

CV.060

Conversion Technical Design

ASI-II Supplier hub

Conversion of Data from fas to supplier Hub

Conversion ID: FRC-C-011

Author: Akshay Nayak

Creation Date: February 21, 2017

Last Updated: April 10, 2017

Version: 1.4

1. **Title, Subject, Last Updated Date, Reference Number**, **and** **Version** are marked by a Word Bookmark so that they can be easily reproduced in the header and footer of documents. When you change any of these values, be careful not to accidentally delete the bookmark. **You can make bookmarks visible by selecting Tools->Options…View and checking the Bookmarks option in the Show region.**

**Approvals:**

1. To add additional approval lines, press [Tab] from the last cell in the table above.

|  |  |
| --- | --- |
| **Name** | **Designation** |
| Becki Memmer | IT |
| Brenton Van Breda | PMO |
| Eric Anton | IT |
| Kathy Naumann | PMO |
| Mathew Ditzler | IT |
| Preeti Wadhwa | PMO |
| Seema Pahwa | PMO |
| Terry Alberts | PMO |
| Tottie Bunye | IT |
| David Carrier | IT SQA |

## Document Control

**Change Record**

1

| Date | Author | Version | Change Reference |
| --- | --- | --- | --- |
| 07-MAR-2016 |  |  |  |
| 21-FEB-2017 | Akshay Nayak | 1.0 | Initial Draft |
| 21-MAR-2017 | Akshay Nayak | 1.1 | Updated as per latest changes in CV040 |
| 10-APR-2017 | Akshay Nayak | 1.2 | Updated with Source and Destination Mapping and added Scenario Description. |
| 12-APR-2017 | Akshay Nayak | 1.3 | Updated data flow diagram. Added attribute group name and attribute name in the mapping table. |
| 10-MAY-2017 | Akshay Nayak | 1.4 | For Franchisee1 Tax Payer Id must be equal to SSN. |

**Reviewers**

| Name | Designation |
| --- | --- |
|  |  |
| Steve Barton | PMO |
| Shabir Firdaus | IT |
| Sanjay Kulshrestha | IT |
| Mark Markovic | IT |
| Zein Sweis | IT |
| Richard Russell | IT |
| Dwayne Wilson | IT |
| Richard Parker | IT |
| Robin Bryant\* (PDSS) | IT |
| Barry Krause (1A Host) | IT |
| Tanja McClain (FranConnect) | IT |
| Scott Barefield\* (LicenseHQ) | IT |
| Todd Cohen (Interfaces) | IT |
| Neeraj Pathania (Mainframe CTS) | CTS |
| Divya Gupta | Deloitte |
| David Carrier | IT SQA |

**Distribution**

| Copy No. | Name | Location |
| --- | --- | --- |
|  |  |  |
|  | Eric Anton | IT |
|  | Matthew Ditzler | IT |
|  | Roland Ehigiamusoe | Deloitte Partner Firm |
|  | Shabir Firdaus | IT |
|  | Ben Hillman | VMA Partner Firm |
|  | Barry Krause | IT |
|  | Sanjay Kulshrestha | IT |
|  | Paul McCollum | IT |
|  | Seema Pahwa | PMO |
|  | Dee Paskonis | OCS Partner Firm |
|  | Richard Russell | IT |
|  | Zein Sweis | IT |
|  | Eric Hunter | OCS Partner Firm |
|  | Ken Milton | Deloitte Partner Firm |
|  | Sam Shalaby | OCS Partner Firm |
|  | David Carrier | IT SQA |

1. The copy numbers referenced above should be written into the **Copy Number** space on the cover of each distributed copy. If the document is not controlled, you can delete this table, the Note To Holders, and the **Copy Number** label from the cover page.

Contents

Document Control 2

Introduction 5

Process Area 5

Conversion Assumptions 6

References 6

Scope 6

Clean-Up Criteria 7

Data Flow Diagram 9

Approach 12

Conversion Tables 15

Ordering of Tables 16

Dependencies 16

Processing Rules 17

Translation Rules 18

Filter Rules 19

Incremental Conversion Filters (optional) 19

Foreign Key Rules 20

Derivation Rules 21

Default Values 22

Conversion Staging Table Creation Program Logic 23

Extract Program Logic 25

Translation Program Logic 26

Load Program Logic 27

Conversion/Validation Program Logic 28

Conversion Program Modules 41

ETL Objects and Connectivity 41

Conversion Programs 41

Automated Conversion Tool Files 41

Conversion Execution Instructions 42

Conversion Validation Strategy 43

Open and Closed Issues for this Deliverable 44

Open Issues 44

Closed Issues 44

Glossary 46

Appendix A (Optional) 47

Appendix B (Optional) 48

1. To update the table of contents, put the cursor anywhere in the table and press [F9]. To change the number of levels displayed, select the menu option Insert‑>Index and Tables, make sure the Table of Contents tab is active, and change the Number of Levels to a new value.

## Introduction

This Conversion Program Design defines the key assumptions, rules, and logic that are needed to create the conversion programs. The conversion program code is not included in this document.

This document is the detailed specification for the conversion of Supplier and Supplier sites from FAS into Oracle Supplier Hub. At a detail-level, this document includes the basic components of conversion design, including:

* Detailed data mapping of attributes
* Transformation rules required for conversion
* Pseudo algorithm of the program flow
* Exception handling and reprocessing of the program in case of any errors faced.

### Process Area

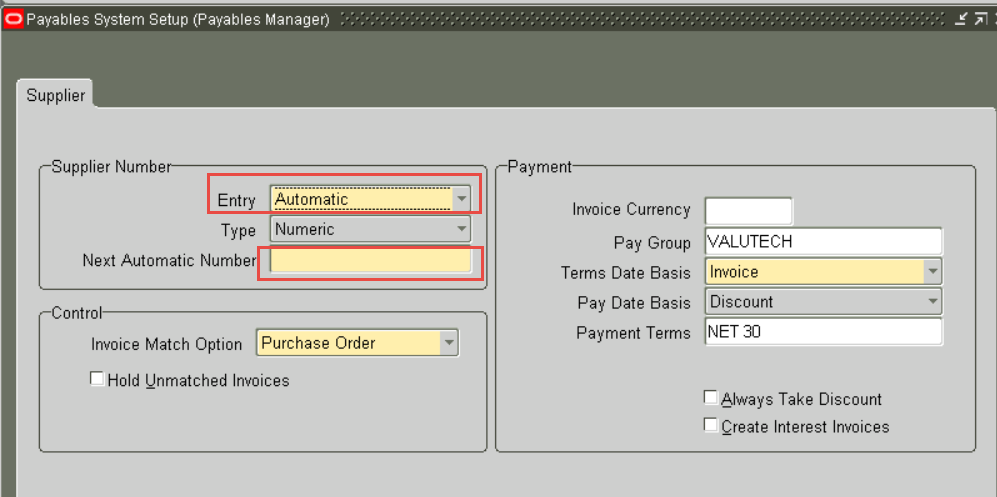
Order to Cash  Forecast to Delivery  Procure to Pay

Record to Report  Idea to Offering  Other (Supplier Hub)

## Conversion Prerequisite

The following application setups must be completed in the Oracle Supplier Hub prior to the data conversion taking place. The details of the below steps can be found in the document Supplier Hub BR100, which must be completed and signed prior to the user acceptance test:

* Necessary setups (including values sets populated with reference data, attribute groups and lookup codes) must be completed in Oracle Site Hub before the conversion program starts:
  + Attribute Group – Franchisee Details
  + Attributes – Franchisee SSN, Date of Birth, First Franchisee Date, Ethnicity, Marital Status and Sex
* Disable the standard Oracle Data Quality Management Sync Program by disabling the profile option HZ: Enable DQM Merge Suggestion at the responsibility level before the conversion begins. Turn off or disable all the business events like create supplier, update supplier events using the standard profile option for the conversion user so as to not trigger any updates to the downstream systems
* Update the Payables System Setup under Payables setup to enable the Supplier number entry to be automatic and choose the sequence number to be used for the conversion.



* Directories where files must be kept must be created.
* Disable following Oracle Alerts that currently triggers and sends notification to Oracle AP during the execution of the conversion program
  + SLCAP - CCEC INVOICES PAYGROUP CHECK
  + SLCAP - FUTURE TERMS DATE
  + SLCAP - NEW VENDOR
  + SLCAP - NULL PAY GROUP ALERT
  + SLCAP Mismatch PayGroup Alert

### References

|  |  |  |
| --- | --- | --- |
| Document Name | Revision | Document Description |
| CV040 FRC-C-011 FAS to Supplier Hub v1.6 |  | [CV040 FRC-C-011 FAS to Supplier Hub v1.6](http://7hub.7-eleven.com/projects/ASI/ProjectTeamSite/tracks2and6/Shared%20Documents/06-Design%20and%20Build/Track%206%20(Franchise%20Management)/CV040%20-%20Draft/Ready%20for%20review/CV040%20FRC-C-011%20FAS%20to%20Supplier%20Hub%20v1.6.docx) |

### Scope

The information in this document outlines the conversion design of Supplier and Supplier Sites information from FAS legacy system to the target system (Oracle Supplier Hub) based on the conversion requirements.

The scope of this conversion object includes the following:

* Data Source – LFASB
* Target System – Oracle Supplier Hub
* Data Entities – Franchisee

| Module | Data Object | Source System | Target System | Data Volume |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Oracle Supplier Hub | Conversion from FAS to Oracle Supplier Hub | FAS | Oracle Supplier Hub | 15,000 |

**Data Selection Criteria**

* All Franchisee both Active and Inactive must be picked from FAS legacy system.

Note: Data extraction logic is not in scope of this document.

* The extract from FAS must contain 2 files with the below details:
  + The current/most updated Franchisee details for all store letter codes/ operators (Franchisee letter codes only and not applicants)
    - File Name format must be ISPFAS\_CURRENT\_<DDMMYYYYHH> (Day, month, year and Hour)
  + History of Franchisee records (changes made to the Franchisee’s during the life of a store letter code/operator and not applicants)
    - File Name format must be ISPFAS\_HISTORY\_<DDMMYYYYHH> (Day, month, year and Hour)
* Data must be presented in CSV file format.
* Data File Name must be having following file format ISPFAS\_CURRENT\_<DDMMYYYYHH> and ISPFAS\_HISTORY\_<DDMMYYYYHH>.
* This file must be placed into shared drive or database server by the legacy team.

**Data Manipulation Criteria**

* The 0 and blank values must be converted to null while loading into the staging table**.**

**Data Exclusion Criteria**

There is no exclusion criteria. All records must be eligible for conversion.

### Clean-Up Criteria

**Pre-Conversion Clean-Up Criteria**

* The SEI business team must validate data extracts through sampling and sign off prior to conversion.
* Planned cleansing activities must occur in the source systems prior to extraction of data by SEI business users.
* During the mock conversion cycles in the SEI development environment, the legacy team must scramble or mask sensitive data for example, Federal Tax ID, SSN etc. prior to generating the extract file.
* Any exceptions that occur during the conversion process because of issues in source data, must be handled by the business manually i.e. the data must be loaded manually and/or entered directly into Supplier Hub.

**Post-Conversion Clean-Up Criteria**

Not Applicable

## Data Flow Diagram

The Supplier and Supplier Site data from legacy system FAS must be brought to Oracle Supplier Hub using this conversion program. Below are the steps involved in the process?

1. Data Extract- Business team must extract data for Supplier and Supplier Sites from FAS in comma delimited format. Data file must be placed in Shared drive or database server by the business team.
2. Loading Data- Data must be picked from CSV files and loaded into staging table using SQL Loader utility.
3. Validation- PLSQL package must be build. PLSQL package must be called from concurrent program. Program must pick all new records or error and failed records from staging table based on the mode in which the program is ran and validate record. Status of the records must be marked Error/Success after completing of the program.
4. Process – Same PLSQL package must be called from Concurrent Program to pick all valid records from the staging table and call API’s to load Supplier and Supplier Site information into Oracle Supplier Hub.
5. Concurrent program will run in below modes.
6. Validate- In this mode PLSQL unit must pick all new records from the staging table i.e. records with Status = ‘N’ and validate record(s).
7. Process- In this mode PLSQL unit must pick all valid records from staging table i.e. records with status as V and import record(s) into Oracle Supplier Hub
8. Revalidate – In this mode PLSQL unit must pick all records which had failed during validation or import in previous run i.e. records with status as E and F and validate records.
   1. F - Stands for records failed during validation.
   2. E - Stands for records failed during import.



**Dependencies on Reference Data**

Not Applicable.

**Dependencies on Master Data**

Not Applicable.

**Dependencies on Other Transaction Data**

Not Applicable.

## Approach

Following list of objects must be created for FAS to Supplier Hub conversion:

1. Type: Table

Name: *SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG*

Schema: *ISPAPPS*

File Name: *SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG\_T.sql*

1. Type: Sequence

Name: *SLC\_ISP\_FASSUPP\_BATCH\_ID\_S*

Schema: *ISPAPPS*

File Name: *SLC\_ISP\_FASSUPP\_BATCH\_ID\_S.sql*

1. Type: Sequence

Name: *SLC\_ISP\_FASSUPP\_RECORD\_ID\_S*

Schema: *ISPAPPS*

File Name: *SLC\_ISP\_FASSUPP\_RECORD\_ID\_S.sql*

1. Type: Synonym

Name: *SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG*

Schema: *APPS*

File Name*: CR\_SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG\_SYN.sql*

1. Type: Synonym

Name: *SLC\_ISP\_FASSUPP\_BATCH\_ID\_S*

Schema: *APPS*

File Name: *CR\_SLC\_ISP\_FASSUPP\_BATCH\_ID\_S\_SYN.sql*

1. Type: Synonym

Name: *SLC\_ISP\_FASSUPP\_RECORD\_ID\_S*

Schema: *APPS*

File Name: *CR\_SLC\_ISP\_FASSUPP\_RECORD\_ID\_S\_SYN.sql*

1. Type: Package

Name: *SLC\_ISP\_FASSUPP\_CNV\_PKG*

Schema: APPS

File Name: *SLC\_ISP\_FASSUPP\_CNV\_SPKG.sql* and *SLC\_ISP\_FASSUPP\_CNV\_BPKG.sql* for package spec and package body respectively.

Package must have all the necessary logic to validate the data from the staging table. Program must also have logic to call API’s to create Supplier in Oracle Supplier Hub.

1. Create control file that must be used to load data from CSV file into staging table.

Control File name must be

* ISPCNV\_SLC\_CURRENT\_FASSUPPLIER.ctl – This must be responsible for loading data from Current file
* ISPCNV\_SLC\_HISTORY\_FASSUPPLIER.ctl – This must be responsible for loading data from History file.

1. Below is the data mapping table.

Data Mapping for Current File:

| Source Field | Target Field | Attribute Group Name/Attribute | Comments |
| --- | --- | --- | --- |
| STORE NUMBER | AP\_SUPPLIER\_SITES\_ALL. VENDOR\_SITE\_CODE |  | Mandatory |
| LTR |  |  |  |
| FRANCHISEE 1: BIRTH DT | POS\_SUPP\_PROF\_EXT\_B. D\_EXT\_ATTR1 | *SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_DOB* |  |
| FRANCHISEE 2: BIRTH DT | POS\_SUPP\_PROF\_EXT\_B. D\_EXT\_ATTR1 | *SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_DOB* |  |
| FRANCHISEE ADDR LINE 1 | AP\_SUPPLIER\_SITES\_ALL.ADDRESS\_LINE1 |  | Mandatory |
| FRANCHISEE CURR CITY | AP\_SUPPLIER\_SITES\_ALL.CITY |  | Mandatory |
| FRANCHISEE CURR STATE | AP\_SUPPLIER\_SITES\_ALL.STATE |  | Mandatory |
| FRANCHISEE CURR ZIP | AP\_SUPPLIER\_SITES\_ALL.ZIP |  | Mandatory |
| FRANCHISEE 1: FIRST NAME | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR5/AP\_SUPPLIERS.VENDOR\_NAME | *SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_FIRST\_NAME* | Mandatory |
| FRANCHISEE 2: FIRST NAME | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR5/AP\_SUPPLIERS.VENDOR\_NAME | *SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_FIRST\_NAME* | Mandatory |
| FRANCHISEE 1: INITIAL | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR6/AP\_SUPPLIERS.VENDOR\_NAME | *SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_MIDDLE\_NAME* | ` |
| FRANCHISEE 2: INITIAL | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR6/AP\_SUPPLIERS.VENDOR\_NAME | *SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_MIDDLE\_NAME* |  |
| FRANCHISEE 1: LAST NAME | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR7/AP\_SUPPLIERS.VENDOR\_NAME | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_LAST\_NAME | Mandatory |
| FRANCHISEE 2: LAST NAME | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR7/AP\_SUPPLIERS.VENDOR\_NAME | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_LAST\_NAME | Mandatory |
| PHONE NUM | AP\_SUPPLIER\_SITES\_ALL.PHONE |  |  |
| FRANCHISEE 1: SSN | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR8 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_SSN2 | Mandatory  SSN must be used to determine if Supplier1 is already existing. |
| FRANCHISEE 2: SSN | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR8 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_SSN2 | Mandatory  SSN must be used to determine if Supplier2 is already existing. |
| INCORPORATING AT CHANGEOVER (Y/N) Flag |  |  | This flag must be used in code to determine if Corporation exists or no. |
| INCORPORATION NAME | AP\_SUPPLIERS.VENDOR\_NAME |  | Mandatory if INCORPORATING AT CHANGEOVER (Y/N) Flag value is Y |
| FRANCHISEE 1: BKGRD | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR1 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_Ethnicity |  |
| FRANCHISEE 1: MARTL | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR2 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_MT |  |
| FRANCHISEE 1: SEX | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR3 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_SEX |  |
| FRANCHISEE 2: BKGRD | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR1 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_Ethnicity |  |
| FRANCHISEE 2: MARTL | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR2 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_MT |  |
| FRANCHISEE 2: SEX | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR3 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_SEX |  |
| FEDERAL ID # | HZ\_PARTIES.JGZZ\_FISCAL\_CODE |  | Mandatory |
| FRANCHISEE 1: ZID |  |  | This field is extracted for future reference. As of now this field is not used anywhere. |
| FRANCHISEE 2: ZID |  |  | This field is extracted for future reference. As of now this field is not used anywhere. |
| FRAN1 ORIGINAL DATE | POS\_SUPP\_PROF\_EXT\_B.D\_EXT\_ATTR2 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_CO |  |
| FRAN2 ORIGINAL DATE | POS\_SUPP\_PROF\_EXT\_B.D\_EXT\_ATTR2 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_CO |  |
| Operator Effective Date Begin |  |  | This field is extracted for future reference. As of now this field is not used anywhere. |
| Operator Effective Date End |  |  | This field is extracted for future reference. As of now this field is not used anywhere. |

Data Mapping for History File:

| Source Field | Target Field |  | Comments |
| --- | --- | --- | --- |
| STORE NUMBER | AP\_SUPPLIER\_SITES\_ALL. VENDOR\_SITE\_CODE |  | Mandatory |
| LTR |  |  | *Not Mapped* |
| FRANCHISEE 1: BIRTH DT | POS\_SUPP\_PROF\_EXT\_B. D\_EXT\_ATTR1 | *SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_DOB* |  |
| FRANCHISEE ADDR LINE 1 |  |  | *Not Mapped* |
| FRANCHISEE CURR CITY |  |  | *Not Mapped* |
| FRANCHISEE CURR STATE |  |  | *Not Mapped* |
| FRANCHISEE CURR ZIP |  |  | *Not Mapped* |
| FRANCHISEE 1: FIRST NAME | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR5/AP\_SUPPLIERS.VENDOR\_NAME | *SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_FIRST\_NAME* | Mandatory |
| FRANCHISEE 1: MIDDLE NAME | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR6/AP\_SUPPLIERS.VENDOR\_NAME | *SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_MIDDLE\_NAME* |  |
| FRANCHISEE 1: LAST NAME | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR7/AP\_SUPPLIERS.VENDOR\_NAME | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_LAST\_NAME | Mandatory |
| PHONE NUM | AP\_SUPPLIER\_SITES\_ALL.PHONE |  |  |
| FRANCHISEE 1: SSN | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR8 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_SSN2 | SSN must be used to determine if Supplier1 is already existing. |
| FRANCHISEE 1: BKGRD | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR1 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_Ethnicity |  |
| FRANCHISEE 1: MARTL | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR2 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_MT |  |
| FRANCHISEE 1: SEX | POS\_SUPP\_PROF\_EXT\_B.C\_EXT\_ATTR3 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_SEX |  |
| FEDERAL ID # | HZ\_PARTIES.JGZZ\_FISCAL\_CODE |  |  |
| FRANCHISEE 1: ZID |  |  | This field is extracted for future reference. As of now this field is not used anywhere. |
| FRAN1 ORIGINAL DATE | POS\_SUPP\_PROF\_EXT\_B.D\_EXT\_ATTR2 | SLC\_ISP\_FRANCHISEE\_DETAILS/SLC\_ISP\_FRANCHISEE\_CO |  |

1. For Franchisee1 and Franchisee2 SSN must be used to determine if Franchisee already exists or no. If Franchisee does not exists in the system then call API’s to create new Franchisee.

If Franchisee is already existing then do not update existing records. Updating franchisee if franchisee information is not part of this conversion object.

1. Vendor Name must follow below naming standard

For Franchisee1 and Franchisee2 vendor name must be

<Franchisee First name><space>< Franchisee Middle name><space>< Franchisee Last Name><space>\_<Supplier Number>

For Corporation vendor name must be

<Incorp Name>\_<Supplier Number>

1. For Corporation Federal Id# must be used to validate if Franchisee for Corporation is already existing. If Corporation information is already exists then do not update. If Corporation information is not existing then create new corporation.
2. It is possible that for there exists a supplier in the system with same Federal Id# as that of current record while conversion program is run. For such case since query must be joining with UDA tables. For existing records since UDA information must be missing and thus program must create franchisee for Corporation.
3. Create Store for every record for Corporation or for Franchisee1 based on the Incorporation Flag.
4. For History file program must be creating only Franchisee1. Not Franchisee2 or corporation or Site number be created for History File.
5. Below are the various scenarios for a given record in Current File.

| Scenario | Converted Data |
| --- | --- |
| Case1:  Incorporation Flag = Y  Corporation information present  Franchisee1 information present  Store Information present | * Verify SSN for Franchisee1 in UDA tables. If Franchisee1 information already exists then do not create Franchisee again. If Franchisee1 Information does not exists then create Franchisee1 * Verify Federal tax# for Corporation in UDA tables. If Corporation information already exists then do not create Corporation again. If Corporation Information does not exists then create Corporation * If Site is not existing then create Site information for Corporation. If Site information is already exists then do not perform anything. |
| Case2:  Incorporation Flag = Y  Corporation information present  Franchisee1 information present  Franchisee2 information present  Store Information present | * Verify SSN for Franchisee1 in UDA tables. If Franchisee1 information is already existing then do not create Franchisee again. If Franchisee1 Information not exists then create Franchisee1. * Verify SSN for Franchisee2 in UDA tables. If Franchisee2 information is already existing then do not create Franchisee again. If Franchisee2 Information is not existing then create Franchisee2. * Verify Federal tax# for Corporation in UDA tables. If Corporation information already exists then do not create Corporation again. If Corporation Information not exists then create Corporation * If Site is not existing then create Site information for Corporation. If Site information already exists then do not perform anything. |
| Case3:  Incorporation Flag = N  Franchisee1 information present  Store Information present. | * Verify SSN for Franchisee1 in UDA tables. If Franchisee1 information is already existing then do not create Franchisee again. If Franchisee1 Information is not existing then create Franchisee1. * If Site is not existing then create Site information for Franchisee1. If Site information already exists then do not perform anything. |
| Case4:  Incorporation Flag = N  Franchisee1 information present  Franchisee2 information present  Store Information present | * Verify SSN for Franchisee1 in UDA tables. If Franchisee1 information is already existing then do not create Franchisee again. If Franchisee1 Information not exists then create Franchisee1. * Verify SSN for Franchisee2 in UDA tables. If Franchisee2 information is already existing then do not create Franchisee again. If Franchisee2 Information is not exists then create Franchisee2. * If Site is not existing then create Site information for Franchisee1. If Site information already exists then do not perform anything. |

1. If there is any error while processing any information for a given store then must mark that rollback information for that record and mark the status of the record as E and populate error message in the error message column.
2. Consolidate Summary of unique Franchisee(s) created and unique stores created must be put in OUT file. Franchisee1\_exists\_flag, Franchisee2\_exists\_flag,Incorp\_exists\_flag,Site\_exists\_flag columns of staging table must be used to display above information.
3. Provide a description of the concurrent program associated with <Subject> customization and include the supplemental information below. If no concurrent program (other than reports addressed above) are included in the customization, then delete this section.

### Conversion Tables

The following tables are going to be populated in the Oracle application for the conversion of Supplier and Supplier Sites

* *SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG*

### Ordering of Tables

Not applicable

### Dependencies

Not applicable

**Foreign Key Dependencies**

Not Applicable.

**Parent/Child Dependencies**

Not Applicable.

**Quick Code Dependencies**

Not Applicable.

**Data Element Domains**

Not Applicable.

**Use of Sequence Generators**

Not Applicable.

## Processing Rules

This section lists the processing rules that are to be used in the conversion of FAS to Supplier Hub:

Not Applicable.

## Translation Rules

* All zero in the data must be converted to null during validation mode. All the dates in format 000000 must be converted to NULL.
* Background, Ethnicity and Sex value received from the input file needs to be translated as per below valueset before passing value to Oracle API’s.
* Code is obtained in file. Code must be replaced with description before calling Oracle API’s.

Lookup

| Type | | Meaning | | | Application | | | | Description | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | |  |  | | |  |
| *SLCISP\_BACKGROUND\_LKP* | *Lookup for Background* | | | *SLC Custom APPS* | | | | *Lookup for Background* | | |
| *SLCISP\_MARITAL\_LKP* | *Lookup for Marital Status* | | | *SLC Custom APPS* | | | | *Lookup for Marital Status* | | |
| *SLCISP\_SEX\_LKP* | *Lookup for Sex* | | | *SLC Custom APPS* | | | | *Lookup for Sex* | | |

Lookup Codes

| Type | | Code | | | Meaning | | | | Description | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | |  |  | | |  |
| *SLCISP\_BACKGROUND\_LKP* | *A* | | | *A* | | | | *American Indian* | | |
| *SLCISP\_BACKGROUND\_LKP* | *B* | | | *B* | | | | *African American (Black)* | | |
| *SLCISP\_BACKGROUND\_LKP* | *H* | | | *H* | | | | *Hispanic* | | |
| *SLCISP\_BACKGROUND\_LKP* | *O* | | | *O* | | | | *Asian Pacific islander* | | |
| *SLCISP\_BACKGROUND\_LKP* | *P* | | | *P* | | | | *Other* | | |
| *SLCISP\_BACKGROUND\_LKP* | *W* | | | *W* | | | | *White* | | |

| Type | | Code | | | Meaning | | | | Description | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | |  |  | | |  |
| *SLCISP\_MARITAL\_LKP* | *M* | | | *M* | | | | *Married* | | |
| *SLCISP\_MARITAL\_LKP* | *S* | | | *S* | | | | *Single* | | |
| *SLCISP\_MARITAL\_LKP* | *A* | | | *A* | | | | *Married* | | |

| Type | | Code | | | Meaning | | | | Description | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | |  |  | | |  |
| *SLCISP\_SEX\_LKP* | *F* | | | *F* | | | | *Female* | | |
| *SLCISP\_SEX\_LKP* | *M* | | | *M* | | | | *Male* | | |

## Filter Rules

Not Applicable.

### Incremental Conversion Filters (optional)

Not Applicable.

## Foreign Key Rules

Not Applicable.

## Derivation Rules

Below is a table listing the derivation rules that are to be used in the conversion of FAS to Supplier Hub:

| Processing Rule | Data Source | Data Element | Data Size/Type | Target Table. Column | Data Size/Type | |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | |
| Derive the *BATCH\_ID* | Custom Sequence to generate a unique value at every run | *SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG* | *NUMBER* | *SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG.* *BATCH\_ID* | *NUMBER* | |
| Derive the *RECORD\_ID* | Custom Sequence for assigning record Id for every record inserted into *SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG* staging table | *SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG* | *NUMBER* | *SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG.* *RECORD\_ID* | *NUMBER* | |

## Default Values

This section lists the default value rules that are to be used in the conversion of FAS to Supplier Hub. The default value rules explain the logic behind why a certain default value has been selected.

| Processing Rule | Data Source | Data Element | Data Size/Type | Target Table. Column | Data Size/Type | |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | |
| Default the PROCESS\_FLAG as NEW for all the records while inserting into the staging tables | Not Applicable | Not Applicable | Not Applicable | SLC\_SM\_PRF\_CNV\_STG .PROCESS\_FLAG | VARCHAR2(255) | |
| Default Vendor\_type\_lookup\_code to FRANCHISEE | Not Applicable | Not Applicable | Not Applicable | Ap\_suppliers. vendor\_type\_lookup\_code | VARCHAR2(30) | |
| Default Vendor\_type\_lookup\_code to TERM\_NAME.  Based on TERM\_NAME pos\_vendor\_pub\_pkg.create\_vendor API must derive TERM\_ID internally | Not Applicable | Not Applicable | Not Applicable | Ap\_suppliers.Terms\_id | NUMBER | |
| Default PAY\_GROUP\_LOOKUP\_CODE to FRANCHISEE | Not Applicable | Not Applicable | Not Applicable | Ap\_suppliers. PAY\_GROUP\_LOOKUP\_CODE | VARCHAR2(25) | |
| Default PAY\_DATE\_BASIS\_LOOKUP\_CODE to DISCOUNT | Not Applicable | Not Applicable | Not Applicable | Ap\_suppliers. PAY\_DATE\_BASIS\_LOOKUP\_CODE | VARCHAR2(25) | |
| Default INVOICE\_CURRENCY\_CODE to USD | Not Applicable | Not Applicable | Not Applicable | Ap\_suppliers. INVOICE\_CURRENCY\_CODE | VARCHAR2(15) | |
| Default PAYMENT\_CURRENCY\_CODE to USD | Not Applicable | Not Applicable | Not Applicable | Ap\_suppliers. PAYMENT\_CURRENCY\_CODE | VARCHAR2(15) | |
| Default start\_date\_active to SYSDATE in pos\_vendor\_pub\_pkg.create\_vendor API | Not Applicable | Not Applicable | Not Applicable | Ap\_suppliers. START\_DATE\_ACTIVE | DATE | |
| Default enabled\_flag to ‘Y’ in pos\_vendor\_pub\_pkg.create\_vendor API | Not Applicable | Not Applicable | Not Applicable | Ap\_suppliers. ENABLED\_FLAG | VARCHAR2(1) | |
| Default PAY\_SITE\_FLAG to Y | Not Applicable | Not Applicable | Not Applicable | ap\_supplier\_sites\_all. PAY\_SITE\_FLAG | VARCHAR2(1) | |
| Default Country to ‘US’ | Not Applicable | Not Applicable | Not Applicable | ap\_supplier\_sites\_all.COUNTRY | VARCHAR2(60) | |
| Deafult Operating Unit name to ‘SLC CONSOLIDATED’.  pos\_vendor\_pub\_pkg.create\_vendor\_site API determine org\_id internally | Not Applicable | Not Applicable | Not Applicable | ap\_supplier\_sites\_all .ORG\_ID | NUMBER | |
| Default Site Name as Home | Not Applicable | Not Applicable | Not Applicable |  | VARCHAR2(30) | |
| Default RECORD\_TYPE value to either HISTORY or CURRENT based on from where data is getting loaded. | Not Applicable | Not Applicable | Not Applicable |  | VARCHAR2(10) | |

## Conversion Staging Table Creation Program Logic

Data extracted from Legacy System must be kept in shared drive or database server.

SQL Loader program must be built to load data from csv file to staging table.

Common concurrent program which was built as part of Track -1 must be reused. As part of this conversion control file must be created. Common loader concurrent program must take control file name and path and data file name and path as parameter.

Data must be loaded into staging table.

Details of the staging table are as bellows.

1. Describe the program logic for the interface table creation program(s) used to convert the business object.   
     
   Include pseudo-code if appropriate.

Schema: ISPAPPS

Table: *SLC\_* *SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG*

Staging table structure used for of conversion logic:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S No. | Column Name | Data type | Not Null? | Default value |
|
| *1* | RECORD\_ID | NUMBER P | *Yes* | *ISPAPPS.SLC\_ISP\_FASSUPP\_RECORD\_ID\_S* |
| *2* | BATCH\_ID | NUMBER |  |  |
| *3* | STORE\_NUMBER | VARCHAR2 |  |  |
| *4* | STORE\_LETTER\_CODE | VARCHAR2 |  |  |
| *5* | STATUS | VARCHAR |  |  |
| *6* | INCORP\_FLAG | VARCHAR2 |  |  |
| *7* | INCORP\_NAME | VARCHAR2 |  |  |
| *8* | FRANCHISEE1\_PARTY\_ID | NUMBER |  |  |
| *9* | FRANCHISEE2\_PARTY\_ID | NUMBER |  |  |
| *10* | INCORP\_PARTY\_ID | NUMBER |  |  |
| *11* | FRANCHISEE1\_BIRTH\_DATE | VARCHAR2 |  |  |
| *12* | FRANCHISEE2\_BIRTH\_DATE | VARCHAR2 |  |  |
| *13* | FRANCHISEE1\_FIRST\_NAME | VARCHAR2 |  |  |
| *14* | FRANCHISEE2\_FIRST\_NAME | VARCHAR2 |  |  |
| *15* | FRANCHISEE1\_MIDDLE\_NAME | VARCHAR2 |  |  |
| *16* | FRANCHISEE2\_MIDDLE\_NAME | VARCHAR2 |  |  |
| *17* | FRANCHISEE1\_LAST\_NAME | VARCHAR2 |  |  |
| *18* | FRANCHISEE2\_LAST\_NAME | VARCHAR2 |  |  |
| *19* | FRANCHISEE1\_SSN | VARCHAR2 |  |  |
| *20* | FRANCHISEE2\_SSN | VARCHAR2 |  |  |
| *21* | FRANCHISEE1\_BKGRD | VARCHAR2 |  |  |
| *22* | FRANCHISEE2\_BKGRD | VARCHAR2 |  |  |
| *23* | FRANCHISEE1\_MARITAL | VARCHAR2 |  |  |
| *24* | FRANCHISEE2\_MARITAL | VARCHAR2 |  |  |
| *25* | FRANCHISEE1\_SEX | VARCHAR2 |  |  |
| *26* | FRANCHISEE2\_SEX | VARCHAR2 |  |  |
| *28* | FRANCHISEE1\_ZID | VARCHAR2 |  |  |
| *29* | FRANCHISEE2\_ZID | VARCHAR2 |  |  |
| *30* | FEDERAL\_ID | VARCHAR2 |  |  |
| *31* | FRANCHISEE1\_ORIGINAL\_DATE | VARCHAR2 |  |  |
| *32* | FRANCHISEE2\_ORIGINAL\_DATE | VARCHAR2 |  |  |
| *33* | EFFECTIVE\_BEGIN\_DATE | VARCHAR2 |  |  |
| *34* | EFFECTIVE\_END\_DATE | VARCHAR2 |  |  |
| *35* | ADDRESS1 | VARCHAR2 |  |  |
| *36* | CITY | VARCHAR2 |  |  |
| *37* | ZIP | VARCHAR2 |  |  |
| *38* | STATE | VARCHAR2 |  |  |
| *39* | PHONE\_NUM | VARCHAR2 |  |  |
| *40* | EMAIL\_ADDR | VARCHAR2 |  |  |
| *41* | ERROR\_MSG | VARCHAR2 |  |  |
| *42* | REQUEST\_ID | NUMBER |  |  |
| *43* | RECORD\_TYPE | VARCHAR2 |  |  |
| *44* | CREATION\_DATE | DATE |  |  |
| *45* | CREATED\_BY | NUMBER |  |  |
| *46* | LAST\_UPDATE\_DATE | DATE |  |  |
| *47* | LAST\_UPDATED\_BY | NUMBER |  |  |
| *48* | LAST\_UPDATE\_LOGIN | NUMBER |  |  |
| *49* | FRANCHISEE1\_EXISTS\_FLAG | VARCHAR2 |  |  |
| *50* | FRANCHISEE2\_EXISTS\_FLAG | VARCHAR2 |  |  |
| *51* | INCORP\_EXISTS\_FLAG | VARCHAR2 |  |  |
| *521* | SITE\_EXISTS\_FLAG | VARCHAR2 |  |  |

## Extract Program Logic

Extraction logic from Source System i.e. FAS is not in scope of this document.

* Data must be extracted in common delimited file.
* All Franchisee’s (Active and Inactive) must be extracted.
* The extract from FAS must contain 2 files as below
  + The current/most updated Franchisee details for all store letter codes/ operators (Franchisee letter codes only and not applicants)
    - File Name format must be ISPFAS\_CURRENT\_<DDMMYYYYHH> (Day, month, year and Hour)
  + History of Franchisee records (changes made to the Franchisee’s during the life of a store letter code/operator and not applicants)
    - File Name format must be ISPFAS\_HISTORY\_<DDMMYYYYHH> (Day, month, year and Hour)
* All dates must be in YYMMDD format
* Data must be extracted in attached format.



* CSV file must be placed in shared drive.

1. Describe the program logic for the upload program(s) used to convert the business object.   
     
   Include pseudo-code if appropriate.

## Translation Program Logic

Not Applicable

## Load Program Logic

* SQL Loader program must be built to load data from csv file to staging table.
* Common concurrent program which was built as part of Track -1 must be reused. As part of this conversion create control file. Common loader concurrent program must take control file name and path and data file name and path as parameter.
* Data must be loaded into staging table.
* Details of the staging table is mentioned in *Conversion Staging Table Creation Program Logic* section
* Concurrent Program Name which must be used for loading data into control file is *SLCUTIL Common Data Loader Conversion Program*
* Control File must be created. Control File Name must be ISPCNV\_SLC\_CURRENT\_FASSUPPLIER.ctl and ISPCNV\_SLC\_HISTORY\_FASSUPPLIER.ctl
* In control file the mode in which data is loaded is in APPEND mode as history of the data loaded is necessary.
* Control file is designed to skip 1 i.e. first record which is the header line of the csv file.

## Conversion/Validation Program Logic

This section describes the logic for the conversion interface/validation programs that must be built to support the conversion of Franchisee and its site from FAS to Supplier Hub.

* PLSQL Store Procedure type Concurrent Program must be built. Concurrent program details with the parameter list is mentioned below.

| Concurrent Program Short Name | | Sequence | Parameter | Description | | | | Value Set | | Default Value | | Display Size | | Concatenated Description Size | | | Prompt |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |  | | | |  | |  |  |  |  | |
| *SLCISPFASSUPPLIERCNV* | *10* | | *P\_PROCESSING\_MODE* | *This must decide in which mode conversion program must be run* | | | | *SLCISP\_CONVMODE* |  | | | *25* | | *25* | | | *Mode* |
| *SLCISPFASSUPPLIERCNV* | *20* | | *P\_BATCH\_SIZE* | *Batch Size* | | | | *NUMBER\_15* | *50000* | | | *15* | | *25* | | | *Batch Size* |
| *SLCISPFASSUPPLIERCNV* | *30* | | *P\_DEBUG\_FLAG* | *Debug Flag* | | | | *YES\_NO* | *NO* | | | *15* | | *25* | | | *Debug Flag* |

Value Set Definition:

| Value Set Name | | Description | | | List Type | | Security Type | | | Format Type | | | Maximum Size | | Validation Type | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | | | |  |  | |  | | |  | |  |  |
| *SLCISP\_CONVMODE* | *Value set for Conversion Modes* | | | *List of Values* | | *No Security* | | | | *Char* | | *30* | | | *Independent* | | |

Value Set Values:

| Value Set Name | | | Value |
| --- | --- | --- | --- |
|  |  | | |
| *SLCISP\_CONVMODE* | | *VALIDATE* | |
| *SLCISP\_CONVMODE* | | *PROCESS* | |
| *SLCISP\_CONVMODE* | | *REVALIDATE* | |

* Concurrent program must be run in below modes.

1. Validate- In this mode PLSQL unit must pick all new records from the staging table i.e. records with Status = ‘N’ and validate record(s).
2. Process- In this mode PLSQL unit must pick all valid records from staging table i.e. records with status as V and import record(s) into Oracle Supplier Hub
3. Revalidate – In this mode PLSQL unit must pick all records which had failed during validation or import in previous run i.e. records with status as E and F and validate record(s).
   1. F - Stands for records failed during validation.
   2. E - Stands for records failed during import.

* PLSQL Package must be built. Package name is SLC\_ISP\_FASSUPP\_CNV\_PKG
* Each row in the file corresponds to one Store information. Thus record must contain information for Franchisee1. If Incorporation Flag is Y then record must contain information for Corporation and optionally record will contain information for Franchisee2.
* If Franchisee is already existing then must program must not create new Franchisee. Franchisee existing condition is checked based on SSN number which must be stored in Franchisee UDA.
* Similarly Corporation uniqueness is checked based on the Federal Id. If Federal Id for a corporation is already existing then new Corporation must not be created.
* Site must be created for the Store and vendor\_site\_code must be equal to the Store Number. Uniqueness of Store is checked based on the vendor\_site\_code. If vendor\_site\_code is matched then new Site is not created.
* If location is already existing then while creating new Site use same location\_id. Program must not create new location.
* Functions and Procedure in the package must be

Functions:

1. SLC\_IS\_DATE\_VALID\_F
2. SLC\_GET\_TRANSFORMED\_DATE\_P
3. SLC\_GET\_LOOKUP\_MEANING\_P
4. SLC\_IS\_DATE\_VALID\_F

Parameters:

p\_in\_date VARCHAR2

Purpose: This function must validate the date passed as parameter. If date is valid function must return Y else function must return N.

Logic:

Perform to\_date operation on date parameter with format as YYMMDD. If date is in valid format there must be no error else function must return error.

1. SLC\_GET\_TRANSFORMED\_DATE\_P

Parameters:

p\_in\_date VARCHAR2

Purpose:

The date passed in the file is of format YYMMDD. If program performs to\_date then for year like 68 , Oracle converts date as 2068.

This function must extract the year part from the code. If year is less than or equal to 17, then this procedure must convert year to 2017 else year must convert date as 19YY.

After performing this transformation date object is retuned.

This function must be called from SLC\_PROCESS\_UDA\_ATTRIBUTES\_P to transform various date values like Franchisee1\_birth\_date, Franchisee2\_birth\_date.

1. SLC\_GET\_LOOKUP\_MEANING\_P

Parameters:

p\_in\_lookup\_type IN VARCHAR2

p\_in\_lookup\_code IN VARCHAR2

Purpose:

Data file is sending code for various fields like Sex, Ethnicity, and Background. While saving in UDA tables program must store meaning.

Thus lookups are created. Details of the lookups are mentioned in the document in table below.

This function returns lookup meaning for the lookup code passed for a given lookup type.

This function must be called from SLC\_PROCESS\_UDA\_ATTRIBUTES\_P to transform various fields like Sex, Ethnicity and Background.

Procedures:

1. SLC\_ASSIGN\_BATCH\_ID\_P
2. SLC\_PRINT\_SUMMARY\_P
3. SLC\_CREATE\_SUPPLIER\_P
4. SLC\_UPDATE\_ORG\_PROFILE\_P
5. SLC\_PROCESS\_UDA\_ATTRIBUTES\_P
6. SLC\_CREATE\_SUPPLIER\_SITE\_P
7. SLC\_VALIDATE\_P
8. SLC\_IMPORT\_P
9. SLC\_MAIN\_P
10. SLC\_POPULATE\_ERR\_OBJECT\_P

Procedures:

1. SLC\_MAIN\_P

Parameters: p\_errbuff OUT VARCHAR2,

p\_retcode OUT NUMBER,

p\_processing\_mode IN VARCHAR2,

p\_batch\_size IN NUMBER,

p\_debug\_flag IN VARCHAR2

Logic:

* If p\_batch\_size parameter is null then set value to 50000
* If p\_processing\_mode is Validate then call assign\_batch\_id passing status as N
* If p\_processing\_mode is Revalidate then call assign\_batch\_id passing status as E and F
* After calling SLC\_ASSIGN\_BATCH\_ID\_P call *SLC\_VALIDATE\_P* to validate the data from the staging tables.
* If p\_processing\_mode is Process then call assign\_batch\_id passing status as V.
* After calling assign\_batch\_id call *SLC\_IMPORT\_P* to import data from the staging tables into Oracle Supplier Hub
* Call slc\_print\_summary\_p procedure to print summary of the concurrent program run into OUT file.

1. SLC\_ASSIGN\_BATCH\_ID\_P

Parameters:

p\_in\_status1 IN VARCHAR2

p\_in\_status2 IN VARCHAR2

p\_in\_status3 IN VARCHAR2

p\_in\_batch\_size IN NUMBER

Logic:

* Assign next sequence value of sequence SLC\_ISP\_FASSUPP\_BATCH\_ID\_S and update BATCH\_ID column for staging table SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG
* Records must be picked based on the status passed to the procedure and rownum restricted by p\_in\_batch\_size
* Batch Id generated must be kept at the package body level to be used by other procedures.

1. SLC\_CREATE\_SUPPLIER\_P

Parameters:

p\_in\_vendor\_name IN VARCHAR2

p\_in\_vendor\_name\_alt IN VARCHAR2

p\_in\_segment1 IN VARCHAR2 DEFAULT NULL

p\_in\_vendor\_type\_lkp\_code IN VARCHAR2

p\_in\_term\_name IN VARCHAR2

p\_in\_pay\_date\_basis\_code IN VARCHAR2

p\_in\_pay\_group\_lookup\_code IN VARCHAR2

p\_in\_invoice\_currency\_code IN VARCHAR2

p\_in\_payment\_currency\_code IN VARCHAR2

p\_in\_jgzz\_fiscal\_code IN VARCHAR2

p\_in\_tax\_reporting\_name IN VARCHAR2

p\_in\_organization\_type IN VARCHAR2

p\_out\_party\_id OUT NUMBER

p\_out\_error\_flag OUT VARCHAR2

p\_out\_err\_msg OUT VARCHAR2

Purpose:

Create Supplier and Party of type Organization.

Logic:

* Assign values passed as parameter to this procedure to local object of type ap\_vendor\_pub\_pkg.r\_vendor\_rec\_type
* Call API *pos\_vendor\_pub\_pkg.create\_vendor* to create vendor.
* Error encountered must be returned to the calling procedure in OUT variable.
* Error message returned by API must be collected from FND\_MSG\_PUB using FND\_MSG\_PUB.GET and assigned to OUT variable.

Note: Every franchisee which is created as Supplier must also be created as party of type Organization in TCA. Party is created by *pos\_vendor\_pub\_pkg.create\_vendor* API internally.

1. SLC\_UPDATE\_ORG\_PROFILE\_P

Parameters:

p\_in\_party\_id IN NUMBER

p\_in\_organization\_name IN VARCHAR2

p\_in\_tax\_payer\_id IN VARCHAR2

p\_out\_error\_flag OUT VARCHAR2

p\_out\_err\_msg OUT VARCHAR2

Purpose:

Update organization name and tax payer id for the party id passed as parameter to this procedure.

Logic:

* Create local variables for apps.hz\_party\_v2pub.organization\_rec\_type and hz\_party\_v2pub.party\_rec\_type.
* Assign values passed as parameter to these variables.
* This procedure must take party\_id, legal status and organization name as in parameter and must call API *HZ\_PARTY\_V2PUB.UPDATE\_ORGANIZATION* to update Organization name and Legal Structure Name of the Franchisee.
* Error encountered must be returned to the calling procedure in OUT variable.
* Error message returned by API must be collected from FND\_MSG\_PUB using FND\_MSG\_PUB.GET and assigned to OUT variable.

1. SLC\_PROCESS\_UDA\_ATTRIBUTES\_P

Parameters:

p\_in\_party\_id IN NUMBER

p\_in\_ssn IN NUMBER

p\_in\_dob IN VARCHAR2

p\_in\_bkgrd IN VARCHAR2

p\_in\_marital IN VARCHAR2

p\_in\_sex IN VARCHAR2

p\_in\_conversion\_source IN VARCHAR2

p\_in\_original\_date IN VARCHAR2

p\_in\_effec\_begin\_date IN VARCHAR2

p\_in\_effec\_end\_date IN VARCHAR2

p\_in\_first\_name IN VARCHAR2

p\_in\_middle\_name IN VARCHAR2

p\_in\_last\_name IN VARCHAR2

p\_out\_error\_flag OUT VARCHAR2

p\_out\_err\_msg OUT VARCHAR2

Logic:

* This procedure must update UDA information into Oracle Supplier Hub.
* Row records i.e *ego\_user\_attr\_row\_obj* must be created and inserted into *ego\_user\_attr\_row\_table* in SYNC mode.
* For all the attributes update ego\_user\_attr\_data\_obj records populate in ego\_user\_attr\_data\_table.
* Call *ego\_user\_attrs\_data\_pub.process\_user\_attrs\_data* API to update UDA information in Oracle Supplier Hub.
* Error encountered must be returned to the calling procedure in OUT variable.
* Error message returned by API must be collected from FND\_MSG\_PUB using FND\_MSG\_PUB.GET and assigned to OUT variable.

1. SLC\_CREATE\_SUPPLIER\_SITE\_P

Parameters:

p\_in\_party\_id IN NUMBER

p\_in\_vendor\_site\_code IN VARCHAR2

p\_in\_addressline1 IN VARCHAR2

p\_in\_addressline2 IN VARCHAR2 DEFAULT NULL

p\_in\_city IN VARCHAR2 DEFAULT NULL

p\_in\_state IN VARCHAR2 DEFAULT NULL

p\_in\_county IN VARCHAR2 DEFAULT NULL

p\_in\_country IN VARCHAR2 DEFAULT NULL

p\_in\_zip IN VARCHAR2 DEFAULT NULL

p\_in\_phone IN VARCHAR2 DEFAULT NULL

p\_in\_ou IN VARCHAR2

p\_out\_error\_flag OUT VARCHAR2

p\_out\_err\_msg OUT VARCHAR2

Logic:

* This procedure must create new Supplier Site for Supplier.
* Create local variables for ap\_vendor\_pub\_pkg.r\_vendor\_site\_rec\_type and APPS.HZ\_PARTY\_SITE\_V2PUB.PARTY\_SITE\_REC\_TYPE.
* If for a supplier a location is already existing then program must assign same location\_id to the new supplier site program is creating for new supplier. If location\_id is NULL then pass address information like Address1, City, State, Zip and other information in ap\_vendor\_pub\_pkg.r\_vendor\_site\_rec\_type record.
* This procedure must accept party\_id, vendor\_site\_code and other address information like AddressLine1, City, State, and Zip as parameter.
* Supplier Site name must be defaulted as Home.
* pay\_site\_flag must be defaulted to Y.
* Call API *pos\_vendor\_pub\_pkg.create\_vendor\_site* to create Supplier Site.
* Error encountered must be returned to the calling procedure in OUT variable.
* Error message returned by API must be collected from FND\_MSG\_PUB using FND\_MSG\_PUB.GET and assigned to OUT variable.

1. SLC\_VALIDATE\_P

Logic:

* This procedure must validate eligible records from the staging table. Procedure must pick records based on the batch\_id assigned by SLC\_ASSIGN\_BATCH\_ID\_P procedure.
* This procedure must be called in 2 modes.

1. Validate
2. Revalidate

* Update error message column NULL for all eligible records.
* In Validate Mode it must pick valid records and in Revalidate Mode program must pick errored and failed records and perform below validation

1. If all 3 field values - Franchisee1 FirstName, Franchisee1 Middle Name and Franchisee1 LastName is null for any record then update the error message in the error\_msg column of the staging table.
2. If Franchisee1 SSN is null for any record then update the error message in the error\_msg column of the staging table.
3. If Incorp Flag is Y for any record then Incorp Name and Federal Id value is mandatory. If not then update the error message in the error\_msg column of the staging table.
4. If either of the field is having value i.e. Franchisee2 FirstName, Franchisee2 Middle Name and Franchisee2 LastName then Franchisee2 SSN value is mandatory. If not then update the error message in the error\_msg column of the staging table.
5. Address1, City, Zip and State is mandatory parameter. If either of the value is NULL then update the error message in the error\_msg column of the staging table.
6. Update 00/00/00 with NULL for all date fields like FRANCHISEE1\_BIRTH\_DATE , FRANCHISEE2\_BIRTH\_DATE, FRANCHISEE1\_ORIGINAL\_DATE, FRANCHISEE2\_ORIGINAL\_DATE, EFFECTIVE\_BEGIN\_DATE, EFFECTIVE\_END\_DATE
7. Update error message in the error\_msg column of the staging table if for any records date format is not correct. Date fields are FRANCHISEE1\_BIRTH\_DATE, FRANCHISEE2\_BIRTH\_DATE, FRANCHISEE1\_ORIGINAL\_DATE, FRANCHISEE2\_ORIGINAL\_DATE, EFFECTIVE\_BEGIN\_DATE, and EFFECTIVE\_END\_DATE.
8. Update error message if value provided for Background, Ethnicity and Sex for Franchisee1 and Franchisee2 is not as per configured lookup value.

* If error message is populated for any record then we conclude that record is failed validation error thus mark the record status as F.
* For all other records for which error\_msg field is NULL then we conclude that validation has passed thus mark the record status as V.
* Call print\_summary procedure to print summary of the Validation in OUT file of concurrent program.

1. slc\_import\_p:

Parameters: NONE

* This procedure must call other procedure like create\_supplier, update\_org\_profile, process\_uda\_attributes and create\_supplier\_site.
* Import must be done using API approach.
* Records must be picked based on the batch id assigned by SLC\_ASSIGN\_BATCH\_ID\_P procedure.

Logic:

For all Valid records perform below thing.

Bulk collect valid records from staging table into local collection.

1. Verify if Franchisee1 already exists. Franchisee1 SSN is unique identifier for a Franchisee1.
2. If Franchisee1 does not exists then

Default

1. Vendor Type Lookup Code to FRANCHISEE
2. Term Name to IMMEDIATE
3. Pay Date Lookup Code to DISCOUNT
4. Pay Group Lookup Code to FRANCHISEE
5. Invoice Currency Code to USD
6. Payment Currency Code to USD
7. Enabled Flag as Y.
8. Call procedure create\_supplier to create Supplier.
9. If there is no error while creating Supplier call update\_org\_profile procedure to update Vendor Name and Legal Status.

Vendor Name must be in below format <First Name> <Middle Name> <Last Name>\_<Supplier Number sequence>

Note: Supplier Number generation must be made Automatic.

Thus while creating Supplier since supplier number must not be available to populate Vendor name thus program must call update\_org\_profile procedure passing vendor name to update Supplier Name.

1. If there is no error while creating updating Org profile then call process\_uda\_attributes to update UDA information.
2. Below UDA information must be updated.
3. SSN
4. Ethnicity
5. Date of Birth
6. Marital Status
7. First Tenure Date
8. Sex
9. Conversion Source
10. If there is no error while updating UDA information then repeat Step 1 to 5 for Franchisee2 with details for Franchisee2 and if Franchisee2 SSN is not NULL. In case of Franchisee2 if Franchisee2 is already existing then program must not perform anything as in case of Franchisee1 program must be calling slc\_update\_org\_profile\_p based on conditions mentioned above in Step 2.
11. If there is no error while creating information for Franchisee 2 and if Incorp Flag is Y then repeat Step 1 to 5 for Corporation.
12. If there is no error then if Incorp Flag is Y then call create\_supplier\_site for party\_id of corporation else call create\_supplier\_site for party\_id of Franchisee1.
13. If there is no error while creating Supplier Site then commit the transaction else rollback the transaction. Since this is a sequential operation if there is any error while importing data for any record program must rollback information created for that record.
14. If there is error while importing the records then log in common error logging table. At the same time mark the record status as E and update error message back in the staging table.
15. For updating NUM\_1099 and SSN for any Supplier follow rule mentioned in below table.

|  |  |  |
| --- | --- | --- |
| Case | Data | Data conversion rules |
| Case-1.  Incorp Flag = N  Franchisee1 details present  Franchisee 2 may or may not be present | SSN1 – Present for Franchisee1  SSN2- If Franchisee2 details is present then SSN2 is available | Since Incorp Flag is N , no Supplier must be created for Corporation.  Franchisee1 must be created with below details  NUM\_1099 must be equal to SSN1.  SSN1 must be updated in UDA for Franchisee2  If Franchisee2 details is available then Franchisee2 details must be created with below details  NUM\_1099 must be equal to SSN2.  SSN2 must be updated in UDA for Franchisee2 |
| Case -2  Incorp Flag = Y  Franchisee1 details present  Franchisee 2 may or may not be present | Federal Id – Present.  SSN1 – Present for Franchisee1  SSN2- If Franchisee2 details is present then SSN2 is available | Since Incorp Flag is Y, Supplier must be created for Corporation.  Supplier number be cerated with below details  NUM\_1099 must be equal to Federal ID  SSN must be null for UDA for the Franchisee  Franchisee1 must be created with below details  NUM\_1099 must be equal to SSN1  SSN1 must be updated in UDA for Franchisee1  If Franchisee2 details is available then Franchisee2 details must be created with below details  NUM\_1099 must be equal to SSN2.  SSN2 must be updated in UDA for Franchisee2 |

1. SLC\_PRINT\_SUMMARY\_P

This program must be called after validate\_p and import\_p to print summary of the records validated, records failed during validation, records failed during import, records successfully imported. This information must be printed in the OUT file of the concurrent program.

Summary format must be:

If program is run in Validate and Revalidate Mode:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Total Number of Records:

Total records validated:

Total records which failed during validation:

If program is run in Process Mode:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Total Number of Records:

Total records successfully imported:

Total records which failed during import:

For each run of concurrent program must insert one summary information in Common Summary table and records in Error table for error encountered.

Executable Definition:

| Executable | | Short Name | Application | Description | | | | Execution Method | Execution File Name | | | Execution File Path | | | Functionality |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |  | | |  |  |  |  |  | |
| *SLCISPFASSUPPLIERCNV* | *SLCISPFASSUPPLIERCNV* | | *SLCUST1* | *Executable file for FAS to Supplier Hub Conversion* | | | | *PL/SQL Stored Procedure* | *SLC\_ISP\_FASSUPP\_CNV\_PKG.slc\_main\_p* | | |  | | | *Executable for concurrent program for FAS to Supplier Hub conversion.* |

| Concurrent Program Short Name | | Sequence | Parameter | Description | | | | Value Set | | Default Value | | Display Size | | Concatenated Description Size | | | Prompt |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |  | | | |  | |  |  |  |  | |
| *SLCISPFASSUPPLIERCNV* | *10* | | *P\_PROCESSING\_MODE* | *This must decide in which mode conversion program must be run* | | | | *SLCISP\_CONVMODE* |  | | | *25* | | *25* | | | *Mode* |
| *SLCISPFASSUPPLIERCNV* | *20* | | *P\_BATCH\_SIZE* | *Batch Size* | | | | *NUMBER\_15* | *50000* | | | *15* | | *25* | | | *Batch Size* |
| *SLCISPFASSUPPLIERCNV* | *30* | | *P\_DEBUG\_FLAG* | *Debug Flag* | | | | *YES\_NO* | *NO* | | | *15* | | *25* | | | *Debug Flag* |

Value Set Definition:

| Value Set Name | | Description | | | List Type | | Security Type | | | Format Type | | | Maximum Size | | Validation Type | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | | | |  |  | |  | | |  | |  |  |
| *SLCISP\_CONVMODE* | *Value set for Conversion Modes* | | | *List of Values* | | *No Security* | | | | *Char* | | *30* | | | *Independent* | | |

Value Set Values:

| Value Set Name | | | Value |
| --- | --- | --- | --- |
|  |  | | |
| *SLCISP\_CONVMODE* | | *VALIDATE* | |
| *SLCISP\_CONVMODE* | | *PROCESS* | |
| *SLCISP\_CONVMODE* | | *REVALIDATE* | |

Lookup

| Type | | Meaning | | | Application | | | | Description | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | |  |  | | |  |
| *SLCISP\_BACKGROUND\_LKP* | *Lookup for Background* | | | *SLC Custom APPS* | | | | *Lookup for Background* | | |
| *SLCISP\_MARITAL\_LKP* | *Lookup for Marital Status* | | | *SLC Custom APPS* | | | | *Lookup for Marital Status* | | |
| *SLCISP\_SEX\_LKP* | *Lookup for Sex* | | | *SLC Custom APPS* | | | | *Lookup for Sex* | | |

Lookup Codes

| Type | | Code | | | Meaning | | | | Description | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | |  |  | | |  |
| *SLCISP\_BACKGROUND\_LKP* | *A* | | | *A* | | | | *American Indian* | | |
| *SLCISP\_BACKGROUND\_LKP* | *B* | | | *B* | | | | *African American (Black)* | | |
| *SLCISP\_BACKGROUND\_LKP* | *H* | | | *H* | | | | *Hispanic* | | |
| *SLCISP\_BACKGROUND\_LKP* | *O* | | | *O* | | | | *Asian Pacific islander* | | |
| *SLCISP\_BACKGROUND\_LKP* | *P* | | | *P* | | | | *Other* | | |
| *SLCISP\_BACKGROUND\_LKP* | *W* | | | *W* | | | | *White* | | |

| Type | | Code | | | Meaning | | | | Description | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | |  |  | | |  |
| *SLCISP\_MARITAL\_LKP* | *M* | | | *M* | | | | *Married* | | |
| *SLCISP\_MARITAL\_LKP* | *S* | | | *S* | | | | *Single* | | |
| *SLCISP\_MARITAL\_LKP* | *A* | | | *A* | | | | *Married* | | |

| Type | | Code | | | Meaning | | | | Description | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | |  |  | | |  |
| *SLCISP\_SEX\_LKP* | *F* | | | *F* | | | | *Female* | | |
| *SLCISP\_SEX\_LKP* | *M* | | | *M* | | | | *Male* | | |

Table Usage:

| Table Name | | Select | | Insert | | Update | | Delete |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | |  | |  | |  |
| *SLC\_ISP\_FAS\_SUPPIER\_CNV\_STG* | *X* | | *X* | | *X* | |  | |
| *EGO\_ATTR\_GROUPS\_V* | *X* | |  | |  | |  | |
| *HZ\_PARTIES* | *X* | |  | |  | |  | |
| *AP\_SUPPLIERS* | *X* | | *X* | |  | |  | |
| *AR\_LOOKUPS* | *X* | |  | |  | |  | |
| *EGO\_ATTR\_GROUPS\_V* | *X* | |  | |  | |  | |
| *EGO\_ATTRS\_V* | *X* | |  | |  | |  | |
| *POS\_SUPP\_PROF\_EXT\_B* | *X* | |  | |  | |  | |
| *AP\_SUPPLIER\_SITES\_ALL* | *X* | | *X* | |  | |  | |

## Conversion Program Modules

### ETL Objects and Connectivity

Not Applicable.

### Conversion Programs

Not Applicable.

| Program Type | Program Name | Description/Purpose | Program Location | Developer | Flat File: File Name and Location (if any) | |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | |  |
|  |  |  |  |  | |  |
|  |  |  |  |  | |  |
|  |  |  |  |  | |  |
|  |  |  |  |  | |  |
|  |  |  |  |  | |  |

### Automated Conversion Tool Files

| Sequence | <Conversion Tool>  Template Name | <Conversion Tool> Map File Name | Script  Name | Data File  Name | Location of Template &  Maps | Target Oracle  Table | Developer | Comments |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

## Conversion Execution Instructions

This section explains how to execute Franchisee Master Conversion.

1. Run Common data loader program passing data file path, data file name, control file name and control file path as parameter. This must load data into staging table.
2. Run the program in Validate mode passing No\_of\_records to validate newly created records.
3. Run the program in Process mode to import data into Supplier Hub.

To reload failed / errored records.

1. Run the program in Revalidate mode passing No\_of\_records to revalidate failed or errored records.
2. Run the program in Process mode to import data into Supplier Hub.

## Conversion Validation Strategy

Data spot checks of a sample of converted records will be completed by SEI business users to ensure that the format of the converted data meets their expectations

## Open and Closed Issues for this Deliverable

1. Add open issues that you identify while writing or reviewing this document to the open issues section. As you resolve issues, move them to the closed issues section and keep the issue ID the same. Include an explanation of the resolution.  
     
   When this deliverable is complete, any open issues should be transferred to the project- or process-level Risk and Issue Log (PJM.CR.040) and managed using a project level Risk and Issue Form (PJM.CR.040). In addition, the open items should remain in the open issues section of this deliverable, but flagged in the resolution column as being transferred.

### Open Issues

| ID | Issue | Resolution | Responsibility | Target Date | Impact Date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### Closed Issues

| ID | Issue | Resolution | Responsibility | Target Date | Impact Date |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 1 | Organization Type setup for the Suppliers. | For Individuals it must be Individual(NON-SERVICE)  For Corporation it must be  CORPORATION/OTHERS  For Corporation whose name is ending with LLP it must be  PARTNERSHIP/LLC/LLP/OTHERS | Divya |  |  |
| 2 | As of now Supplier Number setup is Manual. We want this setup to be changed to Automatic. | Suppler number must be changed to automatic before running the conversion program. | Mayur |  |  |

## Glossary

The available list of glossary terms is available at the below link:

## Appendix A (Optional)

## Appendix B (Optional)