Oracle Insurance Data Foundation Integration With Accounting Hub Cloud Service

User Guide

Release 8.0.7.0.0

September 2020

ORACLE

Oracle Insurance Data Foundation Integration With Accounting Hub Cloud Service

Document Control

Version Number	Revision Date	Change Log
01	February 2019	Updated
02	April 2019	Updated EDS section with Dynamic Principal name and Driver fields. Updated ODI setup validation settings
03	March 2020	Correction to Integration Configuration steps
04	September 2020	Updated section GL and SR Balance Information as part of 8.0.7.3.0 release

Table of Contents

1 About the Guide

This section provides a brief description of the scope, the audience, the references, the organization of the user guide and conventions incorporated into the user guide. The topics in this section are organized as follows:

- Scope of the guide
- Intended Audience
- Prerequisites
- Documentation Accessibility
- Access to Oracle Support
- Related Information Sources
- Acronyms
- Reference

1.1 Scope of the Guide

This user guide explains the features and functions of Oracle Insurance Data Foundation (OIDF) Integration with AHCS. Setup and configuration of the system, pre-packaged business or functional content, technical integration aspects and guidelines on the usage of the system are explained in detail.

Note that additional setup and configuration needs to be performed in AHCS to support accounting rules, journals, and general ledger. Refer to Oracle ERP Cloud documentation on AHCS for details in this regard.

1.2 Intended Audience

This manual is intended for the following audience:

- Technologists: Technical specialists who deal with setup and configuration of the interface between Data Foundation and Accounting Hub Cloud Service, and those responsible for the care and maintenance of the OFSAA instance.
- Business Users: Functional specialists who deal with actuarial sciences, risk management, accounting or a combination of these, seeking to understand details of the interface between Data Foundation and Accounting Hub Cloud Service.

1.3 Prerequisites

- Refer to the DIH Installation Guide for details on OFSAA components, Oracle applications, and environment that constitute technical prerequisites to deploying and using OIDF Integration with AHCS.
- User credentials set up by an authorized user is required before you can access the solution through its user interface.

1.4 Documentation Accessibility

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

1.5 Access to Oracle Support

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

1.6 Related Information Sources

Along with this user manual, you can also refer to the following documents in OHC documentation Library:

- Oracle Financial Services Data Integration User Guide Release 8.0.7
- Oracle Financial Services Data Integration Application Pack Installation Guide Release 8.0.7
- Oracle Financials Cloud Implementing Accounting Hub Guide
- Oracle Financials Cloud Accounting Hub Best Practices Guide

1.7 Acronyms

Acronym	Description
ADI	Application Data Interface
AHCS	Accounting Hub Cloud Service
Apps	Application
DIH	Data Integration Hub
EDD	External Data Descriptor
GL	General Ledger
KM	Knowledge Module
ODI	Oracle Data Integrator
PT	Pass-Through
SR	Supporting References
SLA	Sub-Ledger Application / Subledger Application
UI	User Interface
UCM	Universal Content Manager

1.8 References

- SLA Coverage
- Sub ledgers and Configurations
- SLA Template Validations
- Extract Connectors Structure
- Insert Connectors Structure

- SLA Coverage
- Connectors Structure

2 Introduction

2.1 About Oracle Financial Service Analytical Applications

Oracle Financial Services Analytical Applications (OFSAA) enables financial institutions to:

- Measure and meet risk-adjusted performance objectives
- Cultivate a risk management culture through transparency
- Lower the costs of compliance and regulation
- Improve insight into customer behavior

OFSAA uses industry-leading analytical methods, shared data model and application architecture to enable integrated risk management, performance management, customer insight, and compliance management. OFSAA actively incorporates risk into decision making, enables to achieve a consistent view of performance, promote a transparent risk management culture, and provide pervasive intelligence.

OFSAA delivers a comprehensive, integrated suite of financial services analytical applications for both banking and insurance domains. It includes many applications such as Profitability Management, Asset Liability Management, Customer Insight, and Risk Management.

2.2 About Accounting Hub Cloud Service (AHCS)

AHCS is an accounting integration and reporting platform in oracle cloud that includes products, such as sub-ledger Accounting, Ledger, and Financial Reporting Center. AHC is an accounting integration platform. It standardizes the accounting from multiple third-party transactional systems to consistently enforce accounting policies and meet multiple reporting requirements in an automated and controlled fashion. AHC includes a rules repository to centrally define and maintain accounting rules, a rules transformation engine to create, validate and store the accounting journals, and a detailed accounting repository that is used to reconcile to the source system.

2.3 Objective

Integration of AHC and OFSAA provides a basis for a unified finance and risk architecture to the financial services industry. The benefits of having an out of the box interface between AHC and OFSAA are:

- The OFSAA repository of transaction and reference information becomes a single, unified analytical repository for risk and finance.
- The integration improves consistency in risk and finance data for producing risk-adjusted measures.
- OFSAA customers can take advantage of AHC rules and accounting engines to improve the efficiency and audit ability to transform transactions from financial services systems such as deposits, loan accruals, payments, trades, and withdrawals into accounting.
- Information in the OFSAA foundation such as customer, account, product, branch, and channel information is accessible in AHC, for defining accounting treatments in addition to generating accounting balances. Conditional logic can be used to vary accounting treatments based upon values from transactions and contract balances. The same OFSAA transactions and contract balances are used by AHC to book entries and generate financial balances, which in turn are reused by OFSAA for analytical processing.

- Using AHC balances for analytical processing facilitates the reconciliation of operational risk losses and accounting as prescribed by regulations such as Basel II and III. These accounting balances can be reconciled using the OFSAA Reconciliation Framework. Reconciliation rules can be defined to map AHC balances to product processor information. For example, accounting for provisions and losses can be based upon customer account balances, as provided to OFSAA and read by AHC enhancing the reliability of operational risk and provisioning measurement.

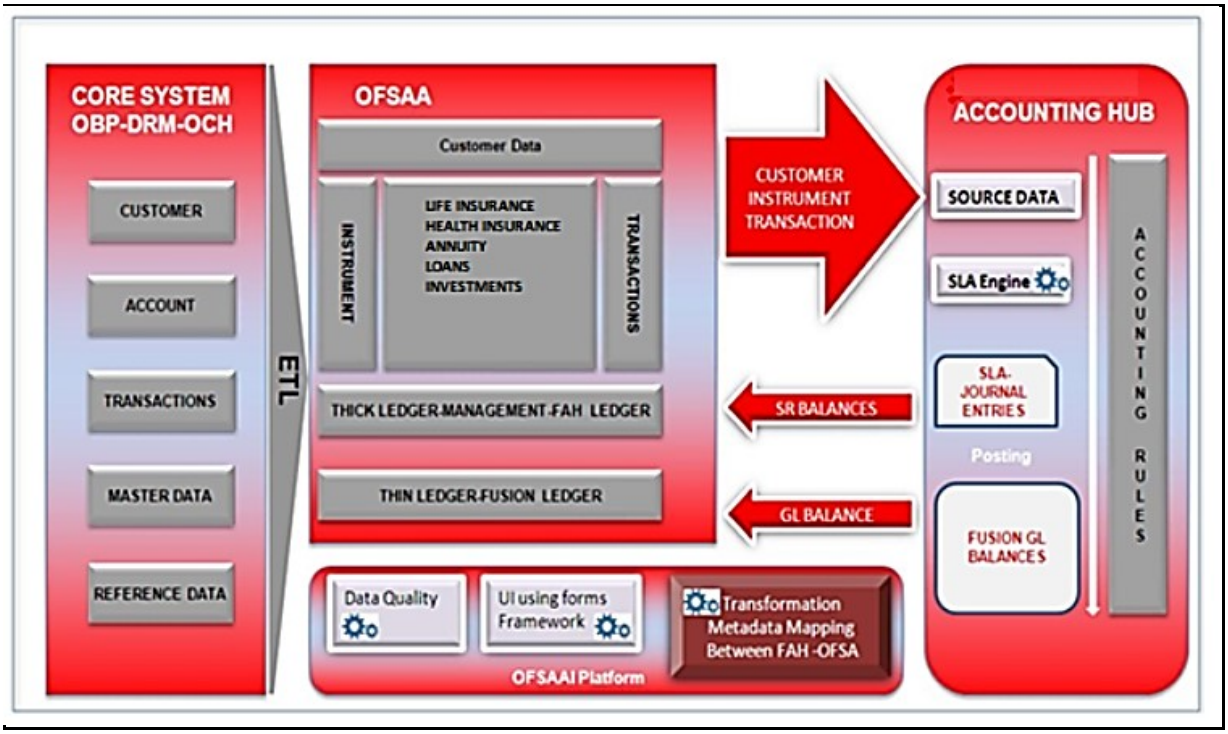
2.4 About Oracle Insurance Data Foundation Integration With Accounting Hub Cloud Service

The major components that can be leveraged from AHC are:

- Pre-Defined Subledger Transaction Object Models
  - Life Insurance
  - Property and Casualty
  - Health Insurance
  - Annuity Contracts
  - Retirement Contracts
  - Investments
  - Policy Loans
  - Policy Admin PT
  - Reinsurance Issued
  - Reinsurance Held
- Auto-generate AHC Subledger registration and transaction feed templates
- Pre-built automated transaction feed to Accounting Hub Cloud Service
- Wizard to customize Subledger Transaction Objects
- Pre-built automated GL Balances feed on Cloud GL to OFSAA.

2.5 OFSAA-AHC Interface Architecture Overview

The OFSAA-AHC Interface Architecture is illustrated in the following diagram:

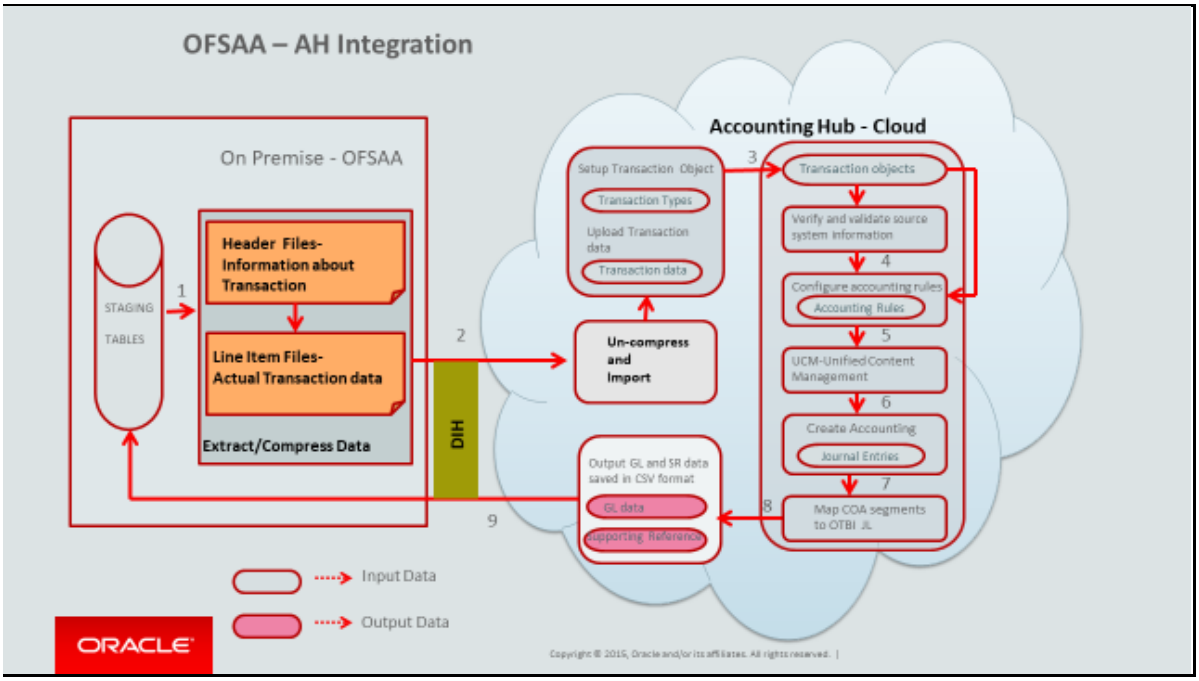


The data flow between OFSAA and AHC is bi-directional. OFSAA includes all the account, contract and transaction information that AHC needs to generate for its Subledger accounting. Additionally, General Ledger and supporting reference balances are required by the various products under the OFSAA suite.

Insurance data is extracted from OFSAA to AHC through DIH connectors (pre-packaged/custom) and is processed within the AHC using accounting rules. The updated GL balances are then extracted, to load General Ledger Balances and Supporting Reference Balances.

The flow from AHC to OFSAA is achieved through pre-built DIH connectors. These connectors extract the GL and SR Balances from AHC and load them into OFSAA staging post which, is available for all downstream applications.

2.6 OFSAA – AHC Integration Overview



The data flow between OFSAA and AHC takes place through files, as AHC is on the cloud.

The integration consists of the following processes:

1. Generate SLA template
2. Extract Data from OIDF for AHC using DIH Connectors
3. Load GL data from AHC using DIH Connectors

AHCS provides a macro-enabled SLA XL template for integrating third-party systems. This template is deployed as part of OIDF Integration with AHCS.

OIDF Integration with AHCS automates the population of the SLA XL template. Post-deployment you must select one of the pre-packaged SLA definitions in OIDF Integration with AHCS and click the Download Template button. OIDF Integration with AHCS makes a copy of SLA XL template with Source System details (which is the Sub Ledger details) and then populates with Transaction types. Header and Line definitions are pre-packaged in OIDF Integration with AHCS. The updated SLA XL template is then manually opened from Source Sheet. Click validate in the Source System sheet. Ensure to enable Marco before validating the template.

In case of errors, fix the SLA definitions in OIDF Integration with AHCS and regenerate the SLA XL template. Refer SLA XL instruction sheet for resolving the error. After the errors are fixed or if there are no errors, from the “source sheet” click the Generate Zip button. The macro generates a zip file with the same name and in the same location as SLA XL template. The zipped files are then manually uploaded into AHC. The file contains metadata of source system that has to be registered, along with the transaction object attributes and transaction types. This must be performed for each SLA defined in OIDF Integration with AHCS. Refer section Registering with AHCS for more details.

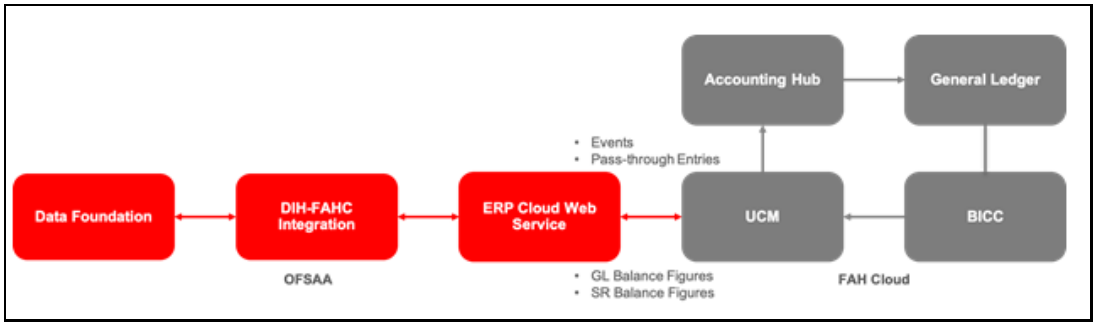
The uploaded zip files register source system, transaction types and define transaction objects (header information and line information) in AHC. AHC has to be configured as per GL accounting requirement.

Post AHC configuration, OFSAA batches are performed using pre-packaged AHC connectors, extract transaction data from OIDF stage tables in ASCII format. The output files are zipped and pushed to AHC through UCM (Universal Content Manager) and the general entries are posted.

On completion of AHC processing two output CSV files are created, one with GL balance and other supporting reference balances. OFSAA batch process is performed using pre-packaged insert connectors and loading data into STG_GL_DATA and STG_MANAGEMENT_LEDGER.

2.7 Data flow for OIDF Integration with AHCS

- Transactions and pass-through accounting information are collated into Data Foundation through DIH Foundation.
- If required, said data is standardized in Data Foundation.
- DIH – Accounting Hub integration extracts relevant data from Data Foundation and prepares ZIP file, with header and line details, and metadata.
- DIH – Accounting Hub integration invokes ERP Cloud web service over REST APIs to a byte-stream ZIP file to ERP Cloud Universal Content Manager.
- Refer https://docs.oracle.com/en/cloud/saas/financ ials/18b/faiac/oracle-accounting-hub-cloud-overview.html#FAIAC307871 for specifics on Accounting Hub Cloud Service.
- Extracts of GL and supporting reference balance figures from Accounting Hub Cloud Service are configured and scheduled in BI Cloud Connector console, as detailed here: https://docs.oracle.com/en/cloud/saas/applications-common/r13-update17d/biacc/biccc-overview.html
- GL and SR balance extracts so configured in BICC produce output files and put those in UCM.
- DIH – Accounting Hub integration invokes ERP Cloud web service over SOAP to byte-stream file produced by BICC above to OFSAA. Refer sections 4.3 and 9 in FAH User Guide for details.
- DIH – Accounting Hub integration loads GL and SR balance figures into appropriate Staging entities in Data Foundation as per mapping information detailed FAH User Guide.



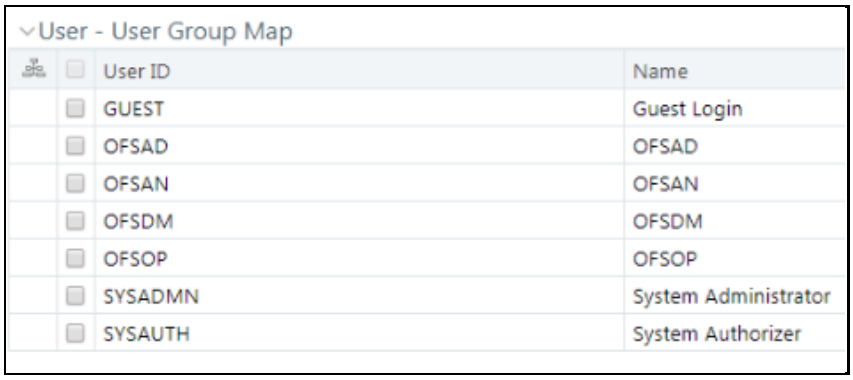
3 Mapping the OFSAA User to AHC User Groups

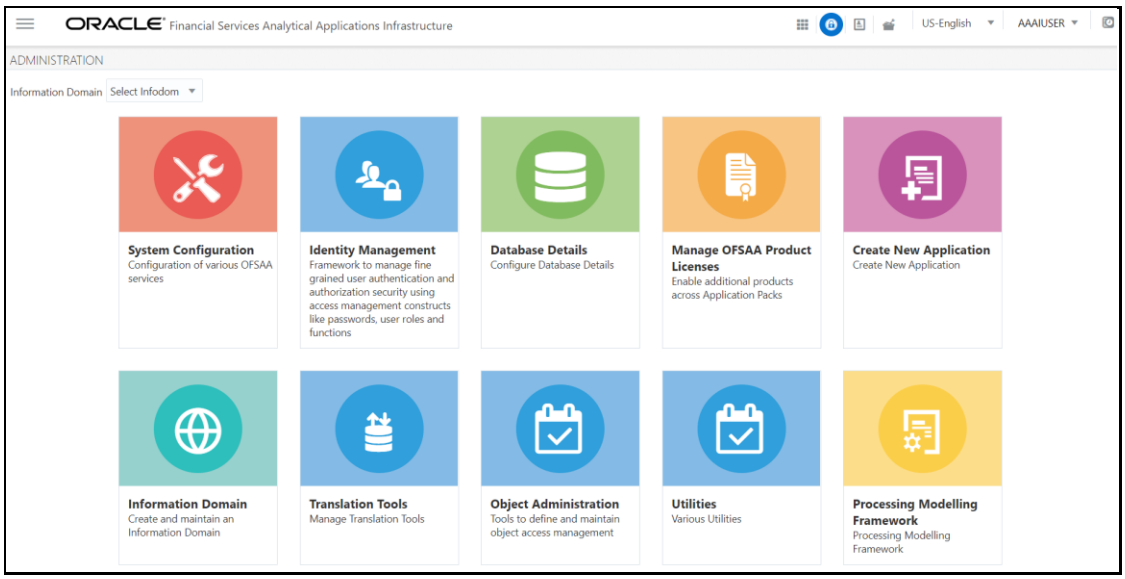
User- User Group Map enables you to map user(s) to a specific user group which in turn is mapped to a specific Information Domain and role. Every user group mapped to the infodom must be authorized. Else, it cannot be mapped to users.

User- User Group Map screen displays fields such as User ID, Name, and the corresponding User-User Group Map. You can view and modify the existing mappings within the User Group Maintenance screen.

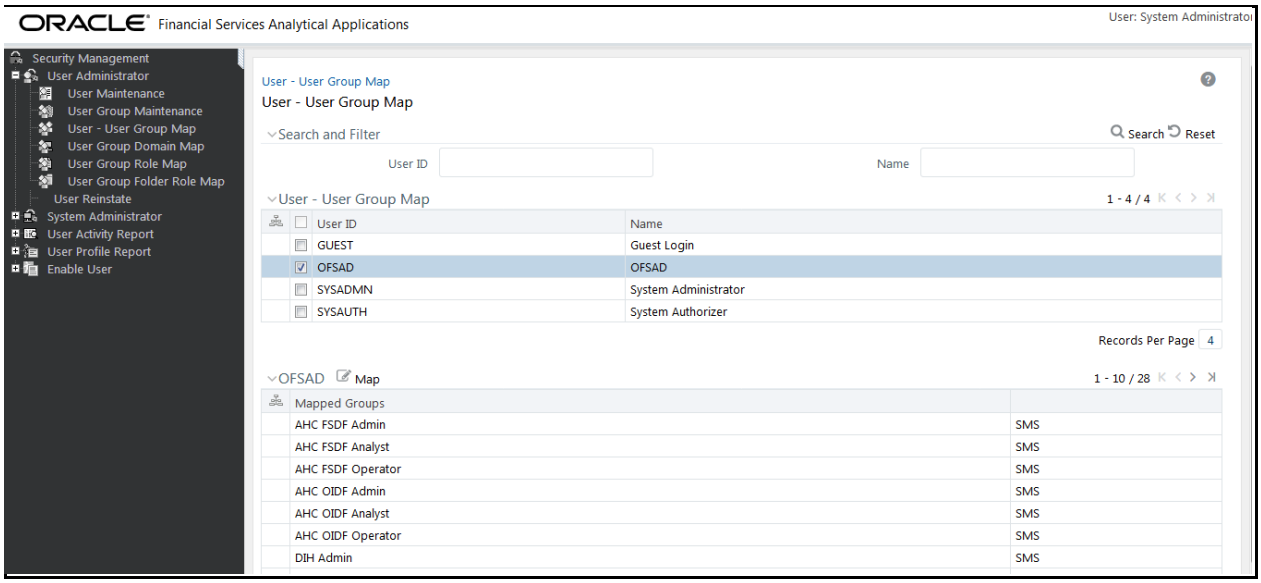
To access User- User Group Map navigate to Identity Management and click Security Management. For details on mapping user to user groups refer to OFSAAI User Guide in OHC documentation library.

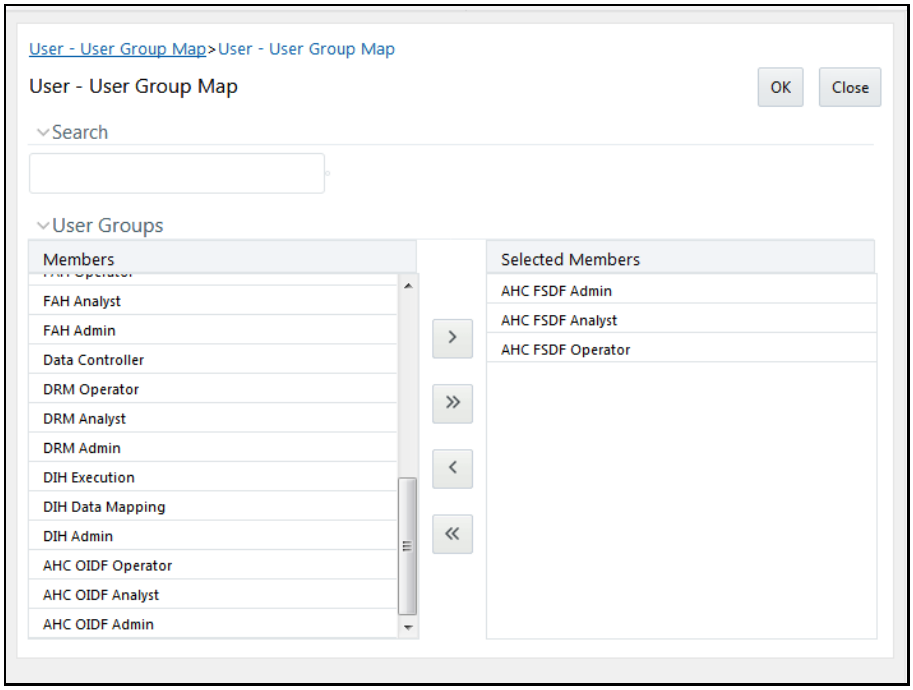
Pre-packaged User Groups for OFSAA - AHC Interface





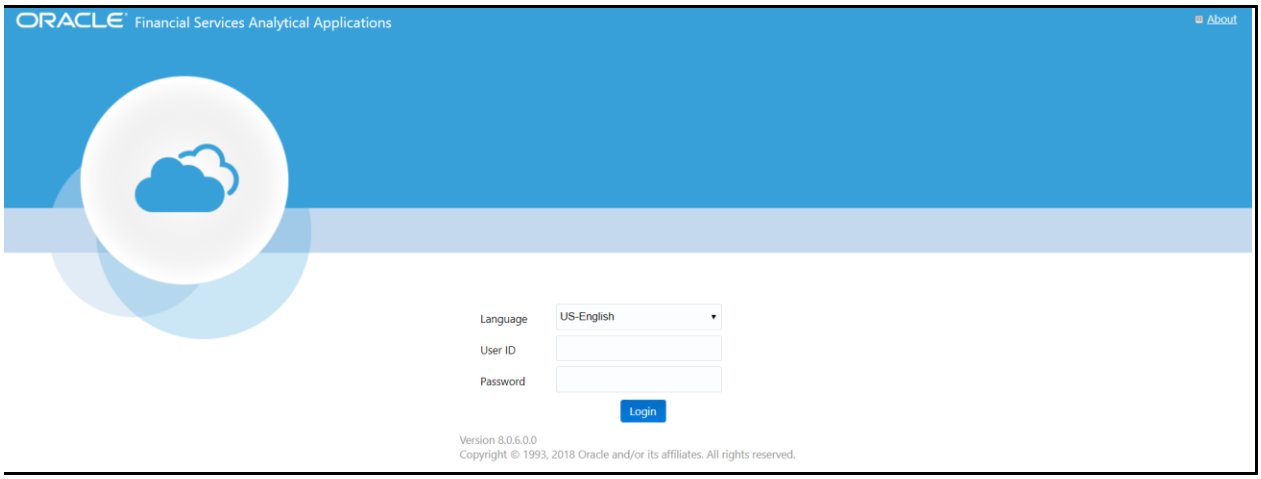
Identity Management


ORACLE Financial Services Analytical Applications
User: System Administrator
Security Management
User Administrator
User Maintenance
User Group Maintenance
User - User Group Map
User Group Domain Map
User Group Role Map
User Group Folder Role Map
User Reinstate
System Administrator
User Activity Report
User Profile Report
Enable User
User - User Group Map
User - User Group Map
Search and Filter
Search
Reset
User ID
Name
User - User Group Map
1 - 4 / 4
User ID
Name
GUEST
Guest Login
OFSAD
OFSAD
SYSADMN
System Administrator
SYSAUTH
System Authorizer
Records Per Page
4
OFSAD
Map
1 - 10 / 28
Mapped Groups
AHC FSDF Admin
SMS
AHC FSDF Analyst
SMS
AHC FSDF Operator
SMS
AHC OIDF Admin
SMS
AHC OIDF Analyst
SMS
AHC OIDF Operator
SMS
DIH Admin
SMS


User - User Group Map>User - User Group Map
User - User Group Map
OK
Close
Search
User Groups
Members
Selected Members
FAH Analyst
FAH Admin
Data Controller
DRM Operator
DRM Analyst
DRM Admin
DIH Execution
DIH Data Mapping
DIH Admin
AHC OIDF Operator
AHC OIDF Analyst
AHC OIDF Admin
AHC FSDF Admin
AHC FSDF Analyst
AHC FSDF Operator

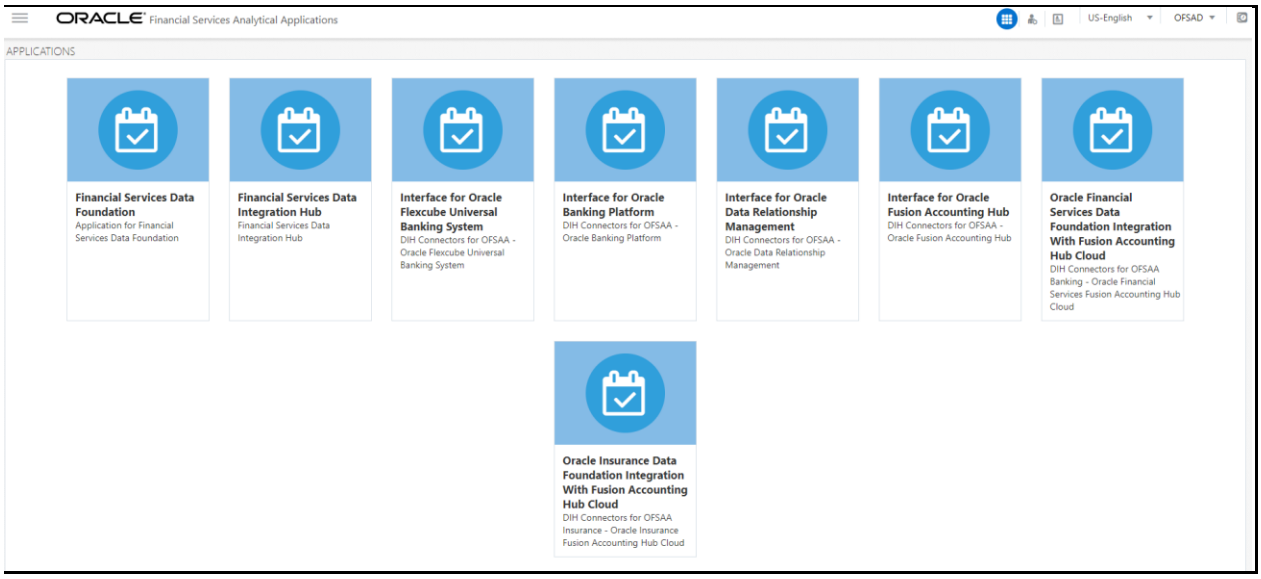
4 Logging into AHC-OFSAA Interface

Access the AHC-OFSAA Interface using your login credentials (User ID and password). The built-in security system ensures that you are permitted to access the window and actions based on the authorization only.

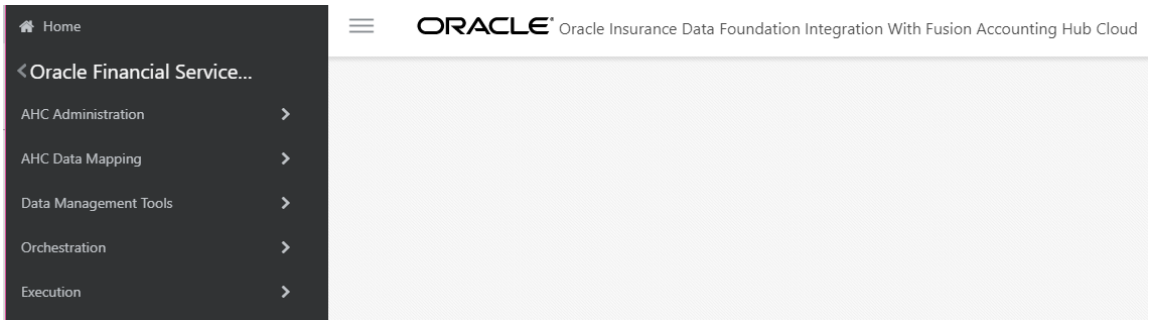


After logging into the application, select Oracle Insurance Data Foundation Integration With Accounting Hub Cloud Service.

NOTE You must be mapped at least any one of the AHC user groups.



The AH-OFSAA landing page is displayed.



5 Component Details

The following components included in this interface:

- Pre-packaged Subledger Application Information
- Template to Register Source System
- Data Integration Hub (DIH) Connectors

5.1 Pre-packaged Subledger Information

The following Subledger applications are pre-defined for the integration:

- Life Insurance
- Property and Casualty
- Health Insurance
- Annuity Contracts
- Retirement Contracts
- Investments
- Policy Loans
- Policy Admin PT
- Reinsurance Issued
- Reinsurance Held

The list of product processors covered under each Subledger Application is provided in the file SLA Coverage.

Each Subledger application is represented as an event class, which can have one or more instruments such as Life Insurance, Annuities, Health Insurance and so on an event class. For each event class, there are predefined accounting events based on expected activities for the Subledger application.

Sub ledgers can have two types of templates.

- Transactions-based: This indicates that AHC receives transaction information which is not pre-formatted for accounting. There are no debits or credits.
- Based on Passthrough accounting entries: There are debits and credits from the source system.

The rules vary based on the templates used.

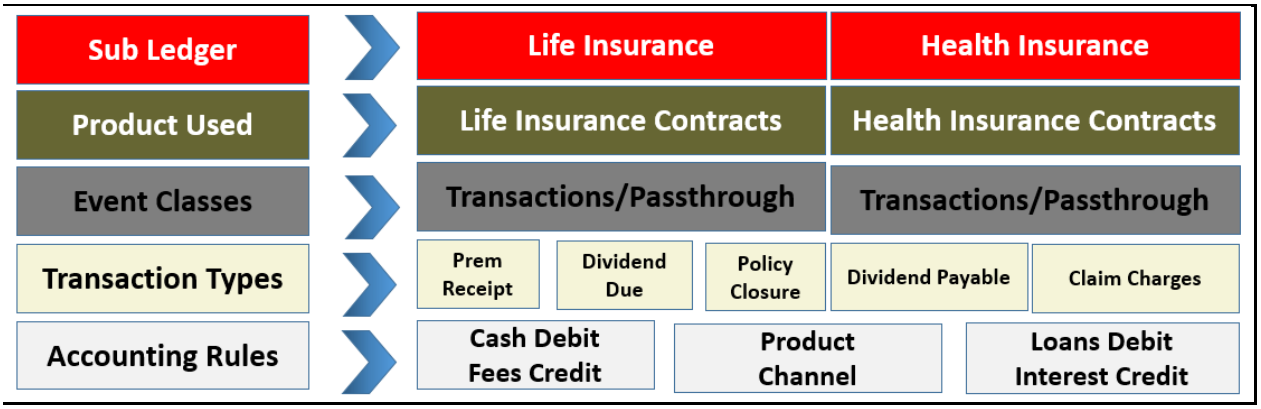
Subledger Application	Type
Life Insurance	Transaction based
Property and Causality	Transaction based
Health Insurance	Transaction based
Annuity Contracts	Transaction based
Retirement Contracts	Transaction based
Investments	Transaction based
Policy Loans	Transaction based
Policy Admin PT	Based on Passthrough accounting entries
Reinsurance Issued	Transaction based
Reinsurance Held	Transaction based

Subledger Information has transaction types mapped, along with the list of transaction and line attributes defined for each Subledger application.

The OIDF data model is registered with AHC using the Excel template. The attributes of the transactions, passthrough entries, and product processors are used by the templates. They are also available for creating new rules configurations to tailor accounting treatments.

You can use these pre-packaged Subledgers or create custom Subledger applications in the Oracle Insurance Data Foundation Integration With Accounting Hub Cloud Service.

An example of the structure for accounting templates for OFSAA integration is illustrated here.



The Life Insurance Subledger application has one product involved: Life Insurance Contracts. Each of these products can have one of the two event classes: Transactions or Passthrough entries. Note that pre-packaged Subledger applications use only one event class per Subledger.

Various transaction types are registered for each of the event classes. For Life Insurance Contracts, transaction types include premium receipt, dividend due, policy closure, dividend payable and claim charges. These transaction types and templates can be later used to configure rules in Accounting Hub Cloud Service.

Supporting references, which provide the ability to create accounting balances at a more detailed level than is provided by the general ledger chart of accounts, are available across all the Subledger

applications provided for the integration. Product and channel are examples of potential supporting references.

5.2 Registering with AHCS

AHCS uses a macro-enabled XLS template to facilitate registration of transaction or event types and creation of a catalog of transaction information, towards preparing accounting rules. The template is defined by AHCS and made available as an XLSM file from your AHCS instance. OIDF Integration with AHCS will populate it with SLA specifications in an automated fashion.

Upon initial use (that is, when you have freshly installed OIDF Integration with AHCS) and after user-configured changes to any aspect of Sub Ledger Applications or their events information (event types, header information, line information or SLA packaging), the following actions need to be undertaken for all or all affected SLAs.

1. Access your instance of AHCS and download the template file `XlaSourceSystemSetup.xlsm`. Refer to the AHCS user guide for details on how this can be achieved.

   The sheets in this XLSM file are as follows:

   a. **Instructions**: This sheet provides a brief overview of the template structure alongside instructions to fill in details and create the final ZIP file which is to be uploaded to your AHCS instance.

   b. **Source System**: This sheet has sections that capture the source system name and transaction types, which are used to register the source system in AHC.

      - Source System – Name and short name are mandatory. This goes as the name for Source, Sub Ledger, Journal Source, Journal Category, and Process Category.
      - Transaction Types – Name and short name for all the event/transaction types must be mandatorily provided in this field.
      - There are ‘Validate’ and ‘Generate ZIP’ buttons available on the Source System sheet. These support functions are detailed in section Registration of SLAs with Accounting Hub Service.

   c. **Transaction Information**: This sheet captures the attributes which you wish to use as header attributes in AHCS. You must enter the Name, Short Name, Domain and Journal Display. By default, three mandatory attributes are displayed, which cannot be removed - these are highlighted in grey.

   d. **Line Information**: This sheet captures the attributes which you wish to use as line attributes in AHCS. You must enter Name, Short Name, Domain and Chart of Accounts Display. By default, three mandatory attributes are displayed, which cannot be removed - these are highlighted in grey.

   Refer to AHCS documentation for further details.

5.3 DIH Connectors

5.3.1 Extract Connectors

The DIH connectors extract data from OFSAA staging tables, which have to be created as part of creating Subledger in the application. These connectors extract data from OFSAA staging tables in ASCII file format per SLA definitions.

There are two External Data Descriptors (EDD) that are created for each Subledger that is saved in OIDF Integration with AHCS. One EDD is for the Header file and another for the Line file. Each EDD

has the same structure as the transaction and line attributes that are configured for a Subledger, respectively. The attributes of the Subledger, which are defined in these EDDs, are mapped to the attributes of the staging tables.

There are pre-defined connectors that are created as part of pre-packaged Subledgers. EDDs and connectors are already provided for pre-packaged Subledgers.

All Header Connectors accept the following input parameters:

- **FIC_MIS_DATE or MIS_DATE** is the Extraction Date
- **LEDGER_NAME** Name of the Primary Ledger is a runtime parameter populated automatically from Subledger details.
- **SLA_CODE** Subledger Code or Short Name is a runtime parameter populated automatically from Subledger details.
- **AH_GAAP_CODE** is a constant Parameter to be updated in DIH Parameters section before executing batch
- **AH_LOAD_RUN_ID** Unique task ID which is auto picked while execution of a task
- **AH_FILE_DATE** is the Extraction Date whose file format should not be altered.

All Line Connectors accept the following input parameters:

- **FIC_MIS_DATE or MIS_DATE** is the Extraction Date
- **SLA_CODE** Subledger Code or Short Name is a runtime parameter populated automatically from Subledger details (For example 'RETAIL_SLA' for Retail Banking)
- **AH_LOAD_RUN_ID** Unique task ID which is auto picked while execution of a task

AH_FILE_DATE is the Extraction Date whose file format should not be altered

For Extract Connectors and mapping details, see the Extract Connectors Structure file section.

5.3.2 Insert Connectors

The connectors have pre-built mappings between AHC source views for GL/SR balances and OFSAA staging entities meant to store ledger data.

The following connectors are provided by AHC for GL and SR balances:

- **AH GL Balances Insert Con**: Connector to load thin ledger balances into the STG_GL_DATA table in OFSAA. This connector accepts these input parameters:
  - **PERIOD_NAME** Period for which GL data is loaded(For Example - '9-Sep'')
  - **FIC_MIS_DATE or MIS_DATE** is the Extraction Date
  - **AH_CONSOLIDATION_FLAG** Runtime parameter and is a single-digit flag (For Example 'C' or 'S').
- **AH SR Balances Insert Con**: Connector to load thick ledger/supporting reference balances into STG_MANAGEMENT_LEDGER tables in OFSAA. This connector accepts these input parameters:
  - **PERIOD_NAME** Period for which GL data is loaded
  - **FIC_MIS_DATE or MIS_DATE** is the Extraction Date

For Connectors and mapping details, see the Insert Connectors Structure file section.

6 Implementation

6.1 Background

Transaction objects in AHC are tables defined for each event class, capturing source transaction data for accounting events. The Create Accounting process gets the source transaction data from the transaction objects to generate journal entries. There are different types of transaction objects indicating whether they are used at the header or line level.

Header sources have the same value for all transaction lines or distributions associated with an accounting event. These sources are associated with a transaction header or with transaction reference data. Line sources have values that can vary by the transaction lines or distributions associated with an accounting event. They must be stored in the transaction objects at the line level.

Two separate header tables, one for each flow type is introduced in the OFSAA staging area for defining header and one Line table is introduced for defining line sources in AHC:

Stage Transaction Header (STG_TXN_HEADER) for event-based accounting

Stage Accounting Entries Header (STG_ACCT_ENTRIES_HEADER) for pass-through accounting

Associated Transaction Line Table for Line accounting. Example- STG_ANNUITY_TXNS for event-based Accounting and STG_ACCOUNTING_ENTRIES for passthrough accounting

6.2 Implementation Guidelines

1. Transaction numbers must match in the header and line files of import data files in AHC. You must carefully create connectors to extract the transaction numbers that have to match accordingly.
2. Ensure you provide the correct ledger name in the OIDF Integration with AHCS screens while configuring the Sub Ledger.
3. Do not execute the connector's batch in OIDF Integration with AHCS till the accounting rules and other required setup are configured.
4. Create connectors with appropriate filters, to extract only the required data from a table. Once a set of transaction numbers are processed in AHC, it does not accept the same transaction numbers again for a Subledger. Hence, you must carefully extract the required data for appropriate accounting.
5. If the SLA template is to be reloaded to AHC after changes, you must be careful while replacing the old configuration and must again create the accounting rules accordingly.
6. Only those event type names which are configured in AHC can be processed for accounting. If any other event type names come as part of data files imported to AHC, the whole file will not be processed.
7. After the SLA template is uploaded and source system is registered, in case of any further changes, ensure the following steps are followed in both OIDF Integration with AHCS and AHC:
    - If you add/edit/delete any of the attribute or transaction types in the AHC screens, but do not update the respective changes in the OIDF Integration with AHCS application, integration may fail as data extract files will still have the unchanged structure of DIH.
    - If you add/edit/delete any of the attribute or event type in the OIDF Integration with AHCS screens, but do not download the new template and re-upload it back to AHC application, integration may fail. In this scenario, the AHC application has a different Sub Ledger configuration and the DIH extract connectors will have a different configuration, which can lead to failure in data import.

- If there are any changes made in the source system/Sub Ledger structure either in OIDF Integration with AHCS or AHC, other application structure must also be changed accordingly. If not, it may lead to integration failure.

8. Through OIDF Integration with AHCS, there is no mechanism to identify those transactions, which is not processed for accounting. You must identify them through AHC screens and re-upload them separately, after making required corrections.

6.3 Transaction Object Population

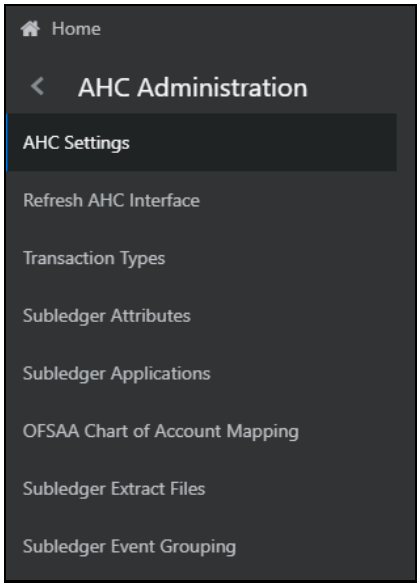
Transaction objects are extracted into Header and Line CSV files and converted to zip format along with the Metadata text file.

1. In Subledger applications, each Subledger will have Header and Line Objects defined along with Transaction types.
2. Header and Line Connectors are created to extract Header and Line data respectively.
3. Connectors are published and an auto-generated batch is executed.
4. After the successful execution of all Tasks in batch, Header information is extracted to XlaTrxH.csv file and Line information to XlaTrxL.csv file.
5. Metadata.txt file is auto-populated with Subledger details.
6. All three files: XlaTrxH.csv, XlaTrxL.csv, and Metadata.txt are auto-converted to zip format which can be uploaded to AHC.

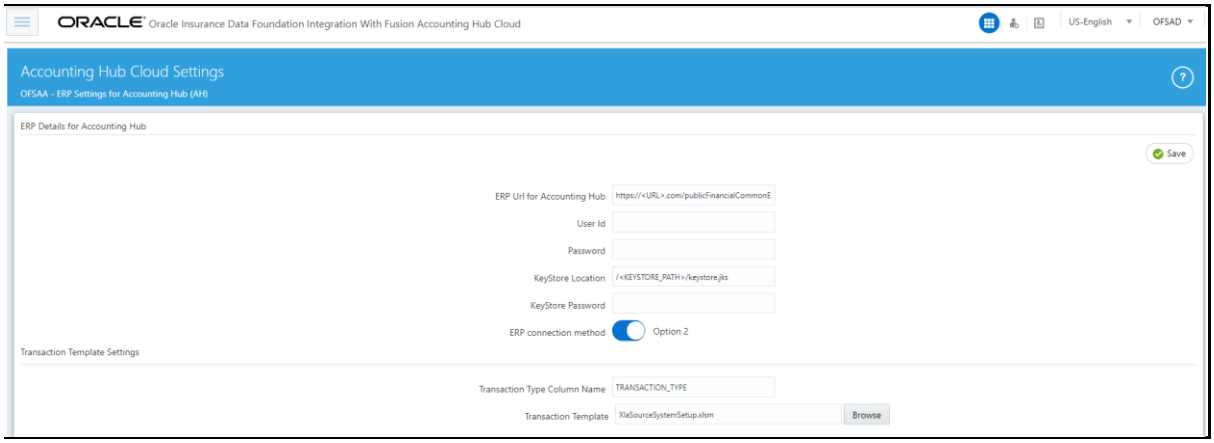
7 AHC Administration

7.1 AHC Settings

1. Navigate to **Oracle Insurance Data Foundation Integration With Accounting Hub Cloud Service** > **AHC Administration** > **AHC Settings**.



The **OFSAA - ERP Settings for Accounting Hub (AH)** screen appears.



2. Enter the required information under ERP Details for Accounting Hub and Transaction Template Settings fields.

   a. Enter the following details under ERP Details for Accounting Hub:

   — URL for Accounting Hub Cloud Service:

   Enter the URL assigned to you as part of your Oracle ERP Cloud AHCS subscription here. This information is specific to your tenancy on Oracle Cloud and subscription to ERP Cloud AHCS instance.

   The URL is in this form:

```
https://<<YourInstance>>/fscmRestApi/resources/latest/erpintegrations
```

Of this URL, the section

"/`fscmRestApi/resources/latest/erpintegrations`" must be left as it is.

Set <<YourInstance>> as assigned to you as part of your Oracle ERP Cloud AHCS subscription.

The captured URL must resemble this example:

```
https://abc.de.fg.oraclecloud.com//fscmRestApi/resources/latest/erpintegrations
```

- User ID
  - Enter User Identification assigned to you for AHCS instance here.
- Password
  - Enter the password for your AHCS instance here.
- KeyStore Location:
  - Enter fully resolved file-path for your KeyStore location here.
- KeyStore Password
  - Enter the password for your KeyStore here.
  - **Note**:
    - The certificate must be imported to KeyStore before connection with AHCS is attempted
    - KeyStore must be saved in one of the following locations:
      - WEB_HOME (when ERP Connection Method is set to Option 1)
      - FIC_HOME (when ERP Connection Method is set to Option 2)
    - See Import Certificate into Keystore section for details on importing Certificate to Keystore
- ERP Connection Method select,
  - Option 1 – This option uploads SLA events extract into UCM in ERP from the OFSA Web Server. The generated file extract is transferred from OFSA Application Server to OFSA Web Server through SFTP. Choose this option if Oracle Java Required Files (JRF) Template is applied while setting up Oracle WebLogic Server Domain. See Oracle Middleware documentation for details on JRF template and WebLogic Server Domain.
  - Option 2 – This option uploads SLA events extract into UCM in ERP from OFSA Application Server. It is recommended to use Option 2 to avoid the additional step of SFTP between OFSA Servers.

b. Enter the following details under Transaction Template Settings:

- Transaction Type Column Name:
  - This is prepopulated with value "TRANSACTION_TYPE", the auto-generated value for Transaction Type column name in Register Transaction Source System step

- If the auto-generated value for Transaction Type column name in Register Transaction Source System step is modified, edit pre-populated text and capture assigned column name here.

— Transaction Template:

- Transaction Template refers to the template contained in the XLSM file downloaded from your instance of AHCS to facilitate registration of transaction or event types. Refer section Registering with AHCS for details.
- Browse and select the transaction template file

  `XlaSourceSystemSetup.xlsm` from the downloaded location in the local machine.

c. Click Save.

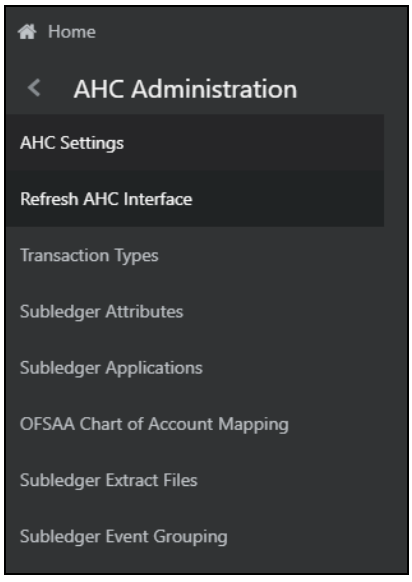
7.2 Refresh AHC Interface

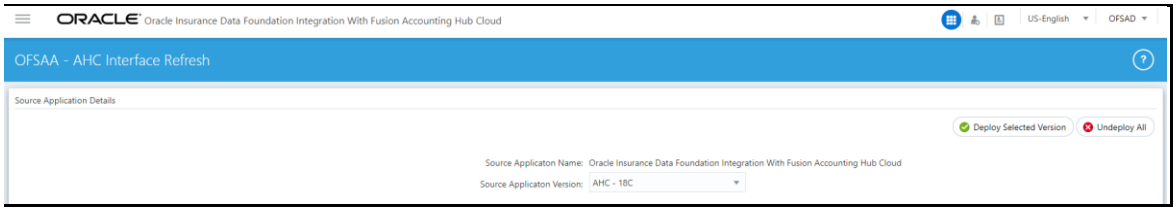
7.2.1 Deploying Connectors

After the pre-requisites are complete, External Data Descriptors and Connectors linking them to Application Data Interfaces, as used for data exchange with AHCS, are ready for deployment through the Refresh AHC Interface menu. These External Data Descriptors and Connectors are listed under the AHC Data Mapping menu once this step is completed.

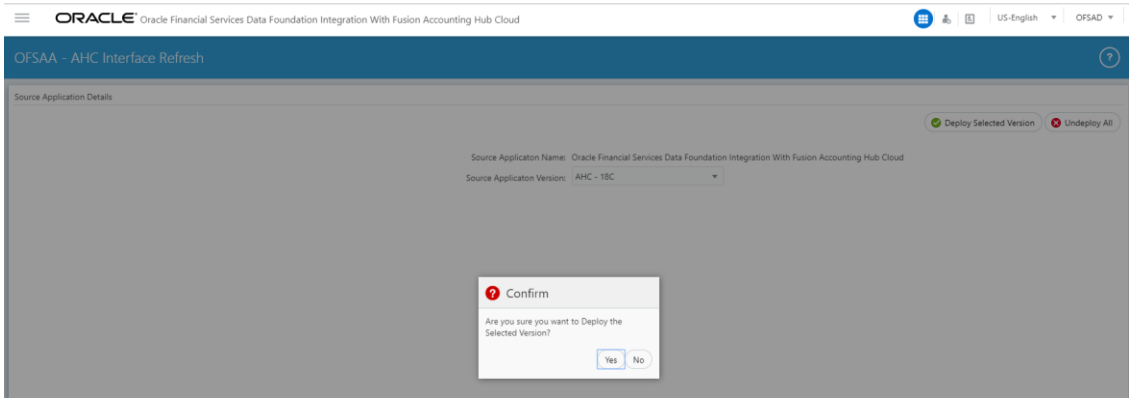
Perform the following steps:

1. Navigate to the AH application interface.
2. Select **AHC Administration**, and click **Refresh AHC Interface**.

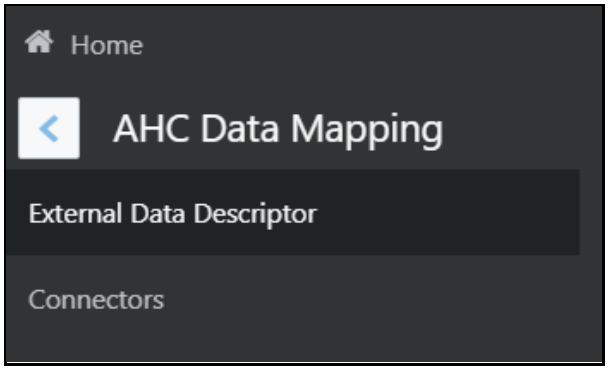


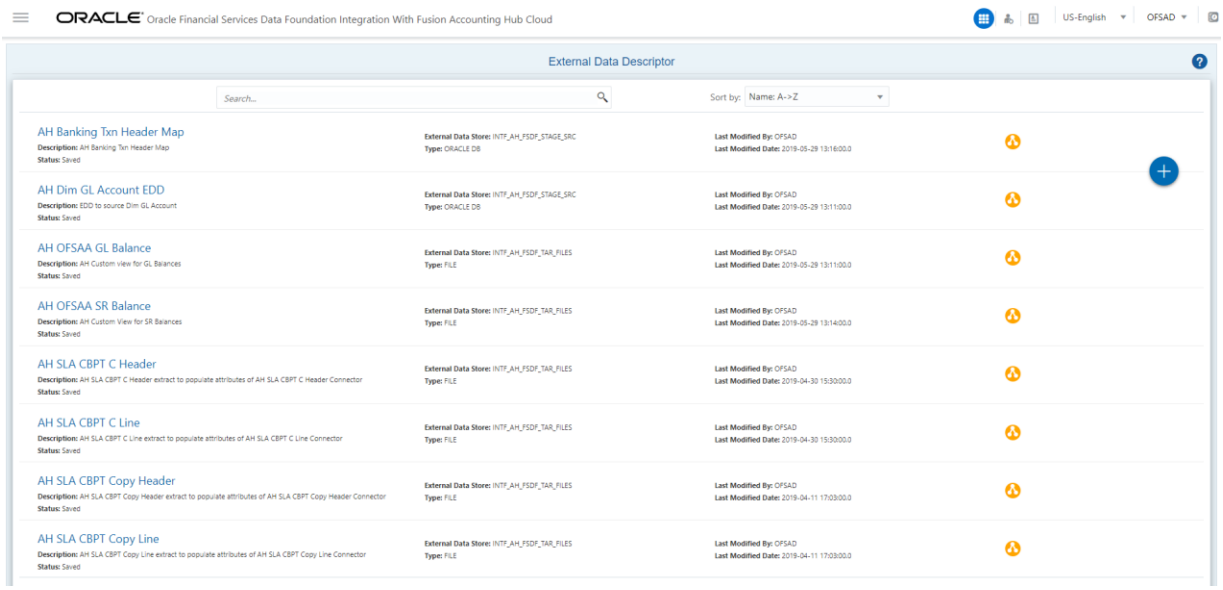


3. Select the **Source Application Version** from the drop-down menu.
    a. This must be the version of AHCS provisioned as part of your subscription.
    b. If the version of AHCS provisioned to you is not listed, contact Support.
4. Click **Deploy Selected Version**. The message, “Are you sure you want to Deploy the Selected Version?” appears.
5. Click **Yes** to proceed. After the deployment is complete, the “Deployment Successful” message appears.

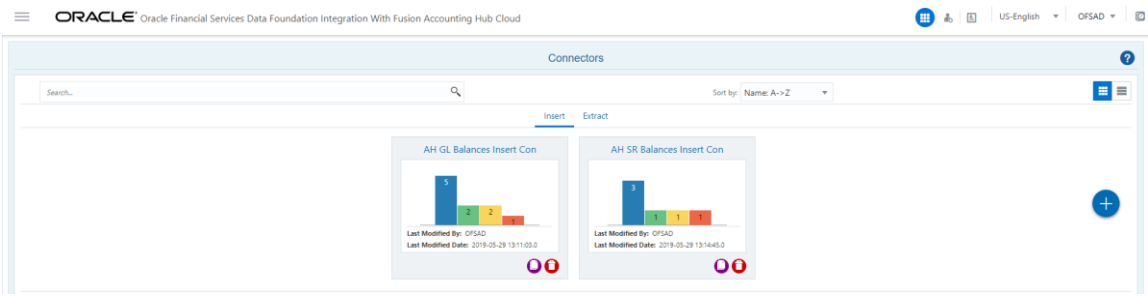


6. Once the system confirms successful deployment, you may navigate to **AHC Data Mapping > External Data Descriptor** and **AHC Data Mapping** > **Connectors** to verify the deployed EDDs and Connectors, as shown in the following screenshots.





AH External Data Descriptor



AH Connectors

7.2.2 Undeploying Connectors

In case there is a need to re-deploy Connectors, they need to be Unpublished, Undeployed and Deployed, post required changes. Undeploy All facility under the AHC Interface Refresh user interface facilitates the undeployment step within this sequence.

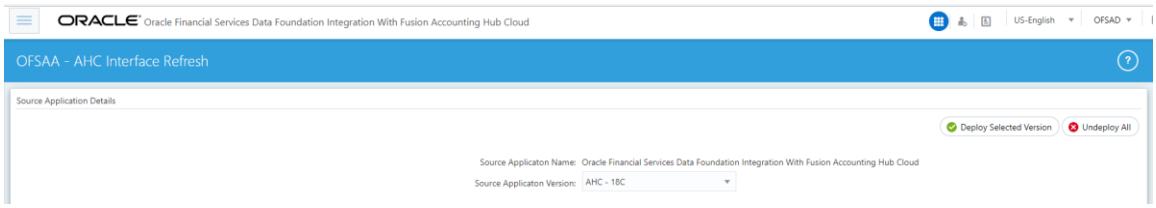
Connectors must be re-deployed only in response to the following situations:

1. Standard Connector definition requires changes or is deployed with changes that need to be maintained.
2. Support has issued instructions to re-deploy Connectors for AHCS integration following or as part of an application patch.

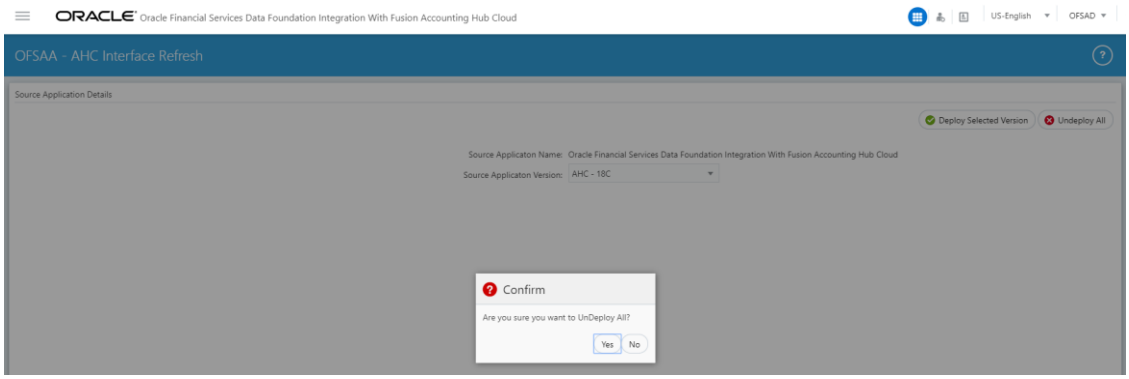
You cannot undeploy Connectors when they are in Published status. To undeploy Connectors, it must be explicitly unpublished. Refer DIH User Guide for details on unpublishing Connectors.

To proceed with undeploying Connectors:

1. Click Undeploy All. The message, “Are you sure you want to Undeploy All?” appears.



2. Click **Yes** to proceed.



3. After the undeployment is complete, the “Successfully UnDeployed All” message appears.

> **NOTE** Interface connectors can be Undeployed only if:
>
> 1. None of the Interface Connectors is in Published state.
> 2. The EDDs deployed through the interface version are not used in any Custom Connectors.

7.3 Transaction Types

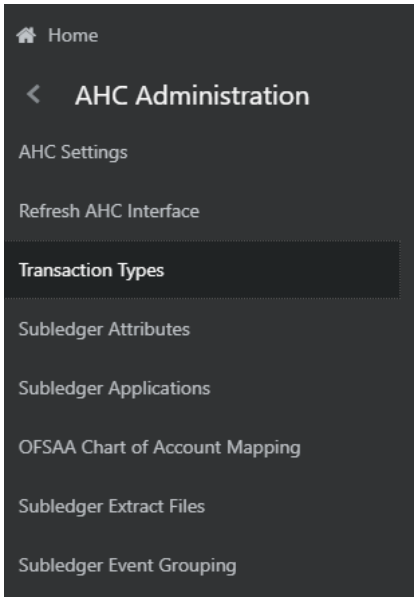
Event or Transaction types, refer to accounting events that are captured when transactions are committed or processed. While creating a Subledger, different transaction types are created so that all the transactions can be classified into one or the other event type and are used for creating journal lines. Examples of transaction types can be withdrawal, deposits, servicing, fees, charges and so on.

Transaction types are required to collect the transaction data and process it into accounting data necessary to form the Subledger information. Data from different source systems come together and get processed depending on these categories to form ledger information.

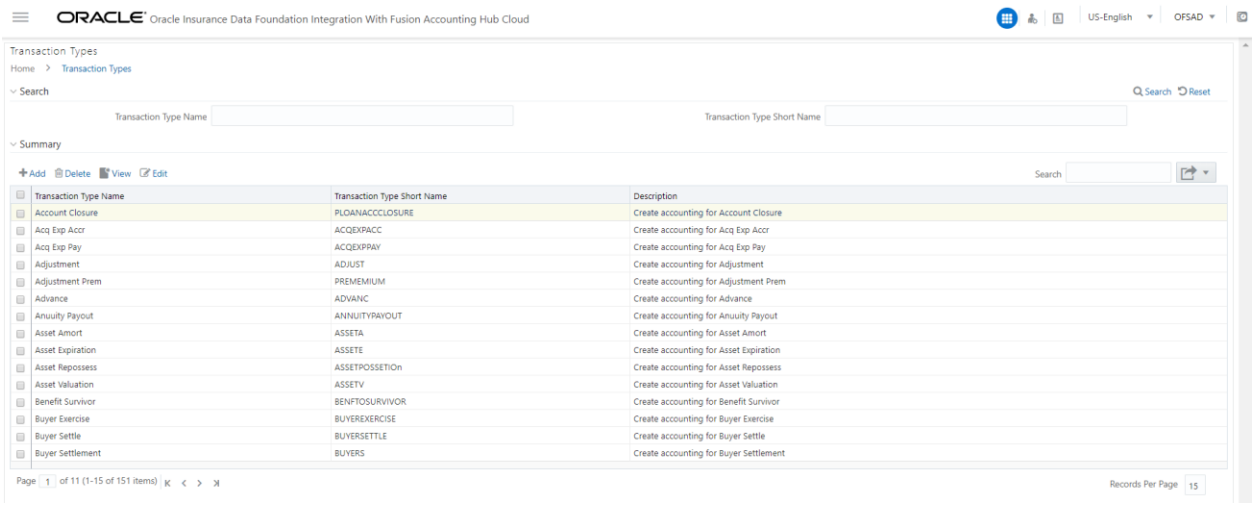
> **NOTE** To use pre-packaged Subledgers, you must use only the pre-packaged transaction types in the source system data.

To understand the transaction types:

1. Navigate to the AHC application interface.

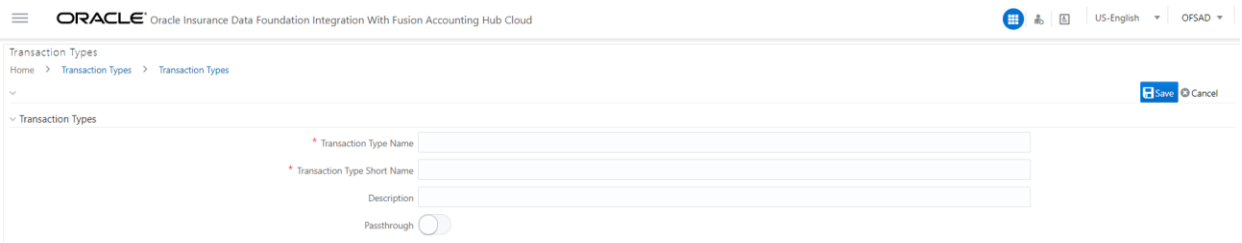


2. Select **AHC Administration**, and click **Transaction types**.



A list of pre-packaged transaction types is displayed on the screen, which you can map with Subledgers accordingly. You cannot edit or delete these pre-packaged transaction types.

3. You can search for Transaction Type Name or Short Name. Enter a part or full name and click the search button to get the filtered result.
4. Under **Summary**, you can Add, Delete, View or Edit the events.
5. Click Add. The transaction types screen appears.
6. Enter the following details to create a new event and click **Save**.



a. Transaction Type Name

b. Transaction Type Short Name

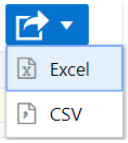
c. Description

d. Passthrough

Note: The following are the validations for the above fields:

Field Name	Limitations/Validations
Transaction Type Name	▪ The name must not have special characters. ▪ Only alphanumeric characters and spaces are allowed. ▪ Must not be more than 15 characters. ▪ The name must not start or end with space. ▪ It must be unique.
Transaction Type Short Name	▪ Short Name must have only Uppercase, numbers and underscores. ▪ Short Name must start only with an uppercase or a number. ▪ It cannot have special characters and space. ▪ Must not have more than 15 characters. ▪ It must be unique.

7. Click Delete if you wish to delete an event.
8. Click View to view the details of the transaction type.
9. Click Edit to edit the transaction type details.

10. Click Export 

to move the screen data to an Excel or CSV format for the reference.

During SLA configuration, the required transaction types from this list can be mapped to the defined SLA.

> **NOTE** It is not possible to edit or delete the transaction types which are pre-packaged in the application as well as those which are mapped to a Subledger. All transaction types lists are stored in the table FSI_ACCNTING_EVENT_TYPE_MASTER.

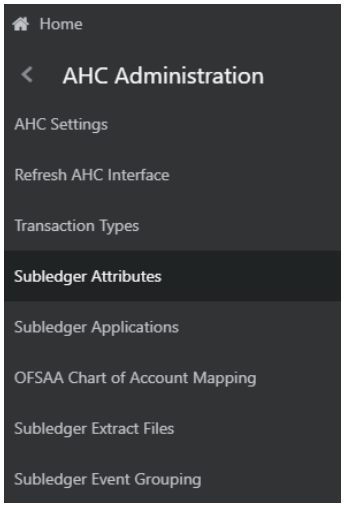
7.4 Sub-Ledger Attributes

The sub-ledger attributes provide a list of the pre-seeded transaction and line attributes, which can be mapped to an SLA while defining it. These attributes cannot be edited. You can also add more attributes to the list of attributes that can be mapped to an SLA during SLA configuration. The ledger balances can be calculated based on these sub-ledger attributes. To accommodate custom attributes, there is an option to add custom attributes that can vary from user to user to configure custom sub-ledger. These custom attributes can help in creating rules and extracting ledger balances in a required way.

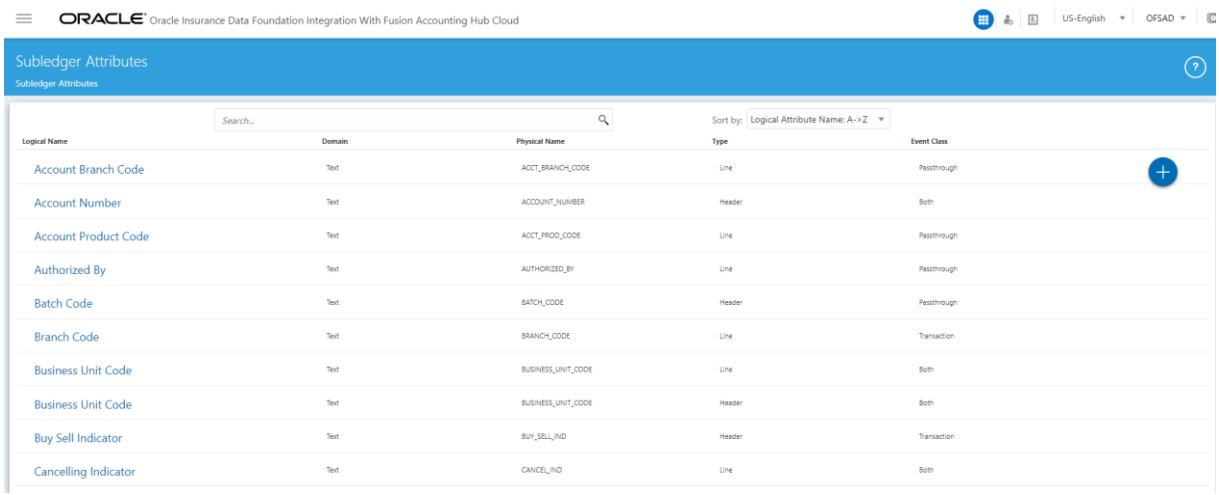
7.4.1 Adding a Sub-Ledger Attribute

To add a Subledger Attribute, perform the following steps:

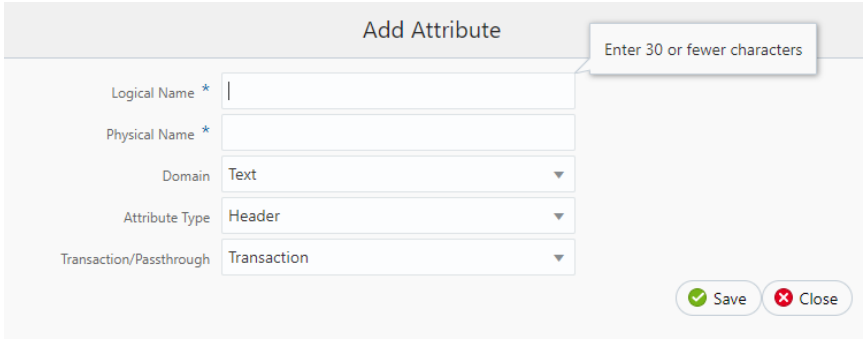
1. Navigate to **AHC Administration** > **Subledger Attributes**.



The Subledger Attributes screen appears with a list of seeded data with Logical and Physical Name.



2. Click  to add a new attribute. The Add Attribute screen appears:



3. Enter the Logical and Physical Name. Physical Name can have only upper case, numbers and alphabets.
4. Select the Domain, Attribute Type and Transaction/Passthrough details from the drop-down list. Transaction/Passthrough is the type of sub-ledger, whether to use this attribute for transaction type of sub-ledger or passthrough type of sub-ledger.
5. If the new attribute added is a Header Type, it appears in the Transaction Information. Whereas, if the new attribute added is a Line Type, it appears in the Line Information in SLA.
6. Click **Save**.

> **NOTE** You can delete only a new attribute added but not a seeded attribute. In case the added attribute is already mapped in SLA and saved then this cannot be deleted.

7.5 Sub-Ledger Applications

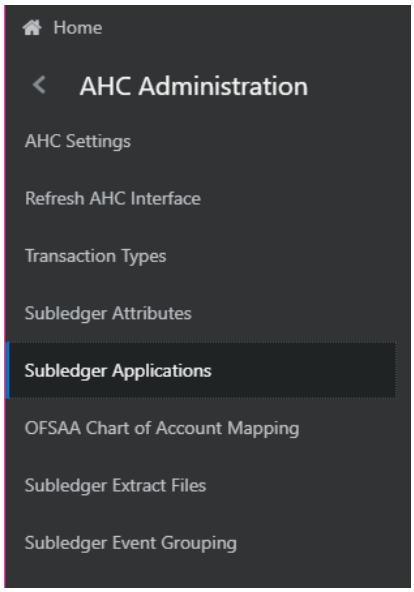
The goal of Subledger accounting is to generate journal entries for transactions that occur in Subledgers. To generate general ledger balances, it is very important to create journal lines and Subledger information. Subledger configuration helps transactional data to be transformed into GL account-level data by defining rules and thus loading the ledger balances. The total of the transactions in the Subledger rolls up into the general ledger.

AHC application has the SLA template through which a Subledger and its details are loaded.

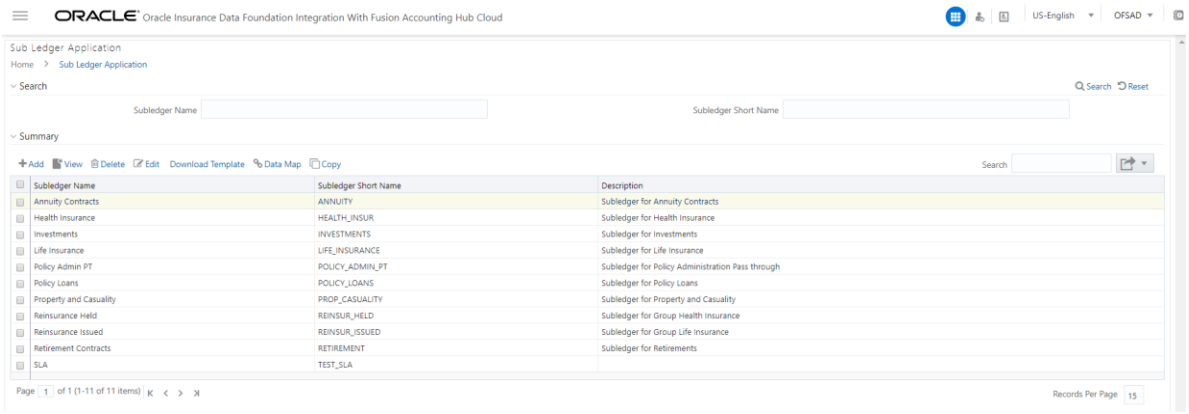
7.5.1 Sub-Ledger Application Summary

To understand the Sub-Ledger Application summary screen:

1. Navigate to the AHC application interface.



2. Select **AHC Administration**, and click **Sub-Ledger Application**.



3. You can search for Sub-Ledger Name or Short Name. A list of pre-packaged Subledgers appears. For more information, see the SLA Coverage list in the previous section.

> **NOTE** Pre-packaged Subledgers are not editable through AHC Sub-Ledger screens.

4. Under Summary, you can Add, Delete, Edit, View Data Map and Download the Sub-Ledger.
5. Click 

to create a new Sub-Ledger.
6. Select one Sub-Ledger and click 

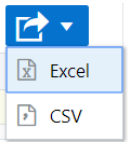
if you wish to delete a Sub-Ledger.
7. Select one Sub-Ledger and click 

the Sub-Ledger details.

> **NOTE** It is possible to publish or unpublish connectors for the pre-packaged Subledgers.

8. Select one Sub-Ledger and click View to view the details of the Sub-Ledger.
9. Select one Sub-Ledger and click Download Template.
10. Select one Sub-Ledger and click Data Map to create and map connectors to the selected Sub-Ledger.
11. Select one Sub-Ledger and click Copy a sub-ledger.
12. After the details are filled, save the Sub-Ledger.
13. Download the SLA template that has to be uploaded to the AHC application.

> **NOTE** It is mandatory to manually validate the template and create a zip file that must be uploaded to the AHC cloud environment. In case of errors during validation, update the required changes in the DIH screens and then download the template again. See the SLA Template Validations section for more information. Sub ledgers saved are stored in the table FSI_SLA_MASTER.

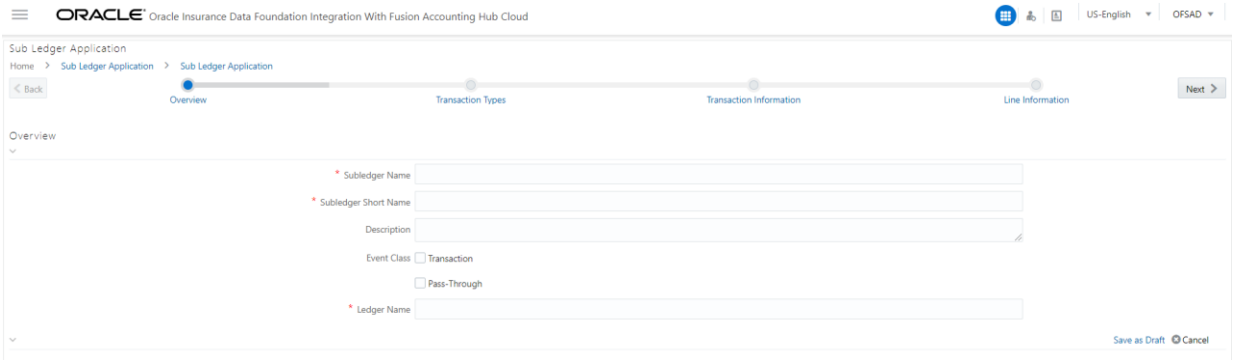
14. Click Export 

to move the summary data to an Excel or CSV format.

7.5.2 Adding a Sub-Ledger Application

To add a Sub-Ledger:

1. Click Add to create a new Sub-Ledger. The following screen appears.



2. Under Overview, enter the following details:
   a. Subledger Name
   b. Subledger Short Name
   c. Description
   d. Event Class
   e. Ledger Name

Note: The following are the validations for the fields mentioned above.

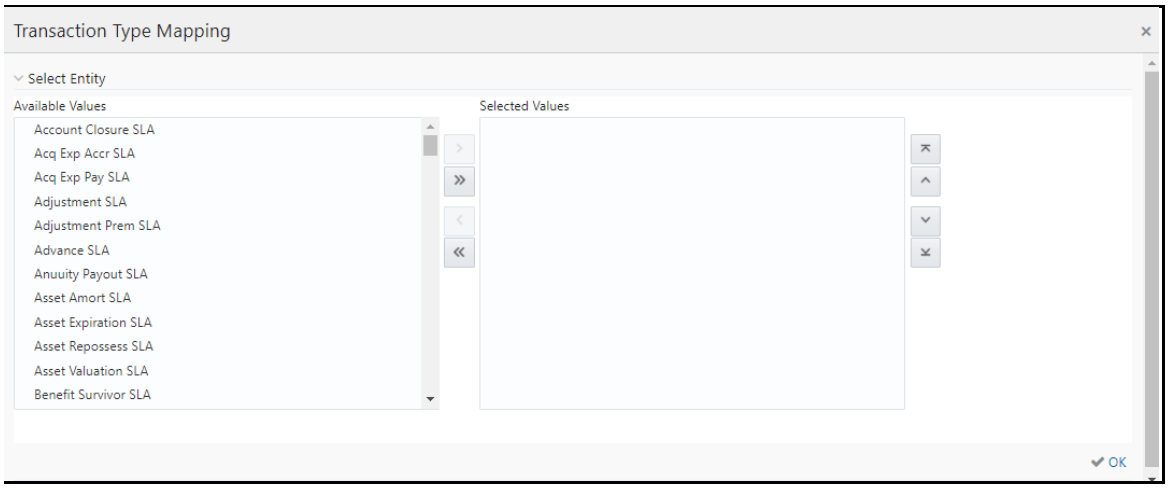
Field Name	Limitations/Validations
Subledger Name	▪ The name must not have special characters. ▪ Only alphanumeric characters and spaces are allowed. ▪ Must not have more than 15 characters. ▪ The name must not start with space. ▪ It must be unique.
Subledger Short Name	▪ Short Name must have only Uppercase, numbers and underscores. ▪ Short Name must start only with an uppercase or a number. ▪ It cannot have special characters and space. ▪ Must not have more than 15 characters. ▪ It must be unique.
Event Class	Can select one or both the options.
Ledger Name	Must not exceed 100 characters. Change the Ledger name to the required name, for the pre-packaged Subledgers, which by default has 'Default Ledger' as the input name.

3. Click **Save as Draft**. The message, "Subledger draft saved successfully" appears.
4. Click **Ok**. The Transaction types screen appears.
5. Under Transaction types, perform the following steps:

   a. Click Transaction Type Mapping. The Transaction Type Mapping screen appears.

   All transaction types that have been created as part of Transaction types, along with the pre-packaged list of transaction types screen appear here on the LHS.

> **NOTE** Transaction types get filtered based on the Event Class selected for the Subledger. Transaction Type to Subledger mappings is stored in the FSI_SLA_EVENT_TYPE_MAP table.

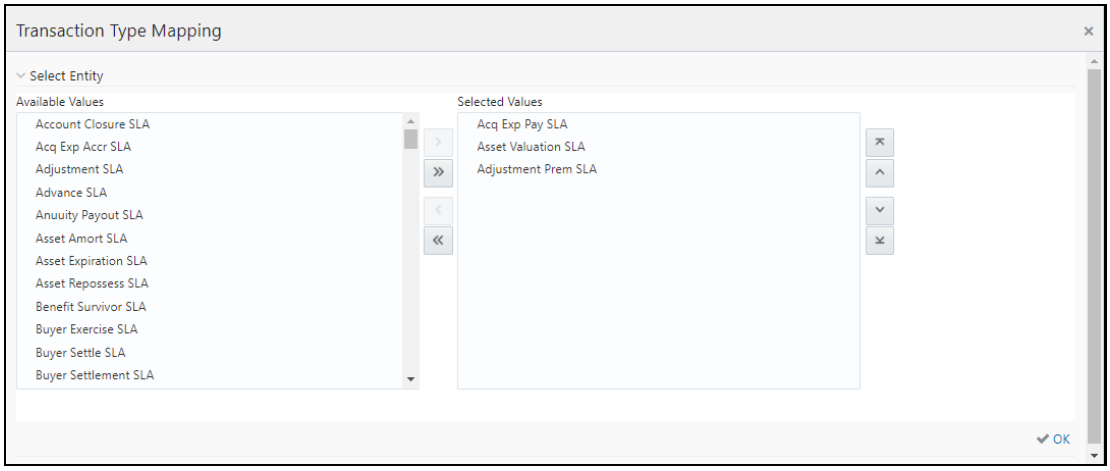


b. Click  to move the required entities to the list of the selected values.

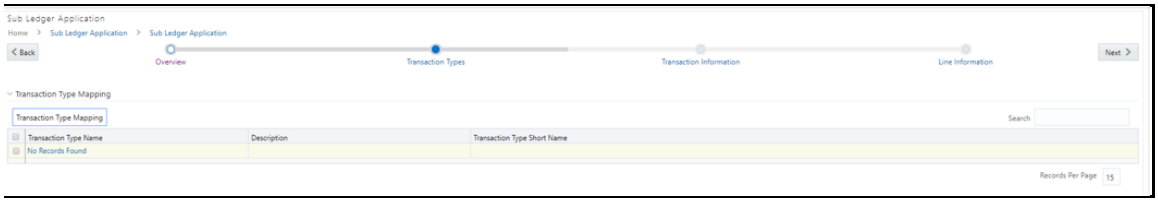
c. Click  to move all the entities to the list of the selected values.

d. You click to move the entities up and down.

e. After the entities are selected, click **OK**.



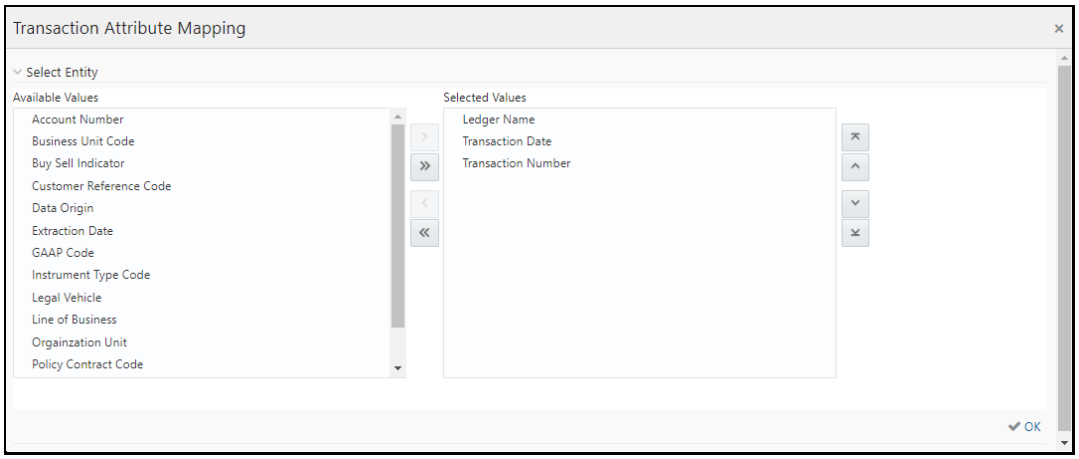
The selected values are now displayed under Transaction Type Mapping.



6. Under Transaction Information, perform the following steps:

a. Three mandatory attributes: Ledger Name, Transaction Date, and Transaction Name, are selected by default and cannot be edited.

b. Click + Transaction Attribute Mapping. The Transaction Attribute Mapping screen appears.



c. The list of transaction attributes is displayed. These are seeded from the FSI_SLA_ATTR_MASTER table.

d. Ledger Name, Transaction Date, and Transaction Name are selected by default.

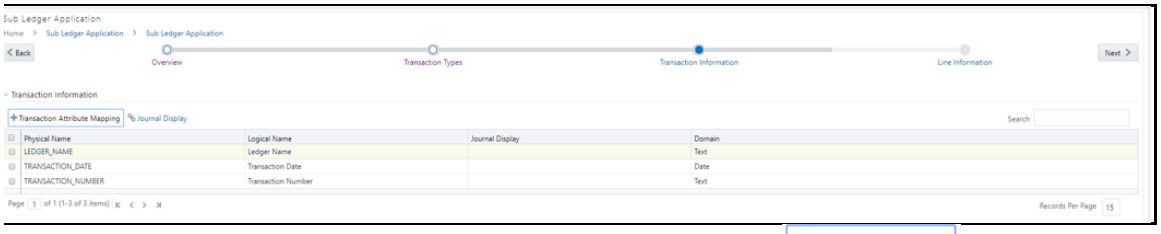
e. Click  to move the required entities to the list of the selected values.

f. Click  to move all the entities to the list of the selected values.

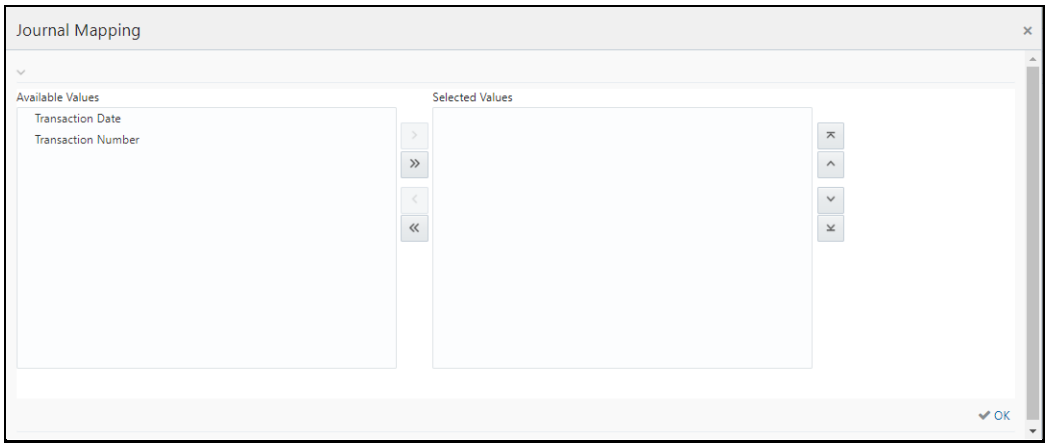
g. You click  to move the entities up and down.

h. After the entities are selected, click OK.

The selected values are now displayed under Transaction Information.



If you wish to display these attributes as part of journal lines, click Journal Display. The Journal Mapping screen appears.



NOTE You cannot assign a journal display as YES to Ledger's name.

i. Click > to move the required entities to the list of the selected values.

j. Click » to move all the entities to the list of the selected values.

k. You click ⤒ ^ v ⤓ to move the entities up and down.

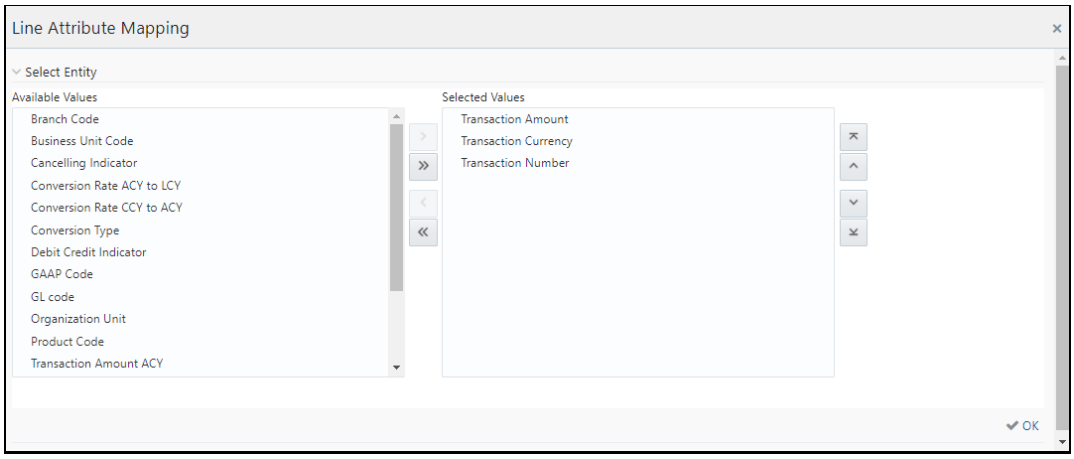
l. After the entities are selected, click OK.

m. Click Next.

7. Under Line Information, perform the following steps:

   a. Three mandatory attributes: Transaction Amount, Transaction Currency, and Transaction Number are selected by default and cannot be edited.

   b. Click + Line Attribute Mapping . The Line Attribute Mapping screen appears.



   c. The list of line attributes is displayed. These are seeded from the FSI_SLA_ATTR_MASTER table.

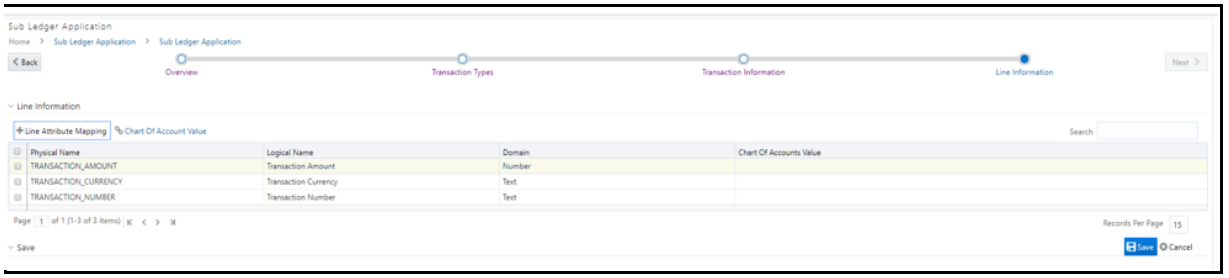
   d. Transaction Amount, Transaction Currency, and Transaction Number are selected by default.

e. Click  to move the required entities to the list of the selected values.

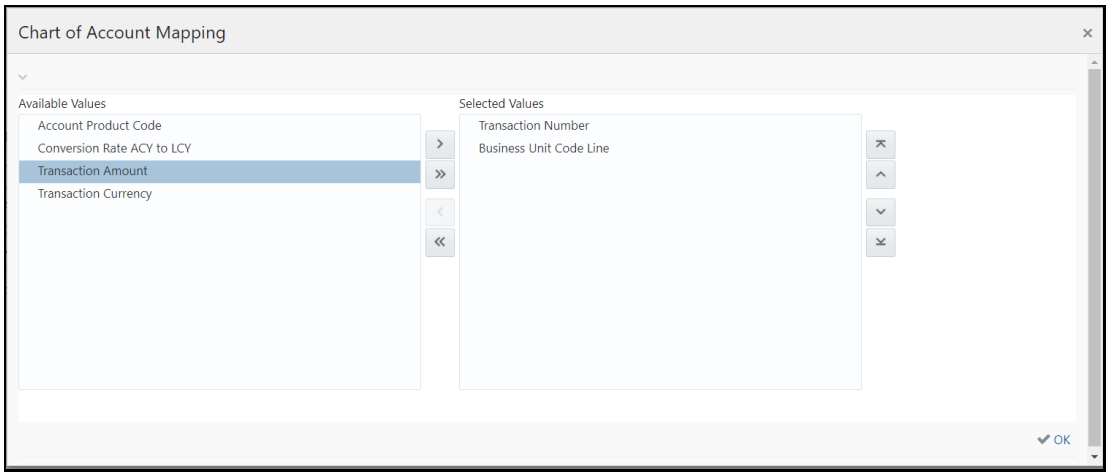
f. Click  to move all the entities to the list of the selected values.

g. You click to move the entities up and down.

h. After the entities are selected, click **OK**. The selected values are now displayed under Line Information.



i. If you wish to display these attributes as part of the chart of account, click Chart Of Account Value. The Chart of Account Mapping screen appears.



NOTE You cannot assign a chart of accounts display as yes to the three mandatory columns.

j. Click  to move the required entities to the list of the selected values.

k. Click  to move all the entities to the list of the selected values.

l. You click to move the entities up and down.

m. After the entities are selected, click OK.

8. Click Save. An entity is now saved under the summary screen.

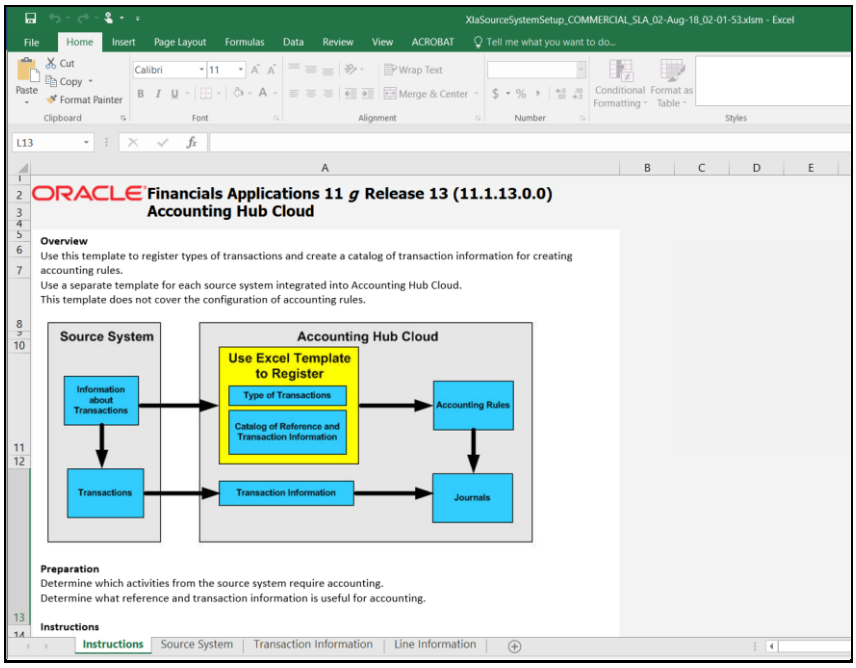
> **NOTE** After the transaction and line, information is mapped and saved, they are stored in the FSI_SLA_TRANSACTION_ATTR and FSI_SLA_LINE_ATTR tables, respectively. Transaction and Line information is already configured for the pre-packaged Subledgers. For information on Pre-packaged Subledger Configurations, refer section Pre-packaged Subledger Information.

7.5.3 Registration of SLAs with Accounting Hub Service

1. Navigate to Sub-Ledger Summary screen to access the list of SLAs.

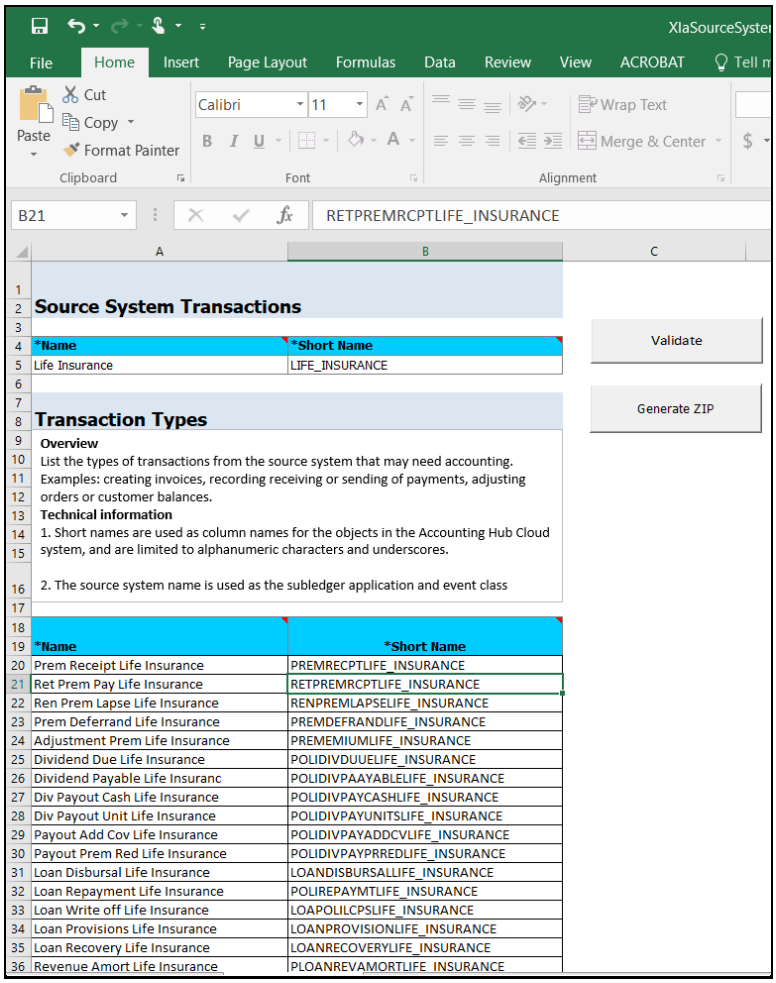
2. From this list of SLAs, select the SLA that needs to be registered.

3. Click **Download Template**. A file with the XLSM extension is downloaded to the client machine. This XLSM file maintains the template in `XlaSourceSystemSetup.xlsm` with Sub-Ledger Application details filled in.

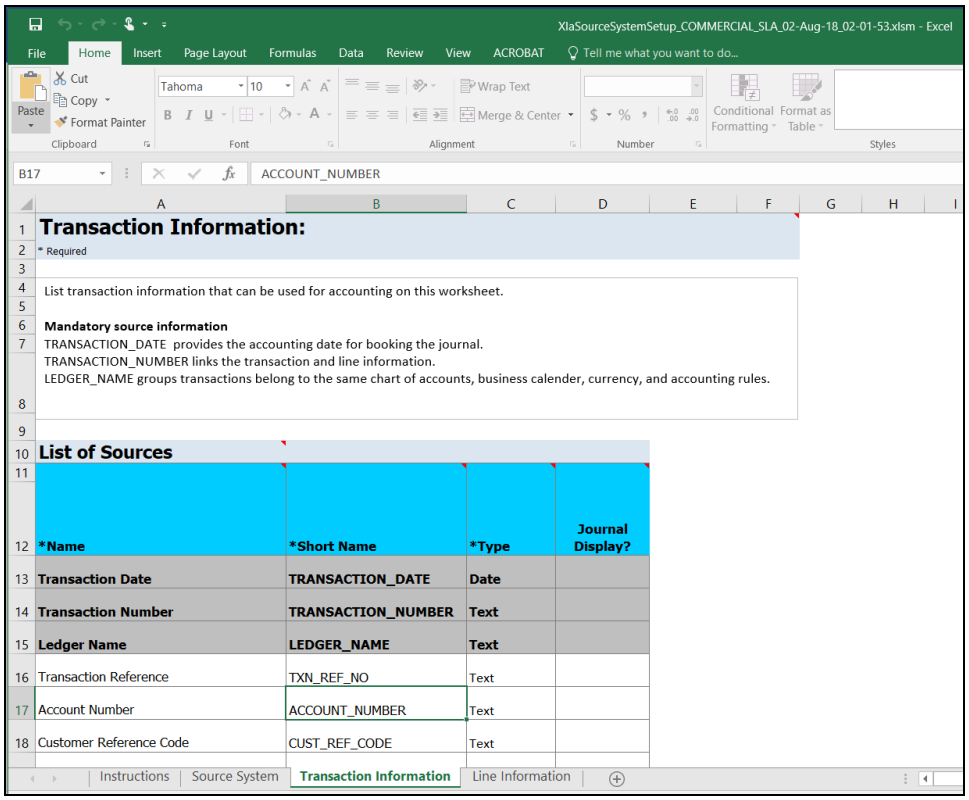


4. In the Instruction screen, all the details are explained.

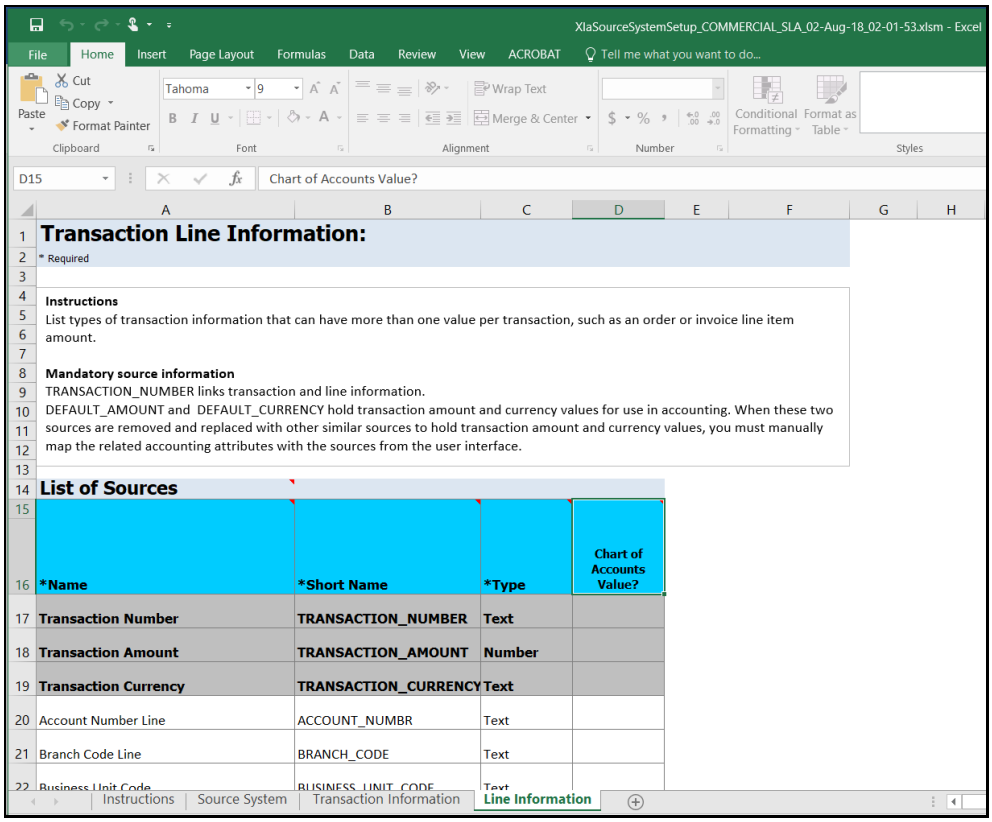
5. In Source System, the name and short name are given in the Sub Ledger Application screen are displayed.



NOTE The name or short name, which appears in the row, must have the event type name along with the Sub Ledger Application name, following EVENT_TYPE_NAME_SLA_NAME pattern, as in the screenshot, above. The Transaction Type name is appended with the SLA name to maintain the uniqueness of transaction types across all Sub Ledger Applications. Short Name is limited by AHCS specifications to 30 characters and those employed by OIDF Integration for AHCS consider this.



6. In Transaction Information, those Transaction Type Names and Short Names against which Journal Display is set to Yes is displayed through the AHCS user interface.



> **NOTE** The three mandatory columns are listed in a greyed area for both transaction and line information.

7. Click **Validate**, in Source System tab of the XLSM file, to check if there are any errors in the template.

> **NOTE** The XLSM files must not be manually modified. If you want to make changes to the files, make the required changes using the DIH user interface and follow the steps to download the template again.

8. Click **Generate ZIP**, in Source System tab of the XLSM file, to compress required content, prepare the ZIP file and upload it to AHCS instance.

> **NOTE** In AHCS, map 'Entered Amount' and 'Entered Currency' in Manage Accounting Attributes user interface to Transaction Amount and Transaction Currency, before proceeding with the configuration of accounting rules.

7.5.4 Steps to Implement Flexible Accounting

References:

- Verified Source Registration file (Verified file for XlaSourceRegistration).

> **NOTE** This is validated in use for registration with AHCS.

- Data template downloaded off AHCS (Verified file for XlaTransactionUploadTemplate).
- Sample files for EDD creation (Header EDD and Line EDD).

7.5.4.1 Instructions

Perform the following instructions:

- Obtain XlaSourceSystemSetup.xlsm from AHCS.
- Configure SLA Attributes in DIH.
- Save to obtain a filled XlaSourceSystemSetup.xlsm file.

> **NOTE** The verified XlaSourceRegistration and XlaTransactionUploadTemplate file have Retail SLA attributes configured, as an example.

- Configure EDDs in DIH.

7.5.4.2 Configuring SLA Attributes

To configure the SLA Attributes, follow these steps:

1. In the AHCS Settings window, update the **Transaction Type** column name to "EVENT_TYPE_CODE", and click **Save**.

2. While configuring SLAs or attributes:
   a. See verified source registration file (XlaSourceRegistration).
   b. Do NOT use the same names as defaulted by AHCS.
   c. Do NOT repeat names across transactions (header) and line attributes.
   d. Do NOT use or include “EVENT_TYPE_CODE” or “TRANSACTION_REVERSAL_FLAG” in the header, in this step.
   e. Do NOT use or include “LINE_NUMBER” in line, in this step.
   f. Excluding mandatory fields, the following counts must be achieved in the files:

Table 1: Transaction (Header)

Text	50
Number	10
Date	10
Long Text	5

Table 2: Line

Text	100
Number	30
Date	10
Long Text	5

   g. In both cases, all SLA attributes, including those you configure to be sent with actual data to AHCS, will figure in the count

3. Locate the filled XlaSourceSystemSetup.xlsm file and open the file.

   Navigate to Validate > Generate ZIP.

4. Register source and create SLA with the zip file in AHCS.
5. Download data template from AHCS for the defined SLA scope.

7.5.4.3 Configuring EDD

To configure EDD, follow these steps

1. See the downloaded data template and prepare EDD specification files referring (Header EDD and Line EDD as examples. Ensure that, “TRANSACTION_REVERSAL_FLAG” is included, if present in the downloaded data template. Also, ensure that “LINE_NUMBER” is included.

> **NOTE** You must carefully apply filters if any while creating connectors to fetch the correct data from a product for a Subledger. Extraction Date filter is already enabled with pre-packaged connectors for executing them.

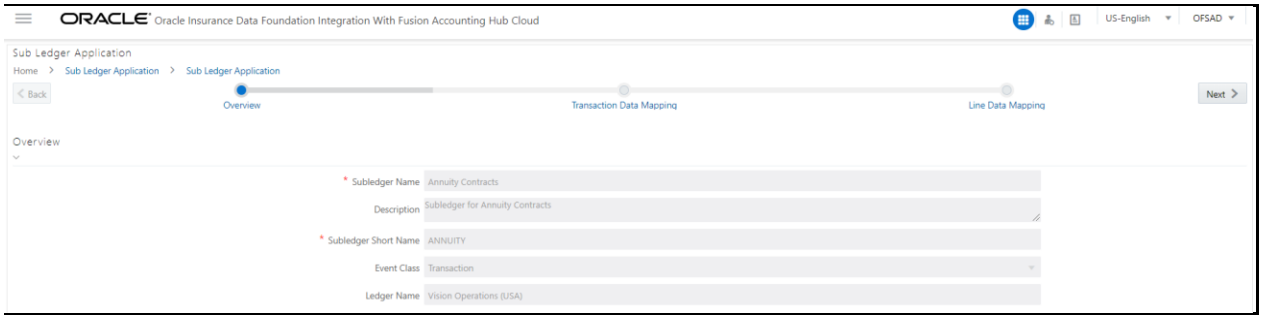
2. Verify and ensure that the order of data elements in Header EDD and Line EDD is the same as that in the downloaded data template.

3. Open transaction (header) EDD created by DIH upon SLA configuration ("AH SLA <<SLA Name>> Header" pattern), upload Header EDD prepared in step 2, save and verify the order.
4. Open line EDD created by DIH upon SLA configuration ("AH SLA <<SLA Name>> Line" pattern), upload Header EDD prepared in step 2, save and verify the order.
5. The interface with your instance of AHCS is now ready for use.

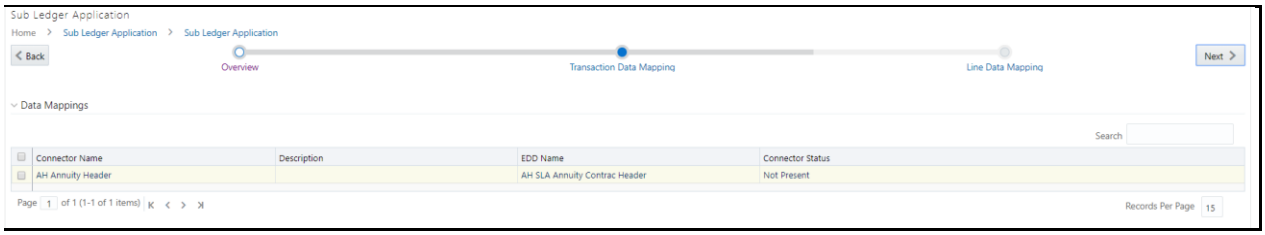
7.5.5 Using Data Map

To create a connector and map it to a Subledger:

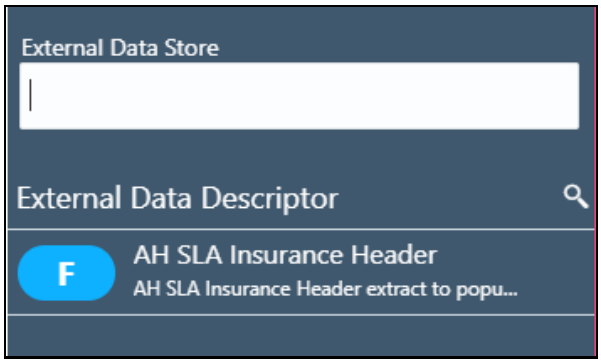
1. Select a row under the Subledger Application summary and click Data Map. The following screen appears. The fields are non-editable under Overview.



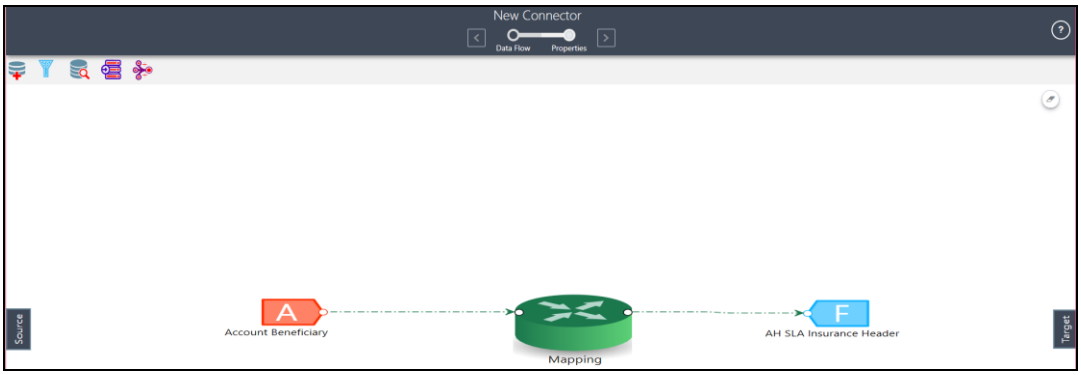
2. Click Next. The Transaction Data Mapping screen appears.



   a. Click **Create** to add a new data mapping. The New Connector screen appears.
   b. Under Target, the required EDD is automatically created following the "AHC SLA <<SLA NAME>> Header" pattern. For example, when Sub Ledger Application is named Commercial Banking, EDD created is "AH SLA Commercial Banking Header". EDD structure will have all the attributes that are mapped to the Sub Ledger Application under Transaction Information.



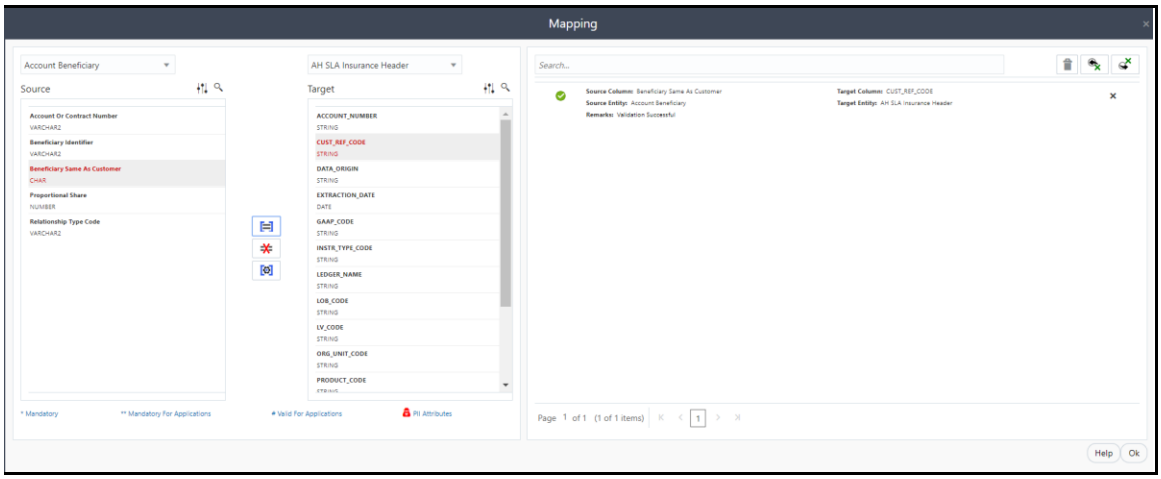
   c. Map the required source and target and save the connector.



NOTE You must carefully apply filters if any while creating connectors to fetch the correct data from a product for a particular Subledger. Extraction Date filter is already enabled with pre-packaged connectors for executing them.

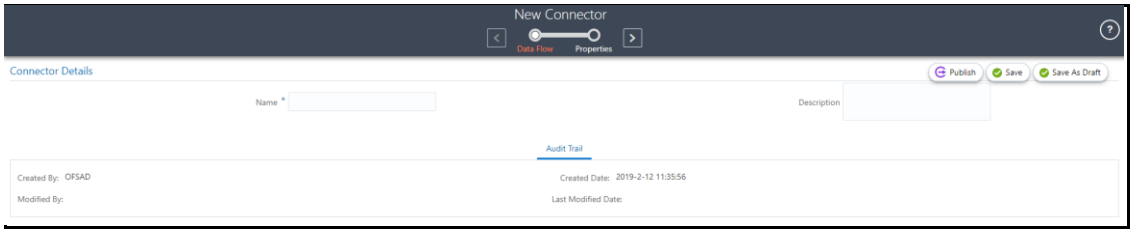
Event Type names must also be filtered depending on the Subledger you configure. If unconfigured, all transaction types are processed in the extract data for a particular Subledger and the accounting process fails.

d. Double-click mapping to map the source and target.

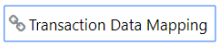


NOTE For more information on mapping, see the Connectors section in the DIH User Guide.

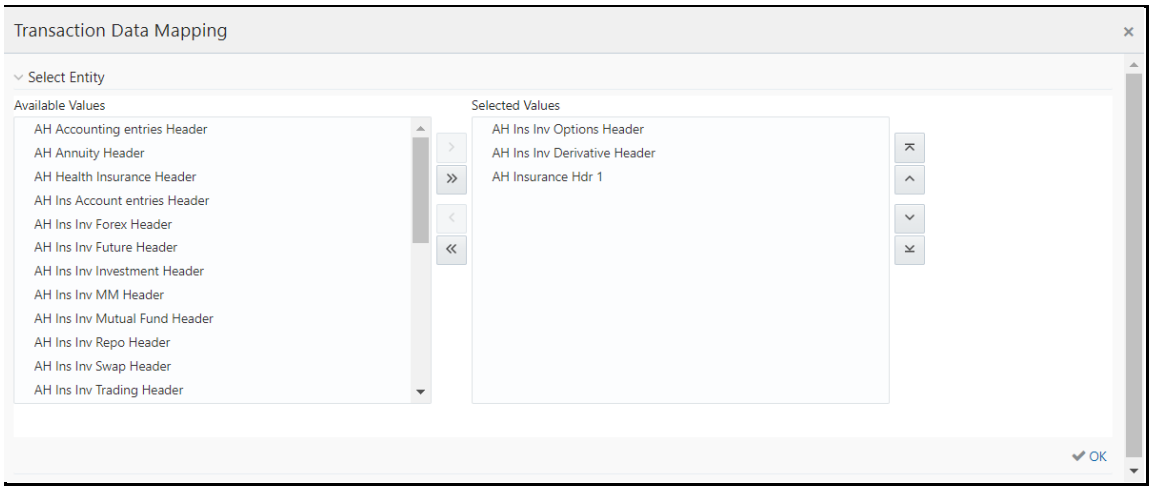
e. Click **Properties**  or  . The Connector Details screen appears.



f. Enter the name and description and click Publish / Save / Save As Draft. You can also publish the connector under the Subledger Data Mappings screen. The published connector is displayed under the Data Mappings screen.

g. Click 

. The Transaction Data Mapping screen appears. It displays all the header related connectors which are saved under AHC. You can add the required connector for the Subledger listed here.



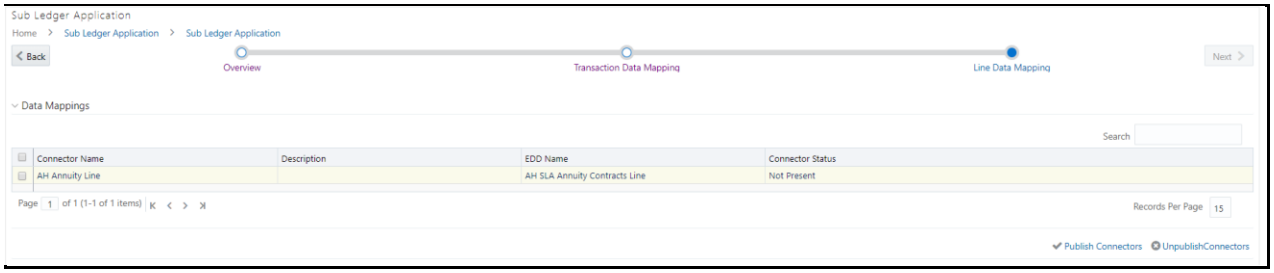
i. Click  to move the required entities to the list of the selected values.

ii. Click  to move all the entities to the list of the selected values.

iii. You click  to move the entities up and down.

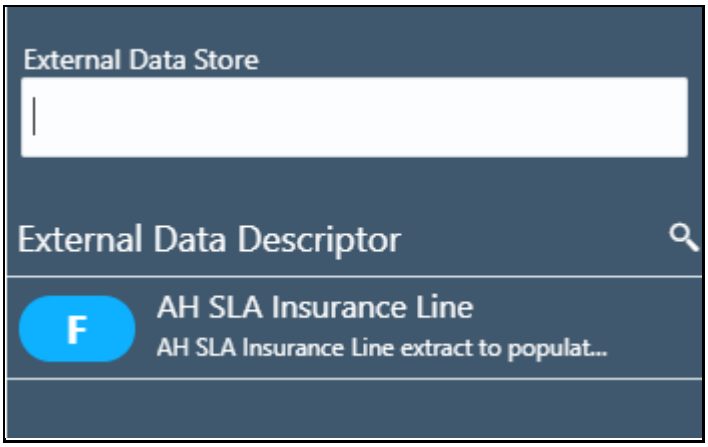
iv. After the entities are selected, click OK.

3. Click **Next**. The Line Data Mapping screen appears.

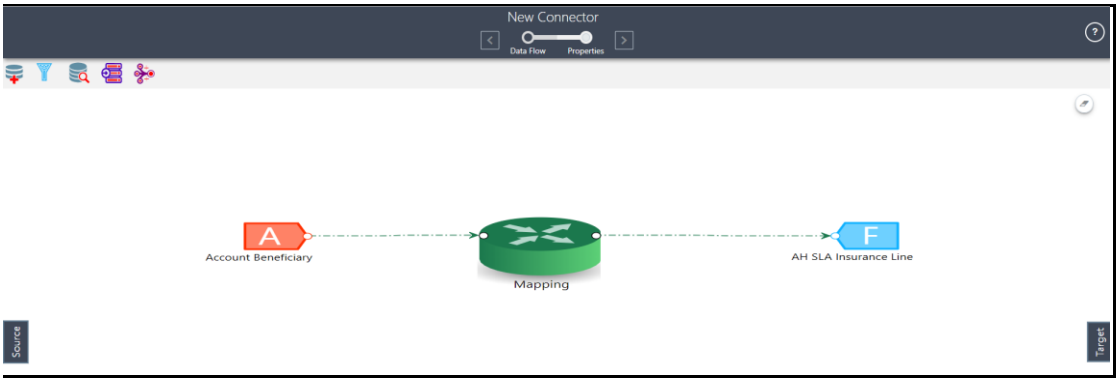


a. Click **Create** to add a new data mapping. The New Connector screen appears.

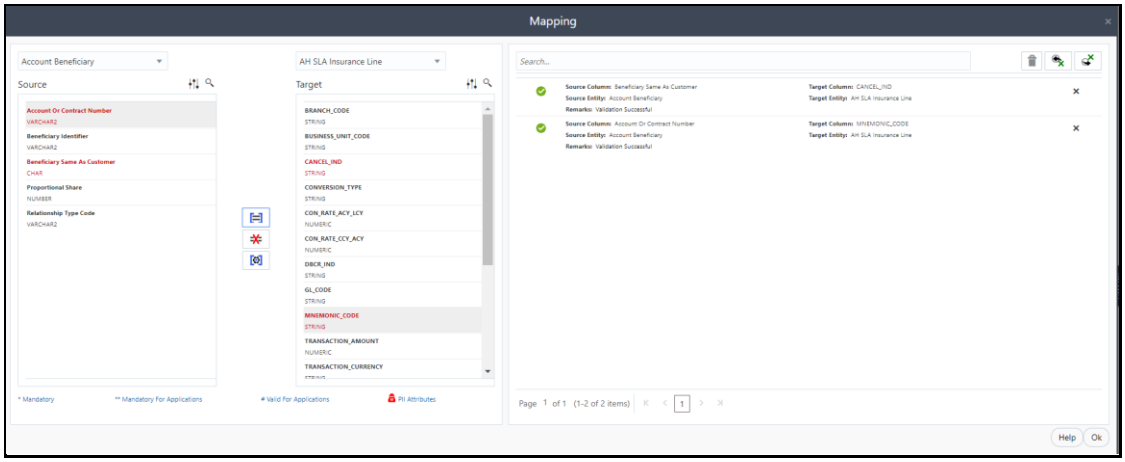
b. Under **Target**, the EDD is automatically created. For example, Subledger's name is Commercial Banking. The EDD is created with AHC SLA as the prefix of the name and Line as the suffix. The full name is “AH SLA Life Insurance Line”. EDD structure will have all the attributes that have been mapped to the Subledger under the Line Information.



c. Map the required source and target, and save the connector.



d. Double-click **Mapping** to map the source and target.



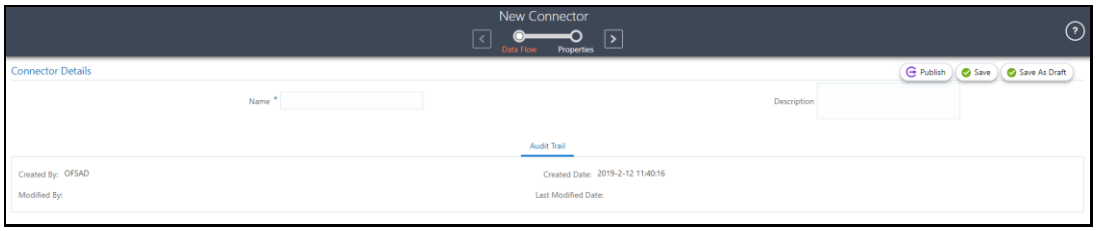
NOTE

You must apply filters if any while creating connectors to fetch the correct data from a product for a particular Subledger. Extraction Date filter is already enabled with pre-packaged connectors for executing them.

For more information on mapping, see the Connectors section in the DIH User Guide.

e. Click **Properties** 

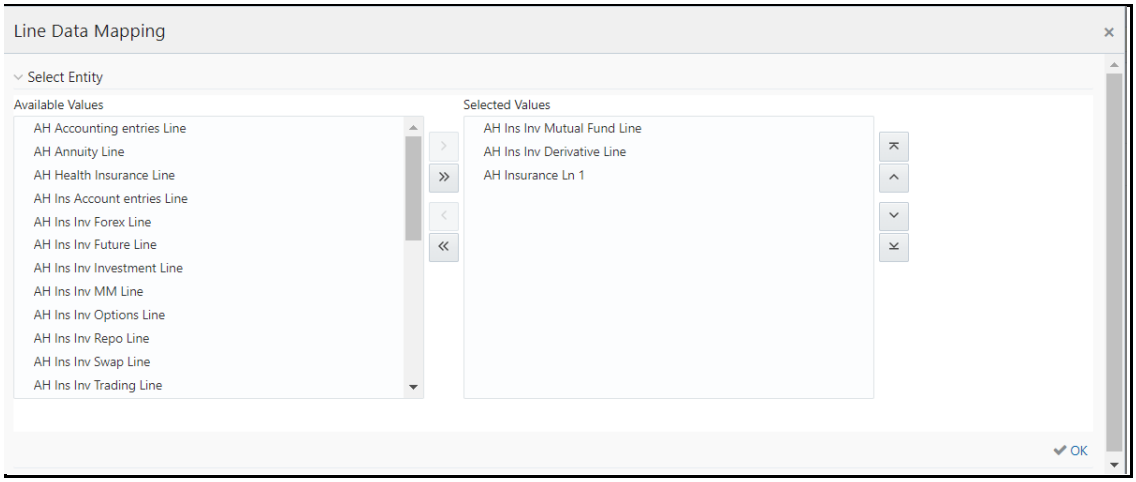
or  . The Connector Details screen appears.



f. Enter the name and description and click Publish / Save / Save As Draft. You can also publish the connector under the Subledger Data Mappings screen. The published connector is displayed under the Data Mappings screen.

g. Click 

. The Line Data Mapping screen appears. It displays all the line related connectors which are saved under AHC. You can add the required connector for the Subledger listed here.



i. Click  to move the required entities to the list of the selected values.

ii. Click  to move all the entities to the list of the selected values.

iii. You click  to move the entities up and down.

iv. After the entities are selected, click OK.

h. Click Save. The message, “SLA and Connector saved successfully” appears.

i. Click 

to publish all the Transaction and Line EDD related connectors. After publishing successfully, a batch is created automatically and the batch name is displayed in the popup up message(<INFODOM>_DIH_AH_<SLA_CODE>).

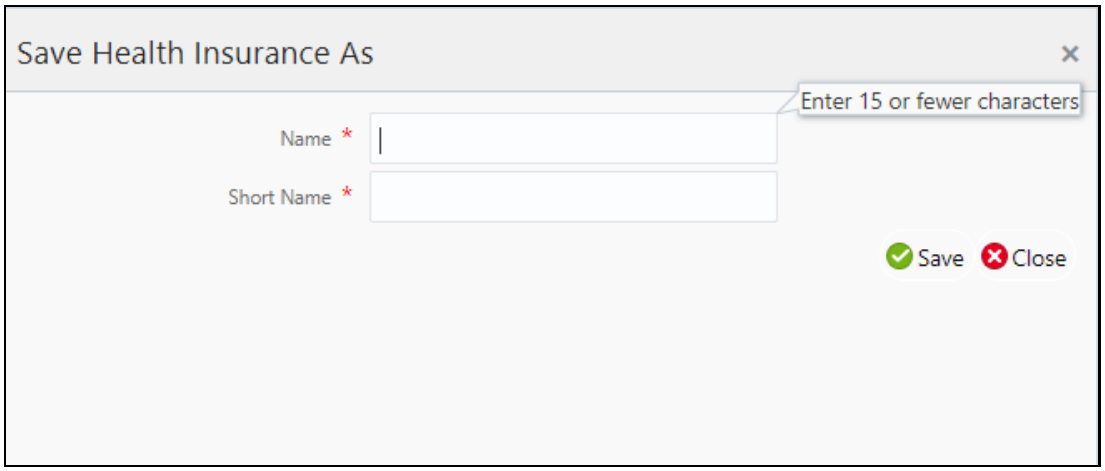
j. In case you wish to unpublish the connector, click UnpublishConnectors.

k. After unpublishing a connector, select the required connector row checkbox and click Edit Data Mapping to edit a required connector.

7.5.6 Copying a Sub-Ledger Application

To copy a Sub-Ledger:

1. Select one **Sub-Ledger** and click Copy a sub-ledger. Save “sub-ledger” As screen appears.



2. Enter the Name and Short Name and click **Save**. This copies all the properties except the name and short name as it has to be unique.

   For example, the connector names before copy are AH Com Bill Contract Header AH Com Casa Header, AH Com Commitment Header and so on. After copying, it changes to AH (copied SLA name) succeeded by Hdr1, Hdr2, and Hdr3.

   Similarly, for line mapping, the name after copy shows as AH <SLAname> Ln1.

3. The EDD name is also replaced with the copied name in the connector screen in the format AH SLA <SLA name> Header and AH SLA <SLA name> Line.

7.6 OFSAA Chart of Account (COA) Mapping

COA segments are pre-defined in the cloud environment before loading data to the STG_GL_DATA and management ledger table through connectors. You must map the relevant COA segments to the OIDF columns for loading data, as defined in the Accounting Hub Cloud Service environment. For example, if COA segment 1 has been mapped to Legal Entity attribute in a cloud environment while defining account rules, the same should be mapped in the DIH screens Legal Entity to COA_SEGMENT1. The mapping screen for the COA segment displays different dimensions to allow you to map them to respective COA segments.

The ledger balance data from AHC comes in the form of COA segment columns. COA segments are defined differently for different users and hence there is an option to select which dimension represents which segment.

The Subledger COA Mapping screen displays the seeded dimension names which are a part of the data model and it is possible to select dimensions against each COA segment as required.

The following list of pre-defined dimensions are needed in the application and which are mapped to COA segments:

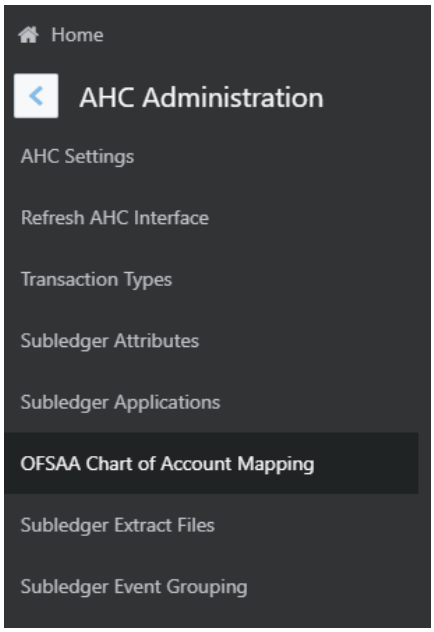
- General Ledger Account Code
- Legal Entity Code
- Account Branch Code
- GAAP Code
- Currency Code
- Product Code
- Organization Unit Code
- Business Unit Code
- Customer Class Code

7.6.1 Mapping Segments

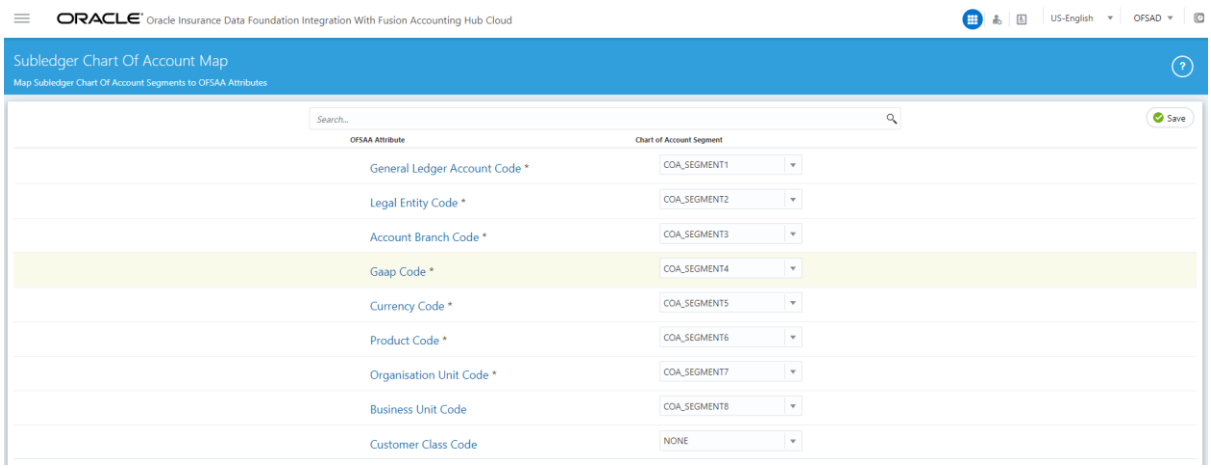
Depending on the COA or accounting principles followed, the OFSAA pre-defined dimensions can be mapped with the list of COA segments.

To map the segments, perform the following steps:

1. Navigate to **Oracle Insurance Data Foundation Integration With Accounting Hub Cloud Service** > **AHC Administration** > **Subledger COA Mapping**.



The Subledger Chart Of Account Map screen appears.



OFSAA Attribute column displays, the list of pre-defined dimensions and Chart Of Account Segment displays, the COA Segment value with a drop-down list of COA segments 'COA_SEGMENT1', 'COA_SEGMENT2' and so on.

Select the 'NONE' option when the dimension is not required to be mapped/used.

2. Select the required COA segments and click **Save**. These segment values are replaced with the dimension columns as mapped in the inbound connector and the data is loaded as per the mapping into the staging tables of the ledger. Once it is saved the Insert connectors automatically get mapped for both GL and SR.

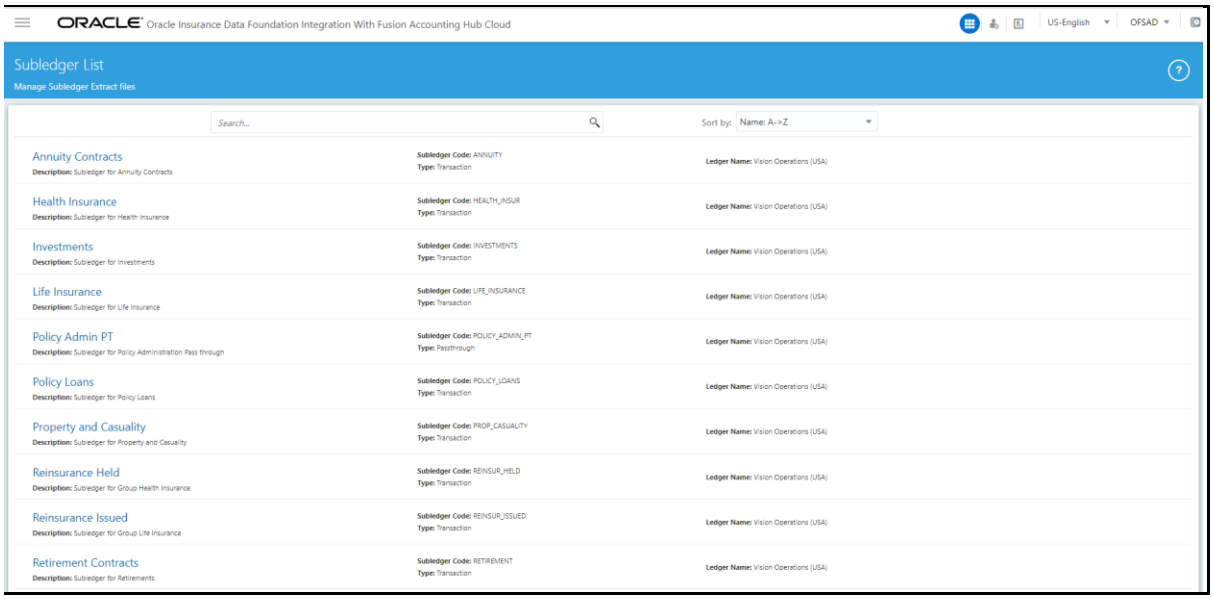
> **NOTE**
>
> - Specify the COA segment as the attributes in OFSAA. For example, if you map the COA Segment 1 to GL Code, all the extracts from FAH with column header as COA Segment 1 are loaded as GL Code in OFSAA.
> - The pre-defined dimensions with * must be mapped to some of the other COA segment to get values, as they are not null columns

7.7 Subledger Extract Files

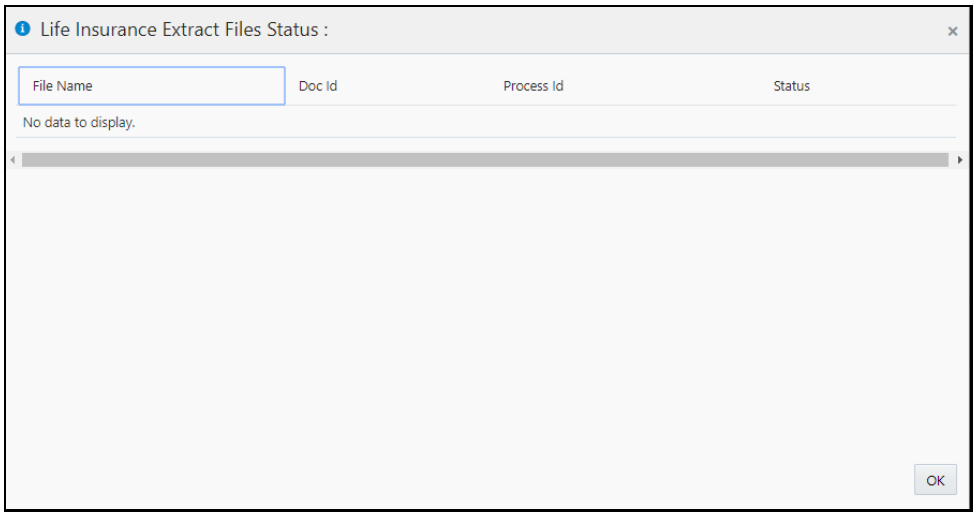
During batch execution of Subledger connectors, by default, the sixth parameter of the last task is set to N (For more information, refer Batch Execution which generates a .zip file after successful execution. When the parameter is set to Y, execution generates a .zip file, uploads it to UCM and triggers the required process automatically.

In both cases, this screen can be used to manage the extracted zip files.

The Subledger Extract Files screen has all the list of Subledgers, where you can select individual Subledger and upload required files to UCM or further process them if files are already uploaded.



When an SLA is selected, a new window opens which contains the list of files uploaded for that SLA.



It has four columns:

- File Name: The files uploaded appear under the File Name column. Click the file name to download the file to a local system.
- Doc ID: For the extracted .zip files, where batch execution is completed with the sixth parameter as N has ‘Upload to UCM’ option in this field for the first time. You can click this and the application uploads the created .zip file to UCM. Once it is uploaded, a unique Doc ID number is created and it appears in this column. For files executed with parameter 6 as Y and parameter 7 as 1, a unique doc ID number appears upfront.
- Process ID: For the extracted zip files, where batch execution is completed with the sixth parameter as Y and the seventh parameter as two, it has a unique Process ID number in this column. Those files, which are executed with the sixth parameter as N or Y (along with the seventh parameter as 1), has the ‘Process File’ option in this field for the first time. Process File will either upload the .zip file to UCM and trigger the process/job ‘Import Accounting Transactions’ or directly trigger the uploaded file with the Import Accounting Transactions process. Click ‘Process File’ and the following actions occur:

- If Doc ID number exists – .zip file, which is already uploaded to UCM under the existing Doc ID, will be triggered for processing and a unique Process ID is created, which appears under this column after a successful trigger.
- If Doc ID says 'Upload to UCM' – .zip file that is generated is automatically uploaded to UCM and triggers the required process. This updates the Doc ID and Process ID with a unique number after successful upload and process respectively.

- Status: The status is updated in a couple of seconds, which reflects the status of the 'Import Accounting Transactions' job.

> **NOTE** Ensure to validate the Host Name and Port Number in the following file in case of any connection errors:
>
> `<FIC_HOME>/ficdb/bin/DIHGenerateZip.sh`

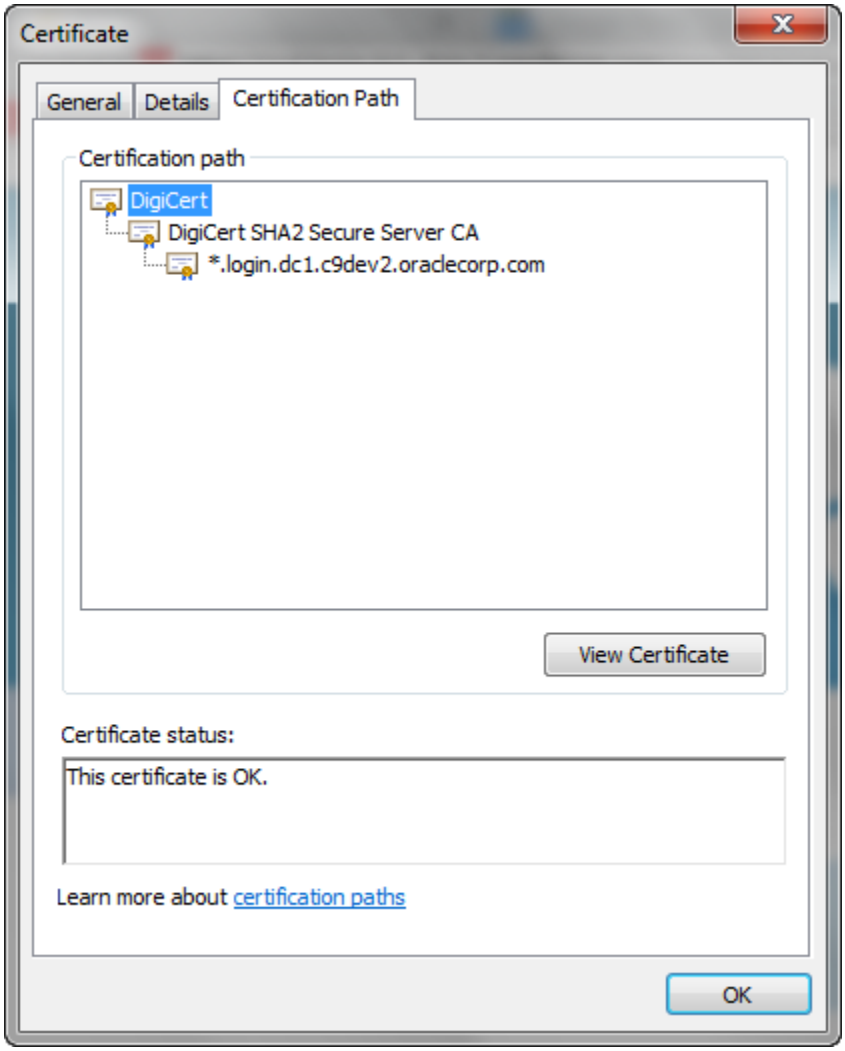
7.8 Import Certificate into Keystore

Export the certificate from the browser to the file using the following steps:

1. Access the AHC SSL URL for any web service using Microsoft Internet Explorer/Mozilla.
2. Click the Security Lock symbol and click View Certificates.

3. On the Certification Path tab, the certificate hierarchy appears.

   Export the top two certificates (DigiCert and DigiCertSHA2SecureServerCA in the following case) to file.



4. On the Certification Path tab, select DigiCert and click View Certificate.
5. On the Details tab, select Copy to File. The Certificate Export Wizard appears.
6. Click Next and enter a name and location for the file you want to export.
7. Change the encoding to Base-64 and click Next.
8. Provide the file name and click Finish.
9. Repeat steps 4 through 8 for the DigiCertSHA2SecureServerCA certificate.

   When using other web browsers, perform similar steps. The navigation path may differ in the web browsers used.
10. Type the following command to import a certificate into keystore:

    keytool -import -trustcacerts -file <filename> -alias <aliasname> -keystore keystore.jks -storepass welcome1
11. Run the following command to verify if the trust store contains the imported certificates:

    keytool -list -v -keystore <filename> -storepass welcome1 | grep -i Verizon

7.9 Subledger Event Grouping

Subledger Event Grouping functionality is introduced to group data based on a few seeded dimensions while extracting and then posting them to AHC. Grouping based on the dimensions and significance. Similarly, while loading the data from AHC to STG_GL_DATA, aggregation based on the dimensions and the basic primary key columns of the GL data table helps in maintaining the uniqueness of the table without any errors.

Grouping functionality includes combining data based on some pre-seeded columns in both header and line-level data. When data is grouped on some dimensions and transaction numbers are not considered for grouping, there is a need to regenerate the transaction numbers. As aggregate columns are different in header and line level, aggregation happens separately for header and line data. However, the only connection between the line and header data is just the transaction number. Hence, to maintain the connection, pair up the corresponding header and line connectors along with a mapping table, which maintains actual transaction numbers with a map to newly generated or replaced transaction numbers.

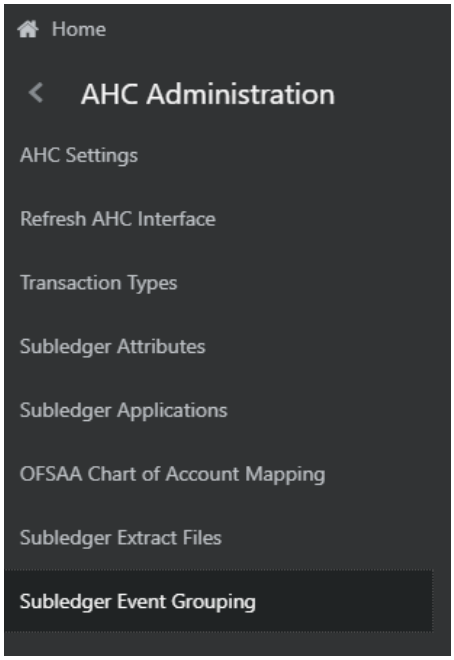
Grouping functionality includes the following components:

- Event Group Summary
- Defining an Event Group
- Managing Group Attributes

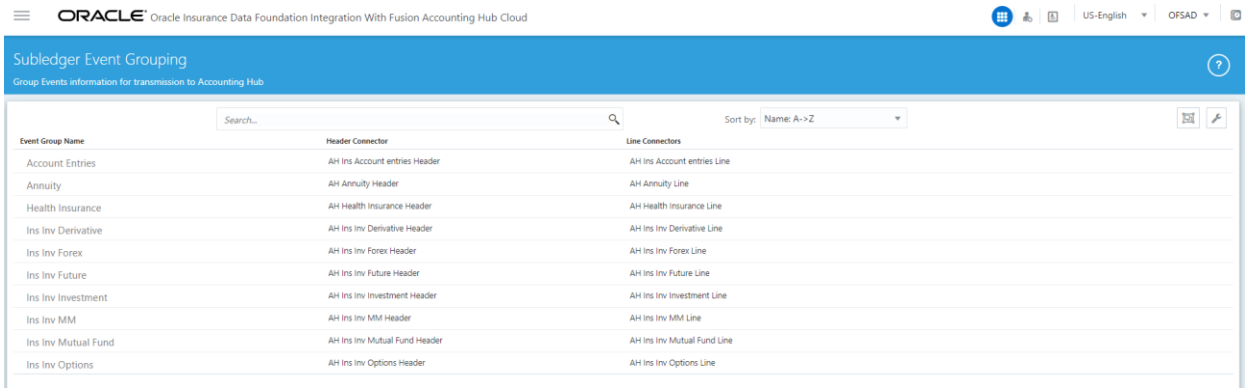
7.9.1 Subledger Event Grouping Summary

To understand the Sub-Ledger Event Grouping summary screen:

1. Navigate to the AHC application interface.



2. Select AHC Administration, and click Subledger Event Grouping. A list of pre-packaged event groups information for transmission to Accounting Hub appears.



The summary screen displays the Event Group Name, Header and Line Connectors:

a. **Event Group Name**: It is the generic name given to identify a header and line connector mapping. Pre-packaged Subledger connectors and name are displayed.

b. **Header Connectors**: Displays the header connector's name.

c. **Line Connectors**: Displays the line connector's name.

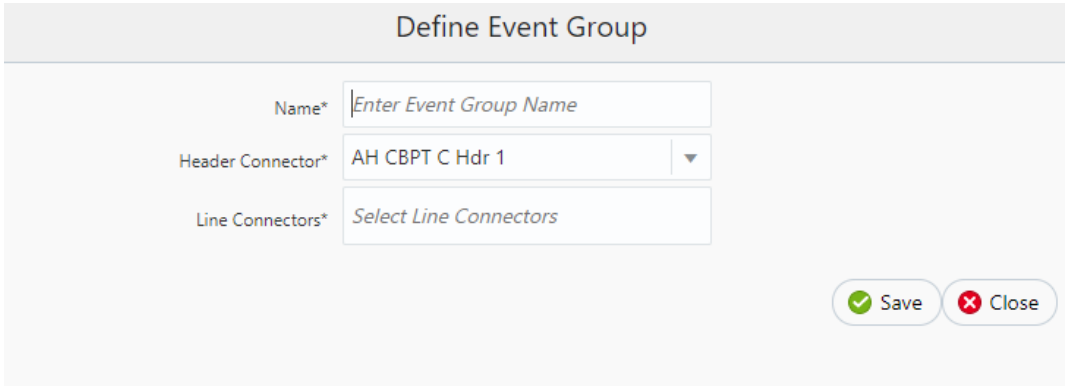
3. You can search for Sub-Ledger Event Grouping Name.
4. Under Summary, you can Define an Event Group, Manage Group Attributes and Delete user-defined group events.
5. Click  if you wish to delete a Sub-Ledger Event Group.

> **NOTE** It is not possible to delete a pre-packaged Sub-Ledger Event Group.

7.9.2 Defining an Event Group

To define an event group, perform the following steps:

1. On the Subledger Event Grouping screen, click  to define event group. The Define Event Group window appears.

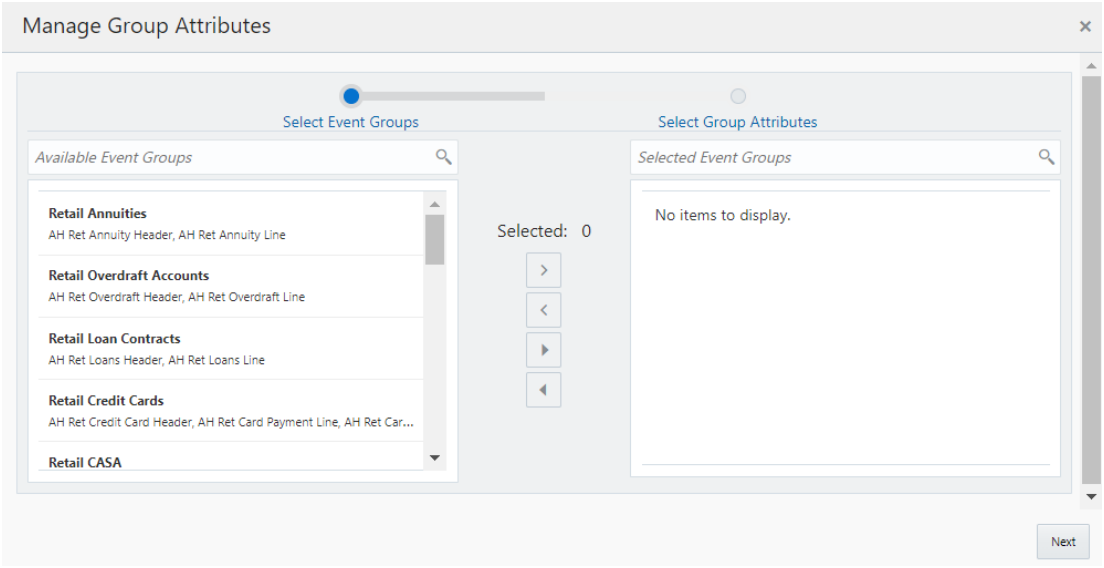


2. Enter the required details and click Save.

7.9.3 Managing Group Attributes

To manage group attributes, perform the following steps

1. On the Subledger Event Grouping screen, click  to manage Group Attributes. The Manage Group Attributes window appears displaying the available Event Groups.



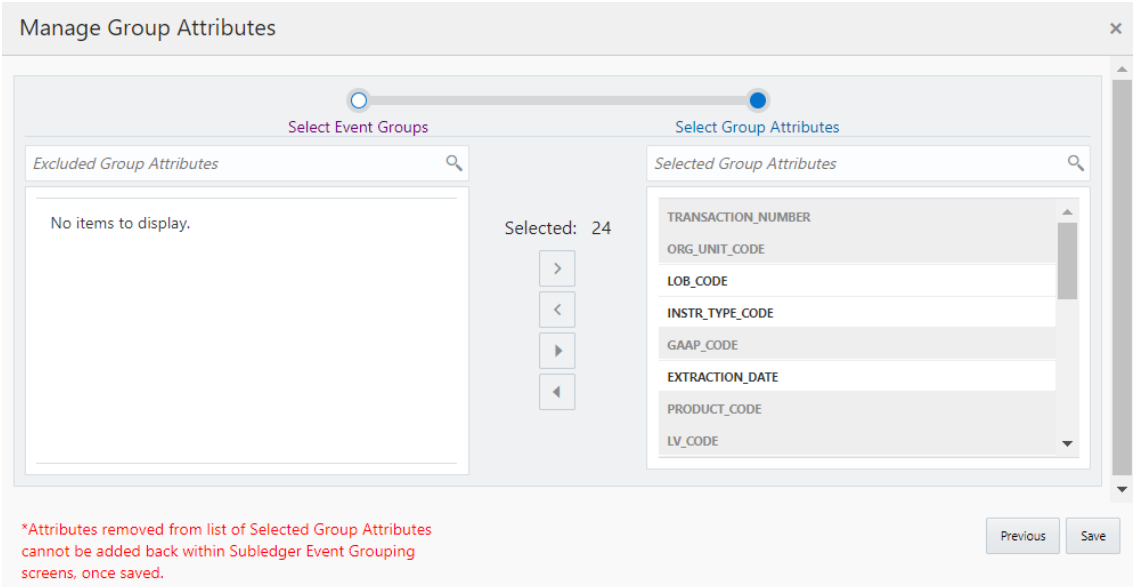
2. Select the Event Groups that are created previously, for which you want to define the group attributes.

   l. Click to move the required event groups to the list of the selected values.

   m. Click to move all the event groups to the list of the selected values.

   n. Click or to move the event groups back one by one or all respectively.

3. After the event groups are selected, click Next to move to the Manage Group Attributes screen.



Here, by default all the columns, which are a part of mappings in the selected event groups are

listed under the selected Group Attributes, that is, they are a part of the group by clause for grouping.

A list of grayed out attributes cannot be removed from the group attribute list, as they are the basic dimensions on which ledger data is required.

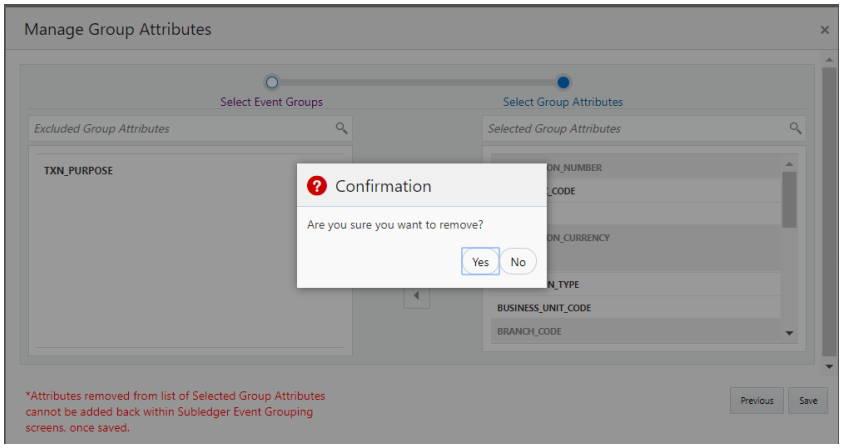
4. You can move the non-mandate columns to LHS and remove them from aggregation or group by clause.

   a. Click to move the non-mandate columns to LHS.

   b. Click to move the non-mandate columns to LHS.

   c. Click or to move the entities back one by one or all respectively.

5. Once all the required columns are confirmed, click Save. A confirmation message appears.



6. Click **Yes**. The Subledger Group Attributes are updated successfully.

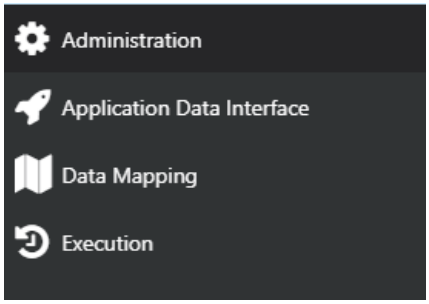
8 Preparing for Connector Deployment

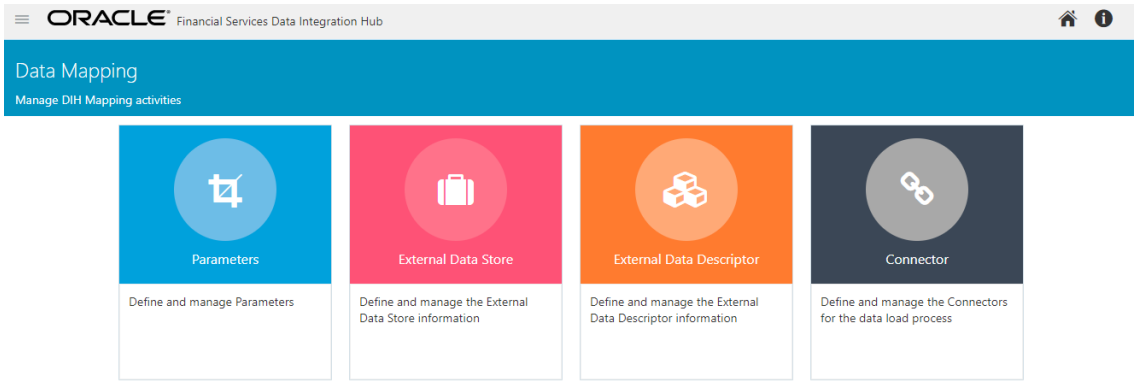
8.1 Configuration Steps

The following configuration steps are specific to OIDF Integration with AHCS and must be undertaken before proceeding to Deploy Connectors.

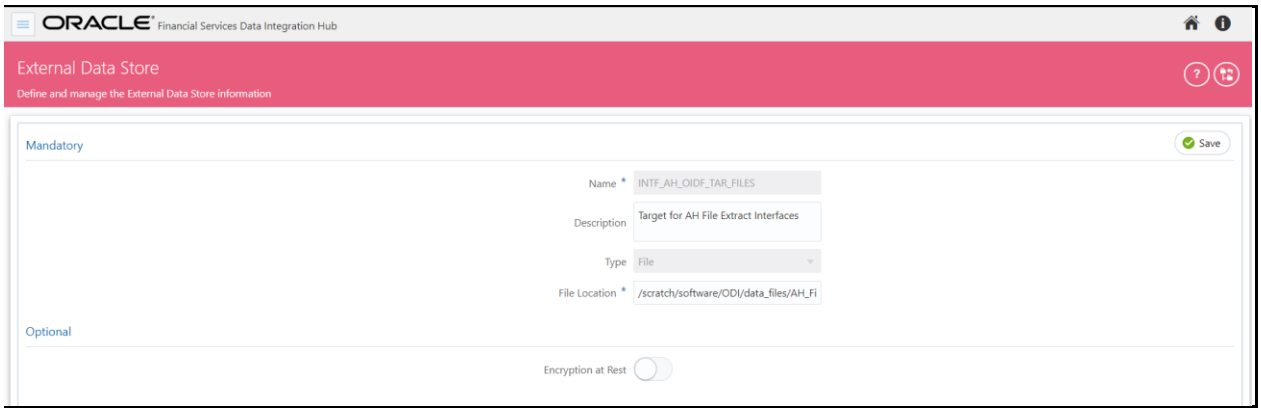
> **NOTE** ODI setup must be completed in DIH Foundation before these tasks are undertaken. Refer DIH User Guide.

1. Navigate to **Financial Services Data Foundation Integration with Accounting Hub Cloud Service** > **Data Mapping** > **External Data Store**.

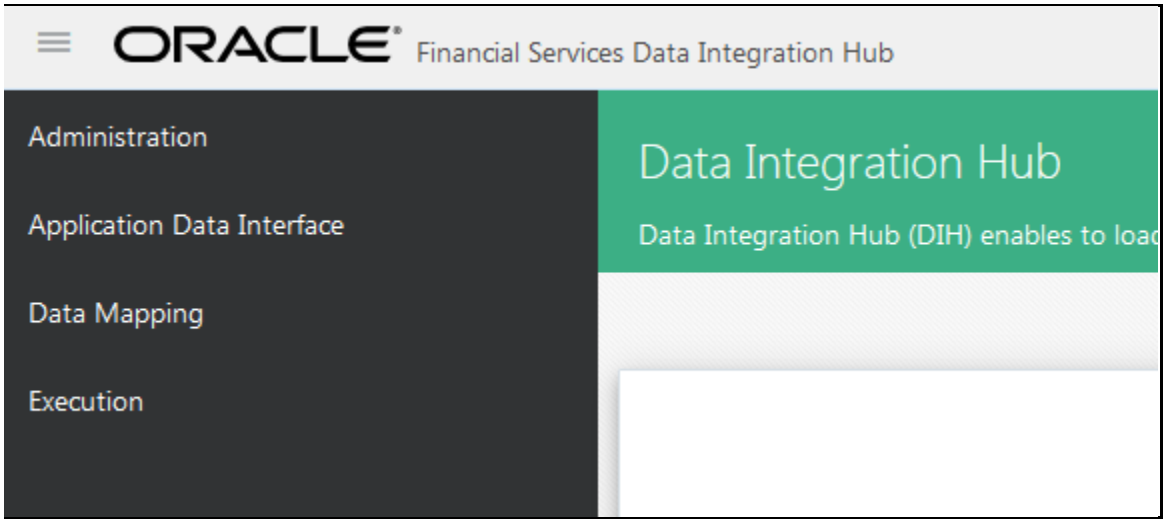




2. In **External Data Store** summary screen, search for EDS name INTF_AH_OIDF_TAR_FILES. Click this EDS to Edit.
    a. Note that this EDS refers to AHCS as a target system for Events information, relevant to Extract Connectors.



3. In the File Location field, specify the location where data files for AHCS integration are to be generated and saved. This file location **must be common to both ODI Agent and OFSAA Server**. This can be achieved in one of the following ways:
    a. ODI Agent is installed or configured in the same server as OFSAA Server
    b. The ODI Server and OFSAA Server have shared file storage with a suitable network mount.
4. Navigate to DIH Application, click the menu.

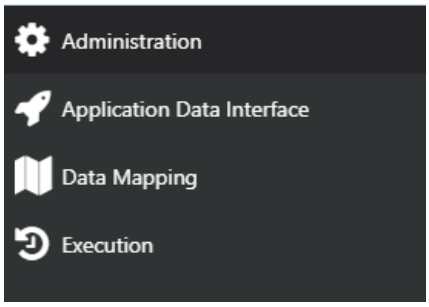


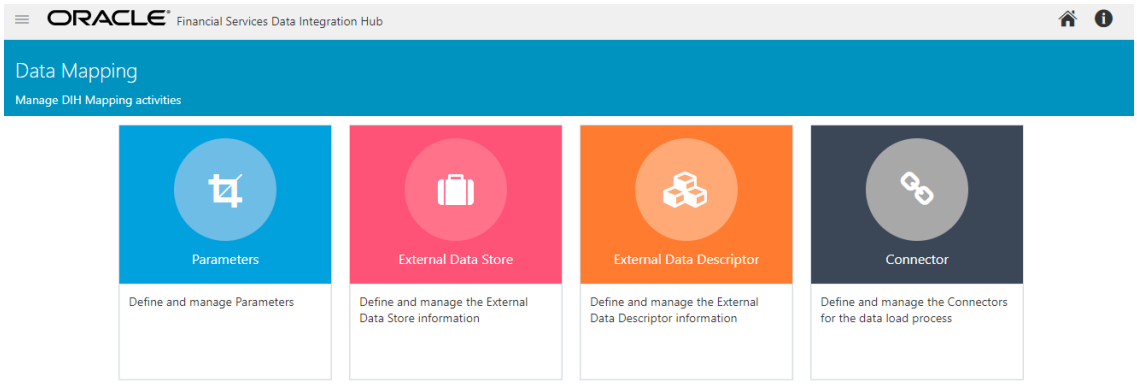
5. Click the Application Data Interface.
6. On the RHS click Refresh ADI.
7. Click Start . This refreshes all the Application Data Interfaces and creates the Application Data Interfaces for all the staging tables present in the model which is being uploaded in the same Infocom.

8. Navigate to DIH Application, click the menu.
9. Click the Application Data Interface.
10. On the RHS click Target Datastore Refresh.
11. Click Start . This refreshes all available target data stores.

> **NOTE** Refer to DIH User Manual to obtain detailed information for any of the steps mentioned above.

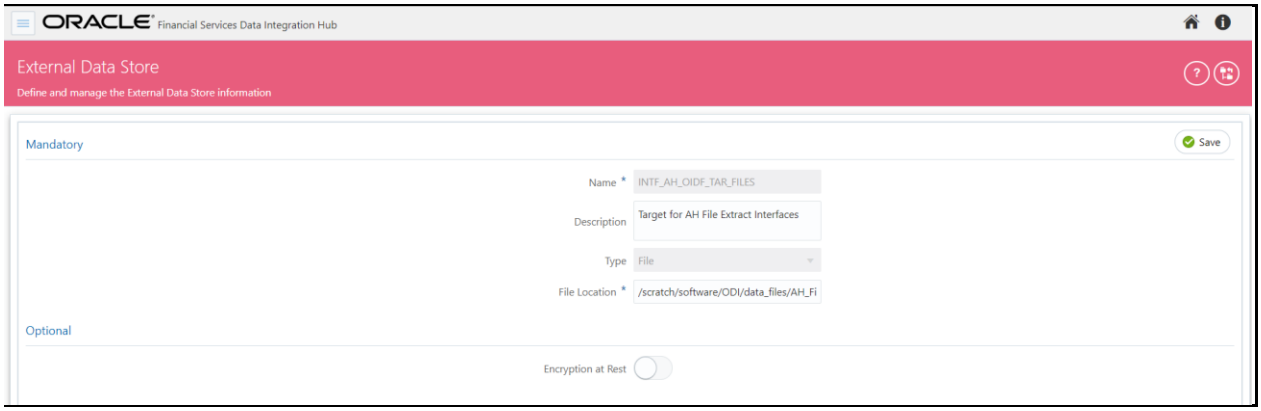
12. Navigate back the Data Store summary, following Financial Services Data Foundation Integration with Accounting Hub Cloud Service > Data Mapping > External Data Store.





13. In External Data Store summary screen, search for EDS name INTF_AH_OIDF_STAGE_SRC. Click this EDS to Edit.

a. Note that this EDS refers to AHCS as a source system for GL and SR balance information, relevant to Insert Connectors.



In the File Location field, enter the location where data files from AHCS are saved. This file location **must be common to both ODI Agent and OFSAA Server**.

9 Obtaining Balance Information from AHCS

This chapter provides details of the process that facilitates balance information against GL Chart of Accounts and with Support Reference information to be extracted from AHCS and ingested into OIDF.

9.1 Configuration and Data Files

This section details the configuration files and data files required before the execution of AHC Connectors.

9.2 Chart of Account to OFSAA Dimension Mapping

See the OFSAA Chart of Account (COA) Mapping section for more information about how this mapping must be done.

9.3 GL and SR Balance Information

The extraction of balance information from your AHCS instance is done through the Oracle BI Cloud Connector. The specific set of balance and statistical balance figures available is specific to your instance of AHCS.

9.3.1 GL and XLA Data CSV Files from AHC

The following files must be downloaded from AHC along with Header Names and must be available under the EDS path before executing the Insert Connectors:

1. ah_gl_balance.csv
2. ah_sr_balance.csv
3. ah_gl_code_combination.csv
4. ah_sr_ref_combination.csv

See the Downloading Files for Insert Connectors section on how to download the CSV files to the EDS path.

9.3.2 Downloading Files for Insert Connectors

The required files can be extracted first to UCM using the BI Cloud console in ERP. An extract must be defined in the BI Cloud console for the following BIV objects.

- FscmTopModelAM.FinXlaBalInqSuppRefBalAM.SuppRefCombinationsPVO
- FscmTopModelAM.FinGlAccountsCodeComboAM.CodeCombinationPVO
- FscmTopModelAM.FinGlInquiryBalancesAM.BalancePVO
- FscmTopModelAM.FinXlaBalInqSuppRefBalAM.SupportingReferenceBalancePVO

The extract path must be chosen as Universal Content Manager (UCM) for the BI Cloud console extract and specified columns must be selected in a specified order according to the above-listed sheets. After configuration, follow these steps to complete the integration:

1. Download a sample extract file for all the BIV Object extracts.

2. If the column order is not matching, then update the column order in the EDDs as per the respective data files.

9.3.3 GL and SR Balance Extraction

After all the previous data file configurations are completed, the following batches can be executed to load the GL and Supporting Reference data into OFSAA.

1. AH_GL_SR_EXTRACT_DOWNLOAD
2. AH_CONNECTORS_70121

> **NOTE** For some unknown reasons, when deployed in the WebLogic server, not all batches are created. Follow these steps to avoid the issue:
>
> 1. Log in to the WebLogic server admin console.
> 2. Navigate to **Console**, select **Services**, and then select **Data Sources**.
> 3. Select **Atomic Schema.DS** and then select **Connection Pool**.
> 4. Update **Statement-Cache-Size** value to 0.
> 5. Restart the WebLogic server and redeploy connectors from the **AHC Refresh Interface** window.

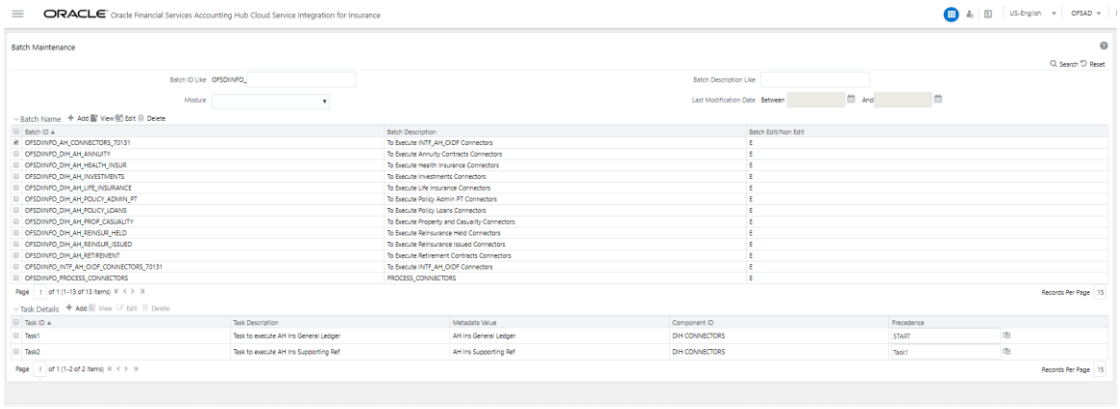
See the DIH User Guide for more information on publishing and executing the DIH Connectors.

9.4 Executing AHC Insert Connectors

After completing the steps in the Chart of Account to OFSAA Dimension Mapping and Obtaining Balance Information from AHCS sections, execute the following steps to consume balance information extracted from your instance of AHCS:

1. From the **Oracle Insurance Data Foundation Integration With Accounting Hub Cloud Service** window, select **Accounting Hub Cloud Service**, select **Orchestration**, and then select **Batch Execution**.
2. Publish **AH GL Balances Insert Con** and **AH SR Balances Insert Con Connectors**.
3. In each execution sequence or period, navigate to **Batch Orchestration** and search for auto-generated batch AH_CONNECTORS.

Figure 1: Batch Maintenance Window



> **NOTE** For some unknown reasons, when deployed in the WebLogic server, not all batches are created. Follow these steps to avoid the issue:
>
> 1. Log in to the WebLogic server admin console.
> 2. Navigate to **Console**, select **Services**, and then select **Data Sources**.
> 3. Select **Atomic Schema.DS** and then select **Connection Pool**.
> 4. Update **Statement-Cache-Size** value to 0.
> 5. Restart the WebLogic server and redeploy connectors from the **AHC Refresh Interface** window.

4. Select **AH GL Balances Insert Con Task** and click **Edit**. The Batch Parameters window is displayed.
5. In **Batch Parameters** specify,
   a. PERIOD_NAME for which GL Balances must be extracted (for example, PERIOD_NAME='Aug-18').
   b. AH_CONSOLIDATION_FLAG (for example, AH_CONSOLIDATION_FLAG='C').
   c. Update the MIS_DATE format to the extracted date format of the ACCOUNTING_DATE column present in the `ah_gl_balance.csv` file.

      For example, if the date is 08-Oct-18, then the format is MIS_DATE='$MISDATE:dd-MMM-yy'.
6. Repeat steps 3 through 5 for **AH SR Balances Insert Con**.
7. Navigate to **Batch Execution**, search for **AH_CONNECTORS**, specify **FIC_MIS_DATE**, and execute the batch.

   See the DIH User Guide for more information on publishing and executing the DIH Connectors.

10 Publishing Events Data to AHCS Using Extract Connectors

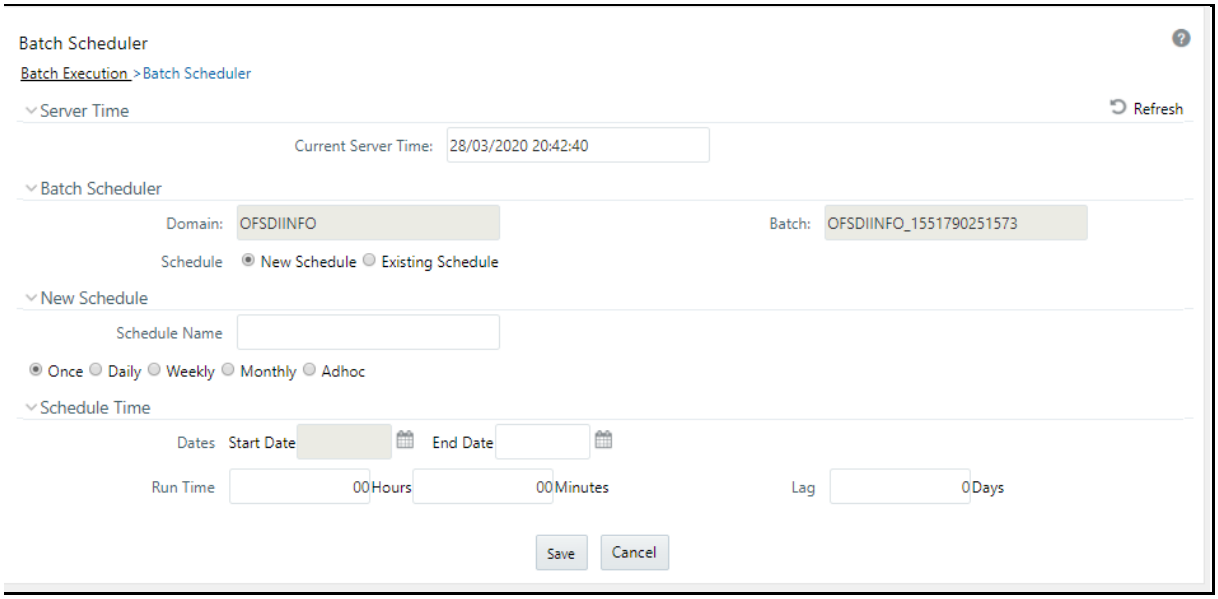
10.1 Batch Execution

> **NOTE** Ensure to have all the required configuration, including accounting rules, completed in your instance of AHCS before attempting to send events data using this process.
>
> Read section Batch Re-Execution carefully before you proceed to Batch Execution.

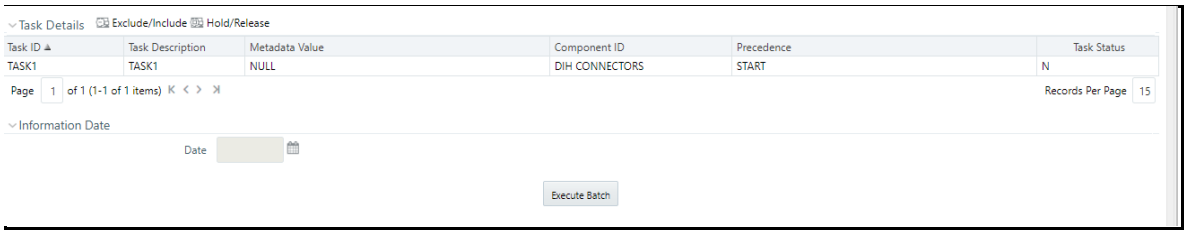
In the Batch Execution window, there are batches created for each Sub Ledger. These batches are created when you publish SLAs, as detailed in the Subledger section of this guide.

To execute a batch, perform the following steps:

1. Select a batch from the list of batches.
2. Click **Schedule Batch**. The Batch scheduler window appears.

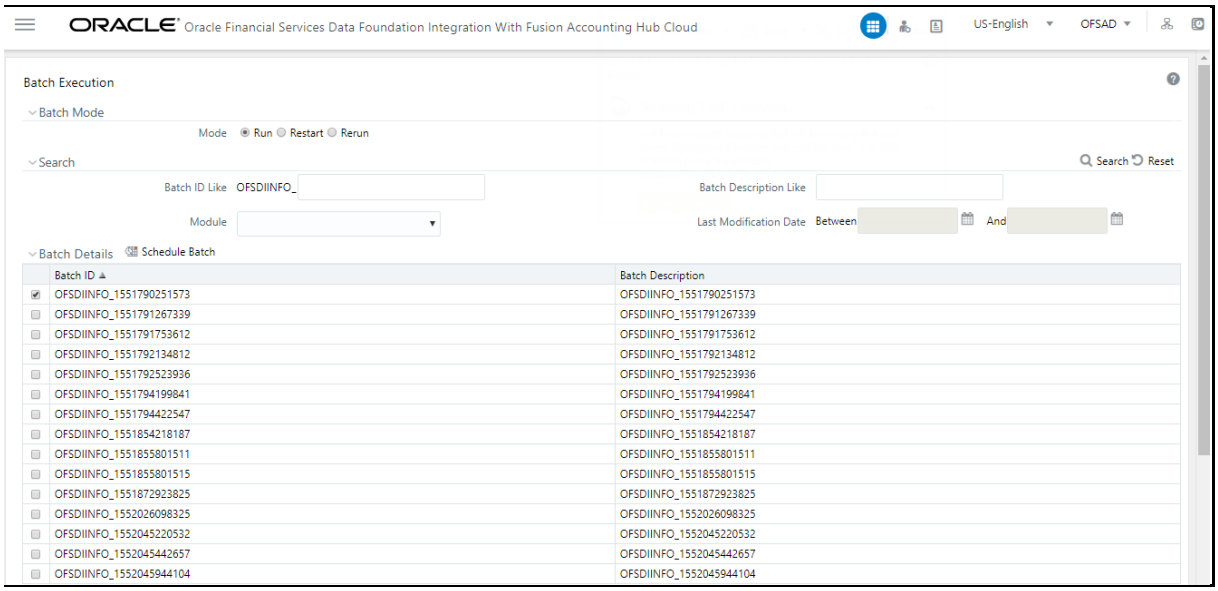


3. Enter the details as of which data must be processed and click **Save**.



4. Click **Execute Batch**.

5. Refer AAI User Guide for details on batch execution, run-time parameters, and monitoring.



Each SLA batch consists of three Connector execution tasks – a Connector execution task that processes Event Grouping, one that prepares the Header file and a third that that prepares the Line file – and a fourth task that runs an executable file. All tasks specified within the batch are to be executed.

This “Run Executable” task performs the following actions:

1. Identifies the extracted Header and Line CSV files.
2. Formats data by removing duplicate Headers rows in both Header and Line files.
3. Generates the Metadata.txt file with SLA details.
4. Creates a zip file including Header, Line, and Metadata files.
5. Refer the following structure to locate and identify ZIP files generated:

```
/<EDS_PATH>/<SLA_CODE>/XlaTransaction_<SLA_CODE>_<MIS_DATE>_<TIMESTAMP>.zip
```

6. For every execution, intermediate files are copied to the temp folder along with a log file.

By default, AHCS disables automated triggering of events data processing once files have been uploaded via its API to UCM. This can be enabled by updating the sixth run-time parameter of the final Run Executable task to “Y” before executing the batch.

With this enabled, the seventh parameter of this task may be set to “1” (whereupon ZIP file is uploaded to UCM and no further action is taken) or “2” (whereupon ZIP file is uploaded to UCM and Import job in AHCS is triggered to process uploaded file) as appropriate for your purposes.

10.2 Batch Re-Execution

OIDF Integration with AHCS does not currently support incremental processing. Each execution causes all events data relevant for the MIS Date specified while executing the aforementioned batches, subject only to filters defined while setting up SLAs / related Connectors, to be extracted from Staging entities and published to AHCS.

In other words, batch processing can be performed only once per SLA for any given MIS Date. Re-execution may cause duplicated events data to be sent to your instance of AHCS, with no provision for automated roll-back.

OFSAAI Support Contact Details

- Contact Infrastructure support at https://flexsupp.oracle.com if you have installed ERM and FCCM applications.
- Raise an SR in https://support.oracle.com if you have any queries related to EPM applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site which has all the revised/recently released documents.