage, configuring batch parameters, and testing the process.

The following steps will enable file import batch processing to run and import the data from the import file to the appropriate application tables in the Oracle Utilities cloud service:

- Setting Up Communication Between Cloud Service and Object Storage
- Configuring File Import Batch Controls
- Testing the Import Process

Setting Up Communication Between Cloud Service and Object Storage

The export process requires authentication and communication between the Oracle Utilities cloud service and Object Storage. See Chapter 15: Object Storage Connection Management for more information about setting up this communication and authentication.

Configuring File Import Batch Controls

The next step is to configure the following parameters of the file import batch control created earlier:

- **File Name**: The name of the file to import.
- **File Path**: the path to the file location in Object Storage where import files will be located. The format for the path is as follows:

  file-storage://<File Location>/<Bucket>

  where:

  - **<File-Location>**: The File Storage Configuration extendable lookup value defined for that file location. This will include the compartment identification.
  - **<Bucket>**: The object storage bucket in the compartment that is defined as part of the File Storage Configuration extendable lookup value.

  For example, the "File-Import" bucket in a compartment that is referenced in the "INT-UPLOAD" File Storage Configuration extendable lookup value can be referenced as:

  file-storage://INT-UPLOAD/File-Import

- Refer to the batch control for information about other parameters.

Testing the Import Process

The last step is to test the import process using the sample data.

To test the import process:

1. Run the file import batch process.
2. Navigate to the **Batch Run Tree** portal and make sure the batch process ran successfully.
3. Check that the records have been added to the FACT table.

Part Four

Oracle REST Data Services

This section describes how to use Oracle REST Data Services with Oracle Utilities cloud services. This includes

- Chapter 18: SQL Developer Web
- Chapter 19: REST APIs

Refer to the Oracle REST Data Services documentation for more information about Oracle REST Data Services.

Chapter 18

SQL Developer Web

Oracle SQL Developer Web is a part of Oracle REST Data Services (ORDS), and is the web-based version of Oracle SQL Developer that enables you to connect to an Oracle database and execute queries and scripts, create database objects, build data models, and monitor database activity.

Oracle Utilities cloud services use Oracle SQL Developer Web to connect to a cloud service database to execute read-only queries on various database schema objects.

Please refer the Oracle REST Data Services documentation for more information about using Oracle SQL Developer Web.

Users must be assigned to the “SQL Developer Web Online User” application role in order to use SQL Developer Web with Oracle Utilities cloud services. See **Pre-Defined Application Roles** in the *Oracle Utilities Cloud Services Administration Guide* for more information about application roles used with Oracle Utilities cloud services.

Access is provided to both CISREAD and STGADM database schemas to perform select/read-only queries.

- The CISREAD schema can be used to perform select and read-only queries of the production schema.
- The STGADM schema can be used to perform select and read-only queries of the staging schema (used with data migration and conversion).
- Oracle REST Data Services can access the production and staging schemas in both production and non-production environments.

Chapter 19

REST APIs

Oracle REST Data Services also provides REST APIs that can be invoked via cURL to connect to an Oracle database and perform operations.

Users must be assigned to the "REST Enabled SQL" application role in order to use REST APIs with Oracle Utilities cloud services. See **Pre-Defined Application Roles** in the *Oracle Utilities Cloud Services Administration Guide* for more information about application roles used with Oracle Utilities cloud services.

The following is an example syntax for cURL command.

```
curl -i -X POST --user <username>:<password> --data-binary "<SQL
statement>" -H "Content-Type: application/sql" -k <Oracle REST URL>
```

Contact your system administrator for Oracle SQL Developer Web and REST service URLs.

No additional configuration is required to use Oracle SQL Developer Web or REST services.

Part Five

Product-Specific Integrations

This section describes product-specific integrations available for use with Oracle Utilities Cloud Services. This includes:

- Chapter 20: Customer Cloud Service Receipt Printing

Chapter 20

Customer Cloud Service Receipt Printing

This chapter describes how to configure Oracle Utilities Cloud Services to support integration with a Point Of Sale (POS) printer for printing of receipts related to the following payment transactions:

- Payment Event
- Payment Event Quick Add
- Payment Quick Add

Refer to the Oracle Utilities Cloud Services *Business User Guide* for more information about these payment transactions.

Configuration to support this functionality includes:

- Oracle Utilities Cloud Services Configuration
- Printer Installation
- Recommended Printer Preferences

Notes:

- The instructions in this document are based on a specific sample printer, the Epson TM-H6000IV USB POS printer.

Oracle Utilities Cloud Services Configuration

Configuration of Oracle Utilities Cloud Services includes the following:

- Configuring the Point of Sale Printer Integration Master Configuration
- Configuring UI Maps and BPA Scripts

Configuring the Point of Sale Printer Integration Master Configuration

To enable printing from the three payment transactions, you must define the following in the Point of Sale (POS) Printer Integration (C1-PointOfSaleIntegConfig) master configuration:

- **Company Name:** Printed at the top of payment receipts.
- **Company Premise**: Used as the source of the company address that is printed at the top of payment receipts.
- **Payment Receipt and Endorsement Messages**: Configure up to 20 payment receipt messages and 10 endorsement messages. Define the messages under an Implementer's Message message category - i.e. either 90000 or 80000.
- **BPA scripts** to launch the **Print** dialog from each type of transaction. The base product provides three sample BPA scripts (one for each payment transaction that supports printing) and corresponding sample BPA scripts:

Processing Type	Sample BPA Script
POS Printing - Payment Event	Payment Event Print (C1-PyEvtPrt)
POS Printing - Payment Event Quick Add	Payment Event Quick Add Print (C1-PyEvQAPrt)
POS Printing - Payment Quick Add	Payment Quick Add Print (C1-PyQAPrt)

Define a BPA script for each of the processing types you want to support.

Note: If your implementation has existing receipt and endorsements messages configured on the Installation record and/or Company Name defined in the override text of message (11,99901), the Update Point Of Sale Printer Configuration (C1-UPPSC) plugin-driven batch process can be run to copy this data into this master configuration. Refer to the C1-UPPSC batch control and its related algorithms for more information.

Configuring UI Maps and BPA Scripts

The BPA scripts referenced on the Master Configuration each reference a UI map that's used to define the print dialog box and to compose the information that is printed on the payment receipt, check endorsement and stub. The base product provides sample UI maps for each of the sample BPA scripts listed above:

Sample BPA Script	Sample UI Map
Payment Event Print (C1-PyEvtPrt)	Payment Event Print Control (C1-PaymentEventPrint)
Payment Event Quick Add Print (C1-PyEvQAPrt)	Payment Event Quick Add Print (C1-PaymentEventQuickAddPrint)
Payment Quick Add Print (C1-PyQAPrt)	Payment Quick Add Print (C1-PmtQuickAddPrint)

The above-mentioned UI maps are designed for local USB point-of-sale printers. This approach differs from previous base samples that were designed for network printing. Note that the prior sample UI maps and BPA scripts will no longer be enhanced starting with this release.

The latest UI map samples print additional payment receipt information related to the: company, cashiering station, payment, tender and the payee. Refer to the UI map in the application for more details.

If your implementation requires additional information to be printed and/or certain information to be composed/printed differently, the sample UI maps and the referencing BPA scripts should be copied and configured accordingly.

Printer Installation

The following printer installation instructions are specific to the Epson TM-H6000IV USB printer that was used to code and test the sample UI maps. If using a different printer brand or model, refer to your printer's installation instructions.

1. Download the printer driver from the manufacturer's site.

   For Epson TM-H6000IV, the location is: https://epson.com/Support/Point-of-Sale/Hybrid-Printers/Epson-TM-H6000IV-with-Validation/s/SPT_C31CB25A8791?review-filter=Windows+10+64-bit
2. In the **InstallShield Wizard**, click **Next**.
3. Accept the terms of the license agreement and click **Next**.
4. Select **Minimum** install and click **Next**.
5. Add the drivers for **Receipt and Endorsement**. Click **Add** for each driver.
6. The available drivers are shown. Look for the **TM-6000IV Receipt** driver to add and click **Next**.
7. Click **Add** again and search for the **TM-6000IV Endorsement** driver. Click **Next**.
8. The download begins. After the drivers are installed and configured successfully, click **Finish**.

Recommended Printer Preferences

To get the best print quality on check endorsements, stubs and payment receipts, the following browser printer preferences are recommended.

- Printer Preferences for Endorsements and Stubs
- Printer Preferences for Payment Receipts
- Browser Printer Settings

Printer Preferences for Endorsements and Stubs

The paper size and font should be reset as follows:

1. From **Settings**, navigate to **Printers and Scanners**. Select the **Endorsement** printer and click **Manage**.
2. Select **Printing Preferences**.
3. On the **Main** tab, set **Resolution** to the highest setting of 160 x 144.
4. Navigate to the **Layout** tab to change the paper size. The default size is 230 x 297 mm. From the **Paper Size** drop-down list, select **User Defined Paper Size**.
5. On the **User Defined Paper Size** window, change the **Paper Size Name** to '80 x 100 mm'. Set **Paper Width** to 80.00 and **Paper Length** to 100.00. Click **Save Paper Size** and click **OK**.
6. The **Layout** tab is displayed with **Paper Size** defaulting to the new paper size, 80 x 100 mm.
7. Navigate to the **Printer Settings** tab to change the font. Select the **True Font Type Substitution** which then displays the **Substitution** options.
8. Select **Substitute** and click **Advanced Settings**.
9. On the **Font Substitution** page, select **Substitute All**.
10. Click **Device Font Name** where the list of available fonts is shown. Select **FontA** and click **OK**.
11. Click **Apply** and then click **OK**.

Printer Preferences for Payment Receipts

No changes are need for the Receipt Printer.

To verify the proper paper size and resolution settings are set:

1. Navigate to **Settings**. Select **Receipt Printer** and click **Manage**.
2. Select **Printing Preferences**.
3. On the **Main** tab, verify that the **Resolution** is set to “180 x 180”. This is set by default.
4. Navigate to **Layout** tab and check that the paper size is set to “Roll Paper 80 x 297”.

Browser Printer Settings

The following is an example of a browser's **Print** dialog. Note that printer settings may slightly vary by browser.

- Firefox Print Settings
- Chrome Print Settings
- Edge Print Settings

Firefox Print Settings

Receipts

Printer Destination: EPSON TM-H6000IV Receipt

Orientation: Portrait

Paper Sixe: Roll Paper 80 x 297 mm

Scale: Fit to Page

Margins: Minimum

Options: Do not select Print headers and footers or backgrounds. Leave both blank.

Endorsements

Printer Destination: EPSON TM-H6000IV Endorse

Orientation: Portrait (Endorsements) - Landscape (Stubs)

Paper Sixe: 80 x 100 mm

Scale: Fit to Page

Margins: Minimum

Options: Do not select Print headers and footers or backgrounds. Leave both blank.

> **Note**: Firefox has no print control setting for Quality as does Chrome and Edge. This makes a difference in the accuracy and quality of the print. There is still some information missing from the endorsement and stub and this is made worse by multiple endorsement messages defined on POS Master Configuration. The more endorsement messages to print on the endorsement, the more likelihood of the endorsement detail being either missing or garbled.

Chrome Print Settings

Receipts

Printer Destination: EPSON TM-H6000IV Receipt

Layout: Portrait

Paper Sixe: Roll Paper 80 x 297 mm

Scale: Fit to Page

Margins: Minimum

Options: Do not select Print headers and footers or backgrounds. Leave both blank.

Endorsements

Printer Destination: EPSON TM-H6000IV Endorse

Layout: Portrait (Endorsements) - Landscape (Stubs)

Paper Sixe: 80 x 100 mm

Quality: 160 x 144 dpi (Endorsements/Stubs only)

Scale: Fit to Page

Margins: Minimum

Options: Do not select Print headers and footers or backgrounds. Leave both blank.

Edge Print Settings

Receipts

Printer Destination: EPSON TM-H6000IV Receipt

Layout: Portrait

Paper Sixe: Roll Paper 80 x 297 mm

Scale: Actual Size 100

Margins: Minimum

Options: Do not select Print headers and footers or backgrounds. Leave both blank.

Endorsements

Printer Destination: EPSON TM-H6000IV Endorse

Layout: Portrait (Endorsements) - Landscape (Stubs)

Paper Sixe: 80 x 100 mm

Quality: 160 x 144 dpi (Endorsements/Stubs only)

Scale: Actual Size 100

Margins: Minimum

Options: Do not select Print headers and footers or backgrounds. Leave both blank.

When you are ready to print a receipt, endorsement or stub, select the appropriate printer from the **Print** drop-down list.

Also, pay close attention to **Orientation/Layout** depending on printing Receipts, Endorsement and Stubs.

- Select the **Receipt** printer to print long, short and duplicate receipts.
- Select the **Endorse** printer to print endorsements and stubs

Part Six

Web Services

This section describes how to use web services with Oracle Utilities Cloud Services. This includes:

- Chapter 21: Web Services in Oracle Utilities Cloud Services

Chapter 21

Web Services in Oracle Utilities Cloud Services

This chapter describes how to access SOAP and REST web services in the Oracle Utilities Cloud Services. This includes:

- Inbound Web Services
- Outbound Messages
- Web Service Catalog on Cloud Services
- Web Service Catalog on On-Premises Applications
- User Rights
- Debugging & Tracing Options

Inbound Web Services

In Oracle Utilities cloud services, inbound web services do not need to be deployed to be accessible. Once an inbound web service is set to active, it is ready to be used and accessed.

Oracle Utilities cloud services support both SOAP and REST services. Implementations can create custom inbound web services but no XSLs can be referenced by the inbound web service.

SOAP Inbound Web Services

This section provides information related to SOAP-based inbound web services.

Accessing a Web Service WSDL on Cloud

You can access and view the WSDL (Web Service Definition Language) of a SOAP service through the cloud service application or by executing a curl command. The WSDL file contains the structure, the schema and security specification for the desired web service.

All WSDLs are secured in Oracle Utilities cloud services. The WSDL URL must include some form of authentication to be accessed, such as a basic username and password.

Users cannot access the WSDL by providing the WSDL URL from a browser or from SOAPUI or any web service testing application. They must use the cloud service application or a curl command.

Cloud Service Application:

Use the following procedure to access a web service WSDL using an Oracle Utilities cloud service application:

1. Select **Admin**, then **Integration**, then **Inbound Web Service**, and then **Search** to access the **Inbound Web Service Query** portal.
2. Search for the SOAP service you wish to access (select "SOAP" from the **Web Service Class** drop-down list)
3. Once the SOAP web service is displayed in the **Inbound Web Service** portal, click the **WSDL** link in the **View WSDL** field.

Curl Command:

Use the following procedure to access a web service WSDL using a curl command:

The curl command format:

```
curl  -k -X GET https://{host}:{port}/{tenant}/{domain}/{appName}/
soap/api/iws/{IWSServiceName}?WSDL  -u username:password
```

where:

`https://{host}:{port}/{tenant}/{domain}/{appName}`	Product Application URL Example of Customer Cloud Service Application URL: `https://cloudenv:port/tenant/prod/ccs`
`/soap/api/iws/`	Fixed text part of the URL
`IWSServiceName`	SOAP Inbound Web Service Name
`username`	User name to login to the application
`password`	Password to login to the application

Getting the Endpoint URL on Cloud

You can obtain the endpoint URL for a SOAP service from the WSDL. Go to the service name part of the WSDL and you get the address location which is the endpoint URL.

The endpoint URL follows this format:

```
https://{host}:{port}/{tenant}/{domain}/{appName}/soap/api/iws/
IWSServiceName
```

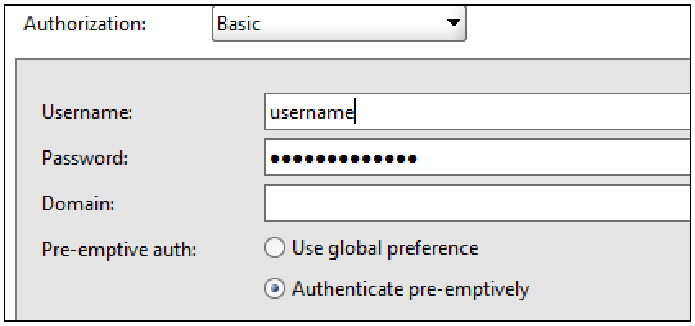
Testing and Using the SOAP Inbound Web Service

There are several resources to use when testing SOAP-based inbound web services.

This is an example of using SOAPUI to test an inbound web service.

1. Access the secured WSDL from the cloud service application or by executing the curl command as described above.

2. Save the WSDL file locally.
3. Create a new SOAP project using the saved local WSDL.
4. Provide the necessary information in the request message.
5. Provide basic authorization information.
    - Select "Basic" from the **Authorization** drop-down list.
    - Enter appropriate values in the **Username** and **Password** fields.
    - Select the **Authenticate pre-emptively** option.



Authorization: Basic

Username: username

Password: •••••••••••••

Domain:

Pre-emptive auth: ○ Use global preference

● Authenticate pre-emptively

6. Test the service as appropriate to your requirements.

REST Inbound Web Services

This section provides information related to REST-based inbound web services.

Accessing REST API Specification

REST API specifications define or describe the reference endpoints in an API. You can access and view the API Specification of a REST inbound web service through the utilities cloud application or by making a REST call.

Cloud Service Application:

Use the following procedure to access a REST API specification using an Oracle Utilities cloud service application:

1. Select **Admin**, then **Integration**, then **Inbound Web Service**, and then **Search** to access the **Inbound Web Service Query** portal.
2. Search for the REST service you wish to access (select "REST" from the **Web Service Class** drop-down list)
3. Once the REST web service is displayed in the **Inbound Web Service** portal, click the **View Specification** link in the **API Specification** field.

REST Call - Cloud Service:

Use the following procedure to access a REST API specification using a REST call with a cloud service:

The endpoint URL to use to make the REST call uses the following format:

```
https://{host}:{port}/{tenant}/{domain}/{appName}/rest/openapi/
{IWSServiceName}
```

The endpoint URL to use to make the REST call to get a v2 Swagger Specification follows this format:

```
https://{host}:{port}/{tenant}/{domain}/{appName}/rest/openapi/v2/
{IWSServiceName}
```

When making the REST call, use the Get method. Refer to the sample URL below.

Method: GET

URL: `https://host:port/tenant/domain/appName/rest/ouaf/openapi/F1-GetIWSWSDL`

REST Call - On-Premises Application:

Use the following procedure to access a REST API specification using a REST call with a on-premises application:

The endpoint URL to use to make the REST call to get an Open API Specification follows this format:

```
https://{host}:{port}/{context}/rest/ouaf/openapi/{IWSServiceName}
```

The endpoint URL to use to make the REST call to get a v2 Swagger Specification follows this format:

```
https://{host}:{port}/{context}/rest/ouaf/openapi/v2/
{IWSServiceName}
```

When making the REST call, use the Get method. Refer to the sample URL below.

Method: GET

URL: `https://host:port/ouaf/rest/ouaf/openapi/v2/F1-GetIWSWSDL`

Getting the Endpoint URL

A REST inbound web service can have multiple reference endpoints to access the resource and multiple methods for each endpoint. An endpoint URL points to a unique inbound web service name and an Operation name and shows the whole path to the resource.

The REST endpoint URL is obtained from the API Specification. It is a combination of the Computed URL and the URI Component.

The endpoint URL follows this format:

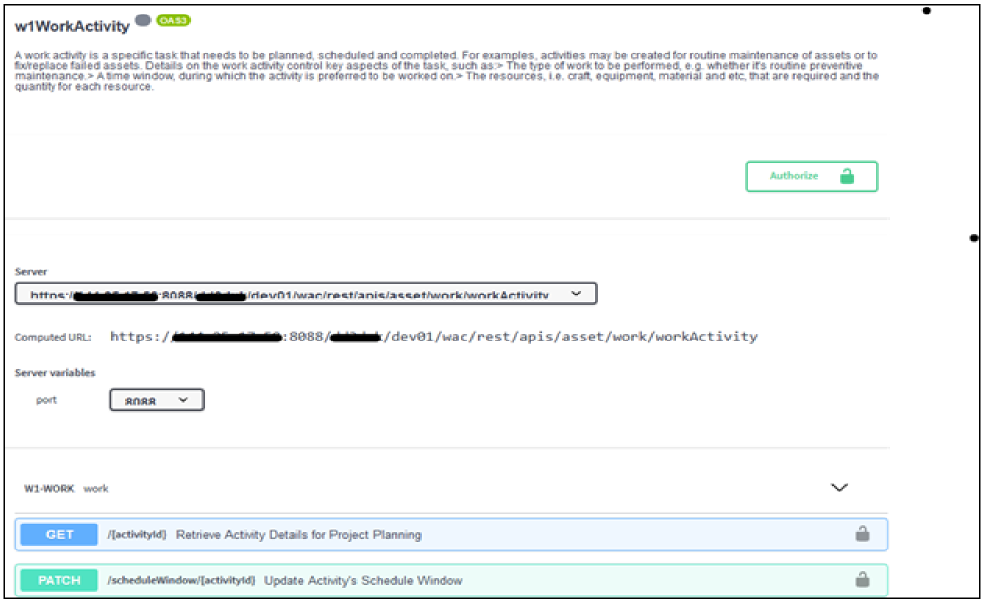
```
{ComputedURL}/{iwsOperationURLComponent}
```

where:

`ComputedURL`	Base URL referring to the common path for the API. The value is obtained from the REST API Specification of the REST inbound web service.
`iwsOperationURLComponent`	Reference a URI Component. The value is obtained from the REST API Specification -Method and URI collection or in the URI Component of the Inbound Web Service - Operations Collection.

Example:

The w1WorkActivity REST inbound web service API Specification has two resource URI Components:



w1WorkActivity OAS3

A work activity is a specific task that needs to be planned, scheduled and completed. For examples, activities may be created for routine maintenance of assets or to fix/replace failed assets. Details on the work activity control key aspects of the task, such as:> The type of work to be performed, e.g. whether it's routine preventive maintenance.> A time window, during which the activity is preferred to be worked on.> The resources, i.e. craft, equipment, material and etc, that are required and the quantity for each resource.

Server: https:/[redacted]:8088/[redacted]/dev01/wac/rest/apis/asset/work/workActivity

Computed URL: https:/[redacted]:8088/[redacted]/dev01/wac/rest/apis/asset/work/workActivity

Server variables

port: 8088

W1-WORK work

GET /{activityId} Retrieve Activity Details for Project Planning

PATCH /scheduleWindow/{activityId} Update Activity's Schedule Window

The endpoint URLs for this service are:

```
https://{hostname}:{port}/{tenant}/dev01/wacs/rest/apis/asset/
work/workActivity/{activityId}

https://{hostname}:{port}/{tenant}/dev01/wacs/rest/apis/asset/
work/workActivity/scheduleWindow/{activityId}
```

About the Computed URL

The Computed URL is the base URL used by the endpoints. A part of the URL is populated in a System Variable called F1_REST_BASE_URL. The F1_REST_BASE_URL system variable would only contain the configuration up to the 'apis' portion of the URL, and the remaining components of the URL would be derived by the application. The system variable supports the difference between formats used with on-premises applications and cloud services.

The Computed URL follows the format below:

For Cloud Applications:

```
https://{host}:{port}/{tenant}/{domain}/{appName}/rest/apis/
{ownerURLComponent}/{resourceCategoryURLComponent}/
{iwsURLComponent}
```

For On-Premises Applications:

```
https://{host}:{port}/{context}/rest/apis/{ownerURLComponent}/
{resourceCategoryURLComponent}/{ iwsURLComponent}
```

where:

`https://{host}:{port}/{tenant}/{domain}/{appName}`	Cloud Product Application URL Example of Customer Cloud Service Application URL: `https://cloudenv:port/tenant/prod/ccs`
`https://{host}:{port}/{context}`	On-Premises Application URL Example of Customer Care and Billing Application URL: `https://hostname:port/ouaf`
`/rest/apis`	Fixed text part of the URL
`ownerURLComponent`	The value is obtained from URI COMPONENT in Owner Configuration for REST services extendable lookup (F1-RESTOwnerURLComponent) where there is an entry for each owner flag. Examples: F1 (Framework): `/common` C1 (Customer): `/customer`
`resourceCategoryURLComponent`	The value is obtained from URI COMPONENT in Resource Category for REST services extendable lookup (F1-RESTResourceCategory) where there is an entry for each resource category. Example: F1-SYSTEM (System): `/system`

Note: The Computed URL for cloud applications may not always be the same as the application URL. Always get the value from the API Specification of the REST Service.

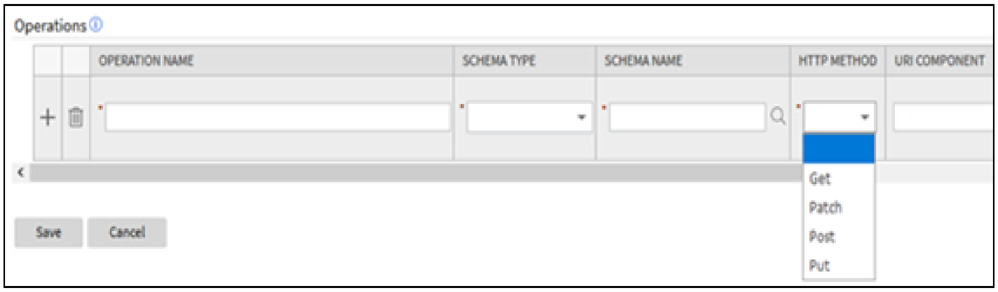
Methods and Parameters

REST inbound web services support the following HTTP Methods:

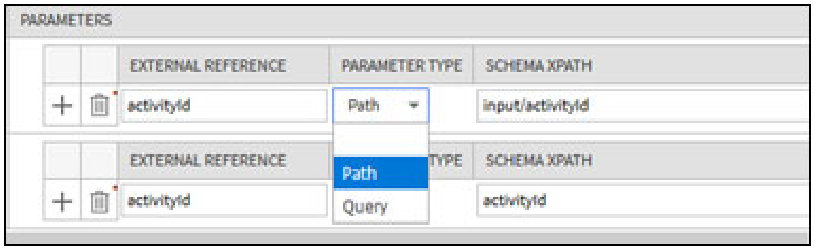
- Get

- Patch
- Post
- Put

The method is defined in the **HTTP Method** drop-down list in the **Operations** section of the **Inbound Web Service** portal.



For the Get, Put and Patch HTTP Methods, "Query" or "Path" parameters can be passed with the endpoint. This is specified using the **Parameter Type** drop-down list in the **Parameters** columns in the **Operations** section of the **Inbound Web Service** portal



EXTERNAL REFERENCE	PARAMETER TYPE	SCHEMA XPATH
activityId	Path	input/activityId
activityId		activityId

- **Path parameters**: These are parameters that are part of the endpoint and are required. Usually they are represented in the endpoint with curly braces.
- **Query parameters**: These parameters are often optional. They are not part of the endpoint but rather are included in the endpoint URL after a question mark, followed by name value pairs.

Example:

```
'../getAccountBills/
{accountId}?startDate=20190101&endDate=20190630'.
```

Testing and Using the REST Inbound Web Service

There are several resources to use when testing REST inbound web services.

This is an example of using POSTMAN to test an inbound web service.

1. Go to the REST API Specification to get the endpoint URL and the HTTP Method for the web service you wish to test.
2. Open POSTMAN and populate the necessary information.
   a. Provide the HTTP Methods and the Request URL
   b. On the **Authorization** tab, choose "Basic Auth Type" and provide the user name and password

c. On the **Header** tab, provide the following key value pair to send and accept json messages.

Key	Value	Description
Accept	application/json	Indicate the response media type that is acceptable
Content-Type	application/json	Indicate the media type of the resource. To send and accept xml messages change the value to "application/ xml".

d. On the **Body** tab, provide the request message. Sample request and response messages can also be found on the REST API Specification by expanding each resource URI.

Outbound Messages

This section describes the setup of components used to send outbound messages by invoking the SOAP or REST service of the target application. The information below assumes outbound message types have been created for each web service.

Make sure the target environment you are accessing is allowlisted in the Oracle Cloud Utilities Entitlement. This is requested by opening a service request for the Oracle Utilities Cloud Operations team.

Using SOAP Services

Setup Message Senders

Invoking a SOAP-based service involves creating one or more new real-time Message Senders.

Use the following procedure to create a Message Sender:

1. From the **Admin** menu, select **Integration**, then **Message Sender**, then **Add**.
2. Enter a unique code and **Description** for the Message Sender.
3. Populate the following values:
   - **Invocation Type**: Real-time
   - **Message Class**: SOAPSNDR (Sender for real-time HTTP/SOAP messages)
   - **Active**: Select the check box
   - **MSG Encoding**: UTF-8 message encoding

4. Select the **Context** tab and set the values for the following context types:

Context Type	Context Value (Sample Values)	Description

HTTP Header	soapAction="process”	Get the value from the soap:operation in the WSDL
		Example:
		`<wsdl:operation name="Get_APPT_SVC">`
		`<soap:operation soapAction="process"/>`

HTTP Login User	Username	User ID to access the service
HTTP Login Password	Password	Password to access the service
HTTP Method	POST	
HTTP Timeout	60	
HTTP Transport Method	SendReceive	
Message Namespace URI	http://oracle.com/GetApptMessage	Get the value from the schema namespace in the WSDL. This entry should be defined when the External System - Outbound Message Type - Namespace Option is set to Configured on the Message Sender

Context Type	Context Value (Sample Values)	Description

HTTP URL 1	@EXT_PUB@ server:port/ servicename	The value should follow the format below:
		`@EXT_PUB@server:port/servicename`
		where:
		@EXT_PUB@ refers to the outbound proxy and append the endpoint url without the https:// protocol.
		
		Get the endpoint url value from the soap:address location in the WSDL
		
		Example:
		`<wsdl:service name="receiveApptReq_PortType">`
		`<wsdl:port name="receiveApptReq_PortType_pt" binding="receiveApptReq_PortType_binding">`
		`<soap:address location="https://server:port/servicename"/>`

SOAP Insert Timestamp (Y/N)	Y	This is only needed when the wsdl policy of the service being invoked is wss_username_token_service policy or any policy that require a timestamp.

5. Click **Save**.

Setup the External System

Associate the outbound message types to their corresponding message senders in an external system.

Define the following details for each outbound message type:

- **Outbound Message Type**: The outbound message type created for the outbound message.
- **Processing Method**: Real-time.
- **Message Sender**: The Message Sender to invoke the SOAP web service.
- **Date/Time Format**: XSD
- **Namespace Option**: Configured on Sender

  Note: For Message Senders using the SOAPSNDR message class, message xsl do not need to be defined just to add the SOAP Envelope.

The SOAPSNDR message class will add the SOAP Envelope before sending the message out.

Using REST Services

Setup Message Senders

Invoking a REST-based service involves creating one or more new real-time Message Senders.

Use the following procedure to create a Message Sender:

1. From the **Admin** menu, select **Integration**, then **Message Sender**, then **Add**.
2. Enter a unique code and **Description** for the Message Sender.
3. Populate the following values:
    - **Invocation Type**: Real-time
    - **Message Class**: RTJSONSNDR (Sender for real-time JSON messages)
    - **Active**: Select the check box
    - **MSG Encoding**: UTF-8 message encoding
4. Select the **Context** tab and set the values for the following context types:

Context Type	Context Value (Sample Values)	Description
HTTP Login User	Username	User ID to access the service
HTTP Login Password	Password	Password to access the service
HTTP Method	POST	
HTTP Timeout	60	
HTTP URL 1	@EXT_PUB@RESTEndpointURL	The value should follow the format below: @EXT_PUB@RESTEndpointURL where" @EXT_PUB@ refers to the outbound proxy and append the REST endpoint url without the https:// protocol.

5. Click **Save**.

Setup the External System

Associate the outbound message types to their corresponding message senders in an external system.

Define the following details for each outbound message type:

- **Outbound Message Type**: The outbound message type created for the outbound message.

- **Processing Method**: Real-time.
- **Message Sender**: The Message Sender to invoke the REST web service.
- **Date/Time Format**: XSD
- **JSON Conversion Method**: Base JSON Conversion

  Note: For Message Senders using the SOAPSNDR message class, message xsl do not need to be defined. SOAPSNDR message class will add the SOAP Envelope before sending the message out.

Web Service Catalog on Cloud Services

For Oracle Integration Cloud to retrieve the REST catalog or SOAP catalog from Oracle Utilities cloud service applications, use the following endpoint URLs.

The endpoint URL for the REST Catalog Service can be obtained by following this format:

```
https://{host}:{port}/{tenant}/{domain}/{appName}/rest/openapi/
iws/catalog
```

The endpoint URL for the SOAP Catalog Service can be obtained by following this format:

```
https://{host}:{port}/{tenant}/{domain}/{appName}/soap/api/iws/
ServiceCatalog
```

Web Service Catalog on On-Premises Applications

For Oracle Integration Cloud to retrieve the REST catalog or SOAP catalog from Oracle Utilities on-premises applications, use the following endpoint URLs.

The endpoint URL for the REST Catalog Service can be obtained by following this format:

```
https://{host}:{port}/{context}/rest/ouaf/openapi/iws/catalog
```

The endpoint URL for the SOAP Catalog Service can be obtained by following this format:

```
https://{host}:{port}/{context}/webservices/builtin/
ServiceCatalog?WSDL
```

User Rights

Web service calls must be authorized for the calling user. In other words, the user must exist as an OUAF User with adequate application services for the underlying services called by the inbound web service, and the debug services. The debug Application Services are F1DEBUG to enable a url with the debug=true setting, and F1USERLOG to view user logs in the online system. Also note that the 'Integration Suite API' has a separate Application Service (C1-INTG-SUITE-API) which is required (when licensed).

You will probably not want to grant any more access to the inbound web service calling user than they absolutely need, so Oracle recommends creating a separate User Group as needed to support very specific access. Check the User Group and Application Services settings for the user.

For example: if the inbound web service is reading an Account, then the user will need read rights on the Account service (CILCACCP).

Debugging & Tracing Options

This section outlines options for tracing and debugging issues when accessing inbound web services.

REST Inbound Web Services

The following debugging and tracing options apply when using REST services:

- REST calls can be made within the application using the **View Specification** link, which will show the curl format of the call, response, etc.
- REST services can be invoked with a debug parameter (`http:<cloud url>//restapi?debug=true`) to show all the debug logs when checking via the application.

  Using the debug parameter will provide additional information in the kibana logs (tech log information) about execution of each step of the scripts.
- To see the user log, the 'calling user' must log into the online application and view User Logs.
- If the **Trace** flag checked on inbound web service, requests and responses are written to the user log.

SOAP Inbound Web Services

The following debugging and tracing options apply when using SOAP services:

- SOAP requests with trace enabled on the inbound web service shows the request message and response message in the user logs, but not in Kibana.
- Using the Debug flag in the soap envelope enables debug mode, but does not enable tracing. The debug flag in soap header can be passed as <debug> or <Debug> and values can be true or yes. For example: `<debug>true</debug>`.
- If the **Trace** flag checked on the inbound web service, requests and responses are written to user log.