

**Oracle Financial Services Data
Foundation Integration with
Accounting Hub Cloud Service**

User Guide

Release 8.1.0.0.0

June 2020

F32441-01

ORACLE
Financial Services

Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud Service User Guide
Copyright © 2021 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information on third party licenses, click [here](#).

Document Control

Version Number	Revision Date	Change Log
01	June 2020	Created the document with instructions for the usage of the OFS Data Foundation Integration with Accounting Hub Cloud Service Release 8.1.0.0

Table of Contents

1	About the Guide	7
1.1	Scope of the Guide	7
1.2	Intended Audience	7
1.3	Prerequisites	7
1.4	Related Information Sources	7
1.5	Abbreviations	7
1.6	Reference Documents	8
2	Introduction	9
2.1	About Financial Service Analytical Applications	9
2.2	About Accounting Hub Cloud Service	9
2.3	Objectives	9
2.4	About Financial Services Data Foundation Integration with Accounting Hub Cloud Service	10
2.5	OFSAA-AHC Interface Architecture Overview	11
2.6	OFSAA-AHC Integration Overview	11
2.7	Data flow for FSDF Integration with AHCS	13
3	Mapping the OFSAA User to AHC User Groups	14
4	Accessing the AHC-OFSAA Interface	16
5	Component Details	17
5.1	Packaged Subledger Application Information	17
5.2	Registering with AHCS	18
5.3	DIH Connectors	19
5.3.1	Extract Connectors	19
5.3.2	Insert Connectors	20
6	Implementation	21
6.1	Background	21
6.2	Implementation Guidelines	21
6.3	Transaction Object Population	22
7	AHC Administration	23

7.1	AHC Settings	23
7.1.1	Refresh AHC Interface	24
7.1.2	Deploying Connectors or Undeploying OFSAA-AHC Connectors	24
7.2	Transaction Types	26
7.3	Subledger Attributes	28
7.3.1	Adding a Subledger Attribute	28
7.4	Subledger Applications.....	29
7.4.1	Subledger Application Summary.....	30
7.4.2	Adding a Subledger Application.....	31
7.4.3	Registration of SLAs with Accounting Hub Service	36
7.4.4	Steps to Implement Flexible Accounting	39
7.4.5	Using Data Map	41
7.4.6	Copying a Subledger Application	43
7.5	Standardization of Product Codes and Mapping with Subledger Applications.....	44
7.5.1	Filtering Data Based on Product Codes.....	44
7.5.2	Managing Product Code Filters	45
7.6	OFSAA Chart of Account (COA) Mapping.....	46
7.6.1	Mapping Segments.....	46
7.7	Subledger Extract Files	47
7.8	Importing Certificate into KeyStore	49
7.9	Subledger Event Grouping.....	50
7.9.1	Subledger Event Grouping Summary.....	50
7.9.2	Defining an Event Group.....	51
7.9.3	Managing Group Attributes	51
8	Preparing for Connector Deployment.....	53
8.1	Configuration Steps.....	53
9	Obtaining Balance Information from AHCS	55
9.1	Configuration and Data Files.....	55
9.1.1	Chart of Account to OFSAA Dimension Mapping	55
9.1.2	GL and XLA Data CSV Files from AHC.....	55
9.1.3	Downloading Files for Insert Connectors	55
9.1.4	GL and SR Balance Extraction	55

9.2	Executing AHC Connectors.....	56
10	Publishing Events Data to AHCS Using Extract Connectors.....	58
10.1	Configuration of Metadata	58
10.2	Batch Execution.....	58
10.3	Batch Re-Execution	60

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 in this user guide.

1.1 Scope of the Guide

This user guide explains the features and functions of FSDF 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 must be performed in AHCS to support accounting rules, journals, and general ledger. See the [Oracle ERP Cloud documentation](#) (**Learn More > Documentation Library**) on AHCS for more details.

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

- See the [DIH Installation Guide](#) for details on OFSAA components, Oracle applications, and environment that constitute technical prerequisites to deploy and use FSDF 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 Related Information Sources

Along with this user guide, you can also refer to the following documents in [OHC Documentation Library](#):

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

1.5 Abbreviations

The following table defines the abbreviations used in this guide.

Table 1: Abbreviations

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	Subledger Application / Subledger Application
UI	User Interface
UCM	Universal Content Manager

1.6 Reference Documents

The following are the reference documents available in this guide.

- [Subledgers and Configurations](#)
- [SLA Template Validations](#)
- [Extract Connectors Structure](#)
- [Insert Connectors Structure](#)
- [SLA Coverage](#)
- [Pre-packaged Subledger Configurations](#)
- [List of Seeded Product Codes](#)

2 Introduction

2.1 About 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 models, 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

Accounting Hub Cloud Service (AHCS) is an accounting integration and reporting platform in Oracle cloud that includes products, such as Subledger 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 Objectives

Integration of AHC and OFSAA provides a basis for a unified finance and risk architecture to the financial services industry. The benefits of having a ready-to-use interface between AHC and OFSAA are as follows:

- 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 are 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 Financial Services Data Foundation Integration with Accounting Hub Cloud Service

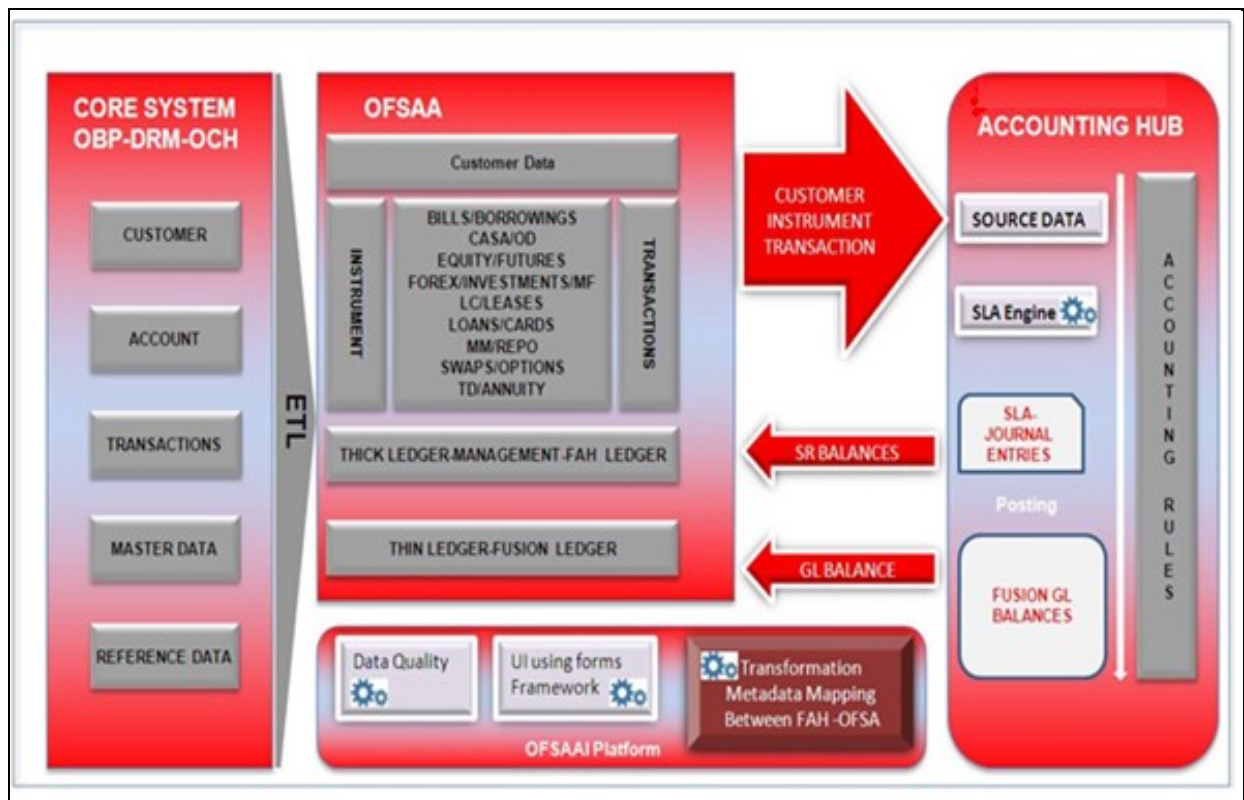
The following are the core constituents or functions of FSDF Integration with AHCS:

- Preconfigured Subledger Applications:
 - Retail Banking
 - Commercial Banking
 - Investment Banking
 - Islamic Banking
 - Treasury
 - Core Banking (using pass-through)
- Automated population of the source system and Subledger Application registration with AHCS.
- The configurable mechanism for grouping events data.
- The prebuilt mechanism for publishing events information to Accounting Hub Cloud Service.
- User interface allowing configuration of Subledger Applications and associated events information.
- The prebuilt mechanism to ingest balance information against the AHCS chart of accounts (GL Balance) and with Supporting Reference information (SR Balance).

2.5 OFSAA-AHC Interface Architecture Overview

The OFSAA-AHC Interface Architecture is shown in the following illustration.

Figure 1: OFSAA-AHC Interface Architecture



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

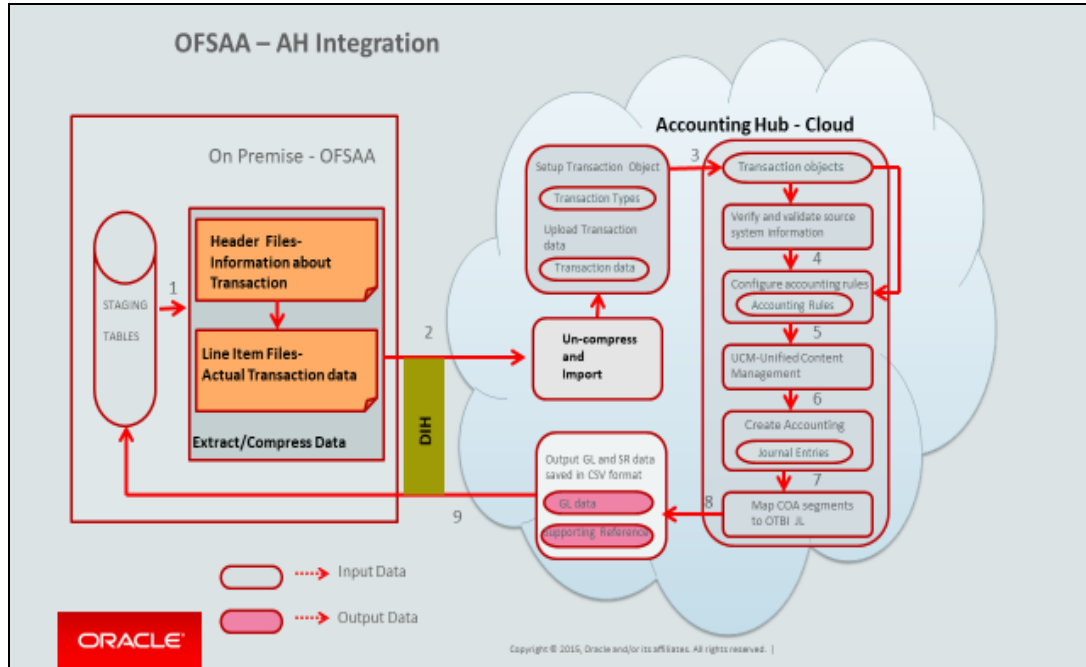
Banking data is extracted from OFSAA to AHC through DIH connectors (pre-packaged or custom) and is processed within the AHC using accounting rules. The updated GL Balances are then extracted, to load General Ledger Balances and Supporting Reference Balances.

The flow from AHC to OFSAA is achieved through prebuilt 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 environment.

Figure 2: OFSAA–AHC Integration



The integration consists of the following processes:

1. Generate the SLA template.
2. Extract Data from FSDf 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 FSDf Integration with AHCS.

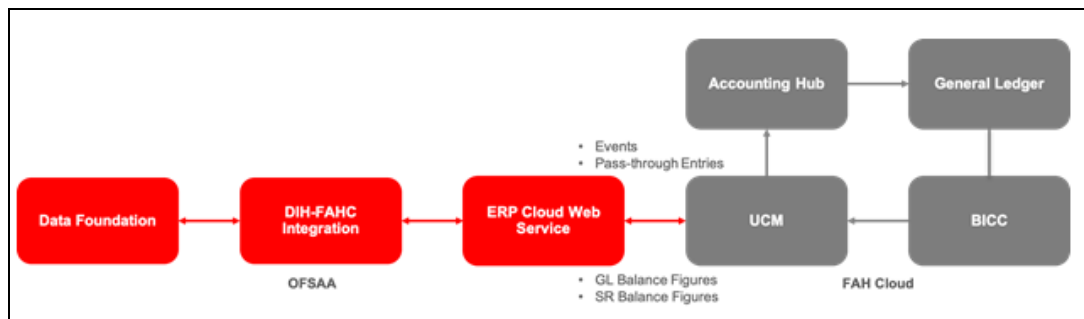
1. FSDf Integration with AHCS automates the population of the SLA XL template.
2. Post-deployment you must select one of the pre-packaged SLA definitions in FSDf Integration with AHCS and click the **Download Template** button in the Subledger Application window.
3. FSDf Integration with AHCS makes a copy of the SLA XL template with Source System details (which is the Subledger details) and then populates with Transaction types.
4. Header and Line definitions are pre-packaged in FSDf Integration with AHCS. The updated SLA XL template is then manually opened from the Source Sheet.
5. Click **Validate** in the Source System sheet. Ensure to enable Marco before validating the template.
6. In case of errors, fix the SLA definitions in FSDf Integration with AHCS and regenerate the SLA XL template. See the SLA XL instruction sheet for resolving the error.
7. After the errors are fixed or if there are no errors, click the **Generate Zip** button from the Source System sheet which is downloaded from the Sub ledger Application window.
8. The macro generates a zip file with the same name and in the same location as the SLA XL template. The zipped files are then manually uploaded into AHC.
9. The file contains metadata of source system that must be registered, along with the transaction object attributes and transaction types. This must be performed for each SLA defined in FSDf Integration with AHCS. See the [Registering with AHCS](#) section for more details.

10. The uploaded zip files register source system, transaction types, and define transaction objects (header information and line information) in AHC. AHC must be configured as per GL accounting requirement.
11. Post AHC configuration, OFSAA batches are performed using pre-packaged AHC connectors, extract transaction data from FSDF 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.
12. 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 FSDF Integration with AHCS

This section explains the data flow for FSDF Integration with AHCS.

Figure 3: Data Flow for FSDF Integration with AHCS



The FSDF Integration with AHCS is as follows:

- 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.
- See [Accounting Hub Cloud Service](#) for specifics.
- 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](#).
- GL and SR balance extracts so configured in BICC produce output files and placed those in UCM.
- DIH: Accounting Hub integration invokes ERP Cloud web service over SOAP to byte-stream file produced by BICC previously to OFSAA. See the *DIH Connectors* and *GL and SR Balance Extraction using AHC Connectors* sections in the [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 in the [FAH User Guide](#).

3 Mapping the OFSAA User to AHC User Groups

User group mapping enables you to map users or users 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.

The User Group Map window displays fields such as **User ID**, **Name**, and the corresponding **Mapped Groups**. You can view and modify the existing mappings within the **User Group Maintenance** window.

To access User Group Mapping, follow these steps:


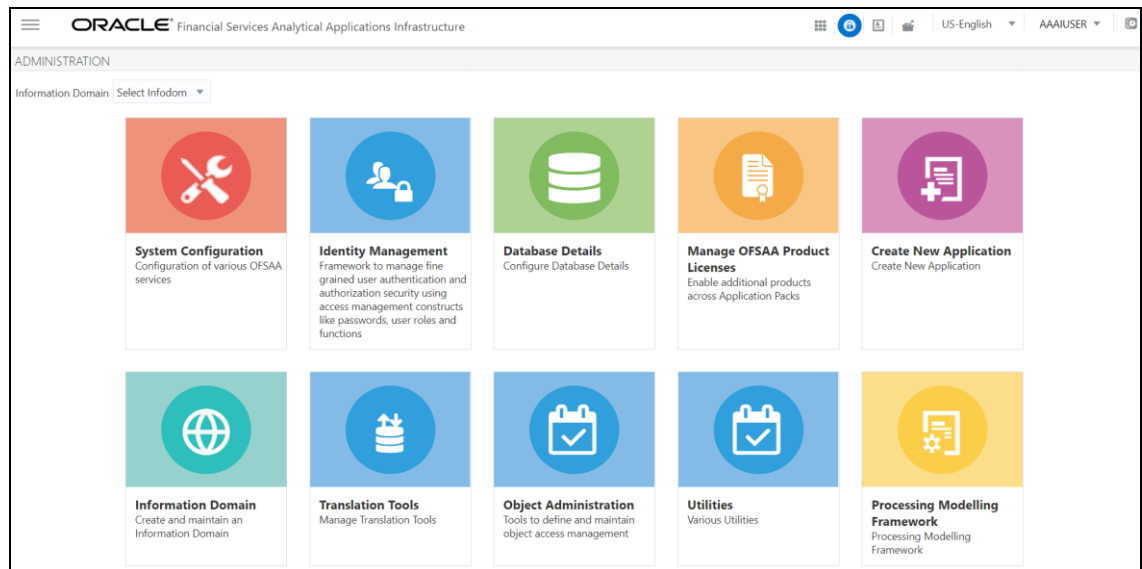
1. From the **Oracle Financial Services Analytical Applications Infrastructure** window, click **Administration**  from the **Header** to display the Administration tools.

Figure 4: OFS AAI Administration Window




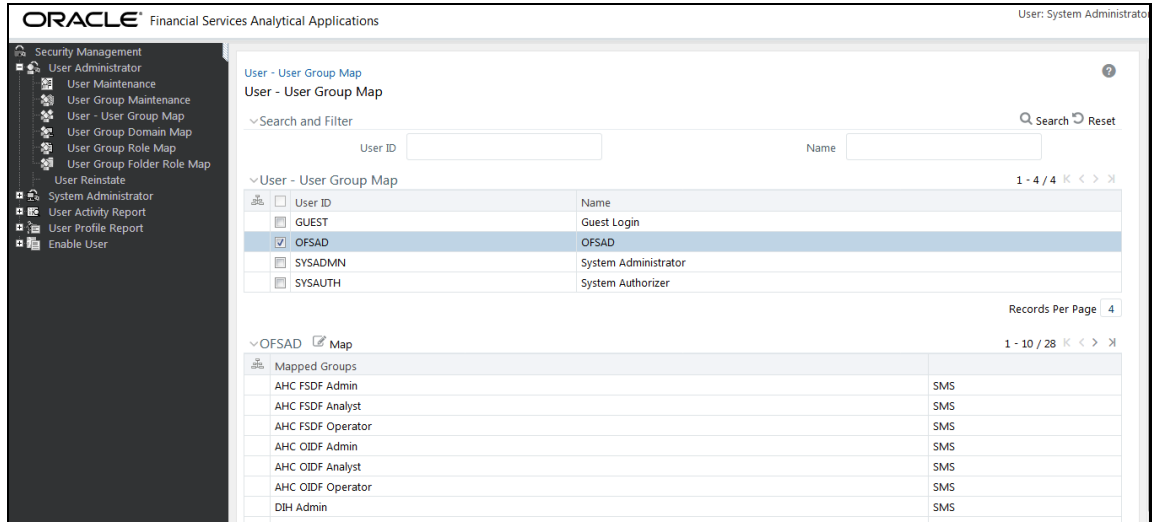
2. Select **Identity Management** from the Tiles menu to view the **Security Management** window or click the **Navigation Menu**  to access the Navigation List.
3. Select **Identity Management** to view the **Security Management** window. It is possible to map the users to the user groups in this window.

Figure 5: OFS AAI Security Management Window



For details on mapping users to user groups, see the [OFSAAI User Guide](#).

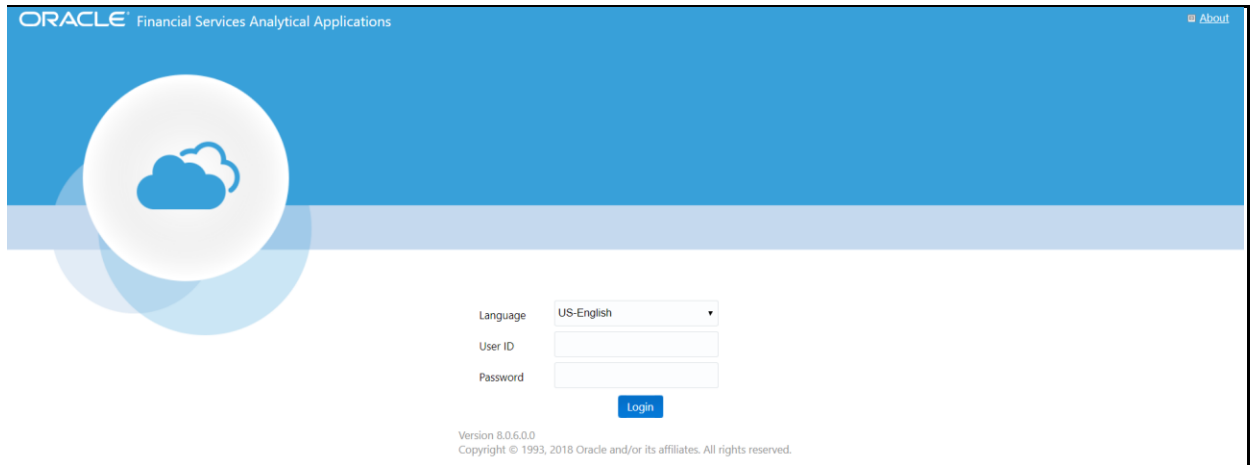
Figure 6: Pre-packaged User Groups for OFSAA - AHC Interface

User - User Group Map	
User ID	Name
<input type="checkbox"/> GUEST	Guest Login
<input type="checkbox"/> OFSAD	OFSAD
<input type="checkbox"/> OFSAN	OFSAN
<input type="checkbox"/> OFSDM	OFSDM
<input type="checkbox"/> OFSOP	OFSOP
<input type="checkbox"/> SYSADMN	System Administrator
<input type="checkbox"/> SYSAUTH	System Authorizer

4 Accessing the AHC-OFSAA Interface

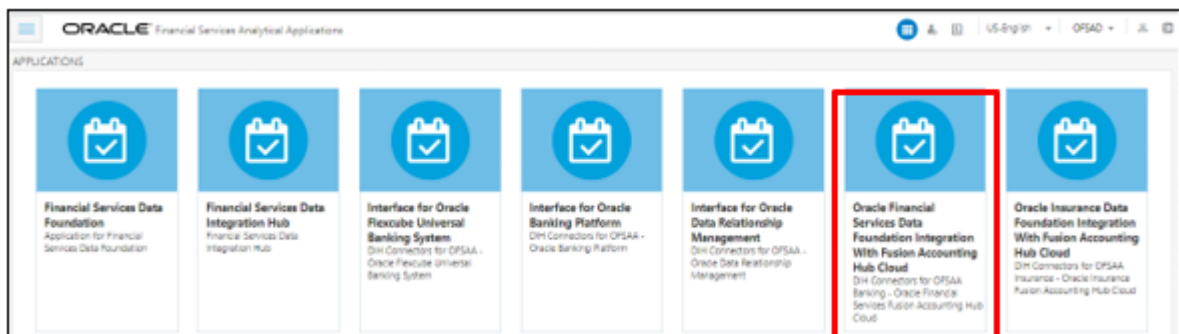
Access the AHC-OFSAA Interface using your login credentials (User ID and Password). The built-in security system ensures that you are permitted to access the window and actions based on the authorization only.

Figure 7: AHC - OFSAA Interface Login Window



After logging into the application, select **Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud**.

Figure 8: Applications Window



NOTE

You must be mapped to any one of the AHC user groups.

5 Component Details

The following components are included in this interface:

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

5.1 Packaged Subledger Application Information

The following Subledger Applications are predefined as follows:

- Retail Banking
- Commercial Banking
- Investment Banking
- Treasury
- Islamic Banking
- Core Banking PT

The list of product types covered under each Subledger Application is provided in the [SLA Coverage](#) file.

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

Subledgers can have two types of templates:

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

The rules vary based on the templates used.

Table 2: Subledger Applications

Subledger Application	Type
Retail Banking	Transaction based
Commercial Banking	Transaction based
Investment Banking	Transaction based
Treasury	Transaction based
Islamic Banking	Transaction based
Core Banking PT	Based on Pass-through accounting entries

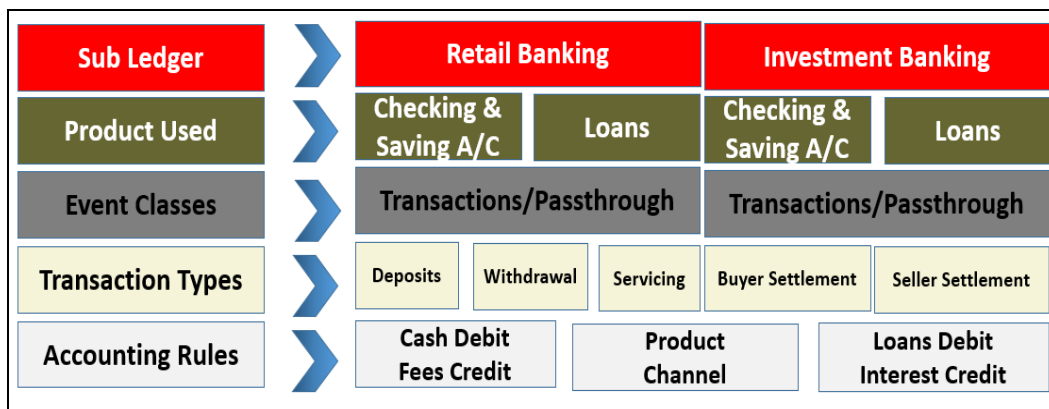
Subledger Information has transaction types mapped, along with the list of transaction and line attributes defined for each Subledger Application.

The FSDF data model is registered with AHC using the Excel template. The attributes of the transactions, pass through 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 Financial Services Data Foundation Integration with Accounting Hub Cloud Service.

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

Figure 9: Structure for Accounting Templates



The Retail Banking Subledger Application has two process products involved: Checking and Savings Accounts and Loans. Each of these products can have one of the two event classes: Transactions or Pass-through 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 checking and savings accounts, transaction types include deposits, withdrawals, and servicing. For loans, transaction types include originations, payments, and closings. 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. FSDF Integration with AHCS populates it with SLA specifications in an automated manner.

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

1. Access your instance of AHCS and download the template file

`XlaSourceSystemSetup.xlsm`. See the [Oracle ERP Cloud documentation](#) (**Learn More > Documentation Library**) on how this can be performed.

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, Subledger, Journal Source, Journal Category, and Process Category.
 - Transaction Types: Name and short name for all the event or transaction types are mandatory in this field.
 - There are **Validate** and **Generate ZIP** buttons available on the Source System sheet. These support functions are detailed in the [Registration of SLAs with Accounting Hub Service](#) section.
 - 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.
2. See the [Oracle ERP Cloud documentation \(Learn More > Documentation Library\)](#) for further details on AHCS.

5.3 DIH Connectors

This section explains about Extract and Insert Connectors.

5.3.1 Extract Connectors

The DIH connectors extract data from OFSAA staging tables, which must 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 FSDF 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 predefined 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 the DIH Parameters section before executing a batch.
- **AH_LOAD_RUN_ID:** Unique task ID which is auto-picked while execution of the task.

- **AH_FILE_DATE** is the Extraction Date whose file format must 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 the task.
- **AH_FILE_DATE**: is the Extraction Date whose file format must not be altered.

For Extract Connectors and mapping details, see the [Extract Connectors Structure](#) file.

5.3.2 Insert Connectors

The connectors have prebuilt mappings between AHC source views for GL or 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 or supporting reference balances into the 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.

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.

For example: STG_ANNUITY_TXNS for event-based Accounting and STG_ACCOUNTING_ENTRIES for pass-through accounting.

6.2 Implementation Guidelines

The implementation guidelines are as follows:

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 FSDF Integration with AHCS screens while configuring the Subledger.
3. Do not execute the connector's batch in FSDF 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. You must carefully extract the required data for appropriate accounting.
5. If the SLA template must be reloaded to AHC after changes, you must cautiously replace the old configuration and 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 is not processed.
7. After the SLA template is uploaded and source system is registered, for any further changes, ensure the following steps are performed in both FSDF Integration with AHCS and AHC:
 - If you add, edit, or delete any of the attribute or transaction types in the AHC screens, but do not update the respective changes in the FSDF Integration with AHCS application, integration may fail as data extract files will still have the unchanged structure of DIH.

- If you add, edit, or delete any of the attribute or event type in the FSDF Integration with AHCS screens, but do not download the new template and re-upload it back to the AHC application, integration may fail. In this scenario, the AHC application has a different Subledger 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 or Subledger structure either in FSDF Integration with AHCS or AHC, other application structure must also be changed accordingly. If not, it may lead to integration failure.
8. Through FSDF 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 the 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 the auto-generated batch is executed.
4. After the successful execution of all Tasks in a batch, Header information is extracted to the `XlaTrxH.csv` file and Line information to the `XlaTrxL.csv` file.
5. The `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

To complete the AHC setup, follow these steps

1. Log into the application and select **Oracle Financial Services Data Foundation Integration With Fusion Accounting Hub Cloud**.
2. Select the **AHC Administration** and then select **AHC Settings**.
3. From the **Data Integration Hub Designer** window, select **Configure** and then select **Settings**. This window captures the ODI set up information. The ODI settings window is displayed. The **OFSAA - ERP Settings for Accounting Hub (AH)** window appears.

Figure 10: OFSAA - ERP Settings for Accounting Hub (AH) Window

The screenshot shows the 'Accounting Hub Cloud Settings' window. The title bar includes 'Accounting Hub Cloud Settings' and 'OFSAA - ERP Settings for Accounting Hub (AH)'. Below the title bar, there is a 'Save' button. The main content area is divided into two sections: 'ERP Details for Accounting Hub' and 'Transaction Template Settings'. The 'ERP Details for Accounting Hub' section contains the following fields: 'ERP Url for Accounting Hub' (https://eadi.fa.dc1.c9dev2.oraclecorp.com/fsc), 'User Id' (finuser1), 'Password', 'KeyStore Location' (/scratch/dihdev/key/keystoreNew.jks), and 'KeyStore Password'. The 'Transaction Template Settings' section contains the following fields: 'Transaction Type Column Name' (EVENT_TYPE_CODE) and 'Transaction Template' (XlaSourceSystemSetup.xism) with a 'Browse' button.

4. Enter the required information in the **ERP Details for Accounting Hub** and **Transaction Template Settings** panes.
 - a. Enter the following details in the **ERP Details for Accounting Hub** pane:

— **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 the following 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.
- See the [Importing Certificate into KeyStore](#) section for details on importing Certificate to KeyStore.

- b. Enter the following details in the Transaction Template Settings pane:

- **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 the prepopulated text and capture the 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. See the [Registering with AHCS](#) section for details.

Browse and select the transaction template file `XlaSourceSystemSetup.xlsx` from the downloaded location in the local machine.

- c. Click **Save**.

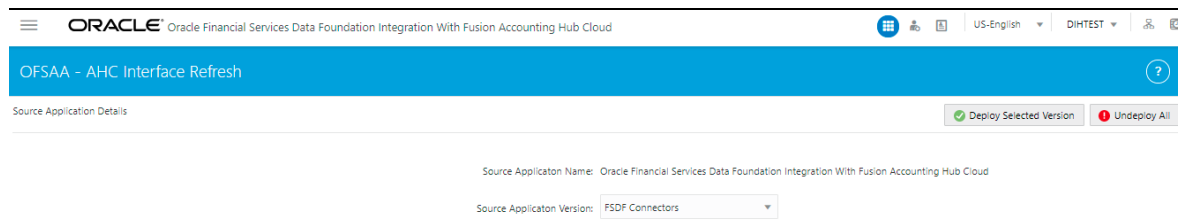
7.1.1 Refresh AHC Interface

7.1.2 Deploying Connectors or Undeploying OFSAA-AHC Connectors

After the prerequisites 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.

To deploy or undeploy OFSAAHCS Connectors using the Refresh DRM Interface window, follow these steps:

1. From the **Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud Service** window, select **AHC Administration** and then select **Refresh AHC Interface**. The **Source Information** window is displayed.

Figure 11: OFSAA – AHC Interface Refresh

2. Select the **Source Application Version** as **FSDf Connectors** from the drop-down list.
3. Click **Deploy Selected Version**. A message is displayed: *“Are you sure you want to Deploy the Selected Version?”*
4. Click **Yes** to proceed.
5. If there is a requirement to re-deploy Connectors, they must be Unpublished, Undeployed, and Deployed, post required changes. The **Undeploy All** facility under the **AHC Interface Refresh** user interface facilitates the undeploy step within this sequence. Connectors must be re-deployed only in response to the following situations:
 - a. Standard Connector definition requires changes or is deployed with changes that must be maintained.
 - b. 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. See the [DIH User Guide](#) for details on unpublishing Connectors.

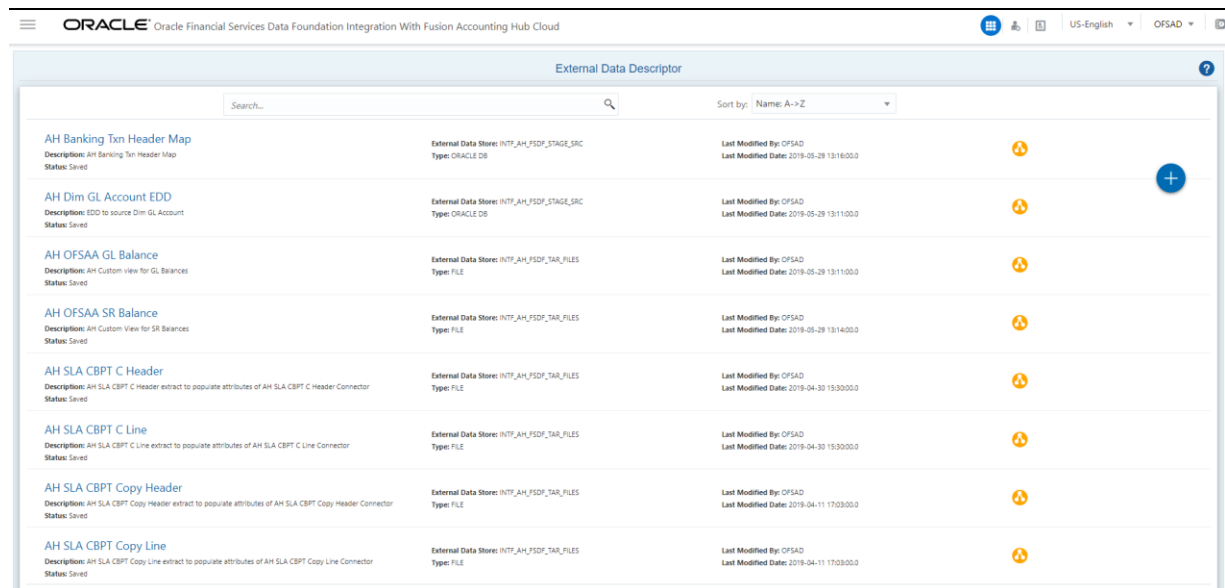
6. Click **Undeploy All**. A message is displayed: *“Are you sure you want to Undeploy the Selected Version?”*
7. Click **Yes** to proceed.

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.

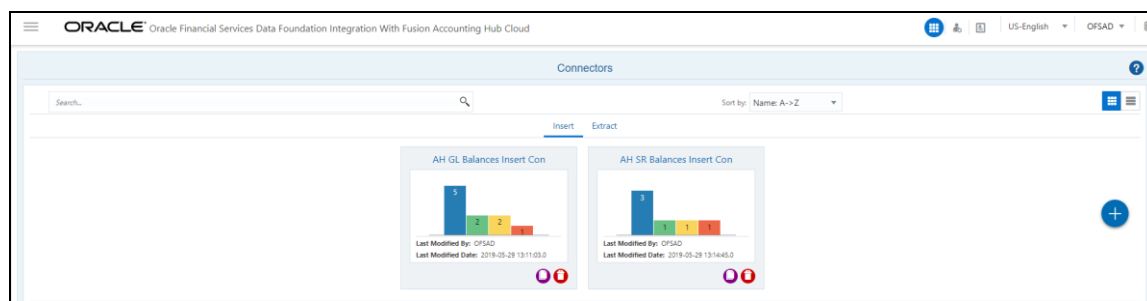
8. To check deployed EDDs and Mappings, select **AHC Data Mapping** from the **Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud Service** window, and then select **External Data Descriptor**.

Figure 12: External Data Descriptor Summary Window



- To check deployed Connectors and Mappings, select **AHC Data Mapping** from the **Financial Services Data Foundation Integration with Accounting Hub Cloud Service** window, and then select **Connectors**.

Figure 13: Connectors Summary Window



7.2 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.

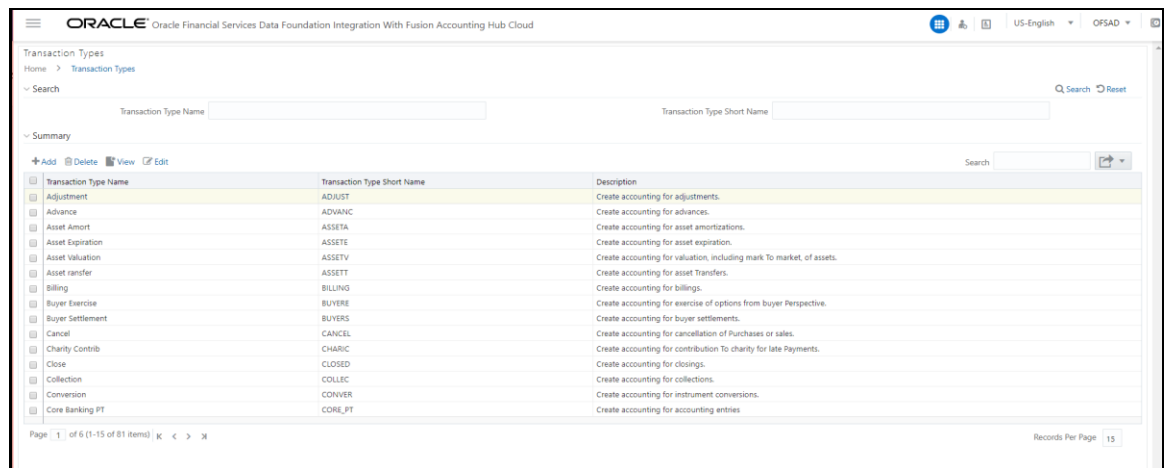
Transaction types are required to collect the transaction data and process it into accounting data necessary to form the Subledger information. Data from different source systems come together and are processed depending on these categories to form ledger information.

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, follow these steps:

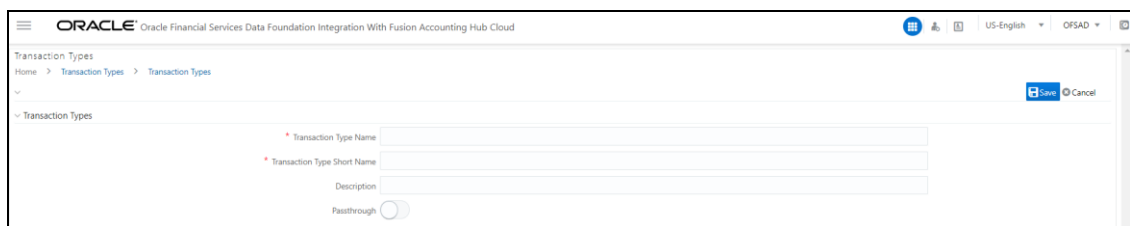
1. From the **Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud Service** window, select **AHC Administration**, and then select **Transaction Types**. The **Transaction Types** window is displayed.

Figure 14: Transaction Types Summary window



2. A list of pre-packaged transaction types is displayed on the window, which you can map with Subledgers accordingly. You cannot edit or delete these pre-packaged transaction types.
You can search for Transaction Type Name or Short Name. Enter a part or full name and click the search to get the filtered result.
3. In the **Summary** pane, you can **Add**, **Delete**, **View**, or **Edit** the events.
4. Click **+Add**. The **Transaction Types** window appears.
5. 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




Figure 15: Transaction Types window



The following are the validations for the fields:

Table 3: Validations

Field Name	Limitations/Validations
Transaction Type Name	<ul style="list-style-type: none"> • Must not have special characters. • Only alphanumeric characters and spaces are allowed. • Must not be more than 15 characters. • Must not start or end with space. • Must be unique.
Transaction Type Short Name	<ul style="list-style-type: none"> • Must have only Uppercase, numbers, and underscores. • Must start only with an uppercase or a number. • Must not have special characters and space. • Must not be more than 15 characters. • Must be unique.

6. Click  if you want to delete a transaction.
7. Click  to view the details of the transaction type.
8. Click  to edit the transaction type details.
9. Click **Export** to move the screen data to an Excel or CSV format for the reference.
10. 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 FSI_ACCNTING_EVENT_TYPE_MASTER table.

7.3 Subledger Attributes

The Subledger Attributes user interface provides a list of seeded header 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, which can be mapped to an SLA during SLA configuration. The ledger balances can be calculated based on these Subledger attributes. To accommodate custom attributes, there is an option to add custom attributes that can vary from user-to-user to configure custom Subledger. These custom attributes can help in creating rules and extracting ledger balances in a required way.

7.3.1 Adding a Subledger Attribute

To add a Subledger Attribute, follow these steps:

1. From the **Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud Service** window, select **AHC Administration** and then select **Subledger Attributes**. The **Subledger Attributes** window is displayed with a list of seeded data with Logical and Physical Name.

Figure 16: Subledger Attributes window

Logical Name	Domain	Physical Name	Type	Event Class
Account Branch Code	Text	ACCT_BRANCH_CODE	Line	Passthrough
Account Number	Text	ACCOUNT_NUMBER	Header	Both
Account Product Code	Text	ACCT_PROD_CODE	Line	Passthrough
Account Reversal Indicator	Text	REVERSAL_IND	Line	Transaction
Authorized By	Text	AUTHORIZED_BY	Line	Passthrough
Batch Code	Text	BATCH_CODE	Header	Passthrough
Branch Code	Text	BRANCH_CODE	Line	Both
Branch Code	Text	BRANCH_CODE	Header	Transaction
Business Unit Code	Text	BUSINESS_UNIT_CODE	Line	Transaction
Business Unit Code	Text	BUSINESS_UNIT_CODE	Header	Both


- Click  to add a new attribute. The **Add Attribute** window is displayed.

Figure 17: Add Attribute window

Add Attribute

Enter 30 or fewer characters

Logical Name *

Physical Name *

Domain

Attribute Type

Transaction/Passthrough

- Enter the **Logical Name** and **Physical Name**. Physical Name can have only numbers, and alphabets in upper case.
- Select the **Domain**, **Attribute Type**, and **Transaction/Passthrough** details from the drop-down list. **Transaction/Passthrough** is the type of Subledger. Use this attribute for transaction type of Subledger or passthrough type of Subledger.
- 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**.
- Click **Save**.

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

7.4 Subledger 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.

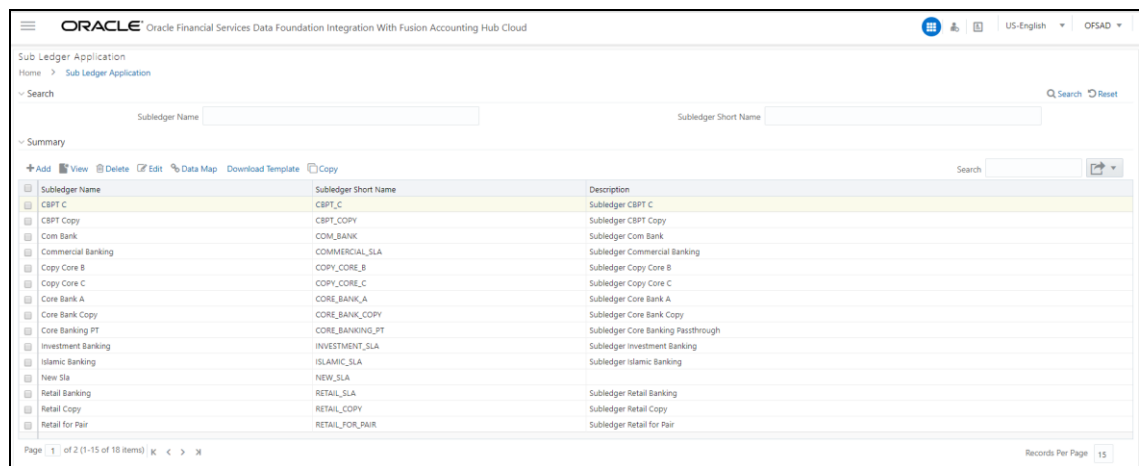
The AHC application has the SLA template through which a Subledger and its details are loaded.

7.4.1 Subledger Application Summary

To understand the Subledger Application summary window, follow these steps:

1. From the **Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud Service** window, select **AHC Administration** and then select **Subledger Applications**. The **Subledger Application** window is displayed.

Figure 18: Subledger Application Summary window




2. You can search for **Subledger Name** or **Subledger Short Name**. A list of pre-packaged Subledgers appears. For more information, see the [SLA Coverage](#) file.

NOTE Pre-packaged Subledgers are not editable through the AHC Subledger window.

3. In the **Subledger Application** window, in the **Summary pane**, you can **Add, Delete, Edit, View Data Map**, and **Download the Sub Ledger**.
4. Click **+ Add** to create a new Sub Ledger.
5. Select one Sub Ledger and click **Delete** if you wish to delete a Sub Ledger.
6. Select one Sub Ledger and click **Edit** the Sub Ledger details.

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

7. Select one Sub Ledger and click **View** to view the details of the Sub Ledger.
8. Select one Sub Ledger and click **Download Template** to download the source template with SLA information.
9. Select one Sub Ledger and click **Data Map** to create and map connectors to the selected Sub Ledger.

10. Select one Sub Ledger and click  Copy a sub ledger.
11. After the details are filled, save the Sub Ledger.
12. Download the SLA template that must be uploaded to the AHC application.

NOTE It is mandatory to manually validate the template and create a zip file, which must be uploaded to the AHC cloud environment. For errors during validation, update the required changes in the DIH windows, and then download the template again. See the [SLA Template Validations](#) section for more information. Subledgers saved are stored in the FSI_SLA_MASTER table.

13. Click **Export** to move the summary data to an Excel or CSV format.

7.4.2 Adding a Subledger Application

To add a Subledger:

1. Click  Add to create a new Subledger. The **Subledger Application** window is displayed.

Figure 19: Subledger Application window

2. 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:

Table 4: Validations

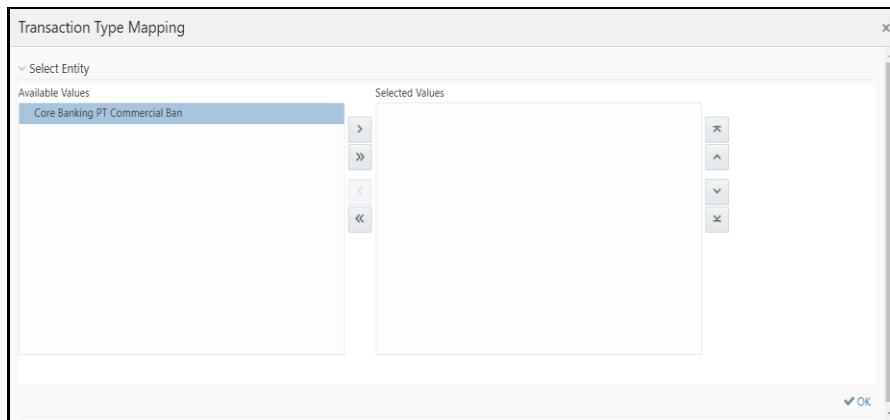
Field Name	Limitations/Validations
Subledger Name	<ul style="list-style-type: none"> • Must not have special characters. • Only alphanumeric characters and spaces are allowed. • Must not have more than 15 characters. • Must not start with a space. • Must be unique.




Field Name	Limitations/Validations
Subledger Short Name	<ul style="list-style-type: none"> • Must have only Uppercase, numbers, and underscores. • Must start only with an uppercase or a number. • Must not have special characters and space. • Must not have more than 15 characters. • Must be unique.
Event Class	<ul style="list-style-type: none"> • Can select one or both the options.
Ledger Name	<ul style="list-style-type: none"> • 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*” is displayed.
4. Click **OK**. The Transaction types window is displayed.
5. In the **Transaction Types** pane, perform the following steps:
 - a. Click [Transaction Type Mapping](#). All the transaction types that are created as part of Transaction Types, along with the pre-packaged list of **Transaction Types (Available Values)** window are displayed on the LHS.

NOTE Transaction Types are 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.

Figure 20: Transaction Type Mapping window



- 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. Click  to move the entities up and down.
- e. After the entities are selected, click **OK**.

The selected values are now displayed in the **Transaction Type Mapping** window.


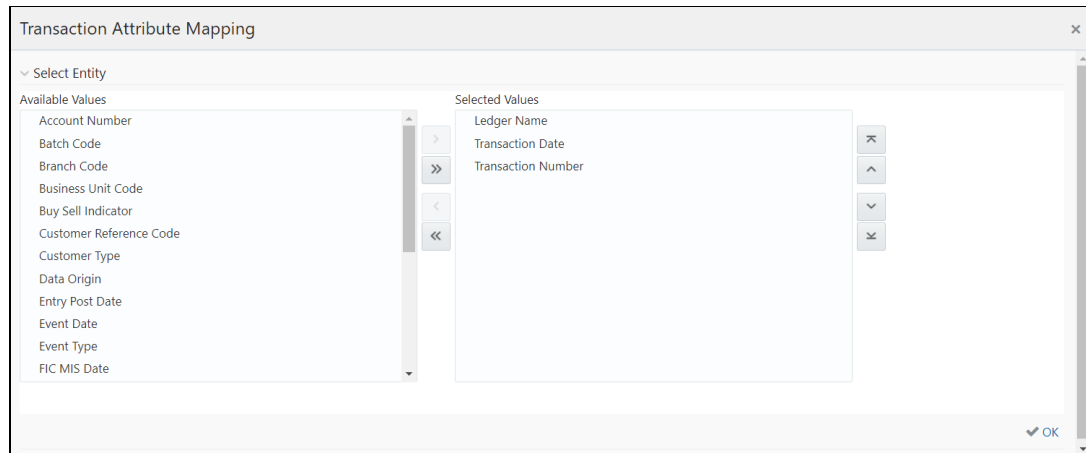
6. In the **Transaction Information** pane, perform the following steps:
 - a. Three mandatory attributes: **Ledger Name**, **Transaction Date**, and **Transaction Name** are selected by default and cannot be edited.
 - b. Click  [Transaction Attribute Mapping](#) . The **Transaction Attribute Mapping** window is displayed.

Figure 21: Transaction Attribute Mapping window






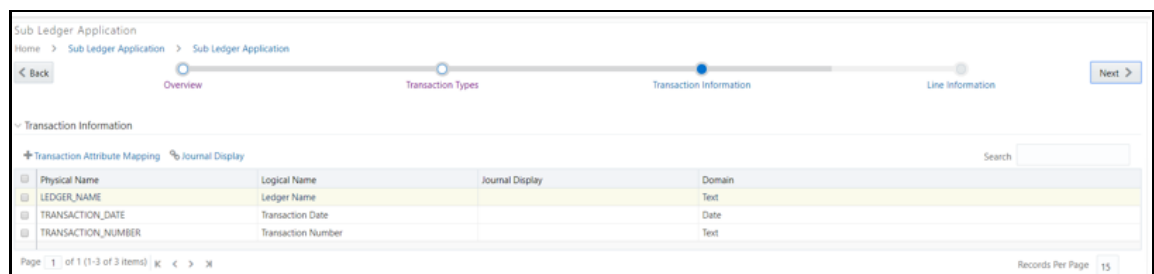
- a. The list of transaction attributes is displayed. These are seeded from the FSI_SLA_ATTR_MASTER table.
- b. Ledger Name, Transaction Date, and Transaction Name are selected by default.
- c. Click  to move the required entities to the list of the **Selected Values**.
- d. Click  to move all the entities to the list of the **Selected Values**.
- e. Click  to move the entities up and down.
- f. After the entities are selected, click **OK**.
- g. The selected values are now displayed in the **Transaction Information** pane.

Figure 22: Transaction Information window



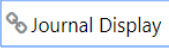
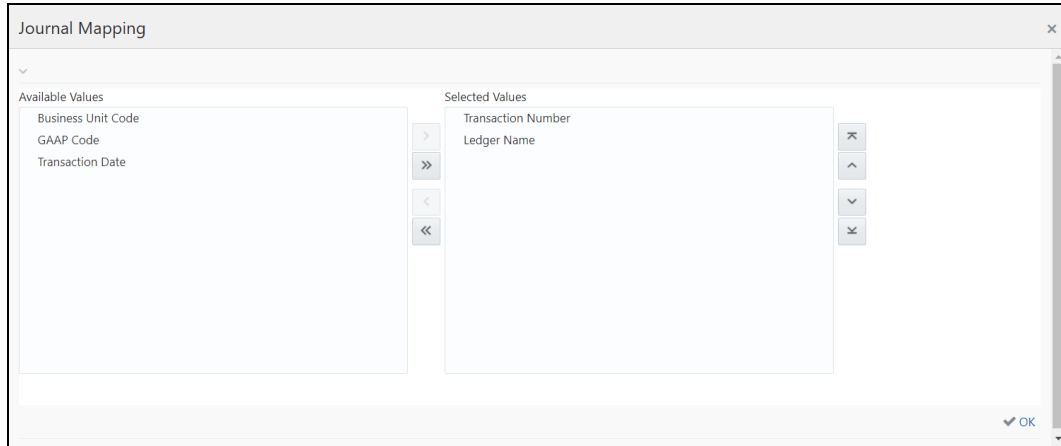
- h. If you wish to display these attributes as part of journal lines, click  . The **Journal Mapping** window is displayed.

Figure 23: Journal Mapping window



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





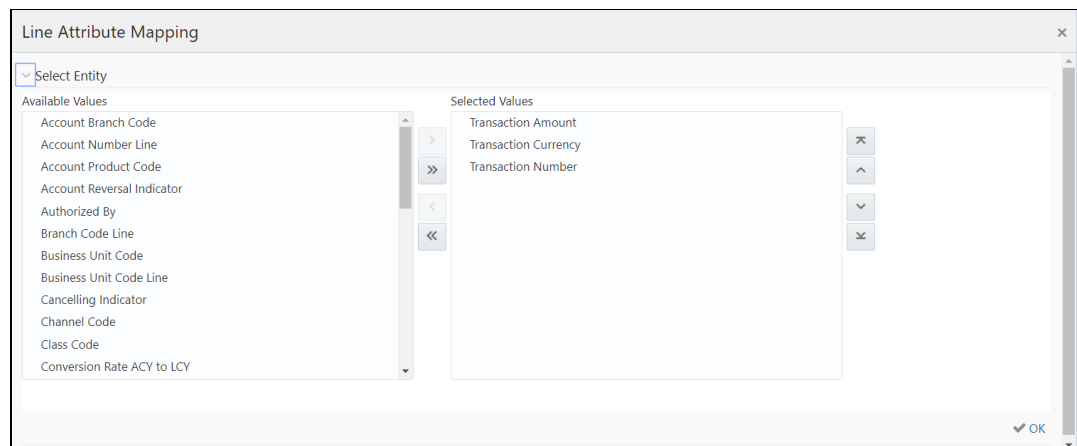
- a. Click  to move the required entities to the list of the **Selected Values**.
 - b. Click  to move all the entities to the list of the **Selected Values**.
 - c. Click  to move the entities up and down.
 - d. After the entities are selected, click **OK**.
 - e. Click **Next**.
7. In the **Line Information** pane, 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** window is displayed.

Figure 24: Line Attribute Mapping window






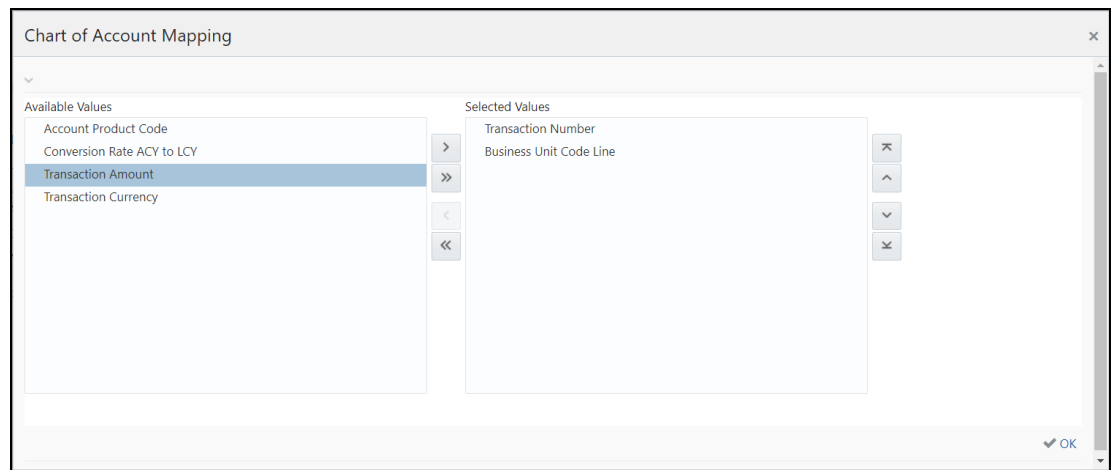



- a. The list of line attributes is displayed. These are seeded from the FSI_SLA_ATTR_MASTER table.
- b. **Transaction Amount**, **Transaction Currency**, and **Transaction Number** are selected by default.
- c. Click  to move the required entities to the list of the **Selected Values**.
- d. Click  to move all the entities to the list of the **Selected Values**.
- e. Click  to move the entities up and down.
- f. After the entities are selected, click **OK**. The selected values are now displayed in the Line Information pane.
- g. If you wish to display these attributes as part of the chart of account, click [Chart Of Account Value](#). The **Chart of Account Mapping** window is displayed.

Figure 25: Chart of the Account Mapping window



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

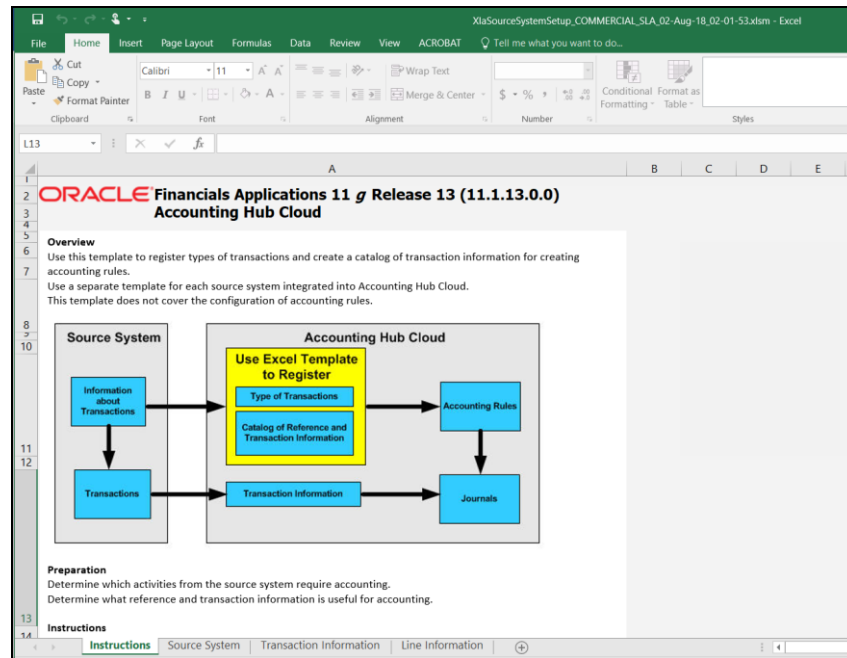
- a. Click  to move the required entities to the list of the **Selected Values**.
 - b. Click  to move all the entities to the list of the **Selected Values**.
 - c. Click  to move the entities up and down.
 - d. After the entities are selected, click **OK**.
8. Click **Save**. An entity is now saved in the summary window.

NOTE After the Transaction Information and Line Information is mapped and saved, they are stored in the FSI_SLA_TRANSACTION_ATTR and FSI_SLA_LINE_ATTR tables, respectively. The Transaction and Line information is already configured for the pre-packaged Subledgers. See the [Pre-packaged Subledger Configurations](#) file.

7.4.3 Registration of SLAs with Accounting Hub Service

1. From the **Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud Service** window, select **Subledger Application**. The **Subledger Applications** summary window is displayed with a list of SLAs.
2. From this list of SLAs, select the SLA that must 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.xlsxm` with Subledger Application details filled in.

Figure 26: XlaSourceSystemSetup.xlsxm



4. In the **Instructions** sheet, all the details are explained.
5. In the **Source System** sheet, the **Name** and **Short Name** are given in the Subledger Application window are displayed.

Figure 27: XlaSourceSystemSetup.xlsm – Source System

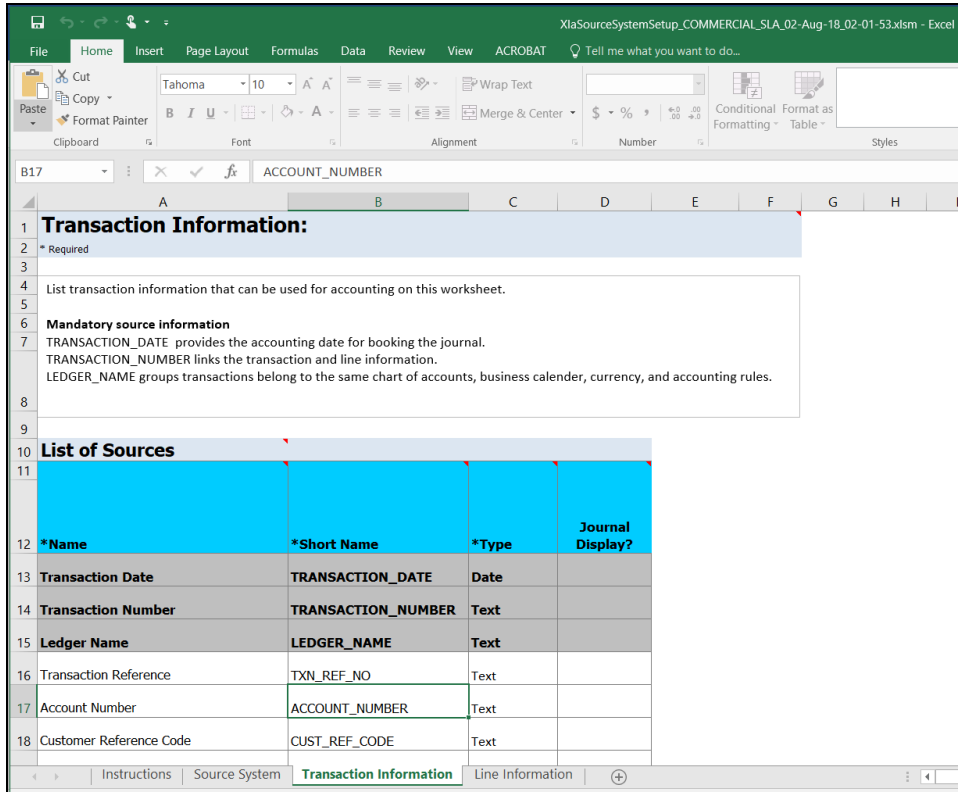
*Name	*Short Name
Adjustment Txn Commercial Bank	ADJUST_T_COMMERCIAL_SLA
Advance Txn Commercial Banking	ADVANC_T_COMMERCIAL_SLA
Asset Amort Txn Commercial Ban	ASSETA_T_COMMERCIAL_SLA
Asset Expiration Txn Commercia	ASSETE_T_COMMERCIAL_SLA
Asset Valuation Txn Commercial	ASSETV_T_COMMERCIAL_SLA
Buyer Settlement Txn Commercia	BUYERS_T_COMMERCIAL_SLA
Close Txn Commercial Banking	CLOSED_T_COMMERCIAL_SLA
Collection Txn Commercial Bank	COLLEC_T_COMMERCIAL_SLA
Conversion Txn Commercial Bank	CONVER_T_COMMERCIAL_SLA
Cost Allocation Txn Commercial	COSTAL_T_COMMERCIAL_SLA
Default Txn Commercial Banking	DEFAULT_T_COMMERCIAL_SLA
Deposit Txn Commercial Banking	DEPOSP_T_COMMERCIAL_SLA
Deposit Returned Txn Commercia	DEPOSR_T_COMMERCIAL_SLA
Disbursement Txn Commercial Ba	DISBUR_T_COMMERCIAL_SLA
Expenses Amort Txn Commercial	EXPENA_T_COMMERCIAL_SLA
Funding Txn Commercial Banking	FUNDIN_T_COMMERCIAL_SLA

NOTE

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

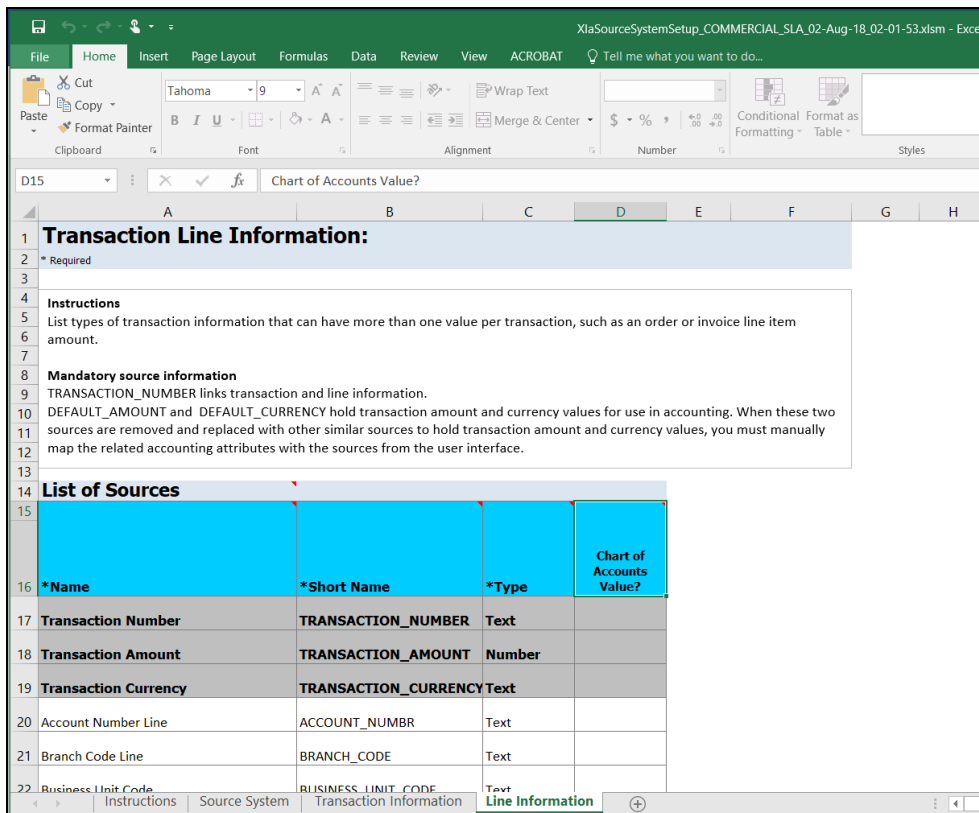
6. In the **Transaction Information** sheet, those Transaction Type Names and Short Names against which Journal Display is set to Yes is displayed through the AHCS user interface.

Figure 28: XlaSourceSystemSetup.xlsm – Transaction Information



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

Figure 29: XlaSourceSystemSetup.xlsm – Line Information



NOTE The three mandatory columns are listed in a greyed area for both Transaction Information and Line Information.

8. Click **Validate**, in the 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.

9. Click **Generate ZIP**, in the 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** window to **Transaction Amount** and **Transaction Currency**, before proceeding with the configuration of accounting rules.

7.4.4 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](#)).

7.4.4.1 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.

7.4.4.2 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 5: Transaction (Header)

Text	50
Number	10
Date	10
Long Text	5

Table 6: 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.

7.4.4.3 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.

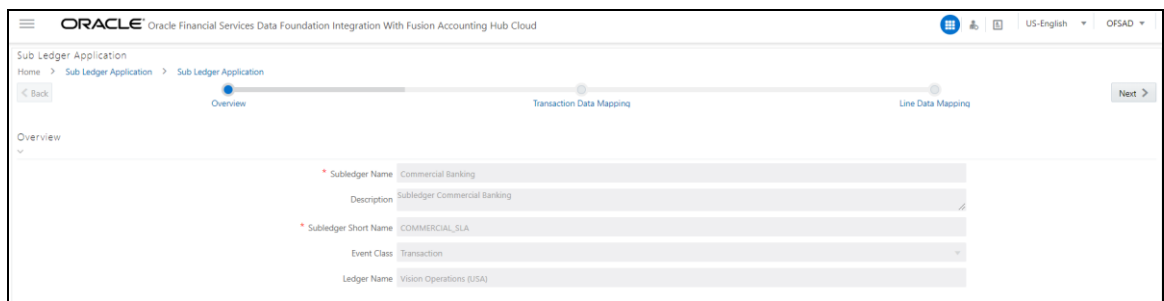
3. Open transaction (header) EDD created by DIH upon SLA configuration (“AH SLA <<SLA Name>> Header” pattern), upload [Header EDD](#) prepared in step 2, save and verify the order.
4. Open line EDD created by DIH upon SLA configuration (“AH SLA <<SLA Name>> Line” pattern), upload [Header EDD](#) prepared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

7.4.5 Using Data Map

To create a Connector and map it to a Subledger Application, follow these steps:

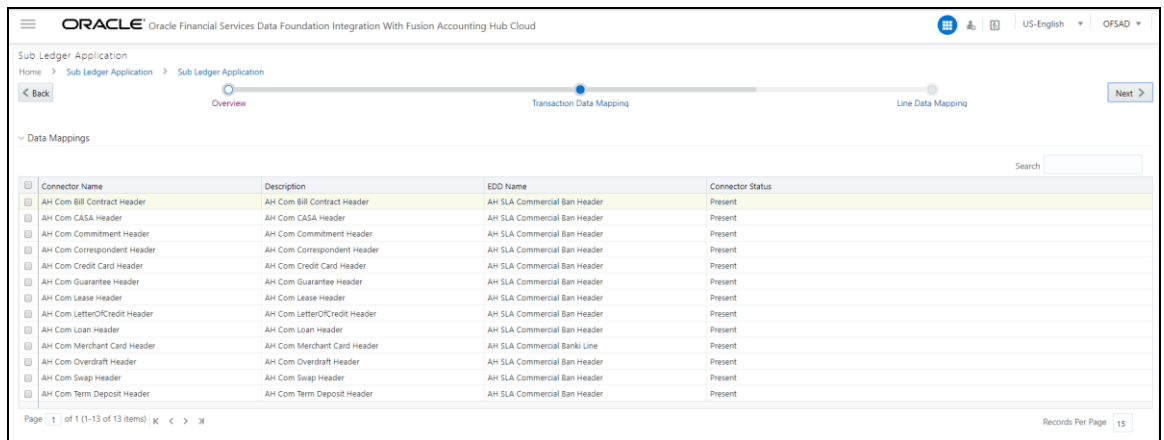
1. From the **Subledger Application** summary window, select the desired SLA, and click **Data Map**. The fields are non-editable in the Overview pane.

Figure 30: Subledger Application Window



2. Click **Next**. The **Transaction Data Mapping** window is displayed.

Figure 31: Transaction Data Mapping Window



- a. Click **Create** to add a new data mapping. The **New Connector** window is displayed.
- b. In the **Target** pane, the required EDD is automatically created following the **AHC SLA <<SLA NAME>> Header** pattern. For example, when Subledger 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 Subledger 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 Subledger. Extraction Date filter is already enabled with pre-packaged connectors for executing them.

- d. Event Type names must also be filtered depending on the Subledger you configure. If not configured, all transaction types are processed in the extract data for a Subledger and the accounting process fails.
- e. Double-click **Mapping** to map the source and target.

NOTE For more information on mapping, see the *Connectors* section in the [DIH User Guide](#).





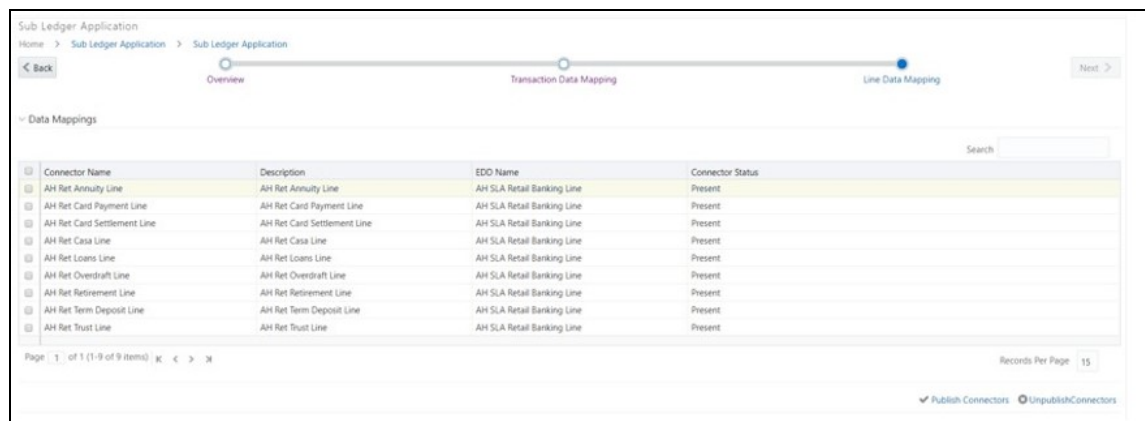







- f. Click **Properties** or . The **Connector Details** window is displayed.
 - g. Enter the **Name** and **Description** and click **Publish, Save**, or **Save As Draft**. You can also publish the connector in the **Subledger Data Mappings** window. The published connector is displayed in the **Data Mappings** window.
 - h. Click **Transaction Data Mapping**. The **Transaction Data Mapping** is displayed with 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** window is displayed.

Figure 32: Line Data Mapping Window




- a. Click **Create** to add a new data mapping. The **New Connector** window is displayed. In the **Target** pane, 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 Commercial Banking Line**. EDD structure has all the attributes that are mapped to the Subledger under the Line Information.
- b. Map the required source and target and save the connector.
- c. 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 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](#).

- d. Click **Properties** or . The **Connector Details** window is displayed.
- e. Enter the **Name** and **Description** and click **Publish**, **Save**, or **Save As Draft**. You can also publish the connector in the **Subledger Data Mappings** window. The published connector is displayed in the **Data Mappings** window.
- f. Click **Line Data Mapping**. The **Line Data Mapping** window is displayed. 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**.
4. Click **Save**. The message, “*SLA and Connector saved successfully*” is displayed.
5. Click  **Publish Connectors** 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 a message (<INFODOM>_DIH_AH_<SLA_CODE>).
6. If you want to unpublish the connector, click  **UnpublishConnectors**.
7. After unpublishing a connector, select the required connector and click  **Edit Data Mapping** to edit a connector.

7.4.6 Copying a Subledger Application

To copy a Subledger, follow these steps:

1. From the **Subledger Application** window, select one Subledger and click  **Copy**. The **Save Subledger** window is displayed.
2. Enter the **Name** and **Short Name** and click **Save**. This copies all the properties except the Name and Short Name as it must be unique.

For example,

Table 7: Connector Details

Connector Name Before Copy	Connector Name After Copy
AH Com Bill Contract Header AH Com Casa Header	AH (copied SLA name) succeeded by Hdr1, Hdr2, and Hdr3
AH Com Commitment Header	

- Similarly, for line mapping, the name after copy shows as AH <SLAname> Ln1.
- The EDD name is also replaced with the copied name in the connector window in the format AH SLA <SLA name> Header and AH SLA <SLA name> Line.

7.5 Standardization of Product Codes and Mapping with Subledger Applications

7.5.1 Filtering Data Based on Product Codes

Data for each Subledger Application is obtained from common Header tables and specific transaction tables corresponding to the types of products relevant to the SLA. Some of these transaction tables or Product Processor Entities contain data relevant to more than one SLA.

In such instances, appropriate filters must be applied as data is extracted from the Product Processor Entities for purposes of each SLA. Subledger Applications packaged with FSDf Integration with AHCS already contain such filters, referring standard product types and codes thereof. See the [seeded product codes](#) file for details on product processor entities and seeded product codes.

Product Processor Entities that contain data relevant to more than a single SLA are listed along with corresponding SLAs as follows.

Table 8: Product Processor Entities

Product Processor Entity	Subledger Applications		
STG_CASA	Retail Banking	Commercial Banking	
STG_LOAN_CONTRACTS	Retail Banking	Commercial Banking	
STG_CARDS	Retail Banking	Commercial Banking	
STG_TD_CONTRACTS	Retail Banking	Commercial Banking	
STG_OD_ACCOUNTS	Retail Banking	Commercial Banking	
STG_SWAPS_CONTRACTS	Investment Banking	Commercial Banking	Treasury
STG_MM_CONTRACTS	Investment Banking	Treasury	
STG_INVESTMENTS	Investment Banking	Treasury	
STG_FUTURES	Investment Banking	Treasury	
STG_FX_CONTRACTS	Investment Banking	Treasury	
STG_CREDIT_DERIVATIVES	Investment Banking	Treasury	
STG_OPTION_CONTRACTS	Investment Banking	Treasury	
STG_REPO_CONTRACTS	Investment Banking	Treasury	
STG_COMMITMENT_CONTRACTS	Commercial Banking	Treasury	

Product Processor Entity	Subledger Applications		
STG_CORRESPONDENT_ACCOUNT	Commercial Banking	Treasury	

A set of product codes must be established in [Managing Product Code Filters](#) such that multiple product or instrument codes in use across your institution are standardized for use across analysis and accounting. FSDF Integration with AHCS provides a seeded list of product codes, each of which is mapped to a Subledger Application in [Managing Product Code Filters](#).

For example, STG_CASA is designed to host data relating to both commercial and retail current and savings accounts, through seeded product codes RETACC and CORPACC. These product codes are, in turn, mapped to Retail Banking and Commercial Banking SLAs. Packaged SLAs have filters on product code predefined.

7.5.2 Managing Product Code Filters

FSDF Integration with AHCS uses a combination of filter clauses explicitly defined in Connectors and additional information maintained in SETUP_MASTER, a configuration entity. While defining SLAs and related information, you can use entries recorded in this entity to facilitate filtered extraction of data from product processor entities to SLAs. You can also use this mechanism to affect such filtered extraction, if packaged product codes, SLA definitions, related information, or a combination of these are modified.

The following tabulation depicts sample entries in SETUP_MASTER to address the following scenario:

- STG_CASA has data with four products codes – Retail Product 1, Retail Product 2, Commercial Product 1, and Commercial Product 2.
- Data corresponding to Retail Product 1 and Retail Product 2 must be mapped to Retail Banking SLA.
- Data corresponding to Commercial Product 1 and Commercial Product 2 must be mapped to Commercial Banking SLA.
- Standardized product codes are RETACC and CORPACC, corresponding to Retail Banking and Commercial Banking SLAs, respectively.

Entries in the following table facilitate the appropriate filtering of data from the STG_CASA product processor entity to the correct SLAs.

Table 9: Sample Entries in SETUP_MASTER

V_COMPONENT_CODE (Actual Product Code)	V_COMPONENT_VALUE (Seeded Product Code)	V_COMPONENT_DESC (Description)
Retail Product 1	RETACC	STG_CASA to Retail Banking
Retail Product 2	RETACC	STG_CASA to Retail Banking
Commercial Product 1	CORPACC	STG_CASA to Commercial Banking
Commercial Product 2	CORPACC	STG_CASA to Commercial Banking

7.6 OFSAA Chart of Account (COA) Mapping

COA segments are predefined 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 FSDF columns for loading data, as defined in the Accounting Hub Cloud Service environment. For example, if COA segment 1 is mapped to Legal Entity attribute in a cloud environment while defining account rules, the same must be mapped in the DIH screens Legal Entity to COA_SEGMENT1. The mapping window for the COA segment displays different dimensions to allow you to map them to respective COA segments.

The ledger balance data from AHC appear 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 window 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 list of predefined dimensions required in the application and which are mapped to COA segments are as follows:

- 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

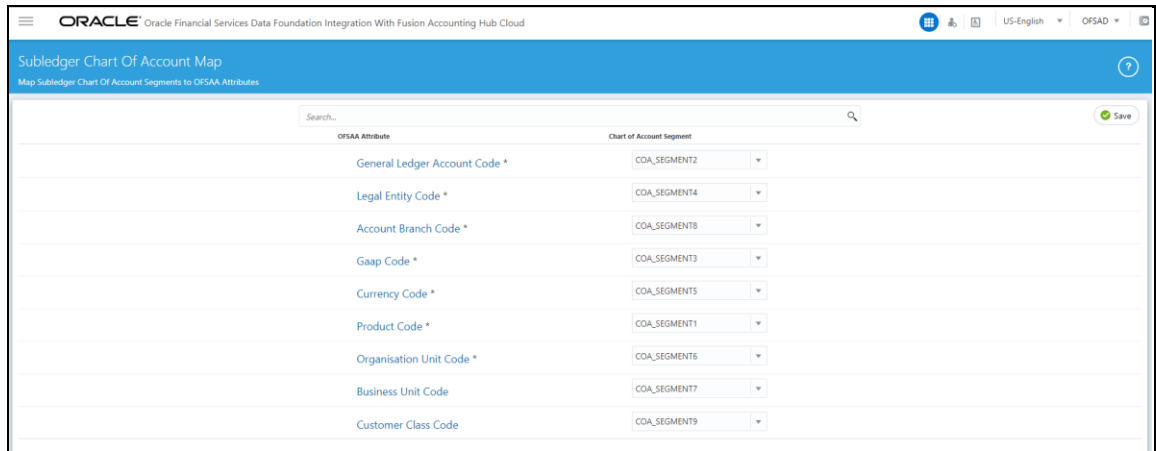
7.6.1 Mapping Segments

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

To map the segments, follow these steps:

1. From the **Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud Service** window, select **AHC Administration** and then select **OFSAA Chart of Account Mapping**. The **Subledger Chart of Account Map** window is displayed.

Figure 33: Subledger Chart of Account Map Window



The **OFSAA Attribute** column displays the list of predefined 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.

2. Select the **NONE** option when the dimension is not required to be mapped or used.
3. Select the required COA segments and click **Save**.
4. 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 loaded as mapped into Data Foundation.
5. After 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 AHCS with column header as COA Segment 1 are loaded as GL Code in OFSAA.
- The predefined 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

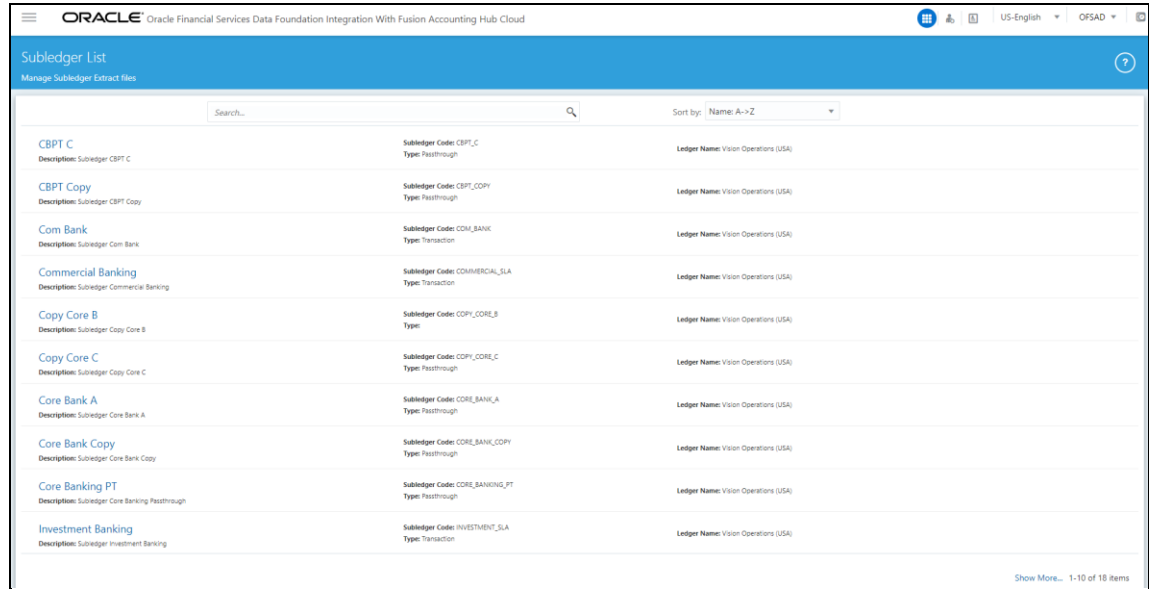
During batch execution of Subledger connectors, by default, the sixth parameter of the last task is set to N (for more information, see [Batch Execution](#) which generates a .zip file) after successful execution. When the parameter is set to Y, execution generates a .zip file, uploads it to UCM, and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

In both cases, the **SubledgerList** window can be used to manage the extracted zip files.

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

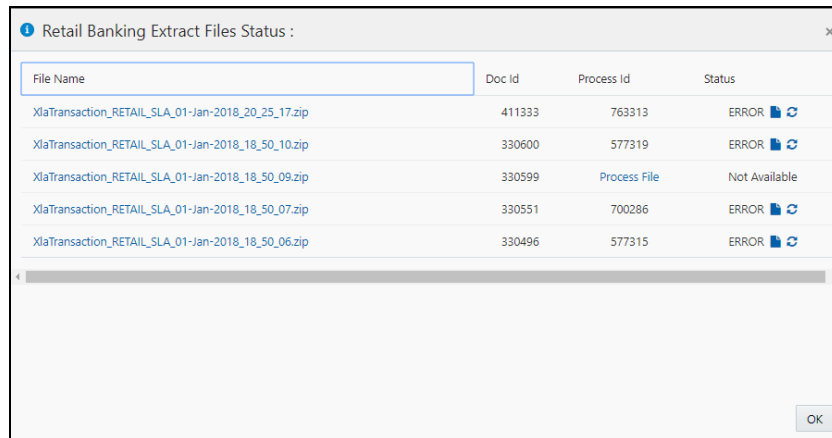
From the **Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud Service** window, select **AHC Administration** and then select **SubledgerList**. The **SubledgerList** window is displayed.

Figure 34: Subledger List Window



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

Figure 35: Subledger Extract Files window



It has four columns:

- **File Name:** The files uploaded are displayed 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. After it is uploaded, a unique Doc ID number is created, and it is displayed in this column. For files executed with parameter 6 as Y and parameter 7 as 1, a unique Doc ID number is displayed.
- **Process ID:** For the extracted zip files, where batch execution is completed with the 6th parameter as Y and the 7th parameter as 2, it has a unique Process ID number in this column. The files which are executed with the **sixth parameter as N or Y** (along with the 7th parameter as 1), has the Process File option in this field is displayed for the first time.

The Process File either uploads the .zip file to UCM and triggers the process or job as 'Import Accounting Transactions'.

or

It directly triggers the uploaded file with the Import Accounting Transactions process.

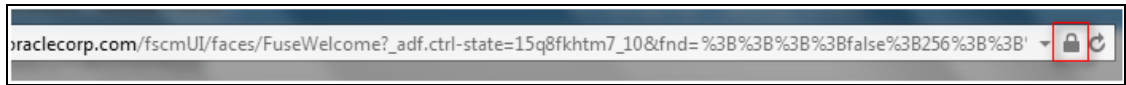
Click **Process File** and the following actions occur:

- If Doc ID number exists: .zip file, which is already uploaded to UCM under the existing Doc ID, is 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.

7.8 Importing Certificate into KeyStore

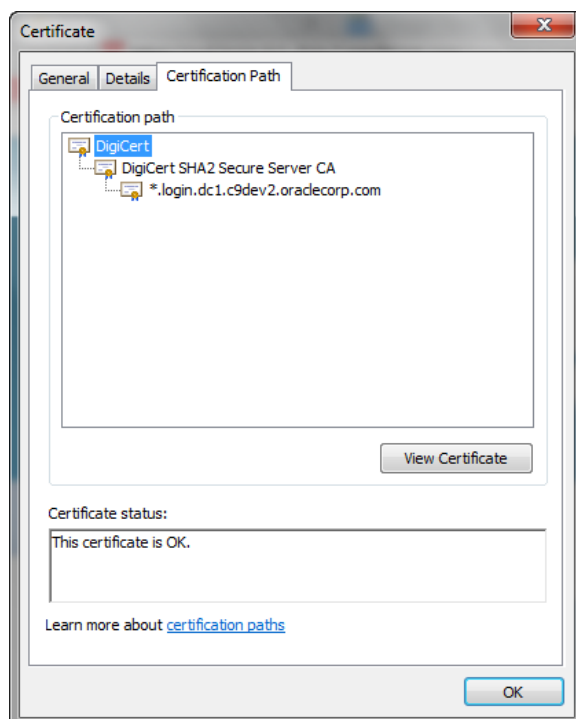
Export the certificate from the browser to the file using the following steps:

1. Access the AHC SSL URL from any web service using Microsoft Internet Explorer or Mozilla Firefox.
2. Click the **Security Lock** symbol and then click **View Certificates**.



3. From the **Certification Path** tab, the certificate hierarchy is displayed. Export the top two certificates (**DigiCert** and **DigiCertSHA2SecureServerCA** in the following case) to KEYSTORE file.

Figure 36: Certificate window



4. From the **Certification Path** tab, select **DigiCert** and then click **View Certificate**.

5. From the **Details** tab, select **Copy to File**. The **Certificate Export Wizard** is displayed.
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. Enter the file name and click **Finish**.
9. Repeat steps 4 through 8 for the **DigiCertSHA2SecureServerCA** certificate.
10. When using other web browsers, perform similar steps. The navigation path can differ in the web browsers used.
11. Run the following command to import a certificate into KeyStore:

```
keytool -import -trustcacerts -file <filename> -alias <aliasname> -  
keystore keystore.jks -storepass welcome1
```

12. 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 is 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 preseeded 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 requirement 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 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 window, follow these steps:

1. From the **Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud Service** window, select **AHC Administration** and then select **Subledger Event Grouping**.

The **Subledger Event Grouping** window is displayed with a list of pre-packaged event groups information for transmission to the Accounting Hub.

Figure 37: Subledger Event Grouping


Event Group Name	Header Connector	Line Connectors
Accounting Entries	AH Accounting entries Header	AH Accounting entries Line
Accounting Entries 1	AH Retail P Copy Hdr 3	AH Retail for Pair Ln 5
Com Bank Pair 10	AH Com Bank Hdr 13	AH Com Bank Ln 14
Com Bank Pair 11	AH Com Bank Hdr 12	AH Com Bank Ln 2, AH Com Bank Ln 13
Com Bank Pair 12	AH Com Bank Hdr 11	AH Com Bank Ln 12, AH CBPT Copy Ln 1
Com Bank Pair 13	AH Com Bank Hdr 10	AH Com Bank Ln 11
Com Bank Pair 2	AH Com Bank Hdr 5	AH Com Bank Ln 5
Com Bank Pair 3	AH Com Bank Hdr 4	AH Com Bank Ln 4
Com Bank Pair 4	AH Com Bank Hdr 3	AH Com Bank Ln 3
Com Bank Pair 6	AH Com Bank Hdr 9	AH Com Bank Ln 10, AH Com Bank Ln 9

2. The **Subledger Event Grouping** window displays the **Event Group Name**, **Header Connector**, and **Line Connectors**.
 - a. **Event Group Name:** It is the generic name given to identify a Header and Line Connector mapping. The pre-packaged Subledger connectors and name are displayed.
 - b. **Header Connector:** 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. You can define an Event Group, Manage Group Attributes, and Delete user-defined group events.
5. Click  if you want 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, follow these steps:

1. From the **Subledger Event Grouping** window, click  to define the event group. The **Define Event Group** window is displayed.
2. Enter the required details and click **Save**.

7.9.3 Managing Group Attributes

To manage group attributes, follow these steps:


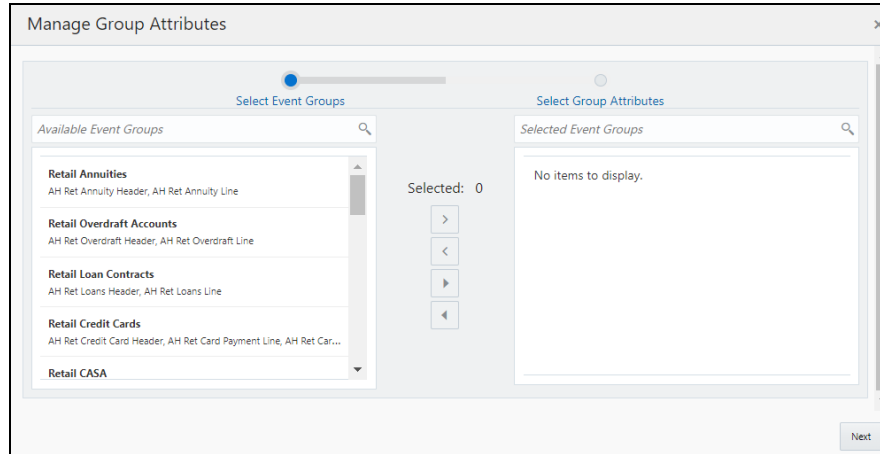




1. From the **Subledger Event Grouping** window, click  to manage group attributes. The **Manage Group Attributes** window is displayed with the **Available Event Groups**.

Figure 38: Manage Group Attributes







2. Select the Event Groups that are created previously, for which you want to define the group attributes.
 - a. Click  to move the required event groups to the list of the Select Group Attributes.
 - b. Click  to move all the event groups to the list of the Select Group Attributes.
 - c. Click  or  to remove the event groups back one by one or remove all, respectively.

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

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-mandated columns to LHS and remove them from aggregation or group by clause.
 - a. Click  to move the non-mandated columns to LHS.
 - b. Click  to move the non-mandated columns to LHS.
 - c. Click  or  to remove the entities back one by one or remove all, respectively.
5. After all the required columns are confirmed, click **Save**. A confirmation message is displayed “Are you sure you want to remove?”.
6. Click **Yes**. The **Subledger Group Attributes** are updated successfully.

8 Preparing for Connector Deployment

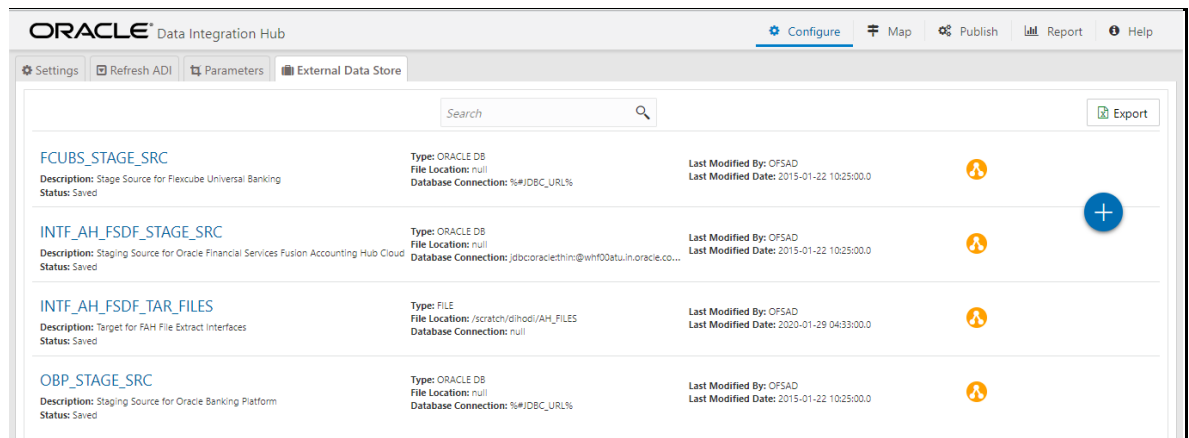
8.1 Configuration Steps

The following configuration steps are specific to FSDF Integration with AHCS and must be performed before proceeding to deploy the Connectors.

NOTE ODI setup must be completed in DIH Foundation before these tasks are undertaken. See the [DIH User Guide](#).

1. From the **Data Integration Hub Designer** window, select **Configure** and then select **External Data Store**. The **External Data Store Summary** is displayed.

Figure 39: External Data Store Summary Window



2. Search for EDS name **INTF_AH_FSDF_TAR_FILES**. Click this EDS to Edit.
3. Note that, this EDS refers to AHCS as a target system for Events information, relevant to Extract Connectors.
4. In the **File Location** field, specify the location where data files for AHCS integration must be generated and saved. Note that this EDS also refers to AHCS as a source system for GL and SR Balance information, relevant to the Insert Connectors. 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.
5. From the **Data Integration Hub Designer** window, select **Configure** and then select **Refresh ADI**. This **Refresh Application Data Interface** summary is displayed.
6. Click **Start** to start the refresh of ADIs. The ongoing ADI refresh is displayed. 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 Infodomain.
7. From the **Data Integration Hub Designer** window, select **Publish** and then select **Refresh Target Datastores**. The **Refresh Target Datastore Interface** summary is displayed.
8. Click **Start** to refresh of the Target Datastore. The ongoing Target Datastore refresh is displayed.

NOTE See the [DIH User Guide](#) for detailed information.

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 FSDF.

9.1 Configuration and Data Files

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

9.1.1 Chart of Account to OFSAA Dimension Mapping

See the section [OFSAA Chart of Account \(COA\) Mapping](#) for more information on how this mapping must be done.

9.1.2 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 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 EDS path.

9.1.3 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.FinXlaBallngSuppRefBalAM.SuppRefCombinationsPVO](#)
- [FscmTopModelAM.FinGIAccountsCodeComboAM.CodeCombinationPVO](#)
- [FscmTopModelAM.FinGInquiryBalancesAM.BalancePVO](#)
- [FscmTopModelAM.FinXlaBallngSuppRefBalAM.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.

9.1.4 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 WebLogic, 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** and 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.

9.2 Executing AHC Connectors

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

1. From the **Oracle Financial Services 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 40: Batch Maintenance Window

Batch ID		Batch Description	
<input checked="" type="checkbox"/>	OFSDIINFO_AH_CONNECTORS_70121	To Execute INTF_AH_FSDF Connectors	
<input type="checkbox"/>	OFSDIINFO_AH_GL_SR_EXTRACT_DOWNLOAD	To Extract GL SR Balances	

Page 1 of 1 (1-2 of 2 items) Records Per Page 15

Task ID	Task Description	Metadata Value	Component ID	Precedence	Task Status
Task1	Task to execute AH GL Balances Insert Con	AH GL Balances Insert Con	DIH CONNECTORS	START	N
Task2	Task to execute AH SR Balances Insert Con	AH SR Balances Insert Con	DIH CONNECTORS	Task1	N

Page 1 of 1 (1-2 of 2 items) Records Per Page 15

NOTE For some unknown reasons, when deployed in WebLogic, not all batches are created. Perform the following steps to avoid the issue:

1. Log in to the WebLogic server admin console.
 2. Navigate to **Console** and 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 Configuration of Metadata

See the [Managing Product Code Filters](#) section for details.

10.2 Batch Execution

NOTE You must have all the required configuration, including accounting rules, completed in your instance of AHCS before attempting to send events data using this process.

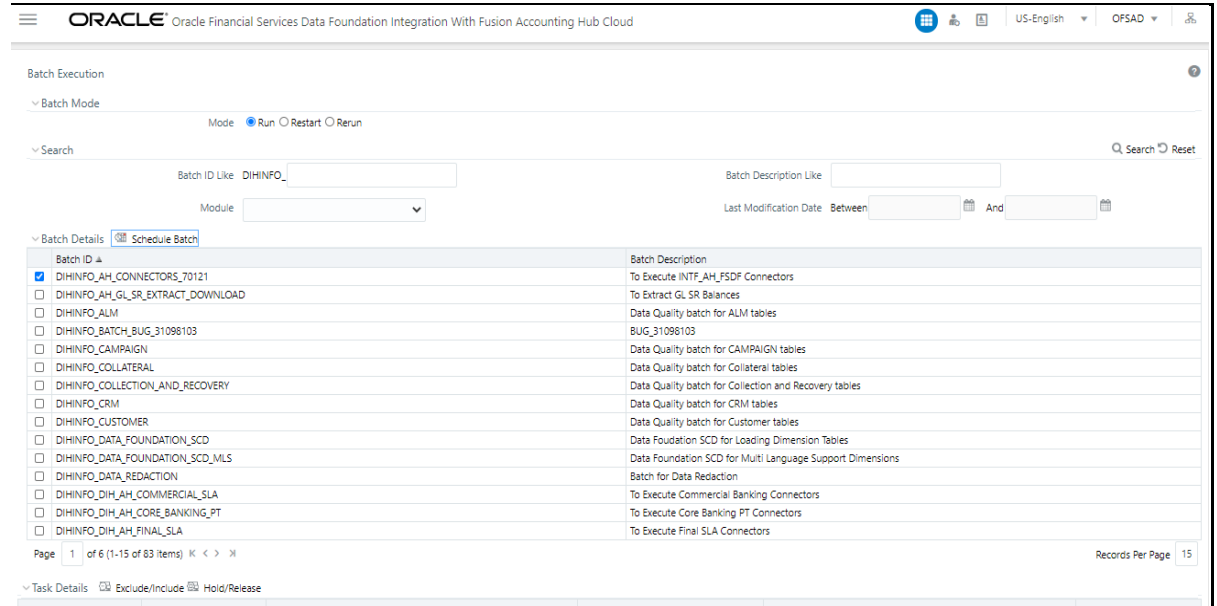
See the [Batch Re-Execution](#) section before you proceed with Batch Execution.

In the **Batch Execution** window, there are batches created for each Subledger. These batches are created when you publish SLAs, as detailed in the [Subledger](#) section of this guide.

To execute a batch, follow these steps:

1. From the **Oracle Financial Services Data Foundation Integration with Accounting Hub Cloud Service** window, select **Execution**, and then select **Batch Execution**.

Figure 41: Batch Execution Window



2. Select a batch from the **Batch Details** table and then click **Schedule Batch**. The **Batch Scheduler** window is displayed.

Figure 42: Batch Scheduler Window

The screenshot shows the Batch Scheduler window with the following details:

- Batch List:** A table with columns 'Batch ID' and 'Batch Description'. The second batch, 'DIHINFO_AH_GL_SR_EXTRACT_DOWNLOAD', is selected. Other batches include 'DIHINFO_AH_CONNECTORS_70121', 'DIHINFO_ALM', 'DIHINFO_BATCH_BUG_31098103', 'DIHINFO_CAMPAIN', 'DIHINFO_COLLATERAL', 'DIHINFO_COLLECTION_AND_RECOVERY', 'DIHINFO_CRM', 'DIHINFO_CUSTOMER', 'DIHINFO_DATA_FOUNDATION_SCD', 'DIHINFO_DATA_FOUNDATION_SCD_MLS', 'DIHINFO_DATA_REDACTION', 'DIHINFO_DIH_AH_COMMERCIAL_SLA', 'DIHINFO_DIH_AH_CORE_BANKING_PT', and 'DIHINFO_DIH_AH_FINAL_SLA'.
- Page:** 1 of 6 (1-15 of 83 items) with navigation arrows.
- Records Per Page:** 15
- Batch Scheduler:**
 - Domain: DIHINFO
 - Batch: DIHINFO_AH_GL_SR_EXTRACT_DOWNI
 - Schedule: New Schedule Existing Schedule
- New Schedule:**
 - Schedule Name:
 - Frequency: Once Daily Weekly Monthly Adhoc
- Schedule Time:**
 - Dates: Start Date: End Date:
 - Run Time: 00 Hours 00 Minutes
 - Lag: 0 Days
- Buttons:** Save, Cancel

3. Enter the details as of which data must be processed and click **Save**.
4. Click **Execute Batch** in **Batch Execution** window.
5. See the [OFSAAI User Guide](#) for details on batch execution, runtime parameters, and monitoring.

Each SLA batch consists of:

- Connector execution task
- Processes Event Grouping
- Prepares the Header file
- Prepares the Line file
- Runs an executable file

All tasks specified within the batch must be executed.

The Run Executable task performs the following actions:

1. Identifies the extracted Header and Line CSV files.
2. Formats data by removing duplicate Header rows in both Header and Line files.
3. Generates the `Metadata.txt` file with SLA details.
4. Creates a `.zip` file including the Header, Line, and Metadata files.
5. See the following structure to locate and identify the `.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 after files are uploaded through its API to UCM. This can be enabled by updating the sixth runtime parameter of the final Run Executable task to Y before executing the batch.

With this enabled, the seventh parameter of this task can be set to 1 (where the .zip file is uploaded to UCM and no further action is taken) or 2 (where the .zip file is uploaded to UCM and import task in AHCS is triggered to process uploaded file) as appropriate for the purposes.

10.3 Batch Re-Execution

FSDf 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 or 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 can cause duplicated events data to be sent to your instance of AHCS, with no provision for automated roll back.

OFSAA Support

Raise a Service Request (SR) in [My Oracle Support \(MOS\)](#) for queries related to the OFSAA 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 that has all the revised/recently released documents.

