Oracle Financial Services Data Foundation Integration With Fusion Accounting Hub Cloud

Copyright © 2019 Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

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

For information on third party licenses, click here.
## Document Control

<table>
<thead>
<tr>
<th>Version Number</th>
<th>Revision Date</th>
<th>Change Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>February 2019</td>
<td>Updated</td>
</tr>
<tr>
<td>02</td>
<td>June 2019</td>
<td>Updated EDS section with Dynamic Principal name and Driver fields. Updated ODI setup validation settings</td>
</tr>
</tbody>
</table>
# 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 Documentation Accessibility ....................................................................................... 7  
  1.5 Access to Oracle Support ............................................................................................. 8  
  1.6 Related Information Sources ...................................................................................... 8  
  1.7 Acronyms ..................................................................................................................... 8  

2 Introduction ....................................................................................................................... 9  
  2.1 About Oracle Financial Service Analytical Applications .......................................... 9  
  2.2 About Accounting Hub Cloud (AHC) ......................................................................... 9  
  2.3 Objective ..................................................................................................................... 9  
  2.4 About Oracle Financial Services Data Foundation Integration With Fusion Accounting Hub Cloud 10  
  2.5 OFSAA-AHC Interface Architecture Overview ....................................................... 11  
  2.6 OFSAA – AHC Integration Overview ....................................................................... 12  

3 Mapping the OFSAA User to AHC User Groups ............................................................ 14  

4 Logging into AHC-OFSA A Interface .............................................................................. 16  

5 Component Details .......................................................................................................... 18  
  5.1 Pre-packaged Subledger Information ......................................................................... 18  
  5.2 SLA Template to Register Source System ................................................................. 19  
  5.3 DIH Connectors .......................................................................................................... 20  
    5.3.1 Extract Connectors ............................................................................................... 20  
    5.3.2 Insert Connectors ............................................................................................... 21  

6 Implementation ............................................................................................................... 22  
  6.1 Background ................................................................................................................. 22  
  6.2 Implementation Guidelines ......................................................................................... 22  
  6.3 Transaction Object Population .................................................................................. 23
7 AHC Administration........................................................................................................24

7.1 AHC Settings .................................................................................................................24
7.2 Refresh AHC Interface .................................................................................................25
7.2.1 Deploying OFSAA-AH Connectors Using Refresh AH Interface ............................25
7.2.2 Undeploying OFSAA-AH Connectors Using Refresh AH Interface .........................27
7.3 Transaction Types ........................................................................................................28
7.4 Sub-Ledger Attributes .................................................................................................31
7.4.1 Adding a Sub-Ledger Attribute .............................................................................31
7.5 Sub-Ledger Applications .............................................................................................33
7.5.1 Sub-Ledger Application Summary ........................................................................33
7.5.2 Adding a Sub-Ledger Application .........................................................................35
7.5.3 Downloading a Template ......................................................................................40
7.5.4 Using Data Map .......................................................................................................44
7.5.5 Copying a Sub-Ledger Application .......................................................................50
7.6 Metadata Configuration for Accounting Filters ........................................................51
7.7 Refer to list of seeded product codes for instrument tables and seeded product codes to be used.OFSAA Chart of Account (COA) Mapping .................................................................52
7.7.1 Mapping Segments ..................................................................................................52
7.8 Subledger Extract Files ...............................................................................................54
7.9 Import Certificate into Keystore ..................................................................................55
7.10 Subledger Event Grouping .........................................................................................57
7.10.1 Subledger Event Grouping Summary ..................................................................57
7.10.2 Defining an Event Group .....................................................................................58
7.10.3 Managing Group Attributes ..............................................................................58

8 Prerequisites for Deploying OFSAA-AHC Connectors ..................................................61

8.1 Additional DIH Configurations for AHC Interface .....................................................61

9 GL and SR Balance Extraction using AHC Connectors ................................................64

9.1 Configuration and Data Files Required ........................................................................64
9.1.1 Chart of Account to OFSAA Dimension Mapping ..................................................64
9.2 GL and XLA csv files from AHC .................................................................................64
9.3 Executing AHC Connectors .........................................................................................64

10 Execution of Extract Connectors ..................................................................................66
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1</td>
<td>Configuration of Metadata</td>
<td>66</td>
</tr>
<tr>
<td>10.2</td>
<td>Batch Execution</td>
<td>66</td>
</tr>
</tbody>
</table>
1 ABOUT THE GUIDE

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

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

1.1 Scope of the Guide

This guide details the tasks that are required to performed in the OFSAA environment. The integration requires effort on the AH system also, for activities such as setting up supporting references, creating jobs to generate accounting entries and so on.

1.2 Intended Audience

Welcome to release 8.0.7.0.0 of the Oracle Financial Services Data Foundation Integration With Fusion Accounting Hub Cloud User Guide. This manual is intended for the following audience:

- ETL Developers: The ETL Developers from the IT Department of the financial services institution, who do the data sourcing.
- Business Analysts: The business analysts from the IT Department of the financial services institution, who do the mapping of the tables.

1.3 Prerequisites

- Data Integration Hub (DIH) should be installed
- OFSAA – Oracle Financial Services Data Foundation Integration With Fusion Accounting Hub Cloud should be installed
- Oracle Data Integrator environment for executing the interfaces

1.4 Documentation Accessibility

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


1.6 Related Information Sources

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

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

1.7 Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADI</td>
<td>Application Data Interface</td>
</tr>
<tr>
<td>AHC</td>
<td>Accounting Hub Cloud</td>
</tr>
<tr>
<td>Apps</td>
<td>Application</td>
</tr>
<tr>
<td>DIH</td>
<td>Data Integration Hub</td>
</tr>
<tr>
<td>EDD</td>
<td>External Data Descriptor</td>
</tr>
<tr>
<td>GL</td>
<td>General Ledger</td>
</tr>
<tr>
<td>KM</td>
<td>Knowledge Module</td>
</tr>
<tr>
<td>ODI</td>
<td>Oracle Data Integrator</td>
</tr>
<tr>
<td>PT</td>
<td>Pass Through</td>
</tr>
<tr>
<td>SR</td>
<td>Supporting References</td>
</tr>
<tr>
<td>SLA</td>
<td>Sub-Ledger Application / Subledger Application</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>UCM</td>
<td>Universal Content Manager</td>
</tr>
</tbody>
</table>
2 Introduction

2.1 About Oracle Financial Service Analytical Applications

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

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

OFSAA uses industry-leading analytical methods, shared data model and applications 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 domain. It includes many applications such as Profitability Management, Asset Liability Management, Customer Insight, and Risk Management.

2.2 About Accounting Hub Cloud (AHC)

Accounting Hub Cloud (AHC) 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 source system.

2.3 Objective

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

- The OFSAA repository of transaction and reference information becomes a single, unified analytical repository for risk and finance.
- The integration improves consistency in risk and finance data for producing risk adjusted measures.
- OFSAA customers can take advantage of AHC rules and accounting engines to improve the efficiency and audit ability of transforming 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 reconciliation of operational risk losses and accounting as prescribed by regulations such as Basel II and III. These accounting balances can be reconciled using the OFSAA Reconciliation Framework. Reconciliation rules can be defined to map AHC balances to product processor information. For example, accounting for provisions and losses can be based upon customer account balances, as provided to OFSAA and read by AHC enhancing the reliability of operational risk and provisioning measurement.

2.4 About Oracle Financial Services Data Foundation Integration With Fusion Accounting Hub Cloud

The major components that can be leveraged from AHC are:

- Pre-Defined Subledger Transaction Object Models
  - Retail Banking
  - Commercial Banking
  - Investment Banking
  - Islamic Banking
  - Treasury
  - Core Banking PT
- Auto generate AHC Subledger registration and transaction feed templates
- Pre-built automated transaction feed to Accounting Hub Cloud
- Wizard to customize Subledger Transaction Objects
- Pre-built automated GL Balances feed from Cloud GL to OFSAA.

NOTE The supported version of AHC is Accounting Hub Cloud 18C.
2.5 OFSAA-AHC Interface Architecture Overview

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

The data flow between OFSAA and AHC is bi-directional in nature. OFSAA includes all the account, contract and transaction information which AHC needs to generate for its Subledger accounting. Additionally, the 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/custom) and is processed within the AHC using accounting rules. The updated GL balances are then extracted, to load General Ledger Balances and Supporting Reference Balances.

The flow from AHC to OFSAA is achieved through pre-built DIH connectors. These connectors extract the GL and SR Balances from AHC and load them into OFSAA staging post which, this is available for all downstream applications.
The data flow between OFSAA and AHC takes place through files, as AHC is on cloud.

The integration consists of the following processes:

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

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

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

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

Refer section SLA Template to Register Source System for more details.

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

Post AHC configuration, OFSAA batches are performed using pre-packaged AHC connectors, extract transaction data from 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.

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.
3 Mapping the OFSAA User to AHC User Groups

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

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

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

Pre-packaged User Groups for OFSAA - AHC Interface

![User-User Group Map](image1.png)

![Identity Management](image2.png)
MAPPING THE OFSAA USER TO AHC USER GROUPS

OFSAA – AHC INTEGRATION OVERVIEW
4 Logging into AHC-OFSAI Interface

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

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

**NOTE** You should be mapped at least any one of the AHC user groups.
The AH-OFSA landing page is displayed below.
5 Component Details

The following components included in this interface:

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

5.1 Pre-packaged Subledger Information

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

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

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

Each Subledger application is represented as an event class, which can have one or more instruments such as 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 pre-formatted 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>
<thead>
<tr>
<th>Subledger Application</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail Banking</td>
<td>Transaction based</td>
</tr>
<tr>
<td>Commercial Banking</td>
<td>Transaction based</td>
</tr>
<tr>
<td>Investment Banking</td>
<td>Transaction based</td>
</tr>
<tr>
<td>Treasury</td>
<td>Transaction based</td>
</tr>
<tr>
<td>Islamic Banking</td>
<td>Transaction based</td>
</tr>
<tr>
<td>Core Banking PT</td>
<td>Based on Pass-through</td>
</tr>
<tr>
<td></td>
<td>accounting entries</td>
</tr>
</tbody>
</table>

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 Fusion Accounting Hub Cloud.

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

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.

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 SLA Template to Register Source System

AHC has the flexibility to register a source system and create objects within it automatically using a standard template format which is macros-enabled. To register a source system and create a Subledger within AHC, download the recommended, macro-enabled, standard template from the cloud application and upload it back with data. The AHCS Integration for Banking helps in populating the standard template with the Subledger data, which is now automated.

Refer to the list of pre-packaged Subledgers and Configurations.

The sheets present in the standard AHC template are described here.

1. **Instructions**: Provides a brief overview of the template structure and instructions to create the final zip file, which has to be uploaded to the AHC environment.

2. **Source System**: Has sections using which you can input 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/transaction types must be mandatorily provided in this section.
   - There are ‘Validate’ and ‘Generate ZIP’ buttons available on the Source System sheet. DIH populates all the required information in this SLA template. The
3. **Transaction Information**: Provide all the attributes, which you wish to use as header attributes in AHC in this section. You must enter the Name, Short Name, Domain and Journal Display. By default, three mandatory attributes are displayed, which cannot be removed; they are highlighted in grey.

4. **Line Information**: Provide all the attributes, which you wish to use as line attributes in AHC in this section. You must enter Name, Short Name, Domain and Chart of Accounts Display. By default, there are three mandatory attributes displayed, which cannot be removed and are highlighted in grey.

The standard template is prepackaged in to installation kit and will be available in the following path after installation:

```
<FIC_HOME>/ficweb/webroot/dihub/extras/fah/slatemplate/XlaSourceSystemSetup.xlsm
```

However, it is recommended to download the latest XlaSourceSystemSetup.xlsm template from the Accounting Hub Cloud. If it is already downloaded then follow below steps before generating war file (before running /ant.sh script):

1. Open it for editing.
2. In Line Information sheet - rename DEFAULT_AMOUNT and DEFAULT_CURRENCY to TRANSACTION_AMOUNT and TRANSACTION_CURRENCY respectively and save it.
3. Backup/Rename the pre-packaged XlaSourceSystemSetup.xlsm template in
   ```
   <FIC_HOME>/ficweb/webroot/dihub/extras/fah/slatemplate/
   ```
directory and copy the downloaded and edited template here.

### 5.3 DIH Connectors

#### 5.3.1 Extract Connectors

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

There are two External Data Descriptors (EDD) that are created for each Subledger that is saved in AHCS Integration for Banking. 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 have been configured for a Subledger, respectively. The attributes of the Subledger, which are defined in these EDDs, are mapped to the attributes of the staging tables.

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

All Header Connectors accept the following input parameters:

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

All Line Connectors accept the following input parameters:
- **FIC_MIS_DATE** or **MIS_DATE** is the Extraction Date
- **SLA_CODE** Subledger Code or Short Name is a runtime parameter populated automatically from Subledger details (For example ‘RETAIL_SLA’ for Retail Banking)
- **AH_LOAD_RUN_ID** Unique task ID which is auto picked while execution of task
- **AH_FILE_DATE** is the Extraction Date whose file format should not be altered.

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

### 5.3.2 Insert Connectors

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

The following connectors are provided by AHC for GL and SR balances:
- **AH GL Balances Insert Con**: Connector to load thin ledger balances into STG_GL_DATA table in OFSAA. This connector accepts these input parameters:
  - **PERIOD_NAME** Period for which GL data is loaded (For Example – ‘9-Sep’)
  - **FIC_MIS_DATE** or **MIS_DATE** is the Extraction Date
  - **AH_CONsolidATION_FLAG** Runtime parameter and is a single digit flag (For Example ‘C’ or ‘S’).

- **AH SR Balances Insert Con**: Connector to load thick ledger/supporting reference balances into STG_MANAGEMENT_LEDGER tables in OFSAA. This connector accepts these input parameters:
  - **PERIOD_NAME** Period for which GL data is loaded
  - **FIC_MIS_DATE** or **MIS_DATE** is the Extraction Date

For Connectors and mapping details, see the [Insert Connectors Structure file section](#).
6 Implementation

6.1 Background

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

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

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

- Stage Transaction Header (STG_TXN_HEADER) for event based accounting
- Stage Accounting Entries Header (STG_ACCT_ENTRIES_HEADER) for pass through accounting
- Associated Transaction Line Table for Line accounting. Example-STG_ANNUITY_TXNS for event based Accounting and STG_ACCOUNTING_ENTRIES for pass through accounting

6.2 Implementation Guidelines

1. Transaction numbers must match in the header and line files of import data files in AHC. You must carefully create connectors to extract the transaction numbers that have to match accordingly.

2. Ensure you provide the correct ledger name in the AHCS Integration for Banking screens while configuring the Subledger.

3. Do not execute the connector’s batch in AHCS Integration for Banking till the accounting rules and other required setup are configured.

4. Create connectors with appropriate filters, in order to extract only the required data from a table. Once a set of transaction numbers are processed in AHC, it does not accept the same transaction numbers again for a Subledger. Hence, you must carefully extract required data for appropriate accounting.

5. If SLA template is to be reloaded to AHC after changes, you must be careful while replacing the old configuration and must again create the accounting rules accordingly.

6. Only those event type names, which are configured in AHC, can be processed for accounting. If any other event type names come as part of data files imported to AHC, the whole file will not be processed.

7. After the SLA template is uploaded and source system is registered, in case of any further changes, ensure the following steps are followed in both AHCS Integration for Banking and AHC:
   - If you add/edit/delete any of the attribute or transaction types in the AHC screens, but do not update the respective changes in the AHCS Integration for Banking application, integration may fail as data extract files will still have the unchanged structure of DIH.
   - If you add/edit/delete any of the attribute or event type in the AHCS Integration for Banking screens, but do not download the new template and re upload it back
to AHC application, integration may fail. In this scenario, 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 source system/Subledger structure either in AHCS Integration for Banking or AHC, other application structure must also be changed accordingly. If not, it may lead to integration failure.

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

### 6.3 Transaction Object Population

Transaction objects are extracted in to Header and Line csv files and converted to zip format along with 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 auto generated batch is executed.
4. After successful execution of all Tasks in batch, Header information is extracted to XlaTrxH.csv file and Line information to XlaTrxL.csv file.
5. Metadata.txt file is auto populated with Subledger details.
6. All the three files: XlaTrxH.csv, XlaTrxL.csv and Metadata.txt, are auto converted to zip format, which can be uploaded to AHC.
7 AHC Administration

7.1 AHC Settings

1. Navigate to Oracle Financial Services Data Foundation Integration With Fusion Accounting Hub Cloud > AH Administration > AHC Settings.

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

3. Enter the required information under ERP Details for Accounting Hub and Transaction Template Settings fields. Enter these details when the extracted Xla Transaction zip file is uploaded to Accounting Hub Cloud through an automated service.

   a. Enter the following details under ERP Details for Accounting Hub:
      
      — ERP URL for Accounting Hub Cloud:
      
      — This Cloud Service is the AH Cloud URL

      For example:

      https://abc.oracle.com/publicFinancialCommonErpIntegration/ErpIntegrationService

      Only update the domain name up to *.com as the latter part (publicFinancialCommonErpIntegration/ErpIntegrationService) must remain the same.

      — User ID
— Password

— KeyStore Location:

— Ensure to import Certificate to Keystore and save it in FIC_HOME (when ERP connection method is set to Option 2) or WEB_HOME (when ERP connection method is set to Option 1) location. See the Subledger Extract Files section to know how to generate a keystore.

— KeyStore Password

— In ERP connection method select,
  o Option 2 - This is a recommended option.
  o Option 1 - This option is to be selected when option 1 fails and if Oracle-JRF Template is applied while setting up weblogic domain.

b. Enter the following details under Transaction Template Settings:

— Transaction Type Column Name: This textbox is prepopulated with value “TRANSACTION_TYPE”. In case you wish to edit, ensure the value in this textbox should be same as the Transaction Type column name, which is auto, generated while Registering Transaction Source System.

— Transaction Template: Browse and select the transaction template.

c. Click Save.

7.2 Refresh AHC Interface

7.2.1 Deploying OFSAA- AH Connectors Using Refresh AH Interface

After the pre-requisites are complete, you can deploy the AHC connectors that map the multiple file EDDs to the corresponding ADIs through the Refresh AH Interface menu. This creates the corresponding External Data Descriptor and Connectors inside the Data Mapping menu of the AH Interface.

To deploy:

1. Navigate to the AH application interface.
2. Select AH Administration, and click Refresh AH Interface.
3. Select required Source Application Version from the drop-down menu.
5. Click Yes to proceed. After the deployment is complete, the “Deployment Successful” message appears.

6. Navigate to AHC Data Mapping > External Data Descriptor and Connectors for checking the deployed EDDs, Connectors and Mappings.
7.2.2 Undeploying OFSAA-AH Connectors Using Refresh AH Interface

You can use the Undeploy All button to undeploy the connectors. Use the undeploying feature in the following scenarios.

7.2.2.1 Deploying Upgraded Source Version

If there is an upgraded source application (AHC) available, you can undeploy the existing version of the connector, and redeploy the same by selecting the available upgraded source version. The current available source version for AHC connector supports AHC version AH – 18C.
NOTE You can deploy only one source version at a time. You need to upgrade the source, undeploy the version, and then deploy the required version.

7.2.2.2 Changes in ODI / External Data Store AHC

If there is a change in the ODI/External Data Store settings, then you can undeploy the connectors. Modify the settings and redeploy to obtain the latest connector settings.

You cannot undeploy the connectors if any of the Connector/External Data Descriptor is in published mode. Unpublish all the Connector/External Data Descriptor before proceeding with undeployment.

To undeploy:

1. Click Undeploy All to undeploy the Connector version. The message, "Are you sure you want to Undeploy All?" appears.

2. Click Yes to proceed.

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

NOTE Interface connectors can be Undeployed only if:

1. None of the Interface Connectors is in Published state.
2. None of the Custom Connectors is created using the EDDs deployed through Interface version.

7.3 Transaction Types

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

In order to use pre-packaged Subledgers, you must use only the pre-packaged transaction types in the source system data.

To understand the transaction types:

1. Navigate to AH application interface.

2. Select AH Administration, and click Transaction types.

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

3. You can search for Transaction Type Name or Short Name. Enter a part or full name and click the search button to get the filtered result.

4. Under Summary, you can Add, Delete, View or Edit the events.

5. Click **Add**. The transaction types screen appears.
6. Enter the following details to create a new event and click Save.

d. Transaction Type Name

e. Transaction Type Short Name

f. Description

g. Passthrough
Note: The following are the validations for the above fields:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Limitations/Validations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Type Name</td>
<td>Name must not have special characters. Only alphanumeric characters and spaces are allowed. Must not be more than 15 characters. Name must not start or end with a space. Must be unique.</td>
</tr>
<tr>
<td>Transaction Type Short Name</td>
<td>Short Name must have only Uppercase, numbers and underscores. Short Name must start only with an uppercase or a number. Cannot have special characters and space. Must not have more than 15 characters. Must be unique.</td>
</tr>
</tbody>
</table>

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

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

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

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

7.4 Sub-Ledger Attributes

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

7.4.1 Adding a Sub-Ledger Attribute

To add a Subledger Attribute, perform the following steps:

1. Navigate to AH Administration > Subledger Attributes.
The Subledger Attributes screen appears with a list of seeded data with Logical and Physical Name.

2. Click to add a new attribute. The Add Attribute screen appears:
3. Enter the Logical and Physical Name. Physical Name can have only upper case, numbers and alphabets.

4. Select the Domain, Attribute Type and Transaction/Passthrough details from the drop-down list. Transaction/Passthrough is the type of subledger, whether to 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.

5. Click Save.

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

### 7.5 Sub-Ledger Applications

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

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

### 7.5.1 Sub-Ledger Application Summary

To understand the Sub-Ledger Application summary screen:

1. Navigate to AHC application interface.

![Home Page](image)

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

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

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

5. Click **Add** to create a new Sub-Ledger.

6. Select one Sub-Ledger and click **Delete** if you wish to delete a Sub-Ledger.

7. Select one Sub-Ledger and click **Edit** to edit the Sub-Ledger details.

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

8. Select one Sub-Ledger and click **View** to view the details of the Sub-Ledger.

9. Select one Sub-Ledger and click **Download Template**.

10. Select one Sub-Ledger and click **Data Map** to create and map connectors to the selected Sub-Ledger.

11. Select one Sub-Ledger and click **Copy** to copy a sub-ledger.

12. After the details are filled, save the Sub-Ledger.

13. Download the SLA template that has to be uploaded to the AHC application.

**NOTE** It is mandatory to manually validate the template and create a zip file, which must be uploaded to AHC cloud environment. In case of errors during validation, update the required changes in the DIH screens and then download the template again. See the SLA Template Validations section for more information. Subledgers saved are stored in the table FSI_SLA_MASTER.
Click Export to move the summary data to an Excel or CSV format.

### 7.5.2 Adding a Sub-Ledger Application

To add a Sub-Ledger:

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

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

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

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Limitations/Validations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subledger Name</td>
<td>Name must not have special characters. Only alphanumeric characters and spaces are allowed. Must not have more than 15 characters. Name must not start with a space. Must be unique.</td>
</tr>
<tr>
<td>Subledger Short Name</td>
<td>Short Name must have only Uppercase, numbers and underscores. Short Name must start only with an uppercase or a number. Cannot have special characters and space. Must not have more than 15 characters. Must be unique.</td>
</tr>
<tr>
<td>Event Class</td>
<td>Can select one or both the options.</td>
</tr>
</tbody>
</table>
### Field Name | Limitations/Validations
--- | ---
Ledger Name | Must not exceed 100 characters. Change the Ledger name to the required name, for the pre-packaged Subledgers, which by default has ‘Default Ledger’ as the input name.

3. Click Save as Draft. The message, “Subledger draft saved successfully” appears.
4. Click Ok. The Transaction types screen appears.
5. Under Transaction types, perform the following steps:
   a. Click `Transaction Type Mapping`. The Transaction Type Mapping screen appears. All transaction types that have been created as part of Transaction types, along with the pre-packaged list of transaction types screen appears here on the LHS.

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

   

   ![Transaction Type Mapping](image)

   b. Click `>` to move the required entities to the selected values list.
   c. Click `>>` to move all the entities to the selected values list.
   d. You click `<<` to move the entities up and down.
   e. After the entities are selected, click OK.
The selected values are now displayed under Transaction Type Mapping.

6. Under Transaction Information, perform the following steps:
   a. Three mandatory attributes: Ledger Name, Transaction Date, and Transaction Name, are selected by default and cannot be edited.
   b. Click Transaction Attribute Mapping. The Transaction Attribute Mapping screen appears.
   c. The list of transaction attributes is displayed. These are seeded from the FSI_SLA_ATTR_MASTER table.
   d. Ledger Name, Transaction Date, and Transaction Name are selected by default.
   e. Click to move the required entities to the selected values list.
f. Click to move all the entities to the selected values list.

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

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

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

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

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

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

m. After the entities are selected, click OK.

n. Click Next.

7. Under Line Information, perform the following steps:
a. Three mandatory attributes: Transaction Amount, Transaction Currency, and Transaction Number are selected by default and cannot be edited.

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

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

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

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

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

g. Click to move the entities up and down.

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

i. If you wish to display these attributes as part of chart of account, click on Chart Of Account Value. The Chart of Account Mapping screen appears.
NOTE You cannot assign chart of accounts display as yes to the three mandatory columns.

j. Click \(\rightarrow\) to move the required entities to the selected values list.

k. Click \(\rightarrow\) to move all the entities to the selected values list.

l. Click \(\uparrow\) \(\downarrow\) to move the entities up and down.

m. After the entities are selected, click OK.

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

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

7.5.3 Downloading a Template

All data filled in Subledger Applications can be downloaded to a template.

NOTE There is an option to download the template and then map the data.

1. Under the Sub-Ledger Summary screen, select a required row and then click \(\checkmark\) Download Template. A file with the .xlsm extension is downloaded with all the Subledger or source system data filled in.
2. In the Instruction screen, all the details are explained.

3. In Source System, the name and short name given in Subledger Application screen are displayed.
NOTE

The name or short name, which appears in the row, must have the event type name along with the Subledger application name. The Transaction Type name is appended with the SLA name in order to maintain uniqueness of transaction types across all the Subledgers. The length is up to 30 characters and the name limit is adjusted to that.

In Transaction Information, all the transaction types name and short name are displayed along with Journal Displayed, if it is selected as Yes.
4. In Line Information, all transaction types name and short names are displayed along with Chart of Account Value, if it is selected as Yes.
AHC ADMINISTRATION
SUB-LEDGER APPLICATIONS

NOTE The three mandatory columns are listed in grayed area for both transaction and line information.
Click Validate, under Source System Transactions to check if there are any errors in the template.

5. Click Validate, under Source System Transactions to check if there are any errors in the template.

NOTE In case, you want to make changes in the template, change the configurations in the DIH screens, and download the template again.

6. Click Generate ZIP, under Source System Transactions to upload this file in AH Cloud.

NOTE Upload the zip file in the right task list within AHC and map the ‘Entered Amount’ and ‘Entered Currency’ in Manage accounting attributes to Transaction Amount and Transaction Currency, before configuring the rules.

7.5.4 Using Data Map
To create a connector and map it to a Subledger:

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

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

a. Click Create to add a new data mapping. The New Connector screen appears.
b. Under Target, the EDD is automatically created. For example, Subledger name is Commercial Banking. The EDD is created with AHC SLA as the prefix of the name and Header as the suffix. Full name is “AH SLA Commercial Banking Header”. EDD structure will have all the attributes that have been mapped to the Subledger under Transaction Information.

![External Data Store](image1)

![External Data Descriptor](image2)

![NOTE](image3)

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

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

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

![Map the required source and target and save the connector.](image4)

d. Double-click mapping to map the source and target.
NOTE  For more information on mapping, see the Connectors section in the DIH User Guide in OHC.

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

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

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

i. Click to move the required entities to the selected values list.
ii. Click ▶️ to move all the entities to the selected values list.

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

iv. After the entities are selected, click OK.

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

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

b. Under Target, the EDD is automatically created. For example, Subledger 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 will have all the attributes that have been mapped to the Subledger under the Line Information.

c. Map the required source and target, and save the connector.
d. Double-click Mapping to map the source and target.
NOTE You must apply filters, if any, while creating connectors in order to fetch the correct data from a product for a particular Subledger. Extraction Date filter is already enabled with pre-packaged connectors for executing them. For more information on mapping, see the Connectors section in the DIH User Guide in OHC.

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

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

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

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

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

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” appears.
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 the popup up message (<INFODOM>_DIH_AH_<SLA_CODE>).

6. In case you wish to unpublish the connector, click **Unpublish Connectors**.

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

**NOTE**  
EDDs are created after saving a Subledger. EDD names are stored in the FSI_SLA_EDD_MAP table. All the transaction and line data mapping connectors saved are stored in the FSI_SLA_DATA_MAPPING table. See the [Connectors Structure](#) section for more information.

### 7.5.5 Copying a Sub-Ledger Application

To copy a Sub-Ledger:

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

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

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

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

3. The EDD name is also replaced with the copied name in the connector screen in the format AH SLA <SLA name> Header and AH SLA <SLA name> Line.
7.6 Metadata Configuration for Accounting Filters

Each subledger takes in data from header tables and corresponding instrument or transaction tables. There are a few instruments, which has data combined for two or more subledgers. There is a need to apply appropriate filters within these instruments data to extract the data according to the subledger type.

Hence, a list of standard product values are included as filters in the seeded subledger connectors. Those instruments, which have a common data for two or more subledgers are identified and listed in the following table:

<table>
<thead>
<tr>
<th>PP table</th>
<th>Subledgers</th>
</tr>
</thead>
<tbody>
<tr>
<td>STG_CASA</td>
<td>Retail Banking</td>
</tr>
<tr>
<td>STG_LOAN_CONTRACTS</td>
<td>Commercial Banking</td>
</tr>
<tr>
<td>STG_CARDS</td>
<td>Commercial Banking</td>
</tr>
<tr>
<td>STG_TD_CONTRACTS</td>
<td>Commercial Banking</td>
</tr>
<tr>
<td>STG_OD_ACCOUNTS</td>
<td>Commercial Banking</td>
</tr>
<tr>
<td>STG_SWAPs_CONTRACTS</td>
<td>Commercial Banking</td>
</tr>
<tr>
<td>STG_MM_CONTRACTS</td>
<td>Treasury</td>
</tr>
<tr>
<td>STG_INVESTMENTS</td>
<td>Treasury</td>
</tr>
<tr>
<td>STG_FUTURES</td>
<td>Treasury</td>
</tr>
<tr>
<td>STG_FX_CONTRACTS</td>
<td>Treasury</td>
</tr>
<tr>
<td>STG_CREDIT_DERIVATIVES</td>
<td>Treasury</td>
</tr>
<tr>
<td>STG_OPTION_CONTRACTS</td>
<td>Treasury</td>
</tr>
<tr>
<td>STG_REPO_CONTRACTS</td>
<td>Treasury</td>
</tr>
<tr>
<td>STG_COMMITMENT_CONTRACTS</td>
<td>Commercial Banking</td>
</tr>
<tr>
<td>STG_CORRESPONDENT_ACCOUNT</td>
<td>Commercial Banking</td>
</tr>
</tbody>
</table>

- Within each instrument table, seeded product codes are created for specific subledger.
- Map the product codes in each of the instrument tables to the seeded list of products provided.

For example, STG_CASA has data relating to both Commercial SLA and Retail SLA. Hence, corresponding seeded list of products are ‘RETACC’ and ‘COMACC’, where you can map those products that must go to Retail subledger to RETACC and that data which should go to commercial subledger to COMACC product respectively.

These mappings should be inserted in to table named SETUP_MASTER in the OFSAA atomic schema.

Sample entries assuming that STG_CASA has four products out of which two must go to Retail Banking Subledger and other two to Commercial Banking

SETUP_MATSER must have the following entries:
<table>
<thead>
<tr>
<th>V_COMPONENT_CODE (Actual Product Code)</th>
<th>V_COMPONENT_VALUE (Seeded Product Code)</th>
<th>V_COMPONENT_DESC (Description)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting Product 1</td>
<td>RETACC</td>
<td>STG_CASA to Retail Banking</td>
</tr>
<tr>
<td>Accounting Product 2</td>
<td>RETACC</td>
<td>STG_CASA to Retail Banking</td>
</tr>
<tr>
<td>Commercial Product 1</td>
<td>CORPACC</td>
<td>STG_CASA to Commercial Banking</td>
</tr>
<tr>
<td>Commercial Product 2</td>
<td>CORPACC</td>
<td>STG_CASA to Commercial Banking</td>
</tr>
</tbody>
</table>

Refer to [list of seeded product codes](#) for instrument tables and seeded product codes to be used.

### 7.7 OFSAA Chart of Account (COA) Mapping

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

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

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

The following list of pre-defined dimensions are seeded in the application and which are mapped to COA segments: General Ledger Account Code

- Legal Entity Code
- Account Branch Code
- Gaap Code
- Currency Code
- Product Code
- Organization Unit Code
- Business Unit Code
- Customer Class Code

### 7.7.1 Mapping Segments

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

To map the segments, perform the following steps:
1. Navigate to Oracle Financial Services Data Foundation Integration With Fusion Accounting Hub Cloud > AH Administration > OFSAA COA Mapping.

The Subledger Chart of Account Map screen appears.

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

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

2. Select the required COA segments and click Save.

These segment values are replaced with the dimension columns as mapped in the inbound connector and the data is loaded as per the mapping into the staging tables of ledger.

Once it is saved the Insert connectors automatically gets mapped for both GL and SR.
NOTE

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

7.8 Subledger Extract Files

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

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

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

When an SLA is selected, new window opens which contains the list of files uploaded for that SLA.
It has four columns:

- **File Name**: The files uploaded appear under the File Name column. Click the file name to download the file to a local system.

- **Doc ID**: For the extracted .zip files, where batch execution is completed with sixth parameter as N has ‘Upload to UCM’ option in this field for the first time. You can click this and application uploads the created .zip file to UCM. Once it is uploaded, a unique Doc ID number is created and it appears in this column. For files executed with parameter 6 as Y and parameter 7 as 1, a unique doc ID number appears upfront.

- **Process ID**: For the extracted zip files, where batch execution is completed with sixth parameter as Y and seventh parameter as two, it has a unique Process ID number in this column. For those files, which are executed with sixth parameter as N or Y (along with seventh parameter as 1), has ‘Process File’ option in this field for the first time. Process File will either upload the .zip file to UCM and trigger the process/job ‘Import Accounting Transactions’ or directly trigger the uploaded file with the Import Accounting Transactions process. Click ‘Process File’ and the following actions occur:
  - If Doc ID number exists – .zip file, which is already uploaded to UCM under the existing Doc ID, will be triggered for processing and a unique Process ID is created, which appears under this column after 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 couple of seconds, which reflects the status of ‘Import Accounting Transactions’ job.

### 7.9 Import Certificate in to Keystore

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

1. Access the AHC SSL URL for any web service using Microsoft Internet Explorer/Mozilla.
2. Click Security Lock symbol and click View Certificates.
3. On the Certification Path tab, certificate hierarchy appears. Export the top two certificates (DigiCert and DigiCertSHA2SecureServerCA in below case) to file.

4. On the Certification Path tab, select DigiCert and click View Certificate.
6. Click Next and enter a name and location for the file you want to export.
7. Change the encoding to Base-64 and click Next.
8. Provide the file name and click Finish.
9. Repeat steps 4 through 8 for the DigiCertSHA2SecureServerCA certificate.
   When using other web browsers, perform similar steps. The navigation path may differ in the web browsers used.
10. Type the following command to import a certificate into keystore:
    
    ```
    keytool -import -trustcacerts -file <filename> -alias <aliasname> -keystore keystore.jks -storepass welcome1
    ```
11. Run the following command to verify if the trust store contains the imported certificates:

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

### 7.10 Subledger Event Grouping

Subledger Event Grouping functionality is introduced in order to group data based on few seeded dimensions while extracting and then posting them to AHC. Grouping based on the dimensions is significant. 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 the combining data based on some pre-seeded columns in both header and line level data. When data is grouped on some dimensions and transaction numbers are not considered for grouping, there is a need to regenerate the transaction numbers. As aggregate columns are different in header and line level, aggregation happens separately for header and line data. However, the only connection between the line and header data is just the transaction number. Hence, to maintain the connection, pair up 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

#### 7.10.1 Subledger Event Grouping Summary

To understand the Sub-Ledger Event Grouping summary screen:

1. Navigate to AHC application interface.

2. Select AHC Administration, and click Subledger Event Grouping. A list of pre-packaged event groups information for transmission to Accounting Hub appears.
The summary screen displays the Event Group Name, Header and Line Connectors:

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

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

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

3. You can search for Sub-Ledger Event Grouping Name.

4. Under Summary, you can Define an Event Group, Manage Group Attributes and Delete user defined group events.

5. Click ![delete icon] if you wish to delete a Sub-Ledger Event Group.

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

### 7.10.2 Defining an Event Group

To define an event group, perform the following steps:

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

   ![Define Event Group Window]

2. Enter the required details and click Save.

### 7.10.3 Managing Group Attributes

To manage group attributes, perform the following steps:
1. On the Subledger Event Grouping screen, click \( \text{ } \) to manage Group Attributes. The Manage Group Attributes window appear displaying the available Event Groups.

2. Select the Event Groups that are created previously, for which you want to define the group attributes.
   
   a. Click \( \text{ } \) to move the required event groups to the selected values list.
   
   b. Click \( \text{ } \) to move all the event groups to the selected values list.
   
   c. Click \( \text{ } \) or \( \text{ } \) to move the event groups back one by one or all respectively.

3. After the event groups are selected, click \( \text{ } \) Next to move to the Manage Group Attributes screen.

Here, by default all the columns, which are a part of mappings in the selected event groups are listed under the selected Group Attributes, that is, they are a part of group by clause for grouping.
A list of grayed out attributes cannot be removed from group attribute list, as they are the basic dimensions on which ledger data is required.

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

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

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

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

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

   ![Confirmation Message]

Click Yes. The Subledger Group Attributes are updated successfully.
8 Prerequisites for Deploying OFSAA-AHC Connectors

8.1 Additional DIH Configurations for AHC Interface

Follow the below steps:

1. Complete the ODI settings using Settings option in DIH Application Administration menu before deploying the interface.
2. Click Settings in Administration screen.

The ODI settings screen is displayed. Update the setting information correctly before proceeding to the deployment of AHC interface connectors.

<table>
<thead>
<tr>
<th>Settings Menu</th>
<th>Values Required</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODI User</td>
<td>User Name used for Login to ODI</td>
<td>SUPERVISOR</td>
</tr>
<tr>
<td>ODI Password</td>
<td>ODI Password for the ODI user to login</td>
<td>odipassword</td>
</tr>
<tr>
<td>Master Repository DB User</td>
<td>Master Repository DB Schema User Name created for ODI</td>
<td>DIHDEV_ODI_REPO</td>
</tr>
<tr>
<td>Master Repository DB Password</td>
<td>Master Repository DB Schema Password</td>
<td>dbpassword</td>
</tr>
<tr>
<td>Master DB Driver</td>
<td>Oracle Driver (Use the Default)</td>
<td>oracle.jdbc.OracleDriver</td>
</tr>
<tr>
<td>Master DB Connection</td>
<td>Oracle Database JDBC URL</td>
<td>jdbc:oracle:thin:@10.184.135.6:1521:DIHDB</td>
</tr>
</tbody>
</table>
### PREREQUISITES FOR DEPLOYING OFSAA-AHC CONNECTORS

#### ADDITIONAL DIH CONFIGURATIONS FOR AHC INTERFACE

<table>
<thead>
<tr>
<th>Settings Menu</th>
<th>Values Required</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Repository</td>
<td>Repository used inside ODI</td>
<td>DIHREP</td>
</tr>
</tbody>
</table>

**NOTE** For AHC Batch Execution there should be a ODI Agent configured on ficdb server and Default Agent should be pointing to this Agent.

3. A source named INTF_AH_FSDF_TAR_FILES is present in External Data Store under DIH Application. Select the entry INTF_AH_FSDF_TAR_FILES to Edit.

4. Enter the details of the folder path where AH files must be extracted. The path must be in the same sever where the ODI Agent is installed and configured. Refer to [DIH User Manual](#) for more details on External Data Store File settings.

5. Navigate to DIH Application, click Menu.

6. Click Application Data Interface.

7. On the RHS click Refresh ADI.

8. Click Start icon. 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 Infodom.
9. Navigate to DIH Application, click Menu.
10. Click Application Data Interface.
11. On the RHS click Target Datastore Refresh.

12. Click on Start icon. This refreshes all available target data stores.

**NOTE** Refer to [DIH User Manual](#) to obtain detailed information for any of the steps mentioned above.
9 GL and SR Balance Extraction using AHC Connectors

This chapter includes:
- Configuration and data files required.
- Execute DIH Connectors

9.1 Configuration and Data Files Required

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

9.1.1 Chart of Account to OFSAA Dimension Mapping

Refer to section Metadata Configuration for Accounting Filters for more information.

9.2 GL and XLA csv files from AHC

The following two files have to be extracted from AHC along with Header Names and data must be placed in EDS path before executing Connectors:

1. ah_gl_balance.csv from fusion.gl_ofsaa_balances_v(All columns)
2. ah_sr_balance.csv from fusion.xla_ofsaa_balances_v(All Columns)

9.3 Executing AHC Connectors

To extract GL balances and SR Balances from AHC after performing the previous steps execute the following below steps:

1. Navigate to Oracle Financial Services Data Foundation Integration With Fusion Accounting Hub Cloud > Orchestration > Batch Maintenance.
3. Go to batch Orchestration and search for auto generated batch AH__CONNECTORS.
NOTE Sometimes due to unknown reasons, when deployed in WebLogic, not all batches are created. Perform the following steps to avoid the issue:

1. Login to WebLogic server admin console.
2. Navigate to Console - >Services- ->Data Sources and select Atomic Schema.DS ->Connection Pool.
3. Update Statement-Cache-Size value to 0.
4. Restart the WebLogic server and redeploy connectors from AHC Refresh Interface screen.

4. Select AH GL Balances Insert Con Task and click Edit.
5. In Batch Parameters specify,
   a. PERIOD_NAME for which GL Balances have to be extracted (for example PERIOD_NAME='Aug-18')
   b. AH_CONSOLIDATION_FLAG (for example AH_CONSOLIDATION_FLAG='C').
   c. Update MIS_DATE format to the extracted date format of ACCOUNTING_DATE column present in the file ah_gl_balance.csv.
      For Example if date is '08-Oct-18' then format will be MIS_DATE='MISDATE:dd-MMM-yyyy'.
6. Repeat step 3 through 5 for AH SR Balances Insert Con.
7. Go to Batch Execution and search for AH_Connectors and specify FIC_MIS_DATE and execute the batch.

Refer DIH User Guide for more information on publishing and executing DIH connectors.
10 Execution of Extract Connectors

10.1 Configuration of Metadata
Refer to section Metadata Configuration for Accounting Filters.

10.2 Batch Execution

NOTE Source system configuration, all the required accounting rules, and other settings must be done in the AHC environment before proceeding to Batch Execution.

In the Batch Execution page, there are batches created for each Subledger after the connectors are published in the Subledger section. You can select the required Batch ID, specify the date for which the data must be extracted, and click Execute Batch to trigger the batch.

All the tasks specified within the Batch ID are executed. Tasks involve executing all the connectors that are mapped to the selected Subledger. Connectors extract the data from the respective source tables and load it into a file format that can be read in AHC.

Every SLA batch contains corresponding ID Map, Header and Line connector tasks and one “RUN Executable” task at the end.

NOTE Id Map connectors (connectors ending with “*Id Map”) are used to populate ‘FSI_TXN_AH_HEADER_MAP’ table which holds actual Transaction Header IDs and regenerated Header IDs mapping. Same table is then used in Header and Line Connectors.

ATTENTION In case the batch is re-executed for the same date and Id Map connector tasks have already been executed once, then make sure to exclude these Id Map connectors and include only Header and Line connector Tasks.

This ‘Run Executable’ task performs the following actions:

1. Identifies the extracted Header and Line csv files.
2. Formats data by removing duplicate Headers rows in both Header and Line files.
4. Creates a zip file including Header, Line and Metadata files.
5. The following structure helps to 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 temp folder along with a log file.

By default, AH ERP service is disabled to automatically upload the zip file to UCM and process. This can be enabled by updating sixth parameter of the last task to “Y” before executing the batch. With AH ERP service enabled, seventh parameter can be set to one (only uploads zip file to UCM) or two (uploads zip file to UCM and trigger Import job to process file) accordingly.
Send Us Your Comments

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

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

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

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