Oracle Utilities Cloud Services

Implementation Guide

Release 26.4
G50319-04
May 2026

ORACLE

Oracle Utilities Cloud Services Implementation Guide, Release 26.4

G50319-04

Contents

6 Data Access and Analytics

7 Information Lifecycle Management

8 Network Integration Guidelines for Integrating Oracle Utilities Cloud Services with External Applications

Index

1

Introduction

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 Customer Program Management Cloud Service
- Oracle Utilities Market Settlements Management Cloud Service
- Oracle Utilities Meter Solution Cloud Service
- Oracle Utilities Rate Cloud Service
- Oracle Utilities Work and Asset Cloud Service

This document includes:

- Implementation Guidelines
- Data Conversion and Migration
- File-Based Integration
- Oracle REST Data Services
- Product-Specific Integrations
- Web Services in Oracle Utilities Cloud Services

Part I
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 Customer Program Management Cloud Service
- Oracle Utilities Market Settlements Management Cloud Service
- Oracle Utilities Meter Solution 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:

- Post-Provisioning Setup
- Security and Access
- Configuration Tools
- Integration Guidelines
- Data Access and Analytics
- Information Lifecycle Management
- Network Integration Guidelines for Integrating Oracle Utilities Cloud Services with External Applications
- Migrating Legacy Custom Tables and Java to Oracle Utilities Cloud Services

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
- Initial Cloud Service Setup - Adding a New Environment to an Existing Cloud Service
- Configure Product Use Metrics - Required
- Language Pack Setup (Optional)
- Oracle Guided Learning Integration (Optional)

Before You Begin

This chapter describes how Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) is used with Oracle Utilities cloud services.

Before you begin configuring Oracle Cloud Infrastructure Identity and Access Management for your implementation, you should review the following Oracle Cloud Infrastructure documentation:

- Oracle Cloud Infrastructure Security Guide
- Security Best Practices
- Securing IAM

We also recommend you review the following technical papers and blog articles related to IAM security:

- Best Practices for Identity and Access Management (IAM) in Oracle Cloud Infrastructure
- OCI IAM Identity Domains Best Practices
- OCI IAM Policies Best Practices

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 dedicated user and group in the OCI IAM Identity Domain for the Process Automation Tool and assign it to all your cloud service environments. Complete a setup by logging into each environment and verify that the user is created. Adjust user security access, if needed, using the K1PAUSER template user as an example.
- Define the initial set of user groups and assign appropriate access rights to each.

- Create Oracle Cloud Infrastructure Identity and Access Management (IAM) with 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 with Identity Domains** 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 Tool user you created previously.
  - Password of that user in IAM with 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

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..20/TEST/TEST01..20/ 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.

Initial Cloud Service Setup - Adding a New Environment to an Existing Cloud Service

In order to setup the newly provisioned environment, customers are required to follow the below mentioned steps:

- Identity Management Setup
  - Create User group(s) for the newly provisioned environment, if needed.
  - Assign roles to access the new environment to the new / existing groups.
  - Add Master Configuration (Identity and Access Management Integration Configuration) for mapping new or existing user groups to the template user(s) in the newly provisioned environment.

  Refer to **Identity and Access Management with Identity Domains** in the *Oracle Utilities Cloud Services Administration Guide* for more information.
- Object Storage Setup
  - Create a new System Account User in OCI IAM Identity Domain representing the new environment.
  - Create a key ring for object storage access
  - Generate key in the key ring value and activate it.
  - Update the extendable lookup (F1-FileStorage) by adding OS-SHARED and OS-APP values referring to respective object storage compartments (with OCIDs of System Account User, Tenancy, Compartment and Namespace)
  - Get the Public Key from the active generated key value and update to the System Account User in the OCI IAM Identity Domain.

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

Process Automation Tool Setup (for a newly provisioned environment)

In order to set up the Process Automation Tool for all of your newly provisioned environments (for example, DEV01), run the following batch job in the newly provisioned 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..20/TEST/TEST01..20/ PROD
    - <Env List> = environment codes separated by comma without spaces, for example: DEV,TEST,PROD

For example, when the job runs in the DEV01 environment for Customer Cloud Service (CCS) the parameters would be as follows:

```
SCRIPT01_DATA: INIT,CCS,DEV01,file-storage://OS-SHARED/CMA-Files,file-storage://OS-SHARED/CMA-Files,DEV,TEST,PROD
```

Process Automation Tool Setup (for existing cloud environments)

If you add a new environment to your cloud service (for example, DEV01), you must run a similar batch job in each existing environment.

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..20/TEST/TEST01..20/ PROD
    - <Added Env List> = environment codes separated by comma without spaces, for example: DEV,TEST,PROD

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

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

Security Setup

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 Administrative User Guide* for more information.

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

Configure Product Use Metrics - Required

This section describes the process for configuring the Product Use Metrics dashboard used with Oracle Utilities Cloud Services.

The Product Use Metrics dashboard displays metrics related to cloud service subscriptions such as customer counts, device counts, channels of an active meters, number of billable services, and other similar data. The data presented on each dashboard is tailored to provide summary information for management. Refer to Product Use Metrics in the Administrative User Guide for more information about this dashboard.

This dashboard leverages a method of pre-staging data known as Product Metrics Snapshots. See Capturing Product Use Metrics in the Administrative User Guide for more information on Product Metrics Snapshots.

> **Note**
>
> Customer must review the Product Use Metrics dashboard on a monthly basis and communicate with Oracle Sales and/or your Customer Success Manager to reconcile the billable service quantities in use with subscribed quantities, as needed.

Configuring the Product Use Metrics dashboard includes the following:

Configure Master Configuration

The Customer Metrics Dashboard Configuration master configuration is used to configure details related to the data displayed on the **Product Use Metrics** dashboard including specific types of data that should be excluded on the dashboard (including billable service agreement types and debt classes), as well as the colors used in the dashboard zones.

You can access the Master Configuration portal by selecting **Admin**, then **General**, then **Master Configuration**.

Select the Customer Metrics Dashboard Configuration master configuration from the **Master Configuration** zone.

Use the **Add** button beside the record to configure for the first time. If a record has already been added, then click the **Edit** button instead. Use the embedded help to guide you through the meaning of each configuration field.

> **Note**
>
> Configuration of excluded data is strongly recommended. If you fail to exclude Billable SA Types and Debt classes, the data captured as billable service quantities may be significantly higher that your subscribed cloud service billable service quantities. Based on the captured data you will be notified by your Customer Success Manager and/or your sales representative about any discrepancies between subscribed billable service quantities and billable service quantities captured on the **Product Use Metrics** dashboard.

Configure Snapshot Business Objects and Algorithms (Optional)

By default, the Device Count Snapshot and Channel Count Snapshot business objects check for active usage subscriptions when deriving the total count of active devices and measuring components. However, in some implementations this is not required or appropriate.

For the Device Count Snapshot (D1-DeviceCountMetric) business object, the usage subscription check can be disabled if needed by setting the "Exclude US Check (Y or N)" parameter on the Create Device Count Metric Snapshot (D1DEVCNTSNP) Enter algorithm on the Complete state of the Snapshot business object to "Y" (this parameter is set to "N" by default).

For Channel Count Snapshot (D1-ChannelCountMetric) business object, the usage subscription check can be disabled if needed by setting the "Exclude US Check (Y or N) parameter on the Create Channel Count Metric Snapshot (D1CHLCNTSNP) Enter algorithm on the Complete state of the Snapshot business object to "Y" (this parameter is set to "N" by default).

Batch Processing

The Product Use Metrics dashboard is configured to be refreshed on a monthly basis. To refresh the data used in the dashboard, Oracle runs the F1-PMTRC (Product Metric Type Monitor) batch process via monthly internal batch streams. This batch process invokes monitoring rules associated with the current state of Product Metric Type records and capture the latest Product Metric Snapshot.

Operational Device Management (Optional for Meter Solution Cloud Service Subscriptions)

Oracle Utilities Meter Solution Cloud Service (MSCS) environment provisioning disables Operational Device Management (ODM) functionality by default. In order to enable the Operational Device Management functionality in Meter Solution Cloud Service, customers must run the Turn On ODM and C2M (D1-OnOdmC2m) BPA script. This script enables X1- and W1-owned product objects in newly provisioned MSCS environments.

For existing Meter Solution Cloud Service customers who would like Operational Device Management functionality enabled require custom configuration. Customers can create a service request to obtain the guidelines for this configuration.

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.

Release	Product	User Interface / Online Help	Language
26.4	Oracle Utilities Customer Cloud Service	User Interface	Arabic, Latin American Spanish, Brazilian Portuguese, Traditional Chinese
		Online Help	Arabic, Brazilian Portuguese, Traditional Chinese
	Oracle Utilities Testing Accelerator	User Interface	Arabic, Latin American Spanish, Brazilian Portuguese, and Traditional Chinese
25.10	Oracle Utilities Customer Cloud Service	User Interface	Arabic, Latin American Spanish, Brazilian Portuguese, Traditional Chinese
		Online Help	Arabic, Brazilian Portuguese, Traditional Chinese
	Oracle Utilities Testing Accelerator	User Interface	Arabic, Latin American Spanish, Brazilian Portuguese, and Traditional Chinese
25.4	Oracle Utilities Customer Cloud Service	User Interface	Arabic, Latin American Spanish, Brazilian Portuguese, Traditional Chinese
		Online Help	Arabic, Brazilian Portuguese, Traditional Chinese
	Oracle Utilities Testing Accelerator	User Interface	Arabic, Latin American Spanish, Brazilian Portuguese, and Traditional Chinese

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', 'PTB', or 'ESA', as the Language Code for Arabic, Brazilian Portuguese, or 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 (for example, 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.

Oracle Guided Learning Integration (Optional)

If you have licensing the Oracle Guided Learning Cloud Service and wish to integrate it with your Oracle Utilities cloud service, create a Service Request (SR) for the Cloud Operations team.

This integration requires that you provide your Oracle Guided Learning ID along with the target cloud service environment(s).

3

Security and Access

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

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 (IAM) with Identity Domains to manage user creation, application access, passwords, and so on. 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 with 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 (such as disabling a user, or reseting 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

Server Logs - Online and Batch in the *Oracle Utilities Cloud Services Live Operations Guide* for more information.

SMTP Integration

The only supported Simple Mail Transfer Protocol (SMTP) service is Oracle Cloud Infrastructure Email Delivery, which is embedded within your cloud service with the following restriction:

- When used with Oracle Utilities cloud services, the OCI Email Delivery Service may only be used to send emails to Oracle, for example to tell them that their batch job failed. Any attempts to use this embedded instance of the OCI Email Delivery Service for other purposes are blocked and/or rejected.
  The following limitations apply to OCI Email Delivery Service subscriptions included with your cloud service:
  - Daily limit of 100 emails / Partition (prod/non-prod)
  - Sending rate of 10 emails per minute
  - 2 MB maximum message size including base64 encoding and headers
  - Maximum of 10 recipients per email (in the TO:, CC:, or BCC: fields)
- In order to send emails to the customers' customers, such as for bill delivery or notification of bill generation, the customer must have a separate **Enterprise** subscription to Oracle Cloud Infrastructure Email Delivery. Refer to the Oracle Cloud Infrastructure documentation for details about configuration of the OCI Email Delivery Service.

Implementation Guidelines

- Customer subscribed to the OCI Email Delivery Service are supported in the same region where your cloud services are deployed. There is no support for Cross region email server domains. @SMTP_SERVER@ will always point to the region where your cloud services are hosted.
- In order to use the Customer subscribed OCI Email, the following are required configuration on the "EMAIL_SENDER" message sender:
  - **SMTP Host Name**: @SMTP_SERVER@ (Default / Do NOT Change)
  - **SMTP From Address**: your fromaddress (Required)
  - **SMTP Username**: your username (Required)
  - **SMTP Password**: your password (Required)

> **Note**
>
> Customers may use F1-EmailService (Business Service) to perform email configuration testing.

- The OCI Email Delivery Service determines the "From Address" as follows:
  1. If the "From Address" is provided in the F1-EmailService business service payload, it will be used to send the email.
  2. If no "From Address" is provided in the business service payload and the message sender is configured, the "SMTP From Address" configured in message sender will be used.

- If no "SMTP From Address" is defined on the provided message sender, the system raises an error.

3. If no message sender present in the payload, we will use the default message sender. If no "SMTP From Address" is defined on the default message sender, the system raises an error.

> **Note**
>
> If you are using the OCI default SMTP Server, you must provide the following values for the "SMTP From Address" based on the type of environment in which your cloud services are deployed:
>
> | Environment Type | "SMTP From Address" Value |
> |---|---|
> | Non-Production | 'noreply@**sh**.utilities-cloud.ocs.oraclecloud.com' |
> | Production | "noreply@**ph**.utilities-cloud.ocs.oraclecloud.com' |
>
> If you are using subscribed OCI Email Delivery, then you should use your own "SMTP From Address" and it should be used along with the User Name and Password defined on the Message Sender configuration.

Refer to Energy & Water SaaS Email Changes with 24C (KB204968) on My Oracle Support for additional information about using OCI Email Delivery with Oracle Utilities cloud services.

Batch Error / Completion Notification

Cloud service customers can use the OCI Email Delivery System to send batch error / completion notifications (based on Batch Control parameter configuration) to individuals or email groups. This functionality can be used for both online batch job submission and / or via batch scheduler batch control configuration.

This functionality does not include email notification on Batch Level of Service algorithms (BLOS). The Health Check portal within the cloud service should be used to view BLOS notifications. Customers can write external probes to pull the BLOS notifications via cloud web services.

Additional Options for Extracting and Sending Data via Email Delivery

Some options for extracting and sending data to external systems via email include the following:

- Create a plug-in driven batch control that writes a file to Oracle Cloud Object Storage, and then write or configure a process (outside of the cloud service) that looks for these files and sends emails based on the extracted data in the file. For example, a process might create a "mail merge" file containing data for a number of entities for external email server consumption.
- Deploy a web service outside of the subscribed Oracle cloud service network that is accessible by your cloud service. Create a plug-in driven batch process or online algorithm that calls the external web service via a Message Sender for each email notification you wish to send.

Content Security Policy

Content Security Policy (CSP) is a feature that helps to prevent or minimize the risk of certain types of security threats. The primary use case for this is to control which resources, in particular JavaScript resources, a document is allowed to load. By enabling this response header, customers get the security benefits related to this.

Customers are expected to run tests to understand the external URLs they use in the application. Customers should fill in the Content Security Policy (CSP) Modification Request form with details and raise a Service Request to the Oracle Utilities Cloud Operations team to get them included in the CSP.

> **Note**
>
> Content Security Policy will be enabled by default in the 26.4 release, so if external URLs are not added to the CSP, it will result in a loss of functionality.

Testing Your Content Security Policy

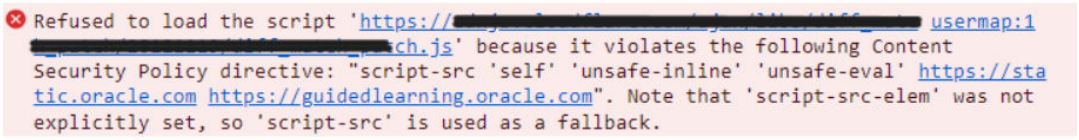
To test your Content Security Policy, use the following procedure:

> **Note**
>
> Users must have F1-DEBUG privileges to be able to enable the csp reporting.

> **Note**
>
> The steps below are based on using Google Chrome.

1. Login into the application using your user ID and password.
2. Modify the URL to add the following parameter: ?cspReportOnly=true Example: http://servername:portno/spl/cis.jsp**?cspReportOnly=true**
3. Refresh your browser. In the browser refresh option choose the "Empty cache hard refresh" option. This will ensure that the cache is deleted and all the contents are freshly loaded.
4. Open the **Settings** menu (Alt + F in Google Chrome), select **More Tools**, and then select **Developer Tools**. Ensure you see the Content-Security-Policy-Report-Only response header for all of your different requests.
5. Navigate through CM modules and modules where there is a possibility to use or access other URLs. For example, in the image below we can see that the error is related to a URL missing in the `script-src` directory.

Figure 3-1 Content Security Policy Error Message



6. Go back to the **Developer Tools** panel and select the **Console** tab. Look for possible errors starting with [Report Only] similar to the image above. For example:

   ```
   [Report Only] Refused to execute inline event handler because it violates
   the following Content Security Policy directive: …..)
   ```

7. Customers can look for the directory name(s) listed in the Content Security Policy Modification Request form. Examples include `img-src`, `script-src`, and so on. Collect these and document like the Sample Completed Content Security Policy Modification Request Form below.
8. Logout of the application.
9. In the browser refresh option, choose the "Empty cache hard refresh‘" option. This will ensure that the cache is deleted and all the contents are freshly loaded. This will ensure to fresh download of the static files without the Content-Security-Policy- Report-Only response header.

Content Security Policy Modification Request Form

Below is the form customers should fill out.

> **Note**
>
> The "Cloud Operations Key" will provide the name of the substitution variable.

Environments names: (example: DEV, TEST, PROD)

Directory Name	Cloud Operations Key	URL That is Causing the Violation (For example, if the URL is https://customer-domain1.com/abc/abc.js, then specify https://customer-domain1.com without the path. Multiple URLs should be delimited by spaces.)
img-src	CSP_URL_IMG_SRC	Used as a substitution variable to allow access to an image from an external source.
script-src	CSP_URL_SCRIPT_SRC	Used as a substitution variable to allow access to a script from an external source.
frame-src	CSP_URL_FRAME_SRC	Used as a substitution variable to allow access to a frame from an external source.

Directory Name	Cloud Operations Key	URL That is Causing the Violation (For example, if the URL is https://customer-domain1.com/abc/abc.js, then specify https://customer-domain1.com without the path. Multiple URLs should be delimited by spaces.)
style-src	CSP_STYLE_SRC	Used as a substitution variable to allow access to a style from an external source.
connect-src	CSP_CONNECT_SRC	Used as a substitution variable to allow access to a network connection from an external source.
font-src	CSP_FONT_SRC	Used as a substitution variable to allow access to a font from an external source.

To override and or append the content security policy (CSP) to use external customer specific urls, images, fonts and so on with in your cloud service environment, substitution variables may be used but this requires a Service Request from the customer to set these parameters to the required external system.

Sample Completed Content Security Policy Modification Request Form

Here is an example of a completed form: ENVIRONMENT: DEV01, TEST

Environments names: (example: DEV, TEST, PROD)

Directory Name	Cloud Operations Key	URL That is Causing the Violation (For example, if the URL is https://customer-domain1.com/abc/abc.js, then specify https://customer-domain1.com without the path. Multiple URLs should be delimited by spaces.)
img-src	CSP_URL_IMG_SRC	https://customer-domain1.com
script-src	CSP_URL_SCRIPT_SRC	https://customer-domain1.com
frame-src	CSP_URL_FRAME_SRC	
style-src	CSP_STYLE_SRC	https://customer-domain1.com https://customer-domain2.com
connect-src	CSP_CONNECT_SRC	
font-src	CSP_FONT_SRC	

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 Accesssection 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. Refer to Migrating Legacy Custom Tables and Java to Oracle Utilities Cloud Services for more information.
User Interface	Add a custom portal.	Yes	
	Extend a product portal with custom zones.	Yes	
	Extend a product multi-query search with custom query options.	Yes	

Category	Option	Supported?	Comment
	Customize a product menu. This includes adding new custom menu lines, hiding and reordering lines.	Yes	
	Add custom indexes to support custom queries	No	This is not supported, however in some special situations, customers can request this by submitting a service request (SR). Please note that this requires proper justification and exceptions can only be made if found justified.
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.
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 Oracle Database Actions toolset to check SQL.

Application Environments

Each cloud service by default comes with three environments designated as Development, Pre-Production (formally called Test), and Production. Pre-Production 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.

Available additional non-production environment types include:

- Additional Pre-Production Non-Production Environment
  - Designed for: production volume activities such as pre-production staging, parallel testing and performance testing (including regular performance regression testing)
  - Size: production size in terms of database storage as well as batch and online capacity
- Additional Functional Test Non-Production Environment
  - Designed for: functional (including regular functional regression), system, integration, and user acceptance testing activities
  - Size: production sized in terms of database storage, and one-third (1/3rd) of the production batch and online capacity (or a minimum Development environment capacity, whichever is larger)
- Additional Training Non-Production Environment
  - Designed for: end-user training and product familiarization type activities
  - Size: development sized in terms of database storage and batch capacity, but with additional online capacity to support up to thirty (30) concurrent online users
- Additional Development Non-Production Environment

- Designed for: development and unit testing activities only
- Size: fixed, minimum development size

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	Pre-Production	Yes	No
PROD	Production	Production	Yes	No
DEV01..DEV20	Development 1 .. 20	Development or Training	No	Yes
TEST01..TEST20	Test 1 .. 20	Pre-Production or Functional Test	No	Yes

Environment Type Use Cases

Cloud service environments have been designed and sized for specific use cases during the implementation and live operation phases. Usage and processing limits are specified in the relevant contract documents (including Service Descriptions), and additional, environment specific, sizing information may additionally be provided via operational notifications.

Production Environment

The Production environment is designed for daily commercial use and production operations of live data during the live operation phase (that is, after go-live). During the implementation phase, the Production environment is intended to be used by customers and implementers for all data conversion and data migration related activities. This allows Pre-Production and Functional Test environments to be reserved for performance and functional testing of your configuration (respectively).

Pre-Production and Functional Test Environments

Both Pre-Production and Functional Test environments are designed for non-production use only, and both allow for a full Production database copy to be loaded.

Pre-Production environments are designed for:

- Performance testing, regular performance regression testing and parallel testing of configuration during the initial implementation phase, and of configuration updates released by customers/implementers during live operations
- Performance regression testing of cloud services after patching or updates
- Pre-production staging activities for large migration data sets added during live operations

Functional Test environments are designed for

- Functional, system integration, and user acceptance of configuration during the initial implementation phase, and for configuration updates released by customers/implementers during live operations.
- Functional regression testing of cloud services after patching or updates.

- Troubleshooting of production issues where production data and production data volumes may be required.

Oracle recommends limiting the number of Pre-Production environments to one due to the higher associated subscription costs. General observation of the existing customer base suggests that the Pre-Production environment included with your base subscription is sufficient for most implementation and live operation scenarios, however additional Pre-Production environments are available for subscription should they be required.

Training and Development Environments

Development environments are small, fixed size environments designed for non-production use only, and are designed for the configuration of cloud service features and unit testing. The Development environment type supports a up to ten (10) concurrent online users.

Training environments are the same size as Development environments, except that they are designed to support up to thirty (30) concurrent online users, which is a common scenario during end-user training or product familiarization sessions.

Additional Environment Resources

Additional data storage and processing resources may be subscribed to separately. These additional resources provide additional processing and/or storage capacity for a specific environment over and above that which is provided as part of the base cloud service. The additional resources that are available for subscription are as follows:

- Additional Batch Threads
- Additional Concurrent Online Users
- Additional Data Storage
- Additional Integration Requests per Minute

Each cloud service is benchmarked to confirm scalability, and is sized according to a reasonable set of assumptions based on customer base observations, however some implementations may require additional processing resources (additional batch threads, concurrent online users or integration requests per minute) and/or additional storage. Common reasons for requiring additional resources include (but are not limited to):

- High volume, high frequency interval read data (specifically, less than 15 minute interval data such as 5 minute interval data)
  - Example: 5 minute interval data involves 3x the data volumes of 15 minute interval data, which means a significant increase to batch processing and storage capacity requirements.
- Effective critical batch window durations of less than 6 hours
- Highly complex configuration, custom integration and/or batch logic
- Inefficient batch process scheduling
- Inefficient configuration, customer integration and/or batch logic which has not been sufficiently performance tested, optimized, and tuned during the implementation phase prior to go-live
- Not properly implementing the recommended Information Lifecycle Management (ILM) business rules
- Failure to ensure that ILM processing occurs regularly and that unnecessary database partitions are regularly dropped
- Storing high volumes of custom information/data (for example via the Java Migration Cloud Service)

- Transitioning to live operations with large numbers of uncorrected data migration validation errors, and/or data that has not been properly sanitized
- Not implementing Information Lifecycle Management protocol

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 the `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_WSCCS_ONLINE	Customer Cloud Service address for web service calls. Customer Cloud Service address for online access.
MSCS_WSMSCS_ONLINE	Meter Solution Cloud Service address for web service calls. Meter Solution Cloud Service address for online access.
WACS_WSWACS_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.
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.

Token	Description
BIP_DEF_PATH	Used as soft parameter value ("Report Absoluted Path") to Analytics Publisher extract algorithm type.
CI_BILL_URL	Used as a substitution variable to allow access to a bill from an external source based on the request URL.
CI_LETTER_URL	Used as a substitution variable to allow access to a Letter from an external source based on the request URL.
DEV_WS, DEV01_WS-DEV10_WSTEST_WS, TEST01_WS-TEST10_WSPROD_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.
- The CI_BILL_URL and CI_LETTER_URL substitution variables can be used with base online bill and letter display algorithm parameters, but this requires a Service Request from the customer to set these parameters to the external system.
- To override and or append the content security policy (CSP) to use external customer specific urls, images, fonts and so on with in your cloud service environment, substitution variables may be used but this requires a Service Request from the customer to set these parameters to the required external system.

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 **Data Fix with Plug-In Driven Batch** in the *Oracle Utilities Cloud Service Live Operations Guide* 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.

5

Integration Guidelines

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

- Integration Methods
- Integration Middleware
- Allowlisting
- Integration Best Practices

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, and so on) 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, and so on. 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 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 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 (KB236177) document on My Oracle Support.

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, and so on.

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, and so on.

Allowlisting

Allowlisting is required to specify allowable access destinations on the public internet. There are networking scenarios documented in detail in Network Integration Guidelines for Integrating Oracle Utilities Cloud Services with External Applications.

IP Allowlisting

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

Inbound Traffic

Inbound traffic is controlled via allow list of IP addresses.

The inbound allow list feature provides a way to allow or deny inbound requests based on user-defined configuration.

By default, all inbound requests coming from all sources are allowed for Oracle Utilities Cloud Services.

Customers are given the capability to override this default behavior. It is possible to limit the sources that are able to perform inbound requests.

If a customer would like to customize or override the inbound allow list behavior, the expected flow is:

- Customers need to determine how they will identify the sources that are allowed to access the resources:
  - via IP address ranges defined as CIDR blocks
  - via VCN OCID - only sources that access the resources via the OCI service gateway
  - both IP and VCN OCID

Outbound Traffic

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

The customer or system integrator can request a DNS (Domain name service) name to be added in the allowlist for outbound interface communication. An allowlist provides access to specified DNS addresses that the Oracle network would otherwise prevent access to. For Oracle Utilities cloud services, a customer or system integrator must request a DNS to be added to the allowlist for outbound communication to all external systems.

Once the requested DNS entry is added to the outbound allowlist, it is a customer responsibility to pro-actively maintain the following requirements:

- TLS / SSL Certificate should be issued by a valid SSL Authority
- Certificate's name(s) must match the server / endpoint name
- Installation of TLS / SSL Certificate should include complete authentication chain
- Expiry / Validation of TLS / SSL Certificate of the endpoint
- Support minimum of TLS 1.2

> **Note**
>
> Customers may use TLS / SSL validations tools such as openssl, TLS / SSL verification websites (such as https://www.ssllabs.com/) to validate the compliance requirements mentioned above.

Configuring IP Allow Lists

To configure IP allow lists, customers must log a service request and follow the steps outlined in the following sections in *Oracle Utilities Cloud Services Cloud Operations Guide* to provide configuration details:

- Inbound: Request for Inbound Allow List
- Outbound: Request for DNS Address to be Added to Outbound Allow List

Integration Best Practices

Oracle Utilities applications integrate with each other or with other external applications to enable new functionalities or to exchange data needed to verify, process, or complete transactions.

This section presents common integration use cases and demonstrates how they can be used efficiently without impacting the overall cloud services performance. This includes:

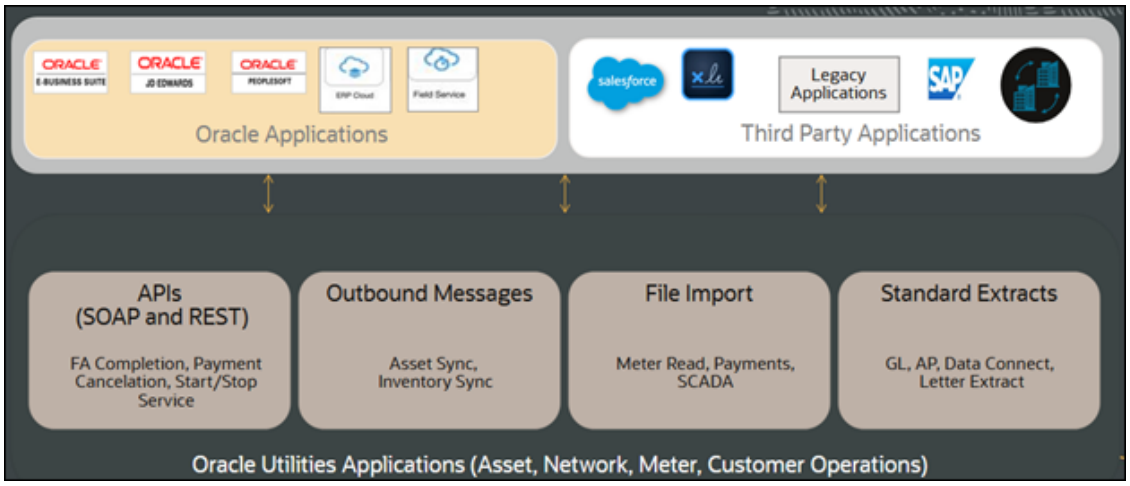
- Primary Integration Methods
- Common Use Cases
- Best Practices

Primary Integration Methods

As noted earlier in this chapter, the primary integration methods supported with the cloud services are:

- Inbound and Outbound Files
- Inbound and Outbound Web Services

Figure 5-1 Integration Options



Inbound and Outbound Files

File processing can be implemented in two primary ways: File Import as Inbound and Standard Extracts as Outbound, depending on the direction of data exchange. The only supported storage with all cloud services is Cloud Object Storage where separate buckets can be created to separate inbound and outbound files.

File Import or Inbound:

Files can be dropped into the Cloud Object Storage and then processed using an Upload Batch Process. There are various fit for purpose upload processes provided by the product. For example, payment upload.

In addition, the product provides a plug-in driven upload. The upload treats one file as one unit of work and streams it to an algorithm that is responsible for parsing the data and adding the data to a table with minimal validations. One run of the upload process can process multiple files, and each file is treated as a unit of work. Multi-threading may occur if there are multiple files.

For more information on the product recommendations related to the responsibility of the upload algorithm, refer to **Uploading Records** in the *Application Framework Administrative User Guide*.

Standard Extracts or Outbound:

An Extract Batch Process can be used to export data from an application object into a file, which is then placed in a designated location for consumption by other systems.

Example: Exporting financial information such as General Ledger (GL) and Accounts Payable (AP) data to be picked up by an external financial system.

Best Practice:

- The recommendation is to split larger files into manageable sizes so that the resulting multiple files can be processed multi-threaded.
- Batch processing should not be executed during peak business hours.
- Use auto archiving feature on cloud object storage to cleanup the buckets for better manageability and improve batch performance.

Inbound and Outbound Web Services

- **API Data Integration**: External or other utilities applications can interface to another utilities application via Inbound Web Services. Using a SOAP or a REST API to communicate and share data between applications. Inbound Web Services can be used for real-time and non-real-time integrations:
  - Real-time Example: Searching for available appointments from an external scheduling tool.
  - Non-real-time Example: Recording in the Customer Information System the details of customer notifications sent out, where the request is submitted via batch, Oracle Cloud Infrastructure (OCI) stream, or message queue.
- **Outbound Messages**: Utilities applications can initiate real time or non-real time outbound communications to other applications.
  - Real-time outbound messages are used for critical or time-sensitive processes.
    - Example: Reporting an outage to an external system as soon as it is logged.
  - Non-real-time outbound messages are typically submitted using a batch monitor or scheduled process.
    - Example: Sending updates to customer information details at scheduled intervals.

Common Use Cases

Use Case 1: Data Synchronization Between Applications

Data is generally maintained in a designated system of record, which serves as the authoritative source for that data. In many cases, this data needs to be synchronized to other systems to support or validate business processes.

Example: Copying service point information from a Customer Information System (CIS) application to an Asset Management Application or Meter Data Management application.

Use Case 2: Data Aggregations

This use case applies where a portal is required to pull data from multiple applications. The goal is to aggregate data from various sources for reporting or contextual display, helping users get a comprehensive view in one place. This improves the user experience by consolidating information (as customer, asset, or meter data) into a single, actionable interface.

Example: Building a customer 360 view or dashboard of different customer related information

Use Case 3: Data Processing and Orchestration

Additional data is needed in another application for data processing and might sometimes involve a middleware application for data orchestration and provide additional validation.

Examples: For a credit card payment to be processed, the credit card must be verified by a payment processing application or when a customer support representative is booking an appointment.

To optimize performance, real-time integration should be used only when batch processing is not feasible.

Example: In a Start Service process, creating a field activity and sending it to another system does not require immediate processing and can be done in non-real-time using a batch job.

Favoring batch over real-time where possible improves scalability, reliability, and system performance.

Use Case 4: Bulk or Batch Data Transfer

Data that does not need to be transferred in real-time or transfer of huge volumes of data can be done by a batch process. Such scenarios optimize and provide better utilization of the application's resources.

Examples: One-time migration of customer information from an old CIS application or periodic transfer of General Ledger (GL) transactions from a CIS application to an Enterprise Resource Planning (ERP) application.

Best Practices

Data Synchronization

Data is maintained and mastered in a designated system of record. Data synchronization to another application is sometimes needed to validate or complete a business transaction.

To perform this synchronization, Business Objects (BOs) are used. A sync request BO defines the behavior and lifecycle of the synchronization. Data is typically synchronized using a batch process. This process should be asynchronous to prevent incorrect handling of a two-phase

commit flow. In this situation, if an error occurs after the message to update the external system is committed but before changes are committed in the current system, then the data sent to the other system is out of sync. Retrying the update may create duplicates in the external system, and if the update is never processed again, then an incorrect record exists in the external system. The correct update flow should be to update the external system only after the changes in the current system are committed, which is by deferring the external update to the corresponding monitor batch process.

This is achieved by using the monitor batch process, which is considered best practice. As such, you should not remove the monitor process on any state of the sync request business objects.

If near real-time processing of sync requests is needed, schedule the sync request monitor jobs frequently.

Configuring Message Senders

Message Senders are used to define how outbound messages are sent out of the application (UPL, security, etc.)

Aside from configuring the HTTP User/Password, HTTP URL, HTTP Method and HTTP Header, make sure the following context values are also populated:

- Making sure customers always set a 'HTTP Timeout' Message Sender Content Value for any outbound scenarios they have.
  - Customers have o/b integrations dependent on an external system but they don't include a timeout, so if the external system is down or responding slowly, our transaction pays the price. HTTP Timeout should be considered to avoid a 2 minute wait if the external system is not responding.
- Populate HTTP Transport Method to
  - SendReceive - When invoking a Synchronous Flow, meaning an immediate response is expected,
  - Send - When invoking an Asynchronous Flow, meaning no response is expected.

Refer to Web Services Best Practices for Oracle Utilities Application Framework (KB704711) on My Oracle Support for more information.

Creating Custom Inbound Web Services (IWS)

When creating a custom Inbound Web Service, ensure it adheres to performance and design standards-such as avoiding full table scans and processing only a small, manageable set of records per call.

Inbound Web Services should be built for efficient, targeted operations rather than bulk processing.

Examples of typical IWS use cases include:

- Passing a service request logged by a customer through a portal into the system.
- Creating a refund transaction in the system based on an external trigger or approval.

Design with scalability in mind to ensure responsiveness and minimize system load.

REST APIs typically operate on a single resource. They may perform basic maintenance actions on the resource, search for data related to the resource, or process a manageable list of resources. When designing a new REST API, a good way of thinking about the intent of a REST operation is to ask these questions - What is being 'operated on' or 'searched for' or

'created' or 'deleted'? Once the resource is determined, you may think through the various operations that can be done on the resource.

There are two main types of "search" operations:

- Search across the entire population of a resource using filters. This is usually a 'starting point' search for top-level objects such as Person, Billing Account or Premise, etc also referred to as 'unbounded' search where the number of results very much depends on the specificity of the search criteria. Such search must cap the number of results to some reasonable number, for example a few hundred rows, and must not offer 'total rows' or 'ordering' capability as this would involve processing the entire list of matches.
- Search across a subset of objects which are related to one particular chosen 'context' object, for example Premises for an Address, Bills for an Account, Cases for a Premise, etc. This is referred to as a 'bounded search' which can be 'paged through'. While the average number of results may be small, it is possible for it to be slightly larger. In these cases, the query should offer pagination with standard 'offset' and 'limit' query parameters. If the caller does not specify a limit, the service must impose a default limit value. Base inbound web services are provided that use this technique.

These search queries must be carefully crafted to perform well, backed by proper indexes and follow an efficient execution plan.

Usage of Inbound Web Services should be limited to the following to fetching / adding/ updating one transaction record for one call:

- Not recommended: Extraction of mass data via inbound web services
- Data migration should not be performed via Inbound Web Services.

Please note:

- Performance tuning of any custom IWS will be the customer's responsibility.
- All the performance testing of any custom integration including file uploads needs to be tested before go-live.

Submit Batch Job through REST API

Customers can use an external batch scheduler and submit the batch job in cloud services via REST API. The Batch Job Submission (F1-SubmitJob) inbound web service is a REST API available with multiple operations to support integration. The REST service creates an entry in the Batch Job Submission table, which is polled by the batch daemon to pick up and execute the job.

Batch job parameters are defaulted from the batch control, when a job is submitted using this script, so only parameters not defined for the batch control need to be provided as input to this script.

Please refer to submitbatchREST.sh for Oracle Utilities SaaS (PNEWS2003) on My Oracle Support for more details.

6

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 only.

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

Oracle Database Actions is also available and included in the service for querying the database (see Oracle Database Actions 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.

Analytics Publisher, as part of your cloud service, is not intended to be used directly as a bill/letter print system, however, it can be used to create reports based on extracted data. Billing/letter data can be extracted from Customer Cloud Service, Customer Care and Billing Cloud Service, and Billing Cloud Service via batch processing with target output files to be stored in Cloud Object Storage. The output file can then be uploaded into Analytics Publisher via an API for bill/letter data reporting.

Analytics Publisher can be configured to write output files to Cloud Object Storage (via Analytics Delivery Channel). This requires configuration of a value for the File Storage Configuration (F1-FileStorage) extendable lookup with the following settings:

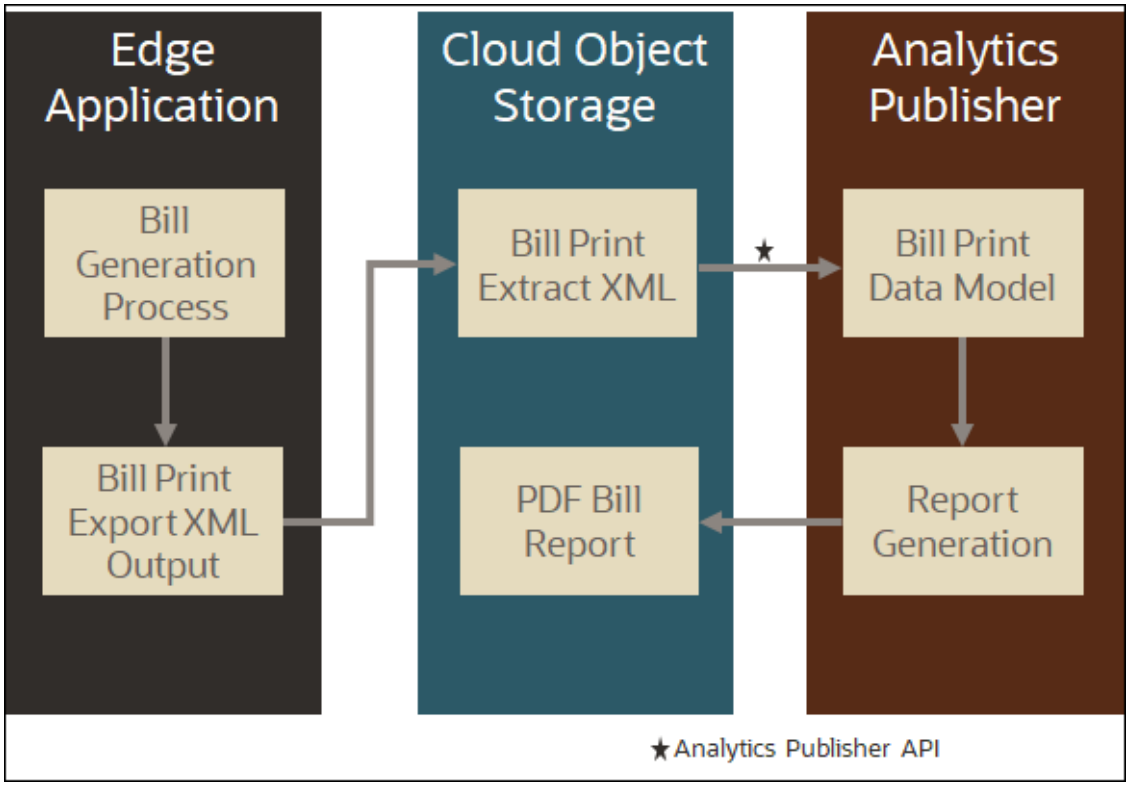
- **File Adapter**: Oracle Cloud Object Storage
- **User**: The user Oracle Cloud ID (ODIC) for the object storage location
- **Tenancy**: The tenancy Oracle Cloud ID (ODIC) for the object storage location
- **Compartment**: The compartment Oracle Cloud ID (ODIC) for the object storage location

- **Namespace**: The namespace for the object storage location
- **Key Ring**: The API signature key used to connect your cloud service to Object Storage.
- **Region**: The region of the object storage tenancy for the connection (Values for this field are defined in the F1_REGION_FLG lookup.
- **Bucket Name Prefix**: Prefix used by Object Storage Bucket configuration
- **Reporting Configuration**: Enabled (checked)

> **Note**
>
> This applies to prior releases provided they are up to date with the latest maintenance pack and hot fix releases.

The following diagram outlines this process:

Figure 6-1 Analytics Publisher - Bill Print



> **Note**
>
> The RTTYPOST batch control is not supported for bill generation using Oracle Utilities cloud services. Instead, customers should use batch-based extracts (such as the Customer Cloud Service bill extract functionality POSTROUT) in conjunction with 3rd party tools/services such as Documaker to ensure smooth, scalable, and successful implementation/operation of their services.

Refer to the following Knowledge Base articles on My Oracle Support for additional information about using Analytics Publisher for generating bills.

- Enabling Online Display Bill And Online Display Letter - Oracle Analytics (KB225626)
- Configuring Object Storage Delivery Channels for Analytics Publisher (KB670136)

Using Analytics Publisher for Bill/Letter Creation

Analytics Publisher is part of Oracle Utilities SaaS cloud service offerings with limited resources and is provided to support operational reports ONLY. Operational reports are reports that extract low to medium data volumes from the Production database to avoid performance-related issues.

> **Note**
>
> The role of Analytics Publisher in terms of high-volume data reports like bill/letter is limited to **formatting** ONLY.

High-volume data reports such as Bill and or Letter Print include four steps:

1. Bill/Letter Generation
2. Bill/Letter Extraction
3. Bill/Letter Formatting
4. Bill/Letter Printing

Bill/Letter Generation

Mass bill/letter generation is always the responsibility of the cloud service application (for example CCS, CCS for Retail, CCBCS and BCS). No reporting tool including Analytics Publisher can be used as a replacement.

Bill/Letter Extraction

Once the bills/letters are generated, the cloud service application provides the bill/letter extract batch background processes that reads the bill/letter tables and create an output file (XML) to cloud object storage. NO reporting tool including Analytics Publisher should be used to extract the bill/letter data from the production database.

Bill/Letter Formatting

The next step is to READ the bill/letter xml files from cloud object storage and perform the bill/letter formatting. Analytics Publisher CANNOT read cloud object storage; the bill/letter xml file should be PUSHED to Analytics Publisher via API. The Analytics Publisher API writes the output file into Analytics Publisher database. After that, Analytics Publisher can be used to format the bill/letter file as per the layout defined in the Bill/Letter Print Report template. The output of the report can be pushed to cloud object storage.

Bill/Letter Printing

The next step is to READ the bill/letter xml files from cloud object storage and perform the bill/letter formatting. Analytics Publisher CANNOT read cloud object storage; the bill/letter xml file should be PUSHED to Analytics Publisher via API. The Analytics Publisher API writes the output file into Analytics Publisher database. After that, Analytics Publisher can be used to

format the bill/letter file as per the layout defined in the Bill/Letter Print Report template. The output of the report can be pushed to cloud object storage.

Examples

Below are some scenarios of what Analytics Publisher can and cannot be used for.

Scenario 1: Analytics Publisher to be used for monthly AR reporting.

Yes, AR reporting is considered as medium data volume report and Analytics Publisher can be used. Furthermore, reports like online bill / letter display (covering single account) are also considered as operational reports.

Scenario 2: Analytics Publisher to be used as bill/letter generation tool.

No. Bill and or Letter generation is always the responsibility of the cloud service and performed via batch background processes (such as BILLING batch process).

Scenario 3: Analytics Publisher to be used as mass data extraction tool.

No. Extraction of mass data cannot be performed via Analytics Publisher. Base Product provides batch background processes (such as POSTROUT) and or Generalized / Specialized data extracts to extract the data out from production database. We also now support GoldenGate replication to synch the Production data with the customer reporting database.

Scenario 4: Analytics Publisher to be used for Bill Printing.

No. Bill Printing is always considered as Bill Print vendor responsibility and out of the scope from SaaS services.

Scenario 5: Analytics Publisher to be used as a Bill formatting tool:

Yes, Analytics Publisher can be used to read the bill/letter print output file and perform the formatting. The report processing time varies based on the complexity of the bill print layout report. Customers / partners may need to adjust the bill generation and extraction process in the edge application to address performance-related issues and to speed up the bill / letter processing time in the Analytics Publisher.

Scenario 6: External and third-party applications like Doc/1, Documaker etc. can be used for Bill formatting.

Yes, Bill print extract output file (available on cloud object storage) can be downloaded and any external/third-party application like Doc/1, Documaker can be used for bill formatting and printing.

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 Oracle Database Actions (see Oracle Database Actions for more information), 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.

Refer to **Database Storage Details** in the *Oracle Utilities Cloud Services Live Operations Guide* for details about viewing database storage details.

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 Oracle Database Actions 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.

> **Note**
>
> Moving files between object storage folders is not supported via Groovy scripting. Refer to Could Not Move Files Around Using Groovy in Enterprise Applications (KB552403) on My Oracle Support for more information.

Uploading and Downloading Files 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.

7

Information Lifecycle Management

This chapter provides information about set up and configuration of Information Lifecycle Management (ILM) with cloud service implementations. This includes:

Information Lifecycle Management Overview

Information Lifecycle Management (ILM) is a set of techniques and technologies available from Oracle that assist in managing the lifecycle of data to support business needs and minimize storage costs. The key to Information Lifecycle Management is working with the business to ensure that data that the business regards as active is available on appropriate hardware whilst data that is less active or dormant, in terms of business activity, is managed in an effective way in terms of storage costs.

Data in the product is added and updated by the business on an ongoing basis. Over time, this data grows and the business activity on individual data will vary with its timeliness, business activity and data retention policies. As data becomes less active in terms of the business, it needs to be stored in a cost-effective way to ensure cost minimization whilst allowing the business to access the data as needed for analysis or auditing purposes.

As the Oracle Database stores and manages data on an ongoing basis, therefore it is logical that Information Lifecycle Management uses the built-in database processes and procedures to offer effective storage management solutions to manage that data.

Initial Configuration During Provisioning

Information Lifecycle Management (ILM) is delivered with some tables configured by default. As part of the Sales process, a sizing spreadsheet is filled out with customer to identify the Initial Partitioning for ILM, which is to plan for historical data (as part of Data Conversion activities) considered for loading into the cloud service application.

Based on the values provided, during the provision process, historical partitions are created to support the implementation, however once environments are provided to the customer, it is the customer's responsibility to make sure that Initial ILM Configuration is reviewed to ensure support for the project requirements, and that ILM batch controls are configured as part of monthly scheduler batch stream.

During provisioning, all historical partitions are created for all ILM enabled tables based on Hybrid Columnar Compression (HCC) type except for last month, current, future month and

PMAX are created as Advance Compression type. For more information on Table compression types, refer to Oracle Database documentation.

> **Note**
>
> Despite the fact that ILM is available in all environments, Prod is the environment where ILM is applicable. Testing of ILM partitioning, and compression can ONLY be performed either in Prod or Prod equivalent sized environments.

Configuration During Implementation

As part of initial provisioning, default partitions are created for the following:

- Customer Care and Billing & Oracle Utilities Application Framework Objects: 48 months
- Meter Data Management Objects: 13 months, with exception of Initial Measurement Data, which is 3 months

> **Note**
>
> Work and Asset Management supports ILM for F1 objects only (W1 prefix tables are not partitioned).

Adjustments to the default partitioning can be made, ideally during implementation (before go-live). Changes after go-live can cause lengthy update periods (downtime), as data may have to be shifted between partitions.

> **Note**
>
> Based on these recommendations, PMax partitions should be empty. If this is the case the partition split will occur quickly. If there is data in the PMax partition because new partitions weren't created during implementation, when data is moved to one new partition, the split will occur quickly, but it will take time to build the index for the new partition, depending on how much data is moved. If a subset of records are moved out of the PMax partition, and some remain, the split could take hours, depending on how much data is present.

New historical partitions are created (as needed) by the "Add Partition" batch process (K1-ILMAD) based on retention period settings. Irrespective of creating partitions for current, future, or previous months, the K1-ILMAD batch process always creates the new partition in Advance Compression. Historical partitions created by K1-ILMAD can be changed to HCC by running the compression batch job.

Loading and/or processing of the data from HCC is slower than Advance compression (only applicable to non-Measurement tables) between 2% to 5%, however after go-live, compression jobs will run effectively.

As part of the implementation, customers are required to configure retention periods in two ILM-related Master Configurations and scheduler batch streams to enable ILM batches run monthly.

ILM-related Master Configurations include the following:

- ILM Configuration
- ILM Configuration - MDM Sub-retention

Customers are not required to run the Add Partition (K1-ILMAD) batch process every month for future month partitions. Oracle automatically adds partitions for entities, every month, on desired Production and Test domains and schemas. This batch process is not run on Development environments. Customers are responsible for running this process on Development environments if needed.

> **Note**
>
> ILM Compression jobs will run internally (like the ILM Add Partition job) by Oracle starting with the 26.10 release. Customers should start running both measurement and non-measurement compression jobs in any test-sized environment, followed by Production environments. Implementations can estimate the impact of compression on production environments by running compression processes in test environments. All ILM applicable table partitions MUST be compressed before upgrading to 26.10 in all Test and Production environments. Failure to do so may have performance impact to the online application following the upgrade to the 26.10 release.

ILM Configuration

The ILM Configuration master configuration contains a **Default Retention Period** parameter as a global default value that will be used if not overridden at the maintenance object level. This configuration also includes ILM Eligibility Algorithms along with associated Crawler Batch controls for each maintenance object.

A **Retention Days** parameter can be set on individual ILM managed maintenance Objects, which override the **Default Retention Period**.

Oracle recommends setting the ILM Master Configuration **Default Retention Period** to 1461 days (or a value override by implementations), then setting the **Retention Period** for Meter Data Management maintenance objects to 400 days (just over 13 months).

For more details on Meter Data Management-specific guidelines, please refer to Partitioning and Data Removal in the *Oracle Utilities Meter Solution Administrative User* guide

ILM Configuration - MDM Sub-Retention

The ILM Configuration - MDM Sub-retention master configuration is provided for specific configuration related to three high-volume Meter Data Management objects: Initial Measurement Data, Activity, and Device Event. This master configuration support defining sub-retention periods in days for each object and further by 'type' to allow for early purge eligibility of non-critical data (by creating sub-partitions).

The ILM Configuration - MDM Sub Retention master configuration will actually set the ILM Retention Period in Days Maintenance Object Option for the three maintenance objects supported: Initial Measurement Data, Activity, and Device Event.

This optional feature allows you to differentiate retention of initial measurements by Interval vs. Scalar as well as by unit of measure (UOM) code, Device Events by Event Category, and Activities by Activity Type. For instance, you could choose to specify a sub-retention value of 90 days for Device Event Types that are not critical, and drop those subsets of records early - while the critical events could be kept for the full retention period of 13 months.

Initially installed partition defaults for sub-partitions are as follows:

- Initial Measurement Data: 30, 60, 90 days
- Activity / Device Event: 30, 60, 90, 400 days.

This is the default setup, but these sub-partitions will be dropped if not used.

Information Lifecycle Management Recommendations

Customer can set the retention period at the maintenance object level using the ILM Retention Period in Days Maintenance Object Option.

Oracle recommends using a value that aligns with the number of partitions. The formula is typically 30.1 * (# of partitions + next month + max).

- Customer Care and Billing and Oracle Utilities Application Framework objects: 48 months (1461 days)
- Meter Data Management objects: 13 months (400 days)
- Initial Measurement Data: 3 months (90 days).

> **Note**
>
> Longer retention periods may have an impact on total database storage needs.

Information Lifecycle Management Go-Live Checklist

The following items are things to consider as your implementation prepares for go-live:

- Refer to **Information Lifecycle Management** in the *Oracle Utilities Cloud Services Live Operations Guide* for details regarding batch processing used with information lifecycle management.
- The data conversion process has set the ILM_DT and ILM_ARCHIVE_SW as 'Y' for historical data. Please make sure that the PMAX partition is empty after the data conversion is completed.
- Make sure that all historical partitions excluding Measurement objects (D1- MSRMT and D1-MSRMT-Log, we do sorting based on MC_ID during compression for better application performance) should be in HCC.
- Cut-over activities may include measurement compression batch processing which will compress Historical Measurement (D1-MSRMT and D1-MSRMT- Log) partitions during cut-over period.
- Drop partition batch processing can be used to drop empty partitions as/if needed to save on database storage.
- Scheduler batch streams are configured to run ILM crawler and compression batch jobs on monthly basis.

8

Network Integration Guidelines for Integrating Oracle Utilities Cloud Services with External Applications

This chapter includes guides for integrating Oracle Utilities Cloud Services with external applications. This chapter includes the following:

Network Integration Guidelines Introduction

In order to integrate Oracle Utilities Cloud Services with an application hosted externally-either in your data center or in a third party data center-you need to understand the networking, authentication and protocol requirements for outbound and inbound communication from and to Oracle Utilities Cloud Services.

This chapter will help you understand the technical and networking requirements for sending and receiving requests from and to Oracle Utilities Cloud Services. It details some possible networking scenarios for integration and provides possible options for a solution.

> **Note**
>
> You must have thorough understanding of general networking concepts along with appropriate Oracle Cloud Infrastructure (OCI) certification or equivalent experience to make efficient use of this chapter, plan, and set up the required networking for integration with Oracle Utilities Cloud Services.

Integrate Oracle Utilities Cloud Services Using Web Services

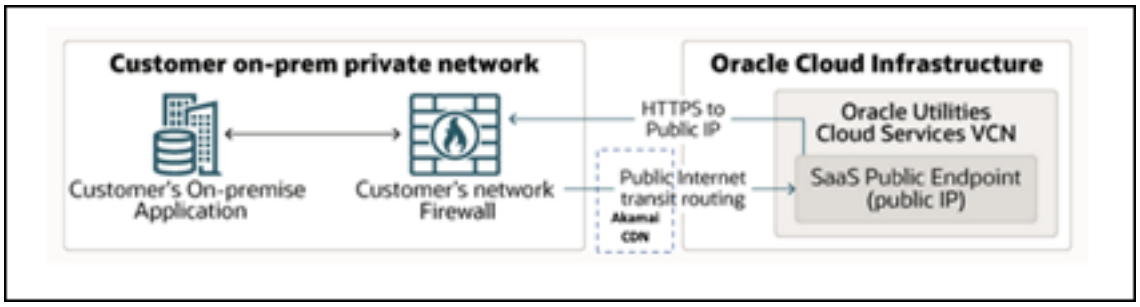
This channel of integration uses web service APIs on both the Oracle Utilities Cloud services and the external applications. Industry standard best practices, protocol and authentication methods have to be used for integration using web services. In order to establish integration between an external application and Oracle Utilities Cloud Services, appropriate networking needs to be setup, so the applications can access each other's APIs securely. The subsequent sections detail various networking options that can be used for setting up the web service-based communication.

You can select any of three architectural options for integrating Oracle Utilities Cloud services with an application hosted in your data center or in a third-party data center. Each of the architecture options provide different levels of security and bandwidth, along with other

variations. The below architecture options apply to forward flow - communication from external applications to Oracle Utilities Cloud Services, and reverse flow - communication from Oracle Utilities Cloud Services to external systems.

Architecture 1: Integrating Through Public (Internet) Web Service APIs

Figure 8-1 Integrating Through Public (Internet) Web Service APIs



In its simplest form integration between application's hosted in your private/corporate network and Oracle Utilities Cloud services can be achieved over the public internet.

Forward Flow

The REST APIs on the Oracle Utilities Cloud services are exposed securely to the public internet through the Akamai content delivery network, so if an on-premise application needs to access the REST APIs, it can do so, as long as the application has access to the public internet, unless VPN Connect or FastConnect are being used.

Akamai is a content delivery network provider with geographically distributed servers that speed up the delivery of web content by bringing it closer to where users are. It currently handles 15-30% of the world's web traffic. Access to Oracle Utilities Cloud Services through Akamai networking adds a layer of security that thwarts malicious attacks such as Distributed Denial-of-Service (DDoS), at the edge, before they reach infrastructure. All communication coming into Oracle Utilities Cloud Services is automatically directed through Akamai network. The DNS of Oracle Utilities Cloud Services automatically resolve to the respective Akamai network end points through which the communication is routed. No additional setup if required in the external application to use Akamai and there is no additional cost to customers of Oracle Utilities Cloud Services, for using Akamai content delivery network.

In order to send communication to Oracle Utilities Cloud services application's web service APIs from an external system hosted outside Oracle Utilities Cloud services, the calling application/service needs to meet these criteria:

- **Access to the web service APIs of Oracle Utilities Cloud services applications**
  The web service APIs of Oracle Utilities Cloud Services can be accessed via public internet. So, an external application trying to connect to Oracle Utilities Cloud Services should have access to the webservice APIs through public internet, unless VPN Connect of FastConnect are being used.
- **OCI IAM Identity Domain token-based authentication or basic Authentication over HTTPS**
  The web service APIs of Oracle Utilities Cloud Services support OCI IAM Identity Domain token based (OAuth) and basic authentication over https to authenticate an incoming request from external applications. Either of the authentication methods can be used for

sending a request to Oracle Utilities Cloud Service's web service APIs, however it is highly recommended that you use OAuth for a more secure and flexible approach.

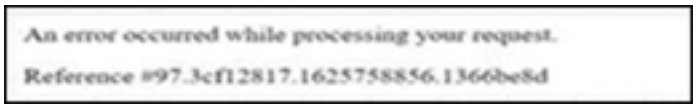
- **Inbound Allowlists for Forward Flow Communications**
  The allow-list option allows you to specify the sources from which communication is allowed into Oracle Utilities Cloud Services. By default communication is allowed from all sources with valid authentication information. You can change this behavior by adding sources to the allowlist. The addition/removal of sources can be done by raising a support ticket to the Oracle Cloud Operations team to update the allow-list (Please refer to *Oracle Utilities Cloud Services Cloud Operations Guide* for more information). Optionally, you can even specify the resources that one or more sources has access to.
  - A source of forward flow request can be identified by:
    - IP address ranges defined as CIDR blocks
    - Both IP and VCN OCID; optionally, you can create and name groups of CIDR blocks and/or VCN OCID through the use of a support ticket.
  - A resource to be accessed can be identified by:
    - Paths depending on the application. For example: /web for online web application; /sql for ORDS, /rest for rest services etc
    - Groups of paths that start with a particular subpath. For example: /rest/ busSvc/K1 for all K1-owned rest service
    - Specific paths. For example: /rest/busSvc/F1-HealthCheck can be configured to be accessible by set of sources Optionally, you can create and name groups of paths through the use of a support ticket.

  The Cloud Operations team defines the inbound allowlist based on the information provide as an Allow Rule. An allow rule would be a mapping between sources and paths such as, which source/sources has access to which path/paths. Each rule is made up of a path group and one or more sources that can access it. Optionally, the customers can request to create and name groups of paths via support ticket.

> **Note**
>
> For creating, updating or removing an allow-list rule, a support ticket needs to be raised.

Important Note:Errors that the Akamai content delivery network encounters, including traffic that was blocked, will be reported as follows:

Figure 8-2 Akamai Error



When Akamai content delivery network reports an error, you'll get a Reference ID in the following format:

```
18.4c96df17.1644609140.2a155db
```

If there is a need, a Service Request (SR) can be raised via My Oracle Support by providing the Reference ID as part of the SR Description.

The first digits before the period give the reason for the error. The most common of those are outlined in the table below:

Error Code	Description
1	Request timed out
18	Request was blocked by WAF
25	Max Redirect Chase. This happens if a request gets in a loop of 3xx redirects
30	Forward SSL Handshake. This happens when Akamai tries to go forward to the origin, but cannot complete the SSL handshake for some reason (usually the origin cert has expired or SNI isn't configured properly)
102	No good forward IP. This happens when Akamai is trying to go forward to the origin, but the origin isn't resolving in DNS

Reverse Flow

Similarly, Oracle Utilities Cloud services can access web service end points that are exposed to the public internet (public IP) such as, if the on-premise application's web service end points are exposed to the public internet, then these can be consumed by Oracle Utilities Cloud services. A firewall in your corporate network may be configured to expose any application's end points to the public internet. Although this forms the simplest possible communication channel, transiting over the public internet requires close consideration of the security, availability, and reliability that the public internet can provide.

For Oracle Utilities Cloud services to make either an outbound call to an external system, the following criteria should be met by the web service APIs of the external system:

- **Accessible API**
  For a request to be made from Oracle Utilities Cloud services to the web service APIs of the external system, those APIs should be accessible via public IP addresses. If your external application APIs are not yet public, the networking administrators on your IT team/ partner/implementer should be able to set up and configure appropriate components so the private APIs are exposed through public IPs. Following are a couple of examples that can help expose the private web service APIs of an external system through public APIs:

  1. Expose the APIs of the external application to public internet, by setting up rules in the firewall.
  2. Create a private endpoint within your OCI VCN to point towards your external system where your application resides.

- **Valid CA-signed Certificate**
  The web service APIs of the external application should have valid CA issued TLS certificate. There is a TLS handshake to check that certificate matches what is on Oracle Utilities Cloud service's outbound allowlist, and if it meets the criteria then Oracle Utilities Cloud services can post outbound requests to the external application.

  If the external application's web service APIs use self-signed certificates, then you need to configure the APIs to support CA-signed certificate. Following are some examples of how this can be done:

  – Use valid CA-signed certificate

- **Basic authentication/OAuth Client Credentials over HTTPS protocol**
  Oracle Utilities Cloud Service can send requests to external applications that support either basic authentication or OAuth client based authentication over https. So, the external application's APIs need to support either of the authentication options for web services. It is highly recommended that you use OAuth for a more secure and flexible approach.
- **SSL Port Number 443**
  If the API of the external application does not have 443 as the port number, then you might have to consider changing the port to 443 or in cases where the port number cannot be changed, you might consider adding additional infrastructure/software as a mediator; for example, adding a reverse proxy that listens on port 443. Oracle Utilities Cloud Service is designed to make calls ONLY to the port 443 as part of all web service reverse flow communication.
- **Outbound Allowlists for Reverse Flow Communications**
  The allow-list option allows you to specify the targets to which communication may be allowed from Oracle Utilities Cloud Services. By default, communication to external interfaces is disabled. You can change this behavior by adding targets to the allowlist. The addition/removal of targets can be done by raising a support ticket the Cloud Operations team to update the outbound allow-list (Please refer to the *Oracle Utilities Cloud Services Cloud Operations Guide* for more information).

  The following information needs to be made available in the support ticket:

  – The named DNS OR URL along with the justification for its allowance.

Architecture 2. Integrating Through VPN Connect

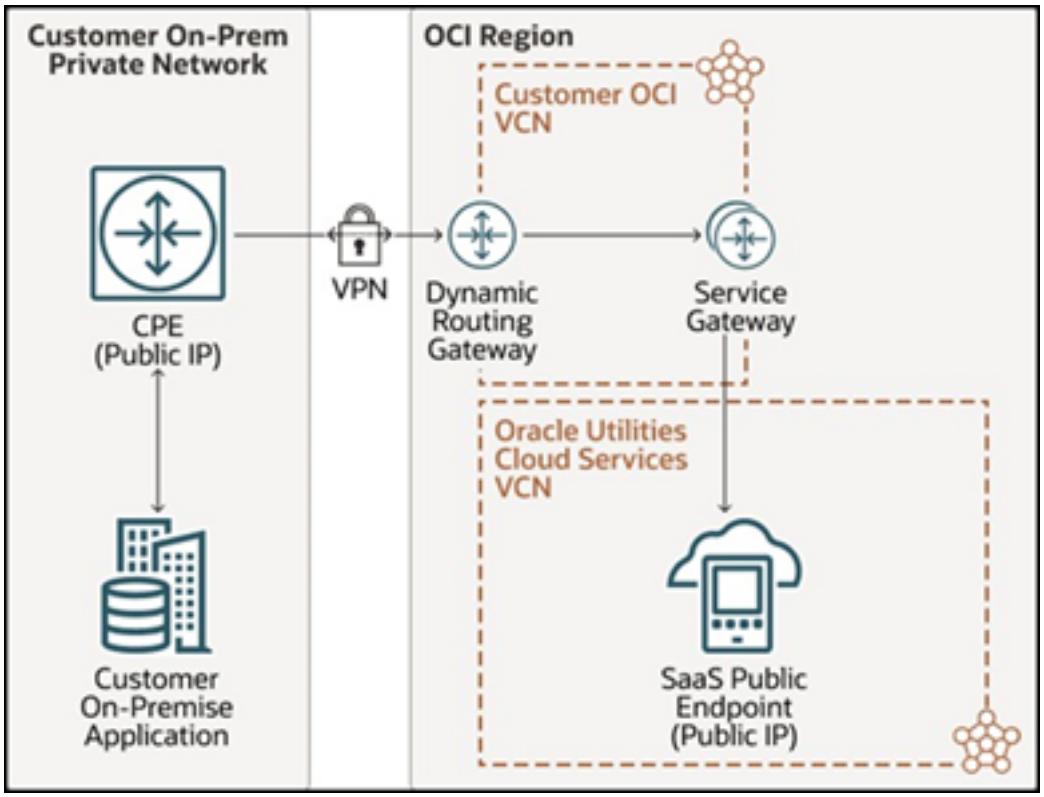
In this architecture, the external application makes web service API calls through the public internet, protected by an extended VPN, which creates a secured IPSec tunnel between your corporate private network and your VCN on Oracle Cloud Infrastructure (OCI). The same setup may be used for securely accessing Oracle Utilities Cloud services and to limit the Oracle Utilities Cloud services access via customer corporate network only.

> **Note**
>
> The end-to-end VPN and other requisite setups can be done by networking administrators in your IT team/partner/implementer.

Refer to When to Use VPN Connect for more information.

Forward Flow

Figure 8-3 Integrating Through VPN Connect - Forward Flow



For forward flows, VPN Connect requires setting up of CPE (Customer Premise Equipment), which interfaces with VPN DRG (Dynamic Routing Gateway) creating a IPSEC Encryption Tunnel over the internet, securing all information flowing through the tunnel. Once the VPN Connect is setup between the external application's data center and your VCN(Virtual Cloud Network) on OCI, communication is routed through the CPE device at the data center of the external application and through the Dynamic Routing Gateway(DRG). Your OCI VCN is not the same as the VCN that hosts Oracle Utilities Cloud Services. So, appropriate setup is required to establish networking between your VCN on OCI and Oracle Utilities Cloud Services VCN, which can be done using a service gateway. Service Gateway is one of the available gateways in OCI VCN that allows for traffic to be routed between two VCNs within OCI.

In order to send communication to Oracle Utilities Cloud services application's web service APIs from an external system hosted outside Oracle Utilities Cloud services, the calling application/service needs to meet the following criteria:

- **Access to the web service APIs of Oracle Utilities Cloud services applications**
  The web service APIs of Oracle Utilities Cloud Services can be accessed via VPN Connect through the use of service gateway in your VCN or via private endpoint (PE). The DNS resolution of service gateway should be set so that the service domain names resolve to home region IPs of Oracle Utilities Cloud Services and NOT that of Akamai content delivery network. The home region IP is the IP of the main data region selected for the tenancy in which the cloud service is deployed. When a cloud account is created, a Home Region is assigned to it. This is the main data region that is linked to that account. Customers are required to raise a Service Request via My Oracle Support to get the Home

Region DNS addresses. Refer to the *Oracle Utilities Cloud Services Cloud Operations Guide* for more information about creating and submitting service requests.

- **OCI IAM Identity Domain token-based authentication or basic authentication over HTTPS**
  The web service APIs of Oracle Utilities Cloud Services support OCI IAM Identity Domain token (OAuth) based and basic authentication over https to authenticate an incoming request from external applications. Either of the authentication methods can be used for sending a request to Oracle Utilities Cloud Service's web service APIs. It is highly recommended that you use OAuth for a more secure and flexible approach.
- **Inbound Allowlists for Forward Flow Communications**
  The allow-list option allows you to specify the sources from which communication is allowed into Oracle Utilities Cloud Services. By default communication is allowed from all sources with valid authentication information. You can change this behavior by adding sources to the allowlist. The addition/removal of sources can be done by raising a support ticket to update the allow-list. Optionally, you can even specify the resources that one or more sources has access to.

  Create a support ticket for adding VCN OCID to the allowlist of Oracle Utilities Cloud Service so that communication coming into Oracle Utilities Cloud Services from only your VCN's service gateway is allowed.

  – A source of forward flow requests can be identified by:
    * IP address ranges defined as CIDR blocks (for access from, other than service gateway)
    * VCN OCID; which is used only for sources that access the resources via the OCI service gateway.
    * Both IP and VCN OCID; optionally, you can create and name groups of CIDR blocks and/or VCN OCID through the use of a support ticket.
  – A resource to be accessed can be identified by:
    * Paths depending on the application. For example: /web for online web application; /sql for ORDS, /rest for rest services etc
    * Groups of paths that start with a particular subpath. For example: /rest/ busSvc/K1 for all K1-owned rest service
    * Specific paths. For example: /rest/busSvc/F1-HealthCheck can be configured to be accessible by set of sources Optionally, you can create and name groups of paths through the use of a support ticket.

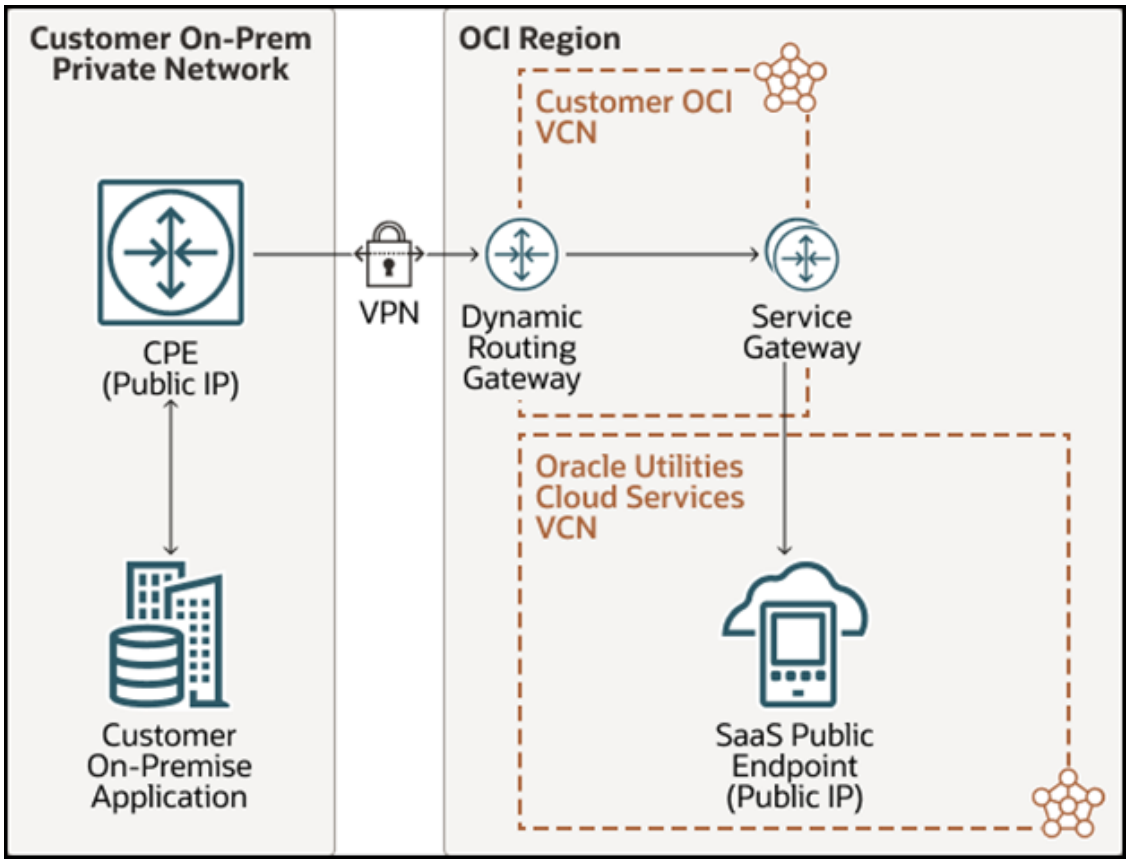
The Cloud Operations team defines the inbound allowlist based on the information provide as an Allow Rule. An allow rule would be a mapping between sources and paths such as, which source/sources has access to which path/paths. Each rule is made up of a path group and one or more sources that can access it. Optionally, the customers can request to create and name groups of paths via support ticket.

> **Note**
>
> A support request must be raised to create, update, or remove an allowlist rule, .

For accessing SaaS applications using the Service Gateway, please refer to **Accessing SaaS Applications Using the Service Gateway** below.

Reverse Flow

Figure 8-4 Integrating Through VPN Connect - Reverse Flow



The Oracle Utilities Cloud service makes web service calls to the external application by using Oracle Utilities Cloud service supported authentication methods such as basic authentication/ OAuth client credentials.

For Oracle Utilities Cloud services to make either an outbound call to an external system, the following criteria should be met by the web service APIs of the external system

- **Accessible API**
  For a request to be made from Oracle Utilities Cloud services to the web service APIs of the external system not accessible via public internet through VPN Connect, those APIs should be accessible via reverse connection endpoints (RCE) from your OCI VCN.
- **Valid CA-signed Certificate**
  The web service APIs of the external application should have valid CA issued TLS certificate. There is a TLS handshake to check that certificate matches what is on Oracle Utilities Cloud service's outbound allowlist, and if it meets the criteria then Oracle Utilities Cloud services can post outbound requests to the external application.
  - Use valid CA-signed certificate
- **Basic authentication/OAuth Client Credentials over HTTPS protocol**
  Oracle Utilities Cloud Service can send requests to external applications that support either basic authentication or OAuth client based authentication over https. So, the external

application's APIs need to support either of the authentication options for web services. It is highly recommended that you use OAuth for a more secure and flexible approach.

- **SSL Port Number 443**
  If the API of the external application does not have 443 as the port number, then you might have to consider changing the port to 443 or in cases where the port number cannot be changed, you might consider adding additional infrastructure/software as a mediator; for example, adding a reverse proxy that listens on port 443. Oracle Utilities Cloud Service is designed to make calls ONLY to the port 443 as part of all web service reverse flow communication.
- **Outbound Allowlists for Reverse Flow Communications**
  The allowlist option allows you to specify the targets accessible via public internet to which communication may be allowed from Oracle Utilities Cloud Services. By default, communication to external interfaces is disabled. You can change this behavior by adding targets to the allowlist. The addition/removal of targets can be done by raising a support ticket with the Cloud Operations team to update the outbound allowlist (Please refer to the *Oracle Utilities Cloud Services Cloud Operations Guide* for more information).

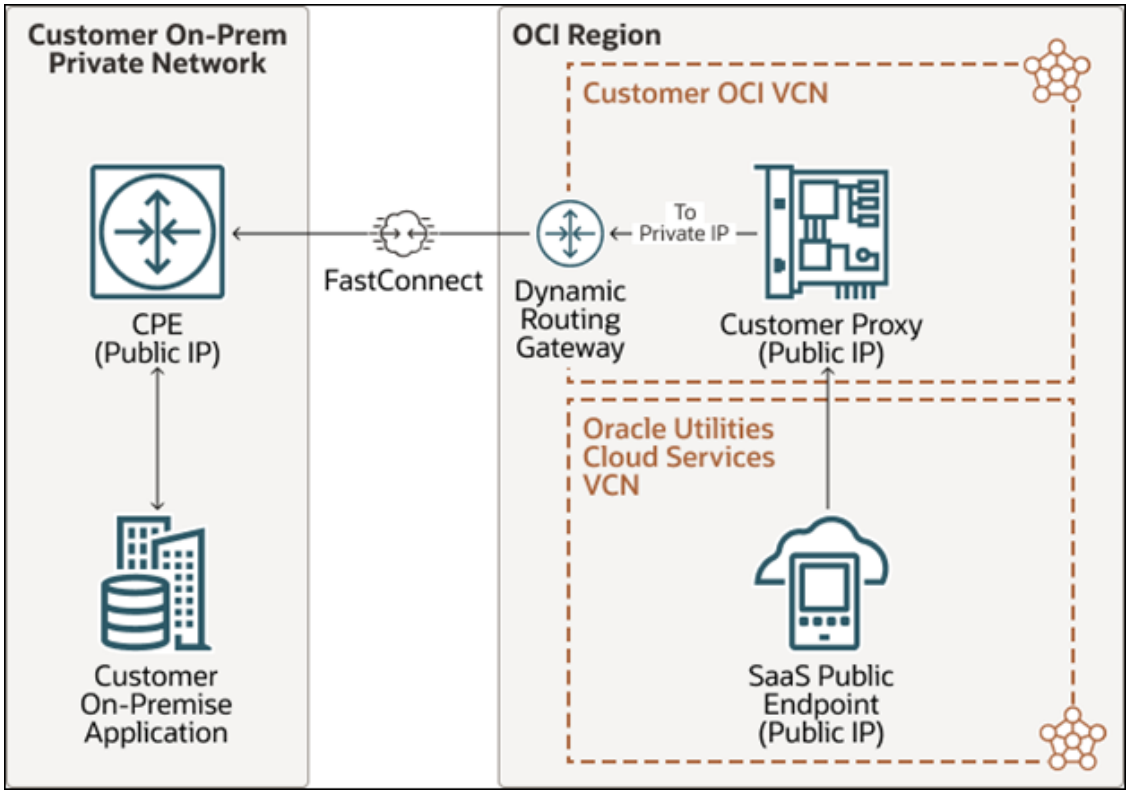
  The following information needs to be made available in the support ticket:

  – The named DNS OR URL along with the justification for its allowance.

Architecture 3. Integrating Through FastConnect for Private Web Service APIs

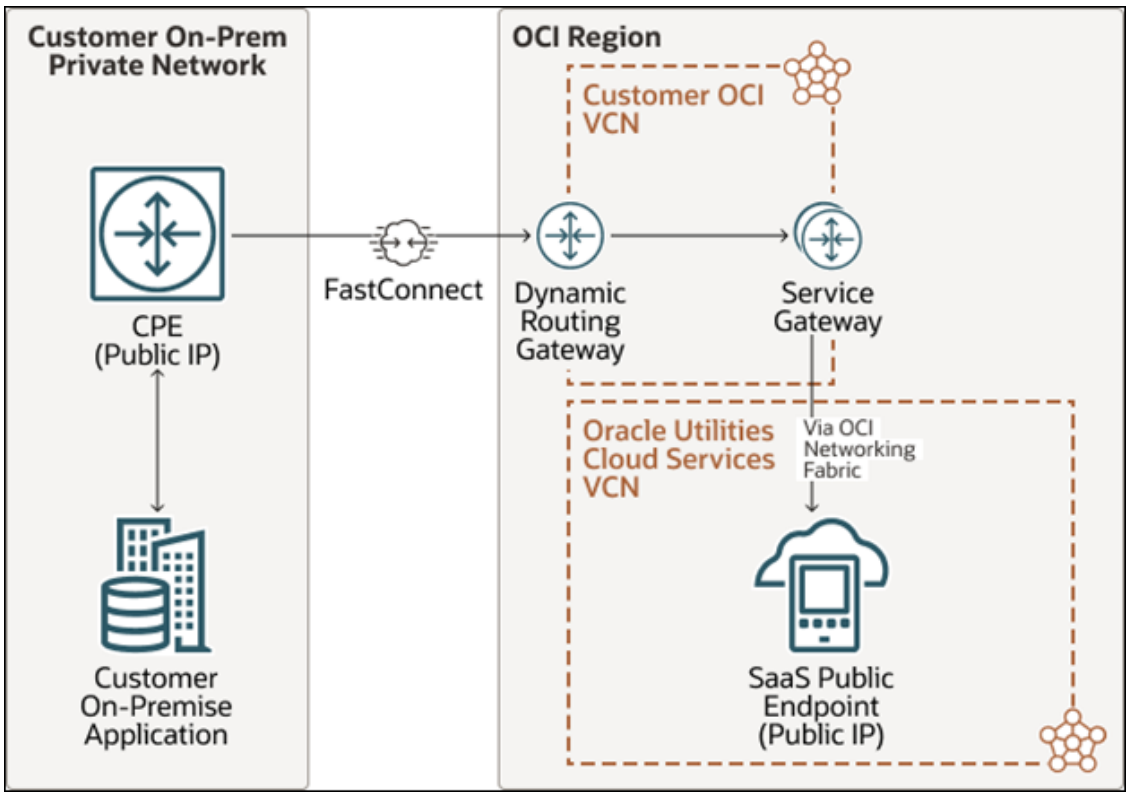
Forward Flow

Figure 8-5 Integrating Through FastConnect for Private Web Service APIs - Forward Flow



Reverse Flow

Figure 8-6 Integrating Through FastConnect for Private Web Service APIs - Reverse Flow



Another private routing option of FastConnect may also be used to connect your private/ corporate network with OCI network(OCI VCN). FastConnect provides an entry point in to OCI for a dedicated private line between your data center and the OCI to enable high bandwidth data transfer over a highly secured channel. FastConnect communication requires FastConnect DRG to be setup on your OCI VCN along with a dedicated line that can connect the CPE with the FastConnect DRG to be set up, which in turn interfaces with the service gateway within you VCN, depending on the direction of the communication. Within OCI networking, communication between Oracle Utilities Cloud Service's VCN (Virtual Cloud Network) and your VCN uses the service gateway depending on the direction of the API Call.

The requirements for connecting external application to Oracle Utilities Cloud Services through FastConnect remain the same as the ones mentioned under the architecture option using VPN Connect. To set up a FastConnect private line between an external data center and OCI, See Understanding Oracle Cloud Infrastructure FastConnect.

Network Scenarios

This section describes four different networking scenarios, based on the above three networking architectures, any of which you might consider when integrating Oracle Utilities Cloud Services with an application hosted externally. To assist you in choosing the appropriate network topology, here we provide a description and pro/con discussion of each scenario.

Use the following table and associated topics to help you decide which networking option best fits your needs.

Scenario	Description	Security	High Availability	Throughput	Cost
1	Connectivity over public internet **without** VPN or FastConnect	TLS only	Relies on connectivity over the internet	Limited	Low setup cost; Low setup cost; OCI data transfer charges may apply
2	Connectivity over public internet with VPN Connect and **without** FastConnect	IPSec, Encrypted	Limited	Typically <250Mbps	Low setup cost; Low setup cost; OCI data transfer charges may apply
3	Connectivity over FastConnect without VPN (VPN may reduce the throughput)	TLS over dedicated private line - Not Encrypted	Redundancy supported - Refer to High Redundancy Best Practices	Port speeds in 1 Gbps, 10 Gbps or 100 Gbps increments	Prominent setup cost; OCI data transfer charges do not apply
4	Connectivity over public internet with VPN (as a fallback) and FastConnect	Depending on the path used for communication (Fast Connect - Not Encrypted; VPN - Encrypted)	Redundancy by Design - Refer to Redundancy Best Practices	Depending on the path used for data transfer	Prominent setup cost; OCI data transfer fees may apply, depending on the path of communication

Although connecting to Oracle Utilities Cloud Service via the internet is the cheaper option to setup, due to its limited security and availability, when transferring secured information as part of product integrations, it might also be the riskier option. Also, the OCI data transfer charges should be taken into consideration when evaluating the networking options. To ensure utmost security and availability, the FastConnect option with a redundant setup of VPN over public internet may be preferred.

The following sections discuss these options in greater detail.

Scenario 1: Connect Over Public Internet Without VPN or FastConnect

You can consider connecting over the public Internet without a VPN or FastConnect when the integration with on-premises application doesn't need high bandwidth or high levels of security. This is illustrated in the diagram in Architecture 1: Integrating Through Public (Internet) Web Service APIs.

Note these considerations:

- Pre-requisites (to be performed by the customer)
  - On-premises application's APIs in customer's network should be publicly accessible through the internet.
  - Application's inside customer's network should have access to public internet.
- Working

  - Oracle Utilities Cloud Services REST APIs are exposed to the public internet, so on-premises applications can use these REST APIs for integrations.
  - Oracle Utilities Cloud Services can call on- premises public (internet facing) APIs for integration.
  - File transfers can be done by using Object Storage, which also has secured public (internet facing) REST APIs.
- Pros
  - Simple setup, lower cost.
- Cons
  - Limited security of data in transit by using TLS, through public internet.
  - No guaranteed availability of connection; network outages between the on-premises data center and Oracle's OCI can occur.
  - Unpredictable throughput; moving large amounts of data can take substantial time
  - OCI data transfer charges may apply.

Scenario 2: Connect Over Public Internet With VPN but Without FastConnect

This scenario covers integration over the public internet with a VPN Connect but not using FastConnect. This is applicable when the integration with on-premises applications doesn't need high bandwidth but requires higher levels of security, with private APIs. This is explained in Architecture-2 diagram and used where additional cost is a factor but network throughput isn't.

Note these considerations:

- Pre-requisites (to be performed by the customer)
  - Appropriate setup needs to be done between the on-premises data center and OCI for the VPN Connect.
  - Service Gateway and/or private endpoint (PE) needs to be setup within customer's OCI VCN to route requests from customer's on-premises data center to Oracle Utilities Cloud Services through the VPN connect.
  - Appropriate setup/configuration needs to be set up to route requests from Oracle Utilities Cloud Services to the private APIs on customer's on- premises data center for both public and reverse connection endpoints (RCE).
  - Redundancy can be planned and the VPN setup should be done accordingly (this is a redundancy best practice).
- Working
  - Oracle Utilities Cloud services REST APIs can be accessed via the VPN Connect route through the service gateway and/or private endpoint (PE), so customer's on-premises applications can use these REST APIs for integrations.
  - Oracle Utilities Cloud services can access both public and private APIs of customer's applications.
  - File transfers can be done by using Object Storage, which also has secured public (internet facing) REST APIs.
- Pros

  - Easy to set up; more secure than public internet option.
  - Redundancy is supported by way of multiple connections and tunnels.
- Cons
  - Service gateway setup and/or private endpoint (PE)/reverse connection endpoint (RCE)
  - Low throughput-typically <250Mbps; moving large amounts of data can take substantial time.
  - OCI data transfer charges may apply.

Scenario 3: Connect Over FastConnect Without VPN

Connect over FastConnect without a VPN when the integration with an on-premises application requires high bandwidth; for example, when you need to transfer large files. This is illustrated in the diagram in Architecture 3. Integrating Through FastConnect for Private Web Service APIs.

Note these considerations:

- Pre-requisites (to be performed by the customer)
  - A dedicated private line between a customer's on-premises data center and OCI.
  - Set up and configuration must be set up so that any private endpoint (PE) and/or reverse connection endpoint (RCE) are exposed to Oracle Utilities Cloud services.
  - Service Gateway and/or private endpoint (PE) needs to be set up within customer's OCI VCN to route requests from customer's on-premises data center to Oracle Utilities Cloud services through the VPN connect.
  - Redundancy can be planned and the FastConnect setup should be done accordingly (Redundancy is a best practice).
- Working
  - Oracle Utilities Cloud services REST APIs can be accessed via the FastConnect and Service Gateway route, so customer's applications can use these REST APIs for integrations.
  - Oracle Utilities Cloud services can access the private APIs of customer's on-premises applications via reverse connection endpoint (RCE) using FastConnect.
  - File transfers are done using Object Storage, which also has REST APIs.
- Pros
  - High bandwidth; secure line.
- Cons
  - Cost of setting up the FastConnect private line.

Scenario 4: Connect Over Public Internet with VPN and FASTConnect

Connect over the public internet with a VPN Connect and FASTConnect when the integration with an on-premises application requires not only high bandwidth, but also needs a fallback mechanism to ensure close to 100% availability. While the fallback mechanism in this case has a lower bandwidth, it ensures that connectivity persists. This is a combination of Architecture 2. Integrating Through VPN Connect and Architecture 3. Integrating Through FastConnect for Private Web Service APIs.

Note these considerations:

- Pre-requisites (to be performed by the customer)
  - A dedicated private line between a customer's on-premises data center and OCI.
  - An appropriate setup and configuration that allows exposure of any public and private endpoint (PE) to Oracle Utilities Cloud Services as public end points.
  - Service Gateway and/or private endpoint (PE) needs to be set up within customer's OCI VCN to route requests from customer's on-premises data center to Oracle Utilities Cloud services through the VPN connect.
  - Redundancy can be planned and the FastConnect setup should be done accordingly (Redundancy is a best practice).
  - Redundancy can be planned and the VPN setup should be done accordingly (Redundancy is a best practice).
- Working
  - Oracle Utilities Cloud services REST APIs can be accessed via the FastConnect or the VPN Connect route, so customer's applications can use these REST APIs for integrations.
  - Oracle Utilities Cloud services can access any public and/or private API of customer's on-premises implementation via FastConnect or the VPN Connect.
  - File transfers can be done using Object Storage's public (internet facing) REST APIs or by connecting to the Object Storage through FastConnect.
- Pros
  - High bandwidth, high availability, and secure.
- Cons
  - Cost of setting up the FastConnect private line.
  - Low throughput of VPN Connect in case FastConnect line becomes unavailable.

Integrate Oracle Utilities Cloud Services using File Transfers

Along with the web services-based integration, Oracle Utilities Cloud services applications also support integration by using file transfers; that is, you can also move data between the Oracle Utilities Cloud services application and the external application by transferring files.

Usually, the Oracle Utilities Cloud services applications process both inbound and outbound files by using batch programs via built-in file adapters to Oracle Object Storage Cloud Service. So, all inbound and outbound file transfers are supported only through Oracle Object Storage Cloud Service. An external application needing to integrate with Oracle Utilities Cloud services application using files should have the capability to push and pull files from the Oracle Object Storage Cloud Service.

Allowing Inbound-Outbound Communication by Using File Transfers

The criteria/methods for moving files to and from Oracle Utilities Cloud services application as part of an integration with external applications is more or less the same. Oracle Object Storage Cloud Service has to be used for pushing and pulling files to and from Oracle Utilities Cloud services application.

The external application needs to be able to integrate with the Oracle Object Storage Cloud Service. Oracle Object Storage Cloud Service has REST APIs available over public internet that can be used by the external application to pull or push files to Oracle Object Storage

Cloud Service. If the external application does not support REST API based file transfers, then an intermediary setup needs to be done to push and pull files from Object Storage. Following are some sample setups for file-based integration, in the absence of REST API capability in the external application:

- If the external application supports SFTP based file transfers, then you can build a custom adapter between the SFTP storage being used by the external application and Oracle Object Storage Cloud Service. The custom adapter can use the Oracle Object Storage Cloud Service's REST APIs to move the files between Oracle Object Storage Cloud Service and the SFTP storage server.
- You can use the Managed File Transfer Cloud Service, which is part of SOA cloud service. Managed File Transfer Cloud Service has built in adapters to Oracle Object Storage Cloud Service. The external application can interface with Managed File Transfer Cloud Service using SFTP.
- Use Oracle Integration Cloud Service for file-based integrations

Understanding Connection Technologies

When integrating between an Oracle Utilities Cloud services application and an application hosted externally, either in your data center or in a third party data center, you can select between three different connectivity technologies, depending on your topology and integration requirements,

Understanding Akamai Networking Usage for Oracle Utilities Cloud Services

Akamai is a content delivery network provider with geographically distributed servers that speed up the delivery of web content by bringing it closer to where users are. It currently handles 15-30% of the world's web traffic. Access to Oracle Utilities Cloud Services through Akamai networking adds a layer of security that thwarts malicious attacks such as Distributed Denial-of-Service (DDoS), at the edge, before they reach infrastructure. All communication coming into Oracle Utilities Cloud Services is automatically directed through Akamai network. The DNS of Oracle Utilities Cloud Services automatically resolve to the respective Akamai network end points through which the communication is routed.

Understanding Oracle VPN Connect

You can use Oracle VPN Connect to connect your corporate/private network to Oracle Cloud Infrastructure (OCI) through public internet. It provides secure communication through a virtual private tunnel via the public internet between the Oracle Utilities Cloud services on OCI and the external application with which it integrates.

Separate VPNs should be setup for each of your data center and each third party data center that you may need to connect to OCI, for integrating external applications. Customers need to setup and configure VPN Connect. VPN Connect is a free service, with no port hour charges. Data transfer cost is covered under networking cloud pricing.

When to Use VPN Connect

You should consider using VPN Connect when the following conditions exist:

- When an external application that needs to be integrated with Oracle Utilities Cloud services is within your corporate private network and you do not plan to expose the APIs to public internet (outbound from Oracle Utilities Cloud services applications) from within your data center or if you application cannot access the public internet (inbound to Oracle Utilities Cloud services) but needs to be integrated with Oracle Utilities Cloud services.

- When you want a secured communication channel over public internet for integrating Oracle Utilities Cloud services with the applications in your data center or third party data center.

Preparing to Use VPN Connect

Before you implement VPN Connect to integrate external applications with Oracle cloud services, do the following:

- Understand the requirements and tasks for setting it up.
- Configure the customer-provided equipment (CPE).
- Plan redundancies that will prevent costly system downtime.

Set Up IPSec VPN Connect:

Setting up an IPSec VPN Connect network between external applications and Oracle SaaS is a multistep process that can be complex unless you understand it thoroughly This topic provides a basic overview of the required steps. For more details, refer to **Setting Up VPN Connect**, referenced in Before Using VPN Connect.

1. Choose the routing type that works best for your implementation. IPSec VPN has two redundant IPSec tunnels and you should configure your customer-provided equipment (CPE) device to use both tunnels. You can select either of two routing types for these tunnels:
   BGP dynamic routing, with which the available routes are learned dynamically through BGP.

   Static routing, with which, when you set up the IPSec connection to the DRG, you specify to your on-premises network the particular routes that you want the VCN to know about.

   Refer to **Routing for the Oracle IPSec VPN**, referenced in Before Using VPN Connect for more details on choosing the routing type.
2. Complete the questionnaire in Setting Up VPN Connect, referenced in Before Using VPN Connect. These questions require such information as your VCN's CIDR, public IP address of your CPE device, the routing type you plan to use, and other pertinent details.
3. Set up the IPSec VPN components:
   - Create your VCN.
   - Create a DRG.
   - Attach the DRG to your VCN.
   - Create a route table and route rule for the DRG.
   - Create a security list and required rules.
   - Create a subnet in the VCN.
   - Create a CPE object and provide your CPE device's public IP address.
   - Create an IPSec connection to the CPE object and provide required routing information.
4. Use the CPE Configuration Helper, which will generate the information for your network engineer to use when configuring your CPE device. For more information, see Using the CPE Configuration Helper and also CPE Configuration, referenced in Before Using VPN Connect.
5. Have your network engineer configure your CPE device.
6. Validate connectivity.

Configure Customer-Provided Equipment

Network engineers require basic information about the inside and outside interfaces of your on-premises device; that is, your customer-provided equipment or CPE. This will allow them to configure these on-premises device at your end of the IPSec VPN so traffic can flow between your on-premises network and VCN.

> **Note**
>
> Oracle has verified specific software for use with VPN Connect; however, listing them here is beyond the scope of this chapter. You can see this list in Verified CPE Devices, referenced in Before Using VPN Connect.

The network engineer will need to take the following steps:

1. Determine the routing requirements. Oracle uses asymmetric routing across the multiple tunnels that make up the IPSec VPN connection. Even if you configure one tunnel as primary and another as backup, traffic from your VCN to your on-premises network can use any tunnel that is "up" on your device.
   Configure the firewalls accordingly. Otherwise, ping tests or application traffic across the connection will not reliably work. If you use BGP dynamic routing with your IPSec VPN, you can configure routing so that Oracle prefers one tunnel over the other.
2. Collect important information about VCN and an IPSec connections multiple IPSec tunnels. This information includes:
   - VCN OCID, which is a unique Oracle Cloud Infrastructure identifier that has a UUID at the end
   - VCN CIDR
   - VCN CIDR subnet mask
   - For each IPSec tunnel, the IP address of the Oracle IPSec tunnel endpoint (the VPN headend) and the shared secret
3. Collect basic information about the inside and outside interfaces of your on-premises device (your CPE).
4. Determine whether the Oracle VPN head ends use route-based tunnels or policy- based tunnels (note, too, that using policy-based tunnels comes with some caveats).
5. Observe these IPSec VPN best practices:
   - Configure all tunnels for every IPSec connection.
   - Have redundant CPEs in your on-premises locations.
   - Consider backup aggregate routes.
6. Test and confirm the connection status. After configuring the IPSec connection, test the connection by launching an instance into the VCN and then pinging it from your on-premises network.

Plan Redundancies

Redundancies in your IPSec VPN Connect implementation ensure reliable performance and minimize or eliminate costly downtime and other stresses on the system. When implementing redundancies, the key is to create backup paths built for efficiency, speed and availability.

To plan for redundancy, consider all the components (hardware, facilities, circuits, and power) between your on-premises network and Oracle Cloud Infrastructure. Also consider diversity, to

ensure that facilities are not shared between the paths. Redundancy considerations are described here:

Component	Considerations
Internet service provider (ISP)	Not all ISPs are the same. Peering relationships from your ISP let your traffic route more efficiently, reducing the latency as it varies over the internet.
Hardware	Enable services with redundant hardware, and ensure that there's no single point of failure anywhere in the path. How will you handle infrastructure maintenance (by Oracle or your own IT department)? Can you tolerate downtime? How much downtime can you tolerate?
Facilities diversity	Do you have redundant power feeds? Do you have diverse telecommunication entry points into your building? Is your equipment in different racks or data centers?
Oracle FastConnect POP diversity	Do you want to terminate both FastConnect circuits into the same point of presence (POP) or into different locations? Note that POP diversity is available only in the Phoenix, Ashburn, Frankfurt, and London regions.
Circuit provider diversity	Are you planning to use diverse carriers? Are your WAN or internet circuits fully diverse, or do they share a POP? Note that having different carriers doesn't mean that the circuits are fully diverse.

To review some useful examples of redundancy planning, see **Redundancy Use Cases** in the *Connectivity Redundancy Guide*, referenced in Before Using VPN Connect.

Before Using VPN Connect

The preceding VPN Connect best practices were culled from the Oracle Cloud Infrastructure documentation, which contains more detailed information on this service. Before using VPN Connect, you should review the following:

- For a VPN Connect prerequisite checklist, see Before You Get StartedBefore You Get Started.
- To understand the VPN Connect setup process, see Setting Up VPN ConnectSetting Up VPB Connect.
- For a list of customer-provided equipment (CPE) devices and their configuration, see Verified CPE DevicesVerified CPE Devices and CPE ConfigurationCPE Configuration .
- For redundancy planning guidelines, see Planning RedundancyPlanning Redundancy.

Understanding Oracle Cloud Infrastructure FastConnect

Oracle FastConnect allows for a dedicated private line, to be installed by a telecommunications provider between an external data center and Oracle Cloud Infrastructure's (OCI) data center. This provides reliable and secured communication through a separate dedicated private line connecting the Oracle Cloud Infrastructure data center and external data center and provides communication capabilities outside the public internet.

Oracle FastConnect can be used with the data center belonging to Oracle Utilities Cloud services application's customer or with third party data center for integrating external applications. FastConnect capability with a third party data center may involve additional

agreements/effort. If FastConnect capability is required, then it needs to be separately licensed and configured by the customer.

When to Use Oracle Cloud Infrastructure FastConnect

Use Oracle Cloud Infrastructure FastConnect whenever a high bandwidth dedicated private line between the data center hosting the external application and the Oracle Utilities Cloud services application is required.

Preparing to Use Oracle Cloud Infrastructure FastConnect

Before you implement Oracle Cloud Infrastructure FastConnect to integrate external applications with Oracle cloud services, do the following:

- Review FastConnect Documentation
- Understand FastConnect Requirements
- Review FastConnect Design Options
- Plan Redundancies

Review FastConnect Documentation

Oracle provides substantial and comprehensive documentation for implementing and using FastConnect which is beyond the scope of this chapter. Please refer to the links in Before Using Oracle Cloud Infrastructure FastConnect to locate this content.

Understand FastConnect Requirements

Before getting started with FastConnect, you need to address a set of implementation requirements.

At a minimum, meet these requirements:

- Have an Oracle Cloud Infrastructure account, with at least one user with appropriate Oracle Cloud Infrastructure Identity and Access Management (IAM) permissions.
- Have the proper Oracle Cloud Infrastructure Identity and Access Management (IAM) permissions (for example, a user in the Administrators group).
- Possess network equipment that supports Layer 3 routing using BGP.
- For colocation with Oracle: Ability to connect using single mode fiber in your selected FastConnect location. Also see Hardware and Routing Requirements.
- For connection to an Oracle partner, you need at least one physical network connection with the partner.
- For connection to a third-party provider, you need at least one physical connection with the provider.
- For private peering only, you need a least one existing DRG set up for your VCN.
- For public peering only:, you need the list of public IP address prefixes that you want to use with the connection. Oracle will validate your ownership of each prefix.

You can find additional information on these requirements in the Hardware and Routing Requirements documentation, referenced in Before Using Oracle Cloud Infrastructure FastConnect.

Review FastConnect Design Options

There are three FastConnect options to choose from: Oracle Provider, Third-Party Provider, and Colocation. This topic will help you to choose the best option for your implementation, based on consideration important points in the design.

You need to consider three main points when planning a FastConnect deployment:

- First, determine the Oracle location to which you want to connect.
  Oracle has various locations around the globe, called Regions, where Oracle Cloud Infrastructure is deployed. Each region has one or two physical entry locations called FastConnect Data Centers (DC). The FastConnect DC is the entry point into the OCI region and is equiped with redundant hardware. The FastConnect DC is where the customer will establish connectivity to. You can find a complete list of FastConnect-enabled regions at Oracle's North America Network Provider and Exchange Partners website, referenced in Before Using Oracle Cloud Infrastructure FastConnect.
- Next, identify the services to which you want to connect via FastConnect (also known as a virtual circuit).
  FastConnect refers to the physical connection between on-prem and Oracle Cloud Infrastructure (OCI). Within FastConnect you will create a virtual circuit to connect to services within OCI. There are two types of virtual circuits (VC):
  - Private peering, which is a virtual circuit when you want to connect on-prem to your Virtual Cloud Network (VCN) within OCI.
  - Public peering, which is a virtual circuit you can use to connect your on- premises system to Oracle Services Network (OSN) and access public services without using the Internet.
- Finally, determine what kind of FastConnect to use:
  Which of the three FastConnect options best meets your needs? To answer that question, you need to consider some different aspects of your design:
  - Where is the customer's data center located?
  - How many data centers the customer wants to connect to OCI
  - What relation does the customer have with providers or carriers?
  - Where can the provider deliver circuits or cross connects to
  - How fast does the customer want FastConnect deployed? Cost Latency Bandwidth

These are key points to consider in the design and choosing the proper FastConnect option. For example, if the customer has a good relation with Provider A but Provider A is not an Oracle Provider, then the next option is to use Third-Party provider or Colocation option. If Provider A can deploy circuits to the FastConnect DC but can't deploy cross connect at the FastConnect DC then customer will have to look at a different provider. If the customer is not at the same address as the FastConnect DC then Colocation is not an option.

This is just a brief outline of how to address the design options for implementing OCI FastConnect. For a more detailed discussion of these design options, see the blog post FastConnect Design, referenced in Before Using Oracle Cloud Infrastructure FastConnect.

Plan Redundancies

Redundancies in your FastConnect implementation ensure reliable performance and minimize or eliminate costly downtime and other stresses on the system. When implementing redundancies, the key is to create backup paths built for efficiency, speed and availability.

While which redundancy best practices you should follow depend on which connectivity model you use, you should always design your network to achieve high availability (HA) and so that it's prepared for these types of disruptions:

- Regularly scheduled maintenance by your organization, your provider (if you're using one), or Oracle.
- Unexpected failures on the part of your networking components, your provider, or Oracle. Failures are rare, but you should plan for them.

For help ensure redundancy, Oracle provides:

- Multiple providers for each region
- Two FastConnect locations for each of the following regions (all other regions have a single FastConnect location)
  - Germany Central (Frankfurt)
  - UK South (London)
  - US East (Ashburn)
  - US West (Phoenix)
- Two routers in each FastConnect location
- Multiple physical connections between each Oracle partner and Oracle (for a given region)

You can find a more comprehensive discussion of best practices specific to a FastConnect connectivity model in FastConnect Redundancy Best Practices, referenced in Before Using Oracle Cloud Infrastructure FastConnect.

Before Using Oracle Cloud Infrastructure FastConnect

The preceding Oracle Cloud Infrastructure FastConnect best practices were culled from the Oracle Cloud Infrastructure documentation, which contains more detailed information on this service. Before using FastConnect, you should review the following documentation:

- For comprehensive FastConnect documentation, see FastConnectFastConnect.
- To understand the FastConnect prerequisites, see FastConnect RequirementsFastConnect Requirements.
- For best practices when designing a FastConnect implementation, see FastConnect Redundancy Best PracticesFastConnect Redundancy Best Practices .

Understanding Private Endpoints (PE)

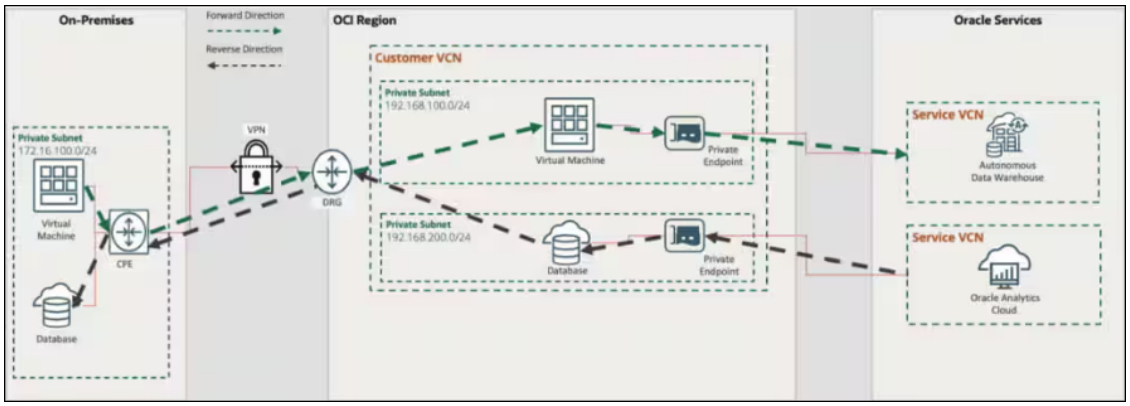
Many Oracle customer cloud networks are designed to prevent outbound connections to the internet. Typical on-premises deployed services are hosted on the enterprise intranet and don’t require public internet access. Migrating these services to the cloud by using the public connectivity model requires customers to enable internet access for their backend clients and punch holes in their firewalls and other security controls. Private connectivity using a service gateway offers a private and secure alternative. Although service gateways enable private access to Oracle services from customer OCI and on-premises networks, this access is over a publicly accessible IP.

Private endpoints allow customers to access Oracle services over a private IP taken from a subnet within their VCN. This capability allows a customer experience where the service is hosted in their own private network, either in their own VCN or on their on-premises site. When allowed, private endpoints support reverse connections to allow Oracle services to initiate connections to instances within the customer’s VCN or on-premises data sources.

The following architectural diagram depicts how customer-to-service (C2S) or forward direction and service-to-customer (S2C) or reverse direction traffic flows through the private endpoint. Within the customer VCN, the private endpoint is a logical representation of the service’s fully

qualified domain name (FQDN) or IP address. Customers access the service using a dynamically allocated free private IP address from their chosen subnet as a target. Private endpoints physically reside outside the customer VCN and are managed on-behalf of the customer by the service provider.

Figure 8-7 Private Endpoints



Customer Obligations

Customers must perform the following activities prior to raising the private endpoint service request.

Step 1: Create a VCN with a private subnet where the private endpoint can be created.

Step 2: Add the necessary policy to allow private endpoint creation on customer tenancy.

Add the following IAM Policy to allow Oracle to create Private endpoint in the compartment in your tenancy.

```
allow service ORACLE_INDUSTRY_SAAS to manage vnics in compartment <Customer
Compartment Name>
allow service ORACLE_INDUSTRY_SAAS to use subnets in compartment <Customer
Compartment Name>
allow service ORACLE_INDUSTRY_SAAS to use network-security-groups in
compartment <Customer Compartment Name>
allow service ORACLE_INDUSTRY_SAAS to inspect work-requests in compartment
<Customer Compartment Name>
```

Example:

```
allow service ORACLE_INDUSTRY_SAAS to manage vnics in compartment JI_PE_POC
allow service ORACLE_INDUSTRY_SAAS to use subnets in compartment JI_PE_POC
allow service ORACLE_INDUSTRY_SAAS to use network-security-groups in
compartment JI_PE_POC
allow service ORACLE_INDUSTRY_SAAS to inspect work-requests in compartment
JI_PE_POC
```

If the customer requirement is to access the cloud services via private network, they need to create a service request with the Oracle Cloud Operations team as describe in **Request for Private Endpoint (PE)** in the *Oracle Utilities Cloud Services Cloud Operations Guide*.

Accessing SaaS Applications Using the Service Gateway

A service gateway lets resources in your VCN privately access specific Oracle services, without exposing the data to the public internet. The resources in the VCN can be in a private subnet and use only private IP addresses. The traffic from the VCN to the service of interest travels over the Oracle network fabric and never traverses the internet. Private endpoint is another way to have private access to oracle cloud services. This document is focusing on Service Gateway only.

Please refer to Access to Oracle Services: Service Gateway in the Oracle Cloud Infrastructure documentation for more information.

By default, all incoming network traffic (Web/SOAP/REST) using Public Internet to Oracle Utilities SaaS applications MUST go through Akamai network. Any direct attempt to access Oracle Utilities SaaS applications bypassing Akamai is blocked.

The only exception to bypass Akamai network is to access Oracle Utilities SaaS applications via Service Gateway using Private Subnet in OCI VCN. This connectivity is routed privately and does not go through Akamai network. This exception is handled via Inbound Allowlist internally managed by Oracle.

Require Customer Actions

1. Create a Request for Inbound Allow List Cloud Operations service request via My Oracle Support and provide the following:
   - VCN OCID that will access the application through Service Gateway.
   - IP address ranges defined as CIDR blocks.
   - Both IP and VCN OCID.

   > **Note**
   >
   > Customers can request for Home Region DNS info, if needed. Please refer to the *Oracle Utilities Cloud Services Cloud Operations Guide* for more details.

2. After confirmation that the service request has been completed, perform the following:
   - For customers using **OCI DNS resolver** with the OCI VCN:
     - Create a Private View for the utilities-cloud.oracleindustry.com Zone. Refer to Akamai - Access to Energy & Water SaaS Applications using Service Gateway (KB500053) on My Oracle Support for more information.
     - Add Service Gateway to the OCI VCN.
     - Add a Route Rule in the private subnet's Route Table that directs Oracle Services traffic through Service Gateway.
   - For customers using Service Gateway with **on-premises DNS resolver**:
     - Override the resolution of Application Hostname to resolve to the Home DNS of the region from the DNS resolver used in their network.
     - Add Service Gateway to the VCN.
     - Add a Route Rule in the private subnet's Route Table that directs Oracle Services traffic through Service Gateway.

Understanding the Differences Between VPN Connect and OCI FastConnect

If you don't need to use a reverse proxy to implement integration between Oracle Utilities Cloud services application and an application hosted externally but still aren't sure whether to use IPSec VPN Connect or OCI FastConnect, understanding the differences between the two will help you make the correct determination.

An IPSec VPN establishes an encrypted network connection over the internet between the external application's data center and your Oracle Cloud Infrastructure virtual cloud network (VCN). It provides low or modest bandwidth and has the inherent variability in internet-based connections.

FastConnect bypasses the internet. Instead, it uses dedicated, private network connections between the external application's network or data center and your VCN.

Further comparisons are here:

Feature	IPSec VPN Connect	OCI FastConnect
Use case	Development, test and small scale production workloads	Enterprise-class and mission critical workloads, Oracle Applications, Backup, DR
Supported Services	All OCI Services within VCN	All OCI Services within VCN
Typical Bandwidth	Typically < 250 Mbps aggregate	Higher bandwidth; increments of 1 Gbps and 10 Gbps ports
Internet	Internet Protocol Security (IPsec)	Border Gate Protocol (BGP)
Routing	Static Routing	Dynamic Routing
Connection Resiliency	active-active	active-active
Encryption	Yes, by default	No, can be achieved using virtual firewall
Pricing	Free for a managed service	Billable Port hours No data transfer charge between availability domains
Service Level Agreement	No Service Level Agreement	99.9% availability Service Level Agreement

9

Migrating Legacy Custom Tables and Java to Oracle Utilities Cloud Services

This chapter provides an overview of the custom Java and custom table migration capability available in selected Oracle Utilities Cloud Services. This chapter includes the following:

Migrating Legacy Custom Tables and Java Introduction

This guide describes how you can migrate custom tables and Java code to the following cloud services:

- Oracle Utilities Customer Care and Billing Cloud Service
- Oracle Utilities Customer Cloud Service

> **Note**
>
> This capability is ONLY supported in the above listed Oracle Utilities cloud services. Cloud services not listed above do not support this capability.

On-premises customers, including Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) customers, are permitted to extend the solutions using both custom tables and custom Java code.

Selected services have been extended to allow an initial load and limited maintenance of both custom tables and custom Java code, in compliance with Oracle policy, to allow on-premises implementations the opportunity to take advantage of the benefits of the Oracle Utilities Cloud Services. Once the custom Java and tables are loaded, the cloud native service capabilities will be used to further extend solutions.

Migration Features

Selected Oracle Utilities cloud services have been extended with new functionality to allow migration of on-premises customers with custom tables and/or custom Java code for extensions. This includes:

- **Oracle Utilities Software Development Kit (SDK) Support**: Customers and partners can continue to use the external Oracle Utilities Software Development Kit (SDK) to build custom tables and/or custom Java code as they do on- premises.

- **Initial Load**: Select Oracle Utilities cloud services include a capability to load custom tables and/or custom Java code from an on-premises implementation as part of a migration.
- **Compliance Checks**: Custom tables and custom Java will be assessed against common standards and compliance with the Oracle Utilities cloud service. This compliance includes assessment against the published allowlist and the security standards employed by the service to preserve cost and risk savings. Any non- compliance will be flagged and not loaded into the service.
- **Maintenance Capability**: Once the custom tables and/or custom Java code is loaded into the Oracle Utilities cloud service, it can be maintained within the tolerances outlined in the service descriptions. The same capabilities used for the initial load are reused for maintenance.
- **Isolation of Custom Objects**: To maximize efficiency and lower risk, the custom Java and/or custom tables are isolated from the main service but accessible from the service. This is to ensure compliance and reduce impact on cloud service activities.
- **CMA/TDM Compatibility**: Once custom Java and tables are loaded into the service, they can be migrated using the Content Migration Assistant (CMA)/Test Data Management (TDM) using custom Migration Plans and Migration Requests.

> **Note**
>
> Test Data Management uses Content Migration Assistant to migrate master and transaction data.

The capabilities ensure compliance and stability whilst allowing custom tables and/or custom Java code to continue to be used within the Oracle Utilities cloud service.

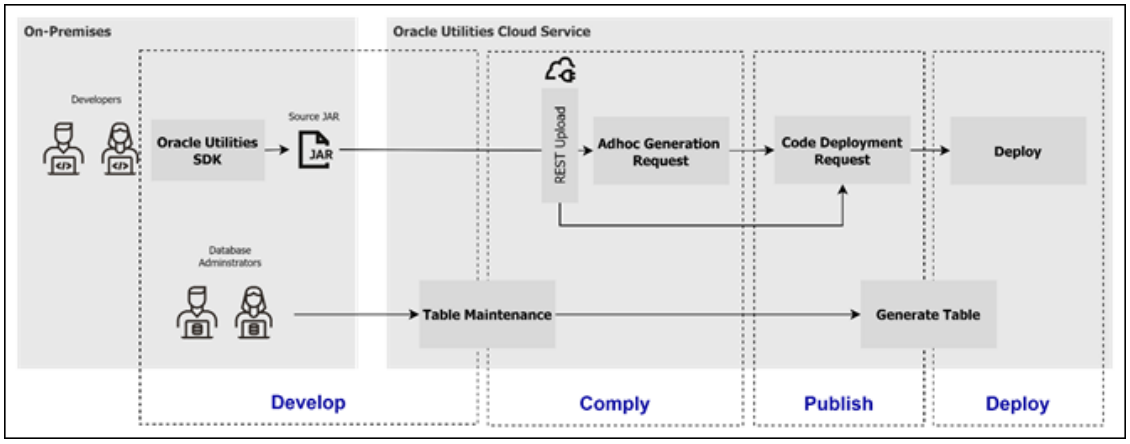
Process

The generic process for migrating custom tables and/or custom Java code is as follows:

- **Development**: Code and tables are developed in an on-premises development instance using database tools and the Oracle Utilities SDK. This is similar in nature to the existing processes used for on-premises implementations. At the end of this process the SDK can export the extensions into a format acceptable to the Oracle Utilities cloud service in JAR format. A set of REST API's are available to upload the custom content to the cloud to process existing customizations.
- **Compliance Checking**: Once the code is uploaded, it must be submitted to compliance checking for suitability for the service. This compliance checking will assess code and tables in respect to compliance against a range of standards used on the Oracle Utilities cloud service. Code that is not compliant with any of the assessed standards will not be deployed to the cloud until remediation is complete.
- **Publishing to Cloud**: Once the code and tables are assessed as compliant with relevant standards they are assembled and loaded into the cloud environment.
- **Deployment**: Once the code and/or tables are loaded into the cloud environment, they are deployed using the cloud infrastructure and are ready for use.

This process is illustrated in the following diagram.

Figure 9-1 Overall Migration/Maintenance Process



Custom Tables in Oracle Utilities Cloud Services

This sections provides details related to the migration of custom tables to Oracle Utilities cloud services. This includes:

- Restrictions
- Implementation Details
- Table Definition and Maintenance Process
- Guidelines for Migrating Tables

Restrictions

To implement custom tables within Oracle Utilities cloud services several restrictions must be considered:

- **Limited Database Object Type Support**: At the present time only the following object types are supported:
  - Tables
  - Columns
  - Constraints
  - Indexes

> **Note**
>
> No other object types are supported at the present time.

> **Note**
>
> Customers using database sequences may convert their sequence code to use the F1-DocumentSequenceAddUpd business service.

- **Limited Data Type Support**: The Oracle Utilities Application Framework uses a subset of the database types available for columns and therefore only this subset is supported. This includes the following data types:
  - CHAR
  - CLOB
  - DATE (includes Date and time)
  - NUMBER
  - VARCHAR2
  - XMLTYPE
- **Naming Conventions are Strictly Enforced**: The Oracle Utilities Application Framework uses a set of database naming conventions for tables and columns that are enforced. Refer to the Oracle Utilities SDK for details. This is enforced to minimize impacts of patches and upgrades on Oracle Utilities cloud services.
- **Limited Scope for Change**: In line with Oracle policy, the relative amount of change on the custom content is limited and is checked as part of compliance. It is expected that further extensions will be implemented using cloud facilities to retain the cost benefits of the service.
- **No Partitioning Support**: The custom tables cannot be partitioned as of 23A release.
- **No ILM Support**: Custom tables cannot implement Information Lifecycle Management (ILM) in the current release. This is in line with current on-premises directions. Managing of the lifecycle of the data in these custom tables is not a responsibility of the service.
- **Custom CMA Content Required**: Content Migration Assistant is used to migrate data on the cloud. Implementations must create custom Migration Plans and Migration Requests to include any custom tables in migrations or test data management.
- **Table Definition Required**: Each table must be defined as a table object within the Oracle Utilities Application Framework to ensure the cloud service is aware of the presence of the table. This definition must include all columns, constraints, and indexes to ensure compliance. Any conflicts with base tables must result in changes to custom tables.
- **No Definition Autoload**: In the present release, there is no autoload capability to transfer the format of the custom tables to the table definition. They must be entered manually using the provided Table maintenance capability.
- **Data Loading Support**: After definition the data can be loaded using the cloud standard data loader process.
- **Truncation and Index Maintenance Support**: For custom table maintenance, use cloud standard truncation and index maintenance processes.
- **Minimal Configuration**: For a custom table to be accepted by the system it must satisfy the following for definitions:
  - The System Table flag must not be set. This is reserved for tables supplied by Oracle only.
  - A primary key for the table must be defined.
  - The primary key index must be defined to the table.
  - The VERSION field must exist in the table.

Implementation Details

When a custom table is implemented in the cloud it is implemented in standard way to maximize efficiency and take advantage of the cloud capabilities including:

- **Tablespace Automatically Allocated**: The tablespace and storage allocations are created automatically as part of the deployment. There is no configuration necessary.
- **Automatically Connected**: The table is available automatically to the product, SQL Developer/Oracle Database Actions and Analytics Publisher as part of the deployment. This includes providing automatic Oracle REST Data Service integration.
- **Isolation from Product**: The table is installed in its own isolated schema to prevent schema leakage into the main service schema.
- **Data Loading**: The data is loaded using the Cloud endorsed method used for other tables. Refer to the Cloud implementation guides for additional information.

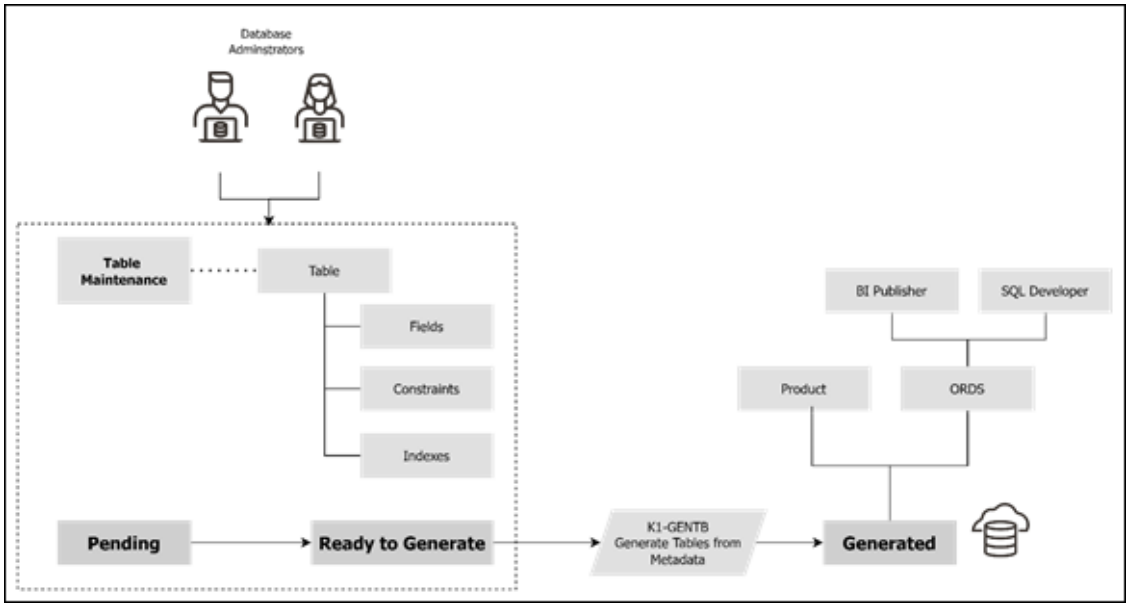
Table Definition and Maintenance Process

The implementation of custom tables follows a set process to allow for definition, assessment, and deployment to Oracle Utilities cloud services. The process is as follows:

- **Table Maintenance**: The tables and related objects are defined to the service via the Table Maintenance function. Each table to be implemented will be defined to the Oracle Utilities Application Framework to ensure correct definition and management by the service. Any maintenance of the definition is performed using this capability. It is recommended that only a subset of approved users perform this action to provide sufficient control. By default, the definition should be set to Pending state when being created. Once the definition is complete, each table entry should be manually transitioned in Ready To Generate status, to progress to the next step.
- **Generation**: The table definition can be generated and deployed using the K1- GENTB batch process, which will implement any table definition in Ready to Generate status. Once deployed to the database the batch process automatically enables access to the service components. Once this is performed the table definition is transitioned to Generated state. Any maintenance changes should be performed after manual transitioning the table object back to the Pending state to repeat the process for changes.

The diagram below illustrates the custom table migration process.

Figure 9-2 Custom Table Migration Process



Guidelines for Migrating Tables

When migrating custom tables to the cloud the following guidelines should be considered to transition as smoothly as possible:

- **Follow Naming Conventions**: Follow the table naming conventions for tables, columns, constraints, and indexes as outlined in the Oracle Utilities Software Development Kit (SDK) and Database Naming Conventions. These are assessed to ensure no conflicts with base objects.
- **Column Naming**: As per the Oracle Utilities Software Development Kit (SDK) and Database Naming Conventions, the column names should conform with standards and any shared columns names with base should adhere to standards. Avoid creating new field definitions if there are field definitions that can be reused to minimize impacts of change.
- **Remove Storage and Technical Details**: The table maintenance will automatically allocate storage and other attributes of the table outside the definition.
- **Enter Table Definition Appropriate for Custom Tables**. Define the table object using the following guidelines:

Parameter	Recommended Settings	Comments
Table	Physical Table Name	Ensure the table name is prefixed and named as per the Oracle Utilities SDK documentation
Table Type	Table	Other types are shown but not supported on the Oracle Utilities Cloud Service at the present time
System Table	No	This is reserved for service tables only and should not be enabled

Parameter	Recommended Settings	Comments
Table Classification	See Comments	For custom tables the following values are supported: – Admin Non-System Table – Master – Transaction
Help URL	See Comments	Not supported for custom tables
Table Volume Type	See Comments	Indicate volume type at creation time to ensure appropriate sizing by service.

> **Note**
>
> Settings not shown may be used as outlined in the documentation.

- **Define Columns**: Define the columns in the table. Ensure each column is defined as a Field with appropriate formats and so on to add to the table definition. Reuse base field definitions as appropriately. If fields conflict with base fields, then you must alter your custom table to mitigate conflict.
- **Define Constraints**: Define all the key constraints including primary key and foreign keys. Conditional foreign keys are only checked if values are present. Once this is specified and saved, the Relationship tab will be populated.
- **Define Indexes**: Define all required indexes for the table including any primary key indexes.

Custom Java in Oracle Utilities Cloud Services

Java is one of the supported extension technologies for on-premises implementations using the Oracle Utilities SDK to build and maintain Java code. Traditionally, the Oracle Utilities SDK also includes a set of utilities to transfer the extensions from the developer's environment to a target environment on-premises.

For customers with investments in existing Java-based extensions, the Oracle Utilities SDK and selected Oracle Utilities cloud services have been enhanced to allow the cloud as a target from the Oracle Utilities SDK to allow customers to transition to Oracle Utilities cloud services.

This sections provides details related to the migration of custom Java code to Oracle Utilities cloud services. This includes:

Restrictions

To allow the custom Java extensions to operate within Oracle Utilities cloud service the code must comply with the following:

- **Oracle Utilities SDK Support**:The java code must be developed using the Oracle Utilities SDK and the extraction utilities used to provide to the cloud.
- **No Compiled Code**: Only uncompiled code is accepted to be loaded onto Oracle Utilities cloud services to be assessed as part of the migration process and ongoing maintenance. Only source JAR files will be accepted as part of the process.
- **No External Libraries**: To ensure compliance and reduce risk, no third-party libraries, other than those provided by the service will be permitted.
- **Code Location**: The custom code must only exist in the com\splwg\cm\domain directory of the JAR file. No other directories or subdirectories are acceptable.
- **Only Java Files are Accepted**: Only .java files are accepted by the interface. All other file types are ignored and not uploaded into the service.
- **Allowlist Compliance**: The Java code will be assessed against the Oracle Utilities cloud service and Oracle Cloud Infrastructure allowlists to ensure code is compliant with the cloud service. Code requiring direct access to memory, database, disk, or threads will not be permitted.
- **Development Environments Only**: The process used to transfer, verify, and build the deployment for the custom Java code can only be performed on development environments only. This feature is disabled, by default, in other non-production environments. This feature cannot be used on Production environments. It is expected to use Content Migration Assistant to migrate deployments across environments.
- **Limited Scope for Change**: In line with Oracle policy, the relative amount of change on the custom content is limited and is checked as part of compliance. This capability is a load and maintain capability only. It is not intended to allow continual extension of the product using Java, just to retain the investment in existing extensions. Partners and customers are expected to use existing cloud friendly capabilities to extend those services past the initial load and maintain levels.

Non-compliance with these restrictions or because of the compliance checking process will result in code being rejected by the Oracle Utilities cloud service and subject to remediation by partners and customers.

Code Migration Process

The cloud migration process has been designed to allow customers to reuse existing development tools and processes but use the cloud as the target environment. The process to migrate code is as follows:

- **Develop Code**: Using the Oracle Utilities SDK, develop Java code against a development version of the product. Once complete, use the SDK utilities as described in the SDK documentation to extract the source in a package for implementation on the cloud. The following additional requirements must be adhered to:
  - Only code in the `com/splwg/cm/domain` namespace is permitted
  - A manifest file should not be provided in the source JAR
- **Ad Hoc Generation Request**: Upload the source using the provided REST API. This will assess the source against the service. This can be used to assess the source prior to uploading to the service. You must assign a Code Id which identifies the upload. The Code Id must only contain alphanumeric characters, dashes, or underscores. Credentials for the

cloud service need to be provided with the command. The uploaded source is set to the "Generated" state if the assessment is successful. The source is assessed against several standards which include:

  - **Allowlist Compliance**: The code is assessed against classes and methods maintained for the Oracle Cloud Infrastructure. This assesses class usage to ensure manipulation of memory, network and file access are within tolerances.

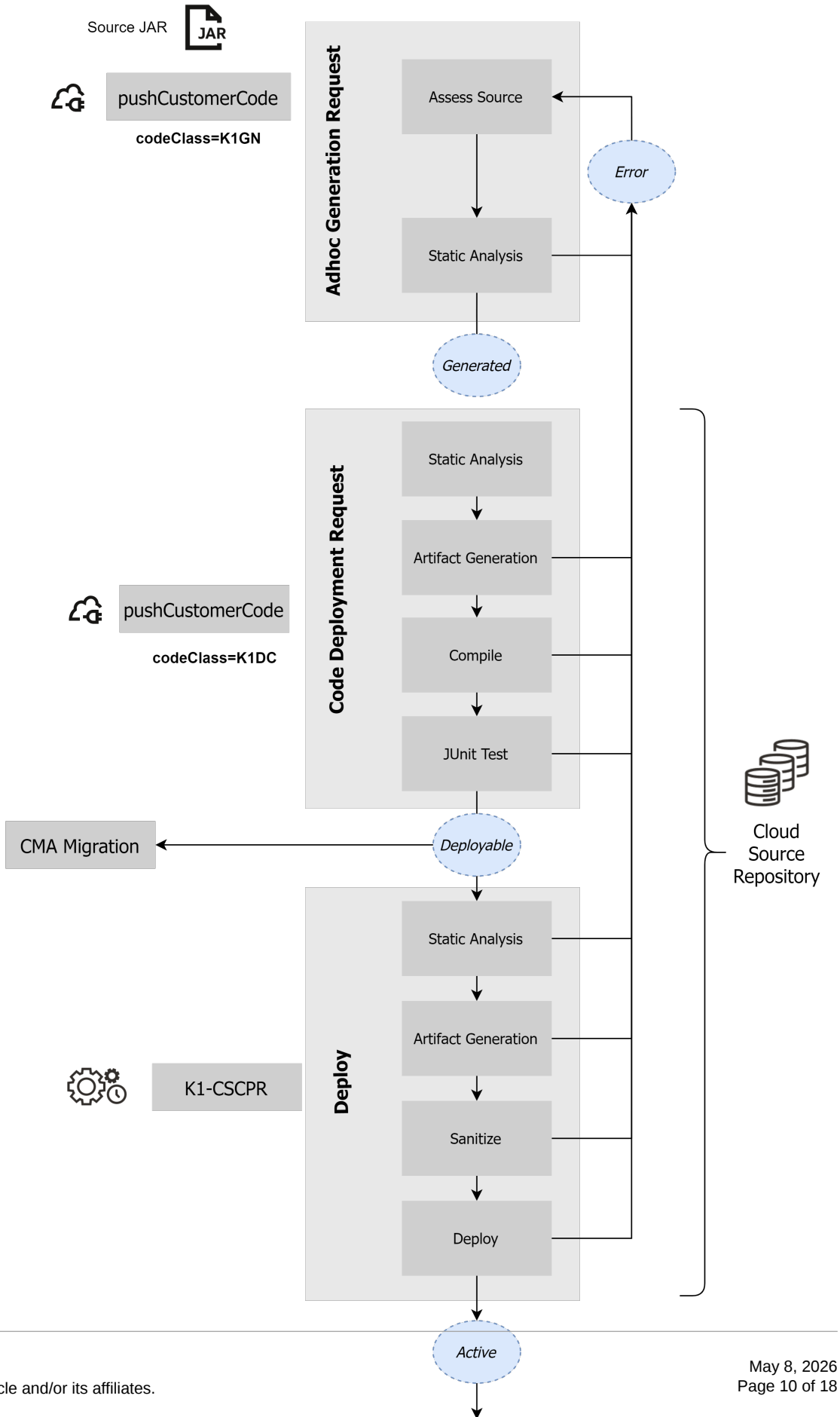
    > **Note**
    >
    > Refer to the Groovy class allowlist for a list of the common classes available.

  - **Malicious Code Checks**: The code will be checked for malicious code within tolerances set by the Oracle Cloud Infrastructure.

- **Code Deployment Request**: This step reassesses, compiles, and assembles the source code into the deployable module that is stored in the database ready for the deployment to the server. This also makes the deployment module available to be migrated using Content Migration Assistant (CMA) to make available to other environments. This step is independent of the Ad Hod Generation Request and loads the code into the cloud source repository. The Code Id used for this step must be unique for each upload.
- **Deployment**: This is the final step of the process that deploys the compiled code into the cloud infrastructure and made available to the service.

The diagram below illustrates the Java code migration and maintenance process.

Figure 9-3 Java Migration and Maintenance Process



Upload and Process API

To upload the source JAR file the following command must be issued from your Oracle Utilities SDK client machine:

```
curl -F "codeClass=<class>" -F "sourceCodeContent=<source>" -i -X POST
https://<serviceurl>/rest/customerCode/pushCustomerCode/<codeid> -u
<user>:<password>
```

The parameters used in this command are as follows:

Parameter	Description
<class>	Process to use for call:•K1GN - Adhoc Generation Request•K1DC - Code Deployment Request
<source>	Provide the value "@cm.jar"
<serviceurl>	Fully qualified URL for development environment to upload source to. **Note**: Only development environments can be used to upload source. This interface will not be available in non-development environments for security reasons.
<codeid>	Code Identifier. The identifier must only contain alphanumerical, dashes, and underscores
<userid>	User Id to use to connect to the cloud service
<password>	Password for user to connect to the cloud service

Additionally provide the following payload information:

Parameter	Comments
sourceCodeContent	Upload the JAR file into this element using a REST client.
commitReference	Commit reference from source control system used in Oracle Utilities SDK.•This should be provided for the initial Adhoc Generation Request (K1GN) to identify the source.•This is required for the Code Deployment Request (K1DC) to identify the source to process.
branchName	Branch name where the code is to be deployed. Only the value main is supported at this time.
changeReference	Optional. Change reference for traceability purposes only. This can be a release identifier for the custom code if necessary.
comments	Optional. Additional comments to be attached to the upload for documentation purposes.

> **Note**
>
> These parameters maybe provided in the REST client used to connect to the service or on the curl command line using the `-F "<parameter>=value"` option.

Preparing and Deploying

Once the code is uploaded and checked and verified, the next stage is to deploy the CM code into its isolated area and enable the new deployment. To do this:

- Run the K1-CSCPR job to deploy all the customizations in the PRIVILEGED threadpool . This must be performed after the Code Deployment Request has been successful. This process will deploy the customizations to the service.

> **Note**
>
> The PRIVILEGED threadpool is the only threadpool supporting Java deployment and can only run this process. Other processes cannot be run in this threadpool.

- Once the job has successfully completed, restart the CM servers where the deployment has occurred. This does not impact the non-CM severs. To do this, start the browser interface with the ?debug=true option as the suffix on the URL. Use the Restart CM Jar Servers to initiate the restart.

Advanced Operations

There are additional operations supported for Java developers that may be useful during development.

Downloading Generated Code from the Cloud

In most cases source code is housed in the Oracle Utilities SDK but it is also possible to get a copy of the generated code artifacts run on the cloud using the following command:

```
curl -X GET https://<serviceurl>/rest/customerCode/getGeneratedCustomerCode/
<codeid> -u <userid>:<password> -J -O
```

> **Note**
>
> Use of the `-O` will output to stdout. Redirect to a file name to save locally.

The parameters used in this command are as follows:

Parameter	Description
<serviceurl>	Fully qualified URL for development environment to upload source to. **Note**: Only development environments can be used to upload source. This interface will not be available in non-development environments for security reasons.
<codeid>	Code Identifier. The identifier must only contain alphanumerical, dashes, and underscores
<userid>	User Id to use to connect to the cloud service
<password>	Password for user to connect to the cloud service

Downloading API Jar files

It is possible to get a copy of the API Jar files used on the cloud using the following command:

```
curl -X GET https://<serviceurl>/rest/customerCode/getApiJars -u
<userid>:<password> -J -O
```

> **Note**
>
> Use of the `-O` will output to stdout. Redirect to a file name to save locally.

The parameters used in this command are as follows:

Parameter	Description
<serviceurl>	Fully qualified URL for development environment to upload source to. **Note**: Only development environments can be used to upload source. This interface will not be available in non-development environments for security reasons.
<userid>	User Id to use to connect to the cloud service
<password>	Password for user to connect to the cloud service

Allow and No Allowlist Of Java Classes

Oracle Utilities cloud service uses prebuilt allowlists to distinguish which classes that are permitted and not-permitted as part of the service. These lists are maintained by Oracle in conjunction with the Oracle Security team in line with Oracle Cloud policy. A full up to date list can be extracted using the API Jar files associated with your release of the service.

In general terms the following types of classes are automatically supported in the Oracle Utilities cloud services:

- Entity classes such as entity data transfer objects (DTO), id classes, language classes, language id classes, and collection classes.
- Lookups
- Algorithm Spot interfaces
- JOBOL record classes

Other classes will be available via the API Jar extract for reference purposes. Any class that is not listed is not available for use in the Oracle Utilities cloud services.

> **Note**
>
> These API's change regularly and therefore is it is recommended to be extracted on a regular basis.

Migration using Content Migration Assistant

After the deployment is built the compiled and deployable code can be migrated using the Content Migration Assistant (CMA) to your target environments.

It is highly recommended to use the Approve capability within Content Migration Assistant within the import process to manually review and approve changes before applying the change

to the target environment. This provides a release approval step to implement fine grained control over change.

> **Note**
>
> After the migration has been applied, the deployment batch process and CM servers restarted must be executed to complete the implementation of the new release.

Additional Step After Major Upgrades

During a major upgrade process, the system may temporarily disable CM (Custom Java Code). This is done to prevent potential conflicts or failures that can occur if custom Java code executes and returns errors during critical upgrade steps.

Once the upgrade is complete and the system has been handed back to the customer, the customer must run the CMA (Configuration Management Assistant). This post-upgrade step ensures that all custom configurations and logic are properly reapplied, validated, and aligned with the upgraded environment, thereby restoring full functionality of the Custom Java Code.

Other Custom Object Types

While the focus of this guide is custom tables and custom Java code, there are potentially other customization object types that need to be remediated before moving to the cloud. The table below outlines supported and non-supported extension types that need to be addressed as part of a migration:

Object Type	Description	Remediation Advice
Database Sequences	Sequences stored on the database used for identifier or other processing	Migrate code to use F1-DocumenSequenceAddUpd business service for sequence maintenance
JSP	These are code-complete screens written exclusively in JSP format	These need to be migrated to use UI Maps with simplified JSP code
JSP User Extensions	These are code snippets for legacy JSP based screens to override behavior on those screens	Not supported at the present time
FILE-PATH	Specifications of raw directories on the machine for batch processing. For example, directories to read or write files	Migrate to use aliases using the Cloud Object Storage Adapter
Environment Settings	Settings in various files in the architecture	Not supported as service provides the environment preconfigured.
Custom Threadpool definitions	Settings and architecture of batch execution threadpools	Migrate to use DEFAULT
Configuration Data (Admin Menu)	Administration menu data	Migratable using Content Migration Assistant
PL/SQL Code	Custom pl/sql functions	Not supported for security reasons
Triggers	Custom code on database	Not supported for security reasons
Other database objects	Other database objects that are not listed in the Custom Table support	Not supported

Object Type	Description	Remediation Advice
Linux/Unix Scripts	Custom scripts to manage the service	Not supported as service takes on that responsibility
Custom WS-Policies	Custom policies for inbound and outbound integration	Not supported as service provides prebuilt security

Database Naming Conventions

This section provides a standard for database objects (such as tables, columns, and indexes) for products using Oracle Utilities Application Framework. This standard is introduced to ensure clean database design, to promote communications, and to reduce errors leading to smooth integration and upgrade processes. Just as Oracle Utilities Application Framework goes thorough innovation in every release of the software, it is also inevitable that the product will take advantage of various database vendors' new features in each release. The recommendations in the database installation section include only the ones that have been proved by vigorous quality assurance processes, field tests and benchmarks.

The following naming standards must be applied to database objects.

Tables

Table names are prefixed with the owner flag value of the product. For customer modification CM must prefix the table name. The length of the table names must be less than or equal to 30 characters. A language table should be named by suffixing _L to the main table. The key table name should be named by suffixing _K to the main table.

It is recommended to start a table name with the 2-3 letter acronym of the subsystem name that the table belongs to.

Some examples are:

- CM_MYTYPE
- CM_MYTYPE_L

> **Note**
>
> A language table stores language sensitive columns such as a description of a code. The primary key of a language table consists of the primary key of the code table plus language code (LANGAGUE_CD).

> **Note**
>
> A key table accompanies a table with a surrogate key column. A key value is stored with the environment id that the key value resides in the key table.

> **Note**
>
> Tables prior to V2.0.0 are prefixed with CI_ or SC_.

Columns

The length of a column name must be less than or equal to 30 characters. The following conventions apply when you define special types of columns in the database.

- Use the suffix _FLG to define a lookup table field. Flag columns must be CHAR(4). Choose lookup field names carefully as these column names are defined in the lookup table (CI_LOOKUP_FLD) and must be prefixed by the product owner flag value.
- Use the suffix _CD to define user-defined codes. User-defined codes are primarily found as the key column of the admin tables.
- Use the suffix _ID to define system assigned key columns.
- Use the suffix _SW to define Boolean columns. The valid values of the switches are 'Y' or 'N'. The switch columns must be CHAR(1)
- Use the suffix _DT to define Date columns.
- Use the suffix _DTTM to define Date Time columns.
- Use the suffix _TM to define Time columns.
- Use the suffix _AREA to define XML Type and CLOB columns

Some examples:

- ADJ_STATUS_FLG
- CAN_RSN_CD

The following database column types are supported:

- CHAR
- CLOB
- DATE (includes Date and time)
- NUMBER
- VARCHAR2
- XMLTYPE

Indexes

Index names are composed of the following components:

`<prefix><datatype><number><indextype><suffix>`
Index components include the following:

Component	Description and Comments
<prefix>	All indexes must be owned appropriately. For client specific implementation indexes, use CM as the prefix to avoid conflicts with base product indexes.
<datatype>	For base product use only. Indicates type of data stored in table. Valid values:•C indicates the table is an Administration/Control Table•M indicates the table is a master table.•T indicates the table is a transaction table.
<number>	Unique three-digit number that makes the index name unique for a table. All indexes for the same table should share the same number.

Component	Description and Comments
<indextype>	Type of index. Valid values:•P indicates that this index is the primary key index•S indicates that this index is a non-primary index
<suffix>	Suffix to support multiple indexes on a table using the following rules:•0 (zero) as a suffix is reserved for the primary index•1 - 9 is reserved for each non-primary index on a table.

Examples:

- XC001P0
- XT206S1
- CM206S2

Frequently Asked Questions - Migrating Custom Tables and Java

There are several common questions that will help you clarify the process for this process.

Does this apply to all Oracle Utilities cloud services?

No, this capability is designed for existing Oracle Utilities Customer Care and Billing and Oracle Utilities Customer to Meter on-premises implementations to migrate to the cloud quickly. It will not be extended past that scenario.

What is the code identifier used for?

With each upload you should specify a code identifier to identify the code you are uploading for each step. This can be any valid release identifier applicable to the developer or code being uploaded. The format of the identifier must only include alphanumeric, underscores, and dashes. Spaces and special characters are not supported as part of identifiers.

How do I deal with compliance failures?

The process has been designed to assess the custom Java source code and custom tables against standards outlined in the Oracle Utilities SDK and cloud service standards to ensure compliance with the cloud and help ensure high levels of security are maintained. If there is non-compliance detected this will be reported by the various utilities and the suspect code is not implemented. It is recommended to re-mediate the code at the source and re-upload the source for reassessment till it is fully compliant.

Can I add new Java and Tables after I migrate?

No, the design of the capability is to quickly migrate to the cloud, maintain your existing extensions, and then use the cloud facilities for new extensions going forward to retain and maximize cost benefits of moving to the cloud.

Is the custom code and table ring fenced?

Yes, the service architecture has been designed to isolate extensions to a different but integrated part of the architecture for security isolation and prevent instability. The architecture uses the cloud infrastructure to provide sufficient access to the extensions that is transparent to the extension but allow the service to fully use the extensions. It combines good security practices while supporting extensions.

There are lots of extension types that are not supported, why?

The main reason that the Oracle Utilities Cloud Services save costs is that there are inbuilt capabilities that replace work that was traditionally done on-premises. There are prebuilt and transparent configurations for the architecture that run the service and your extensions seamlessly. In those cases, operational based extensions are not needed as part of the transition and included as part of the service.

What is the PRIVILEGED threadpool?

The PRIVILEGED threadpool is a special threadpool optimized for processing Java code for deployment. This process is design only to run in this threadpool and this threadpool has been designed only for this process. It is highly recommended not to run any other process in this special threadpool as it is not designed for other processes and may cause failures.

Why do the processes require multiple steps?

The migration and maintenance processing has been designed to be as close as the same process as you are familiar for on-premises but take advantage of the cloud infrastructure. It provides flexibility, traceability, and efficiency in deploying custom java and custom tables reducing outage time, if needed. The multi-step process allows partners and customers assess and react to changes as they are being implemented.

Does it support multiple developers?

Yes, the process has been designed to be flexible enough for multiple developers to upload and add their changes to the service via the Adhoc Generation Request stage. Each developer should use a unique code identifier to delineate code. Each change is cumulative and added to the deployment as necessary. Overlaps in the scope of code identifiers will result in replacing overlaps with the latest upload. Only one developer should run Code Deployment Request to assemble the deployment.

Part II
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:

10

Data Conversion and Migration 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:

An Overview of Data Conversion and Migration

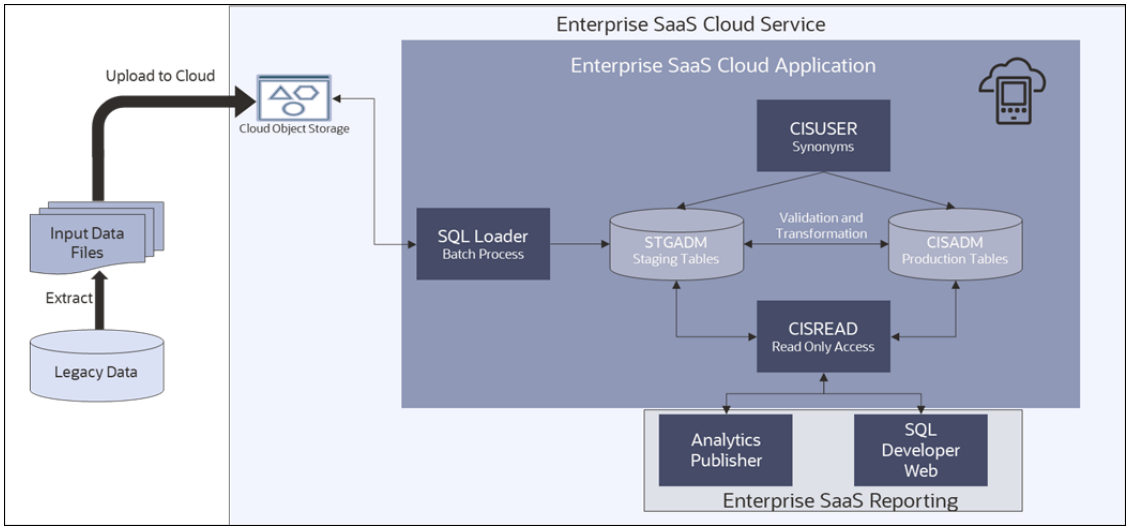
This topic provides an overview of data conversion and migration, including:

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.

Figure 10-1 Cloud Data Migration - High Level Process



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
- Create custom scheduler batch streams to orchestrate the data migration jobs with required dependencies
- Rehearse data upload and fine-tune configurations and/or legacy data extract, if needed
- Create reconciliation reports in Analytics 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 during project implementation is the best candidate for conversion.

In the newly provisioned instance, the staging area in the database is created according to application specifications. The Analytics Publisher instance and Oracle Database Actions / 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".

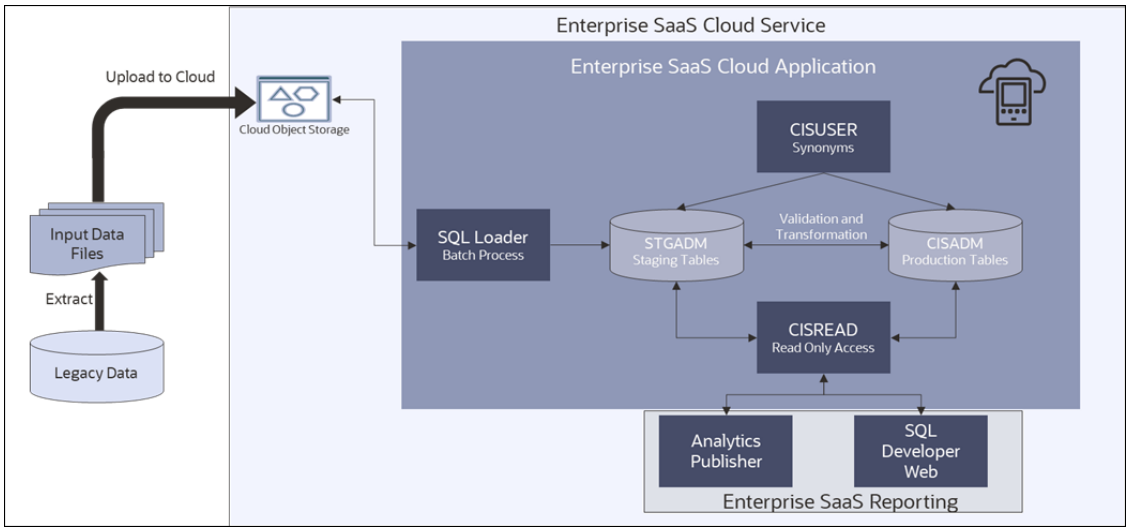
Data Conversion and Migration on Cloud

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

- Provided by Cloud Service Foundation
- Provided by Applications

The "SQL Loader Batch Process" 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.

Figure 10-2 Cloud Data Migration - High Level Process



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 user interface.
  - 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 Analytics Publisher that is connected to the database with read- only access to the staging and production schema tables.
- The instance of Oracle Database Actions 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.

Data Conversion and Migration - Terms and Definitions

The processes described in this section use the following terms:

Term	Definition
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.

Term	Definition
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

Data Conversion and Migration - Types of 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.

Data Conversion and Migration - 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.

Data Conversion and Migration - 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*

This document 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.

11

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 other chapters in the Data Conversion and Migration section of this guide for more detailed information about data conversion and migration.

This chapter includes:

- Data Conversion Approach
- Data Conversion and Migration Go-Live Checklist

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 other chapters in the Data Conversion and Migration section 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 other chapters in the Data Conversion and Migration section 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 other chapters in the Data Conversion and Migration section for more information.

Data Conversion and Migration Go-Live Checklist

The following items are things to consider as your implementation prepares for go-live:

- Make sure that up-to-date configuration has been applied to the Conversion environment.
- Make sure that no Object Validation / FK Validation errors exists (in both the Staging and Production schemas).
- Make sure each Customer Account opening financial balance is reconciled with the ending balance in legacy systems.
- Make sure overall account receivable balances are reconciled with the General Ledger.
- Make sure database indexes and statistics have been updated and gathered (in both the Staging and Production schemas).
- Make sure conversion is disabled in the Production environment.
- Make sure you have submitted a Clone service request to get the copy of your Production environment in any of your Test environments with the Cloud Operations team and the process has been completed before you start using your Production environment.
- Make sure activities that transpired during the cutover period (such as Meter Reads, Payments, Field activities, and so on) have been completed.
- Make sure access to the Switch Schema application service is limited to administrator users only.
- Refer to the Information Lifecycle Management Go-Live Checklist in the Information Lifecycle Management chapter.

12

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.

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

This chapter includes the following data conversion and migration scenarios:

- 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

> **Note**
>
> - The tables in the following topics 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.

Refer to **Initial Master Data Conversion** in the *Administrative User Guide* for information about batch processes that can be used when migrating data from Oracle Utilities Customer Care and Billing to Oracle Utilities Customer to Meter.

Legacy Customer Information System to C2MO

Customer Care and Billing Data		Meter Data Management Data		Operational Device Data	
Person	Upload from Legacy CIS	Contact	Migrate Person		
Account	Upload from Legacy CIS				
Service Agreement (SA)	Upload from Legacy CIS	Usage Subscription (US)	Migrate Service Agreement		
Premise	Upload from Legacy CIS				
Service Point (SP)	Upload from Legacy CIS	Service Point (SP)	Migrate Service Point	Node	Included
Meter	Upload from Legacy CIS	Device	Migrate Meter	Asset	Included
Item	Upload from Legacy CIS	Device	Migrate Item	Asset	Included
Service Point Meter History	Upload from Legacy CIS	Install Event	Migrate Service Point Meter History	Asset Node	Included
Service Point Item History	Upload from Legacy CIS	Install Event	Migrate Service Point Item History	Asset Node	Included

Legacy Customer Information System to C2M

Customer Care and Billing Data		Meter Data Management Data	
Person	Upload from Legacy CIS	Contact	Migrate Person
Account	Upload from Legacy CIS		
Service Agreement (SA)	Upload from Legacy CIS	Usage Subscription (US)	Migrate Service Agreement
Premise	Upload from Legacy CIS		
Service Point (SP)	Upload from Legacy CIS	Service Point (SP)	Migrate Service Point
Meter	Upload from Legacy CIS	Device	Migrate Meter
Item	Upload from Legacy CIS	Device	Migrate Item
Service Point Meter History	Upload from Legacy CIS	Install Event	Migrate Service Point Meter History
Service Point Item History	Upload from Legacy CIS	Install Event	Migrate Service Point Item History

Legacy Meter Data Management to MDM/ODM

Meter Data Management Data		Operational Device Management Data	
Contact	Upload from Legacy MDM		

Meter Data Management Data		Operational Device Management Data	
Usage Subscription (US)	Upload from Legacy MDM		
Service Point (SP)	Upload from Legacy MDM	Node	Synchronize from MDM
Device	Upload from Legacy MDMBind Device IDs	Asset	Upload from Legacy MDMBind 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	Migrate Person
Account	Import from CCB		
Service Agreement (SA)	Import from CCB	Usage Subscription (US)	Migrate Service Agreement
Premise	Import from CCB		
Service Point (SP)	Import from CCB	Service Point (SP)	Migrate Service Point
Meter	Import from CCB	Device	Migrate Meter
Item	Import from CCB	Device	Migrate Item
Service Point Meter History	Import from CCB	Install Event	Migrate Service Point Meter History
Service Point Item History	Import from CCB	Install Event	Migrate Service Point Item History

Customer Care and Billing to C2MO

Customer Care and Billing Data		Meter Data Management Data		Operational Device Management Data	
Person	Import from CCB	Contact	Migrate Person		
Account	Import from CCB				
Service Agreement (SA)	Import from CCB	Usage Subscription (US)	Migrate Service Agreement		
Premise	Import from CCB				
Service Point (SP)	Import from CCB	Service Point (SP)	Migrate Service Point	Node	Included
Meter	Import from CCB	Device	Migrate Meter	Asset	Included
Item	Import from CCB	Device	Migrate Item	Asset	Included
Service Point Meter History	Import from CCB	Install Event	Migrate Service Point Meter History	Asset Node	Included

Customer Care and Billing Data		Meter Data Management Data		Operational Device Management Data	
Service Point Item History	Import from CCB	Install Event	Migrate Service Point Item History	Asset Node	Included

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

Note

Customers are required to update the **Business Object** field in the following tables:

- D1_SP: from "D1-ServicePoint" to "X1D-ServicePoint"
- D1_SP_TYPE: from "D1-ServicePointType" to "X1D-ServicePointType"

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

Customer Care and Billing Data		Meter Data Management Data		Operational Device Management Data	
		Device Configuration	Import from MDM		
		Measuring Component	Import from MDM		
		Install Event	Import from MDM	Asset Node	Import from ODM

> **Note**
>
> Customers are required to update the **Business Object** field in the following tables:
>
> - D1_SP: from "D1-ServicePoint" to "X1D-ServicePoint"
> - D1_SP_TYPE: from "D1-ServicePointType" to "X1D-ServicePointType"

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

> **Note**
>
> Customers are required to update the **Business Object** field in the following tables:
>
> - D1_SP: from "D1-ServicePoint" to "X1D-ServicePoint"
> - D1_SP_TYPE: from "D1-ServicePointType" to "X1D-ServicePointType"

13

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_PERCI_PER _NAMECI_PER_ID ….etcData 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 1234, IND, Doe,… CI_PER 5678, IND, Moon,...CI_PER 9063, BUS, ABC Corp,..CI_PER _ID 1234, SSN,72346781CI_PER _ID 5678, SSN, 87635241CI_PER _ID 9063, EIN, 09182835CI_PER _ID 9063, TID, 82528555CI_PER _NAME 1234, Doe, MaryCI_PER _NAME 5678, Moon, Barry

CLOB Data in a Secondary File

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

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

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

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

Multiple Data Files for Single Table or MO Upload

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

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

14

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

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 Customer Cloud Service

In this scenario a new customer using a legacy CIS application is upgrading to Oracle Utilities Customer Cloud Service. The steps outlined below apply to Master data only.

Step	Description	Remarks
1	Transform legacy person, account, premise, service point and service agreement to CC&B person, account, premise, service point and service agreement.	This step is performed outside of the cloud service.
2	Transform legacy meter/item, meter/item configuration, and register to CC&B meter/item, meter/item configuration, and register.	This step is performed outside of the cloud service.
3	Transform legacy meter/item history to CC&B SP meter/item history.	This step is performed outside of the cloud service.
4	Extract files based on the CC&B master entities (person, account, premise, service point, service agreement, meter/items, and meter/item history).	This step is performed outside of the cloud service.
5	Upload master data files into Oracle Object Storage.	This step is performed outside of the cloud service.
6	Run the SQL Loader batch process to load data from the master data files into the staging schema objects.	
7	Generate Keys, perform Validations, and insert data into the production schema.	
8	Perform validation in the production schema	
9	Run the Admin migration (X1-MIGAD) batch process	
10	Migrate CC&B Person, Service Point, and Service Agreement, to MDM Contact, Service Point, and Usage Subscription	
11	Migrate CC&B Meter/Item (including meter/item configuration and register) to MDM Device/ODM Asset	
12	Migrate CC&B SP Meter/Item History to MDM Install Event/ODM Asset Node	

Customer Care and Billing Upgrading to Oracle Utilities Customer Cloud Service

In this scenario an existing Customer Care and Billing customer is upgrading to Oracle Utilities Customer Cloud Service. The steps outlined below apply to Master data only.

Step	Description	Remarks
1	Export CC&B database to a single instance database.	This step is performed outside of the cloud service.

Step	Description	Remarks
2	Perform database upgrade to match your cloud service supported version.	This step is performed outside of the cloud service.
3	Extract master data files based on CC&B master entities from the upgrade database, and upload files to Oracle Cloud ObjectStorage.	This step is performed outside of the cloud service.
4	Run the SQL Loader batch process to load data from the master data files into the production schema.	
5	Perform validation in the production schema	
6	Run the Admin migration (X1-MIGAD) batch process	
7	Migrate CC&B Person, Service Point, and Service Agreement, to MDM Contact, Service Point, and Usage Subscription	
8	Migrate CC&B Meter/Item (including meter/item configuration and register) to MDM Device/ODM Asset	
9	Migrate CC&B SP Meter/Item History to MDM Install Event/ODM Asset Node	

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

In this scenario an existing customer using integrated CC&B and MDM applications is upgrading to Oracle Utilities Customer Cloud Service. The steps outlined below apply to Master data only.

Step	Description	Remarks
1	Export CC&B database to a single-instance database.	This step is performed outside of the cloud service.
2	Perform database upgrade to match your cloud service supported version.	This step is performed outside of the cloud service.
3	Export MDM database to a single-instance database.	This step is performed outside of the cloud service.
4	Perform database upgrade to match your cloud service supported version.	This step is performed outside of the cloud service.
5	Extract master data files based on CC&B master entities from the upgrade database, and upload files to oracle cloud object storage.	

Step	Description	Remarks
6	Extract master data files based on MDM master entities from the upgrade database, and upload files to oracle cloud object storage.	
7	Run the SQL Loader batch process to load data from the master data files (CC&B and MDM) into production schema	
8	Update the **Business Object** field in the following tables:•D1_SP: from "D1-ServicePoint" to "X1D-ServicePoint" object.•D1_SP_TYPE: “D1-ServicePointType” to “X1D-ServicePointType”.	
9	Perform validations in the production schema	
10	Convert MDM meters to ODM assets.	The following steps only apply if ODM functionality is included.
11	Synchronize MDM service points to ODM service points.	
12	Synchronize MDM install events to ODM asset dispositions.	
13	Create the MDM device identifiers from ODM asset identifiers.	

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

In this scenario an existing customer using integrated CC&B, MDM and ODM applications is upgrading to Oracle Utilities Customer Cloud Service. The steps outlined below apply to Master data only.

Step	Description	Remarks
1	Export CC&B database to a single-instance database.	This step is performed outside of the cloud service.
2	Perform database upgrade to match your cloud service supported version.	This step is performed outside of the cloud service.
3	Export MDM database to a single-instance database.	This step is performed outside of the cloud service.
4	Perform database upgrade to match your cloud service supported version.	This step is performed outside of the cloud service.
5	Export ODM database to a single-instance database.	This step is performed outside of the cloud service.
6	Perform database upgrade to match your cloud service supported version.	This step is performed outside of the cloud service.

Step	Description	Remarks
7	Extract master data files based on CC&B master entities from the upgrade database, and upload files to oracle cloud object storage.	
8	Extract master data files based on MDM master entities from the upgrade database, and upload files to oracle cloud object storage.	
9	Extract master data files based on ODM master entities from the upgrade database, and upload files to oracle cloud object storage.	
10	Run the SQL Loader batch process to load data from the master data files (CC&B, MDM, and ODM) into production schema	
11	Update the **Business Object** field in the following tables:•D1_SP: from "D1-ServicePoint" to "X1D-ServicePoint" object.•D1_SP_TYPE: “D1-ServicePointType” to “X1D-ServicePointType”.	
12	Perform validations in the production schema	

Customer Care and Billing Upgrading to Oracle Utilities Customer Care and Billing Cloud Service

In this scenario an existing customer using CC&B applications is upgrading to Oracle Utilities Customer Care and Billing Cloud Service. The steps outlined below apply to Master data only.

Step	Description	Remarks
1	Export CC&B database to a single-instance database.	This step is performed outside of the cloud service.
2	Perform database upgrade to match your cloud service supported version.	This step is performed outside of the cloud service.
3	Extract master and transactional data files based on the CCB entities from the upgraded database and upload files to Oracle Cloud Object Storage.	
4	Run the SQL Loader batch process to load data from the master and transactional data files into the production schema objects.	

Step	Description	Remarks
5	Perform validation in the production schema	

Customer Care and Billing and Oracle Utilities Meter Solution Cloud Service Upgrading to Oracle Utilities Customer Cloud Service

In this scenario an existing customer using CC&B and MSCS applications is upgrading to Oracle Utilities Customer Cloud Service (CCS). The steps outlined below apply to Master data only.

Step	Description	Remarks
1	Export CC&B database to a single-instance database.	This step is performed outside of the cloud service.
2	Perform database upgrade to match your cloud service supported version.	This step is performed outside of the cloud service.
3	Take a clone of the MSCS Production environment and migrate it to CCS	This step is performed by Oracle Cloud Operations via a Service Request.
4	Review and Create missing CC&B-related table partitions and migrated the CC&B configuration to CCS.	
5	Extract master and transactional data files based on the CC&B entities from the upgraded database and upload files to Oracle Cloud Object Storage.	
6	Run the SQL Loader batch process to load data from the master and transactional data files into the production schema objects.	
7	Perform validation in the production schema	
8	Run the Admin Migration (X1-MIGAD) batch process.	
9	Migrate the CC&B Person, Service Point, and Service Agreement to MDM Contact, Service Point, and Usage Subscription.	
10	Migrate CC&B Meter/Item (including Meter/Item configuration and register data) to MDM Device/ODM Asset.	
11	Migrate CC&B SP Meter/Item History to MDM Install Event / ODM Asset Node.	

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. The steps outlined below apply to Master data only.

Step	Description	Remarks
1	Transform legacy contacts, service points, and usage subscriptions to MDM contacts, service points, and usage subscriptions.	This step is performed outside of the cloud service.
2	Transform legacy devices, device configurations, measuring components to MDM devices, device configurations, and measuring components.	This step is performed outside of the cloud service.
3	Transform legacy install events to MDM install events.	This step is performed outside of the cloud service.
4	Extract master data files based on the MDM master entities (contacts, service points, usage subscriptions, device, device configuration, measuring components and install events)	
5	Upload master data files into Oracle Cloud Object Storage	
6	Run the SQL Loader batch process to load data from the master data files into the staging schema objects.	
7	Generate Keys, perform Validations, and insert data into the production schema.	
8	Perform validation in the production schema	
9	Convert legacy meters to ODM assets.	The following steps only apply if ODM functionality is included.
10	Bind the MDM devices to ODM assets.	See Binding Meter Data Management Device IDs and Operational Device Management Asset IDs.
11	Bind the ODM assets to MDM devices.	See Binding Meter Data Management Device IDs and Operational Device Management Asset IDs.
12	Synchronize MDM service points to ODM service points.	

Step	Description	Remarks
13	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. The steps outlined below apply to Master data only.

Step	Description	Remarks
1	Export MDM database to a single-instance database.	This step is performed outside of the cloud service.
2	Perform database upgrade to match your cloud service supported version.	This step is performed outside of the cloud service.
3	Extract master data files based on the MDM master entities (contacts, service points, usage subscriptions, device, device configuration, measuring components and install events).	
4	Upload master data files into Oracle Cloud Object Storage	
5	Run the SQL Loader batch process to load data from the master data files into the production schema objects.	
6	Perform validation in the production schema	
7	Convert MDM meters to ODM assets.	The following steps only apply if ODM functionality is included.
8	Create MDM device identifiers from ODM asset identifiers.	
9	Synchronize MDM service point to ODM service point.	
10	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. The steps outlined below apply to Master data only.

Step	Description	Remarks
1	Export MDM database to a single-instance database.	This step is performed outside of the cloud service.
2	Perform database upgrade to match your cloud service supported version.	This step is performed outside of the cloud service.
3	Export ODM database to a single-instance database.	This step is performed outside of the cloud service.

Step	Description	Remarks
4	Perform database upgrade to match your cloud service supported version.	This step is performed outside of the cloud service.
5	Extract master data files based on the MDM master entities (contacts, service points, usage subscriptions, device, device configuration, measuring components and install events).	
6	Extract master data files based on the ODM master entities (Asset, Asset Node)	
7	Upload master data files for MDM and ODM into Oracle Cloud Object Storage	
8	Run the SQL Loader batch process to load data from the master data files for MDM and ODM into the production schema objects.	
9	Perform validation in the production schema	

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

Step	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_EVT(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_EVT(D1IE000K)
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 DataValidation	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)

Step	Contact	Device	Device Configuration	Measuring Component	Service Point	Usage Subscription	Install Event
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
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
X1-IDATD	Populate Device Identifier from Asset

Asset Conversion

Conversion Sequence

Step	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 From Legacy CIS and/or Meter Data Management

Migrating measurement or meter data to 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.

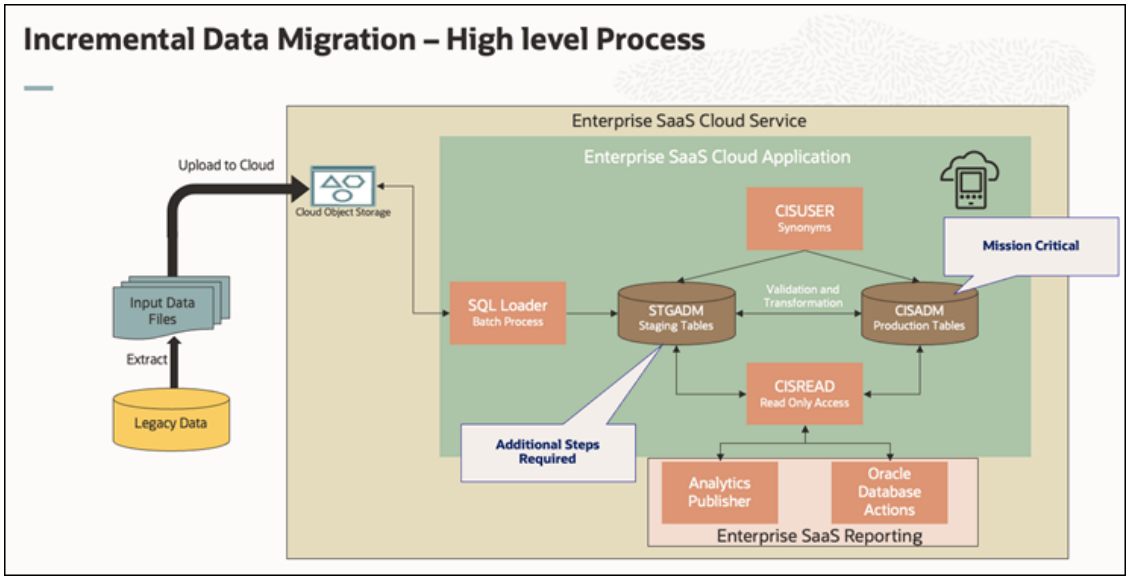
Incremental Conversion

Incremental data conversion enables the controlled migration of additional (delta) data into a live production environment after the initial data conversion has been completed. This approach is typically used when new records must be introduced without disrupting existing operations.

Because production environments contain active business data, incremental conversion is executed with strict safeguards, governance, and planning.

The following diagram provides a high-level overview of this process.

Figure 14-1 Incremental Data Migration - High Leve Process



This section outlines some common use cases.

Use Case 1: Adding Data to a Single Table (With Existing References)

Scenario: New records need to be added to a table (Table 1), where:

- The table has relationships (foreign keys) to another table (Table 2)
- Table 2 data is already fully converted and exists in production
- No new records are required in Table 2

Key Characteristics:

- Only one table receives new data
- All references point to already available production data
- No changes to existing records

Required Steps:

1. Preparation
   - Identify new (delta) records for Table 1
   - Ensure all foreign key references exist in production (Table 2)
   - Perform required configuration updates (if any)
2. Staging Cleanup
   - Truncate staging version of Table 1 and its key mapping
3. Data Preparation
   - Prepare clean data files containing only new records
4. Key Generation
   - Generate new system keys
   - Use controlled parameters (such as startRowNumber offset) to avoid duplicates

5. Data Load
    - Load data into staging
    - Insert into production

Use Case 2: Adding Data Across Multiple Related Tables

Scenario: New data must be added to multiple related tables (Table 1 and Table 2), where:

- Both tables contain new records
- Relationships exist between them
- Keys are system-generated and must remain consistent

Key Characteristics:

- Multiple tables updated together
- Requires synchronized key generation
- More complex dependency management

Required Steps:

1. Preparation
    - Identify delta records for both tables
    - Prepare data maintaining relationships
2. Backup Existing Key Mapping
    - Extract key mapping data from staging
    - Store safely for rollback
3. Staging Cleanup
    - Truncate staging tables and key mapping tables
4. Key Generation
    - Generate keys for both tables
    - Ensure sequencing avoids overlap with production
5. Data Load
    - Load new records into staging
    - Insert into production
6. Restore Key Mapping
    - Reload previous key mappings (append mode)
    - Maintain continuity of historical mappings

Important Functional Limitation Across Both Use Cases

- Only new data insertion is supported
- No updates or deletions of existing production data are allowed
- This ensures stability and protects production integrity

Keys to Success for Incremental Conversion

Successful incremental conversion in a production environment depends on careful planning, strong governance, and disciplined execution. The following factors are critical to ensuring a smooth and risk-free process:

1. Detailed Planning and Design
   - Define a granular execution plan covering each step of the conversion
   - Clearly identify:
     - Data scope (delta records only)
     - Dependencies between tables
     - Sequence of execution
   - Review and validate the plan with all stakeholders before execution
2. Strong Data Validation
   - Ensure all foreign key relationships are valid
   - Validate that:
     - No duplicate records are introduced
     - Data aligns with production standards
   - Perform pre- and post-load reconciliation
3. Robust Key Management
   - Use controlled key generation strategies to avoid overlaps
   - Maintain accurate key mapping tables
   - Always back up key mappings before execution
4. Mock Runs in Production-Like Environments
   - Conduct multiple dry runs in environments similar to production
   - Validate:
     - Performance
     - Data integrity
     - End-to-end execution timing
   - Use discoveries to refine the final approach
5. Performance and Capacity Planning
   - Reassess system capacity and sizing based on incremental data volume
   - Monitor:
     - Batch processing times
     - System load
   - Optimize indexes and database performance as needed
6. Controlled Execution Window
   - Schedule activities during a planned maintenance window (if required)
   - Minimize impact on business operations

- Ensure all teams are aligned and available during execution

7. Backup and Rollback Readiness
   - Take complete backups before starting conversion
   - Define a clear rollback strategy
   - Ensure ability to restore:
     - Key mappings
     - Affected datasets
8. Governance and Stakeholder Coordination
   - Ensure close coordination across:
     - Project teams
     - Technical teams
     - Operations teams
   - Maintain clear communication before, during, and after execution
9. Controlled Enablement of Conversion Mode
   - Enable conversion capabilities only when required
   - Disable immediately after completion
   - Prevent accidental or unauthorized execution
10. Experienced Execution Team
    - Engage a skilled data migration team with experience in:
      - Production data handling
      - Conversion tools and processes
    - Proactively manage risks such as:
      - Duplicate key generation
      - Index inconsistencies
      - Data integrity issues

15

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-CNVGA 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.

> **Note**
>
> - 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 purged.

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
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.CRE_DTTM

Table Name	ILM DT Initial Load
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
CI_FT	CI_FT.CRE_DTTM
CI_APPR_REQ	CI_ADJ.ILM_DT only when CI_ADJ.ILM_ARCH_SW='Y'

16

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

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 an 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

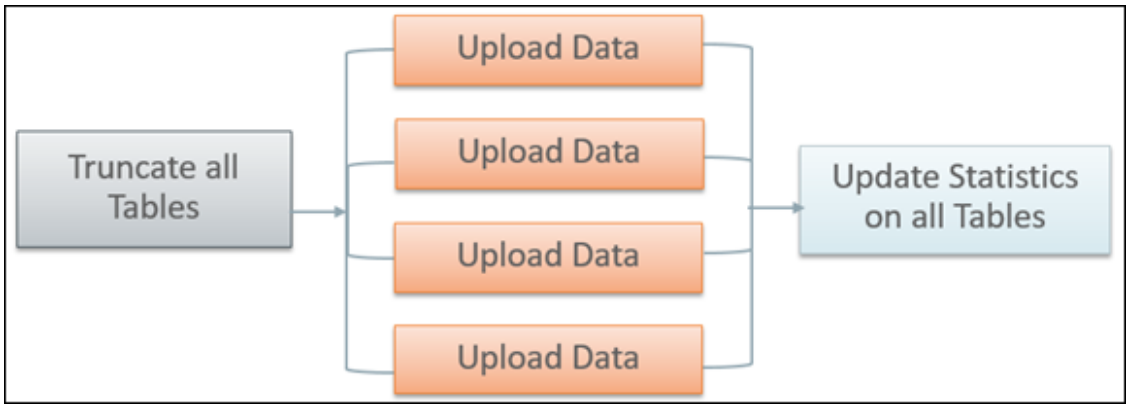
The SQL Loader is running in multiple threads and therefore it is not performing table truncation before loading (command APPEND). Hence, the target tables should be truncated prior to the load. For better performance the indexes have to be disabled before the load and re-enabled/statistics updated after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

Single Table Upload

Figure 16-1 Single Table Upload

Multiple Table or Maintenance Object Upload

Figure 16-2 Multiple Table or Maintenance Object Upload

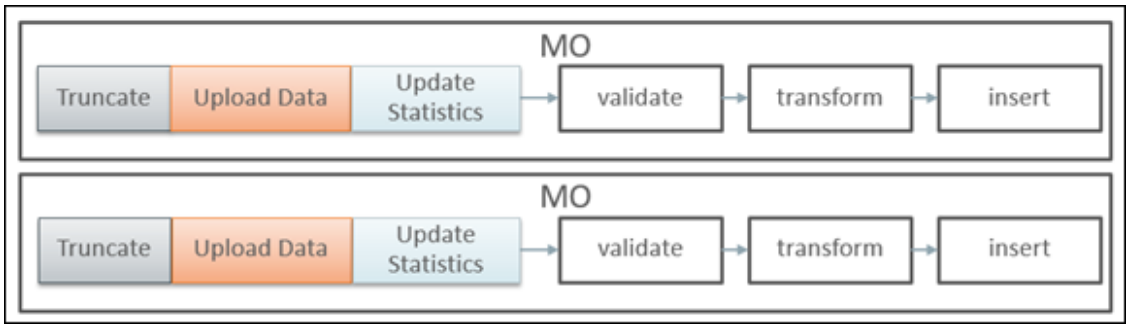


There are multiple strategies to orchestrate the entire conversion run and to build the optimal sequence of the conversion processes. Below are some of the many possibilities:

- 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

Figure 16-3 Full Conversion Chain per MO, Parallel Run



Upload All + Subsequent Validate/Transform MOs

Figure 16-4 Upload All + Subsequent Validate/Transform MOs

17

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 maintenance objects, create new Conversion Instruction (Conversion Task Type) and add the Override Instruction for Table or MO on *Conversion Data Upload* Master Configuration
- To make a non-converted table eligible for conversion, add it to the Override Conversion Eligibility list on *Conversion Data Upload* Master Configuration

> **Note**
>
> 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.

Issue	Details
Provide input data files with CLOB data IF NECESSARY.	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.**
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 don'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.
Using Batch Schedulers	Avoid running adhoc jobs Use the SaaS or an external batch scheduler to orchestrate the batches. Do not use template batch controls; use custom batch controls for specific tasks.

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 theK1-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 III
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:

- Object Storage Connection Management
- File Export Sample Implementation
- File Import Sample Implementation

18

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 Administration 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 Administration 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 Administration* 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 Key**: The region of the object storage tenancy for the connection (Values for this field are defined in the F1_REGION_FLG lookup.
  - **Bucket Name Prefix**: The Prefix used by Object Storage Bucket configuration
  - **Reporting Configuration**: Checkbox that enables the Delivery Channel Buckets section, where you can define the list of buckets configured in this compartment that are available for selection as delivery channels in external reporting tools.

Refer to **External File Storage** in the *Oracle Utilities Application Framework Administrative User Guide* and the *Oracle Utilities Cloud Services Administration 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.

19

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

The export process requires authentication and communication between the Oracle Utilities cloud service and Object Storage. See 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.

20

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 **Uploading Records** in the *Oracle Utilities Application Framework Administrative User Guide*.

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:

Figure 20-1 Sample Import File

SampFlatFileUpload6.txt

```
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 the Batch Control - File Upload Algorithm Entity:

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 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 IV
Oracle REST Data Services

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

- Oracle Database Actions
- REST APIs

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

21

Oracle Database Actions

Oracle Database Actions (formerly known as 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 Database Actions 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 Database Actions.

Users must be assigned to the "SQL Developer Web Online User" and "REST Enabled SQL" application roles in order to use Oracle Database Actions 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.

> **Note**
>
> Oracle Database Actions allows customers to save SQL Statements via Worksheets. However version upgrades and/or clearing the browser cache will delete any saved worksheets, so we recommend that you save all your worksheets outside the browser on your local devices.

22
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.

OAuth-based authorization is also supported. You may create a dedicated OAuth Client for the access to Oracle REST Data Services REST API via self-service. See the *Oracle Utilities Cloud Services Administration Guide* for more information.

The service runs the SQL statements against Oracle Database and returns the result to the client in a JSON format. You can use the HTTPS POST method to access the REST- Enabled SQL service.

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 Database Actions and REST service URLs.

No additional configuration is required to use Oracle Database Actions or REST services.

Use of this feature is subject to the following:

- Basic Authentication and OAuth are supported
- Read Only Access
- Supports Data Access Language (DAL) that includes retrieval of Performance Hub Report and so on
- Only supports retrieval of small sets of data
- SQL Query timeout limit is 5 minutes
- Output will be in JSON format

Part V
Product-Specific Integrations

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

- Customer Cloud Service Receipt Printing

23

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:

- Application Configuration
- Printer Installation
- Recommended Printer Preferences

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

Application 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 - 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 UI maps mentioned in the table above 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 the 2.9.0.0.0/22A 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+11
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

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 VI
Web Services

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

- Web Services in Oracle Utilities Cloud Services

24

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

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:

Parameter	Description
`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:

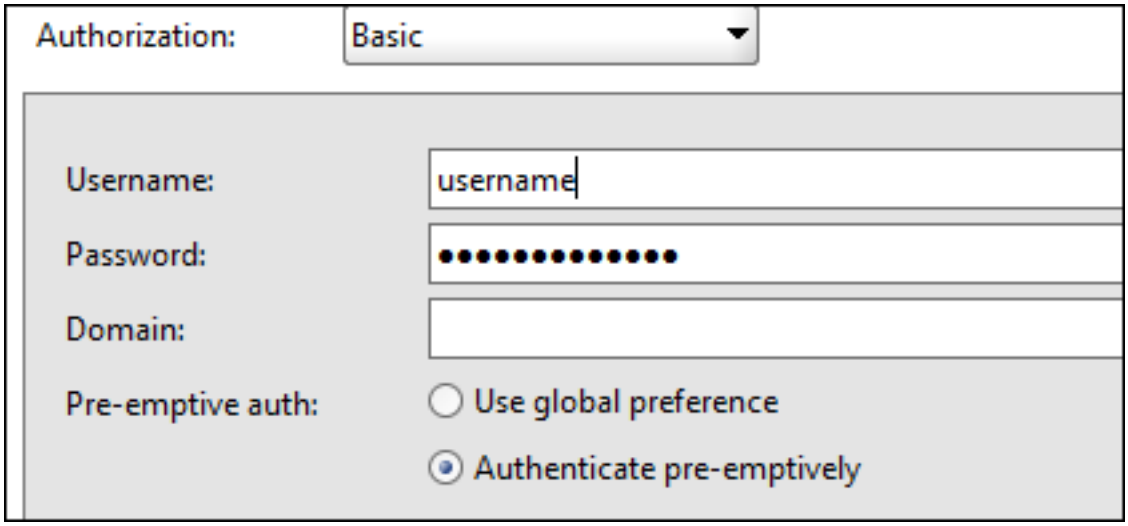
```
https://{host}:{port}/{tenant}/{domain}/{appName}/soap/api/iws/IWSServiceName
```

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.

Figure 24-1 Authorization Dialog



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

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}`

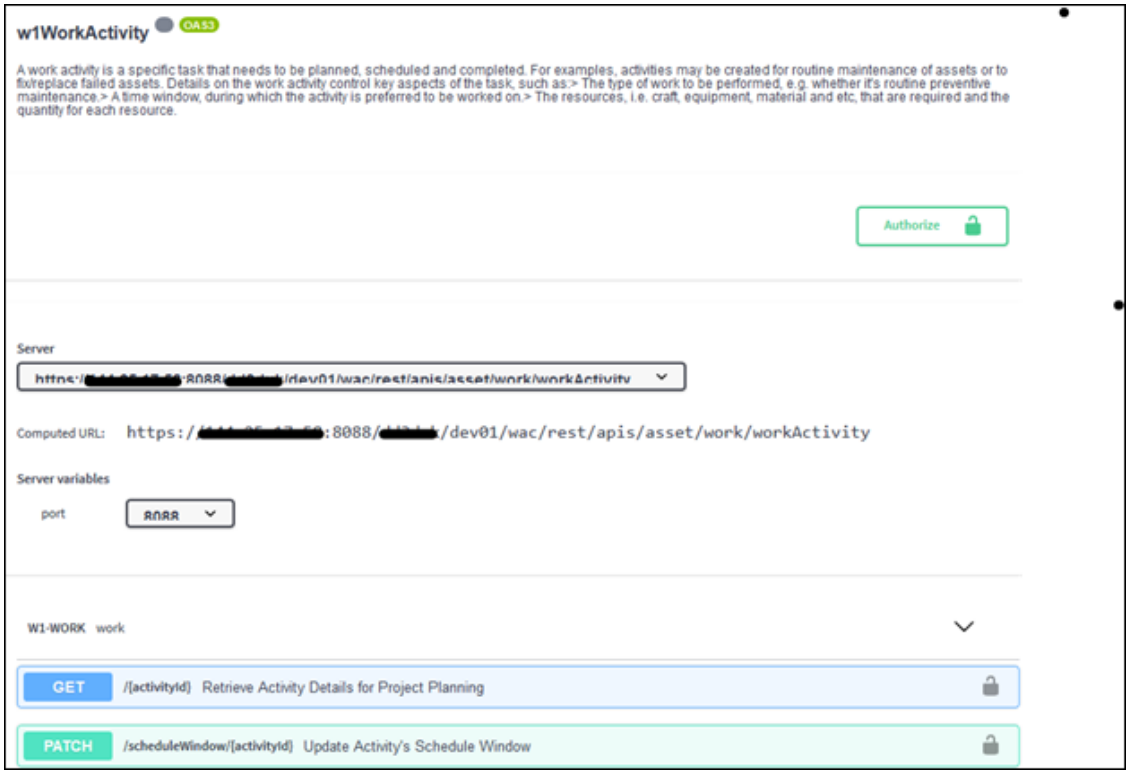
where:

Component	Description
`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:

Figure 24-2 URI Components



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:

Component	Description
`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`

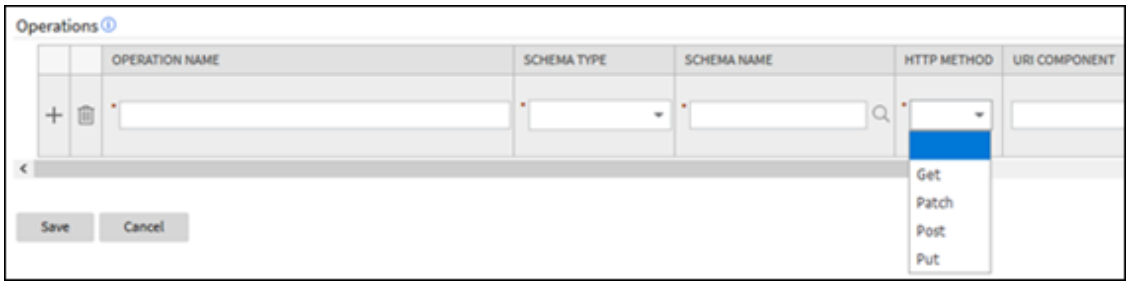
Methods and Parameters

REST inbound web services support the following HTTP Methods:

- Get
- Patch
- Post
- Put

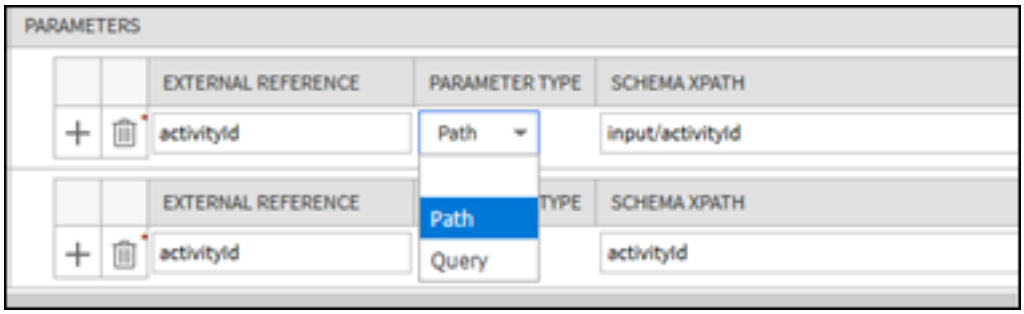
The method is defined in the **HTTP Method** drop-down list in the **Operations** section of the **Inbound Web Service** portal.

Figure 24-3 Inbound Web Service - Operations



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

Figure 24-4 Inbound Web Services - Operations Parameters



- **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.

Outbound Messages 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

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.

Outbound Messages 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@RESTEndpointU RL 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, and so on.
- 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.

Glossary

Index