Shaw Cloud ERP Program

Technical Specifications (Interface)

LOG-INT-18   
Oracle ERP Cloud to Logfire Interface: Point Of Sales

Document Control Information

Document Information

|  |  |
| --- | --- |
| Document Identification | **Shaw\_MD070-120\_LOG-INT-18** |
| Document Name | **Shaw Interface MD70 -120 LOG-INT-18 1.0 Point of Sales (Logfire to Cloud).docx** |
| Project Name | **Shaw Cloud ERP Program** |
| Client | **Shaw Communications** |
| Document Author | Neha Sharma |
| Document Version | 1.0 |
| Document Status | Draft |
| Date Released |  |

Document Edit History

| Version | Date | Additions/Modifications | Prepared/Revised by |
| --- | --- | --- | --- |
| 1.0 | 06-MAY-2020 | initial draft | Neha Sharma |
| 1.1 | 30-SEP-2020 | CHANGE REQUEST:CHG0290262  CHANGES ARE IN THE FOLLOWING SECTION:  2.3  2.4  2.5.1, 2.5.4, 2.5.5  2.6, 2.6.5,2.6.6 | Koushik Gudhi |
|  |  |  |  |

Document Review/Approval History

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution of Final Document

The following people are designated recipients of the final version of this document:

| Name | Organization/Title |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

Table of Contents

[1 Interface Overview 1](#_Toc39691754)

[1.1 Objective 1](#_Toc39691755)

[1.2 Scope 1](#_Toc39691756)

[1.3 Module List 1](#_Toc39691757)

[1.4 Assumptions 1](#_Toc39691758)

[1.5 Dependencies and Prerequisites 1](#_Toc39691759)

[1.6 Acronyms 1](#_Toc39691760)

[1.7 References 2](#_Toc39691761)

[2 Technical Specification 3](#_Toc39691762)

[2.1 Technical Overview 3](#_Toc39691763)

[2.2 Approach 3](#_Toc39691764)

[2.3 Object List 3](#_Toc39691765)

[2.4 Interface Architecture 4](#_Toc39691766)

[2.5 Data Object Definitions 6](#_Toc39691767)

[2.5.1 Data Source and Data File layout 6](#_Toc39691768)

[2.5.2 Tables and View Usage 7](#_Toc39691769)

[2.5.3 Data Mapping/Translations 7](#_Toc39691770)

[2.5.4 Custom Table/View Definition 8](#_Toc39691771)

[2.5.5 Procedures and Functions 9](#_Toc39691772)

[2.5.6 Other Objects 9](#_Toc39691773)

[2.6 Program Logic 10](#_Toc39691774)

[2.6.1 Input Parameters 10](#_Toc39691775)

[2.6.2 Data Sorting, Sequencing and Selection 10](#_Toc39691776)

[2.6.3 Starting Event(s) for the interface 10](#_Toc39691777)

[2.6.4 Validation and Error Handling 10](#_Toc39691778)

[2.6.5 Pseudo Logic 11](#_Toc39691779)

[2.6.6 SQL Statements 11](#_Toc39691780)

[3 Other Considerations 14](#_Toc39691781)

[3.1 Restart Strategy 14](#_Toc39691782)

[3.2 Crash Recovery 14](#_Toc39691783)

[4 Open and Closed Issues 15](#_Toc39691784)

[4.1 Open Issues 15](#_Toc39691785)

[4.2 Closed Issues 15](#_Toc39691786)

# 

# Interface Overview

## Objective

This document forms a basis for the detailed technical design for Point Of Sales.It captures all technical details to address the business requirements given in the functional specification “**LOG\_INT\_18 - Point of Sales (Cloud to Logfire)**” for developing **the “SHAW OM PointOfSales from Cloud to LogFire (1.0)”.** This document is intended to provide the developer with necessary information to implement effective and accurate “Point Of Sales (Cloud to Logfire)” design and build.

## Scope

The scope of this document is to describe the process, how Oracle ICS will process the data from

Oracle application to logfire system.

## Module List

Module involved/impacted: Oracle ERP cloud – Inventory Management

## Assumptions

* The interface will bring the correct Inventory Transaction details from Oracle Cloud that is staged into Logfire.

## Dependencies and Prerequisites

* The required application configuration / setup for both the source and target systems is complete prior to unit testing this integration.
* All data translations are identified and mapped.

## Acronyms

| Acronym | Meaning | Description |
| --- | --- | --- |
| API | Application programming interface | The interface that a computer system, library or application provides in order to allow requests for services to be made of it by other computer programs, and/or to allow data to be exchanged between them. |
| DBCS | Data base cloud service | Database hosted on cloud |
| SOAP | Simple object access protocol | A [protocol](https://en.wikipedia.org/wiki/Protocol_(computing)) specification for exchanging structured information in the implementation of [web services](https://en.wikipedia.org/wiki/Web_service) in [computer networks](https://en.wikipedia.org/wiki/Computer_network). |

## References

|  |  |  |  |
| --- | --- | --- | --- |
| Document Name | Revision | Document Description | Document Location URL |
| LOG-INT-18 1.0 Point of Sales (Logfire to Cloud).docx | 1.0 | Document includes all the functional requirements to implement the point of sales to logfire integration |

# Technical Specification

## Technical Overview

This document defines the technical component details required to implement the point of sales integration to logfire. This Technical Specifications Shaw Interface MD70 LOG-INT-18 Point of Sales document complements the Functional Specifications LOG\_INT\_18 – Point of Sales (Cloud to Logfire) document for Point Of Sales Integration from Oracle to Logfire and the set should be considered as the complete detailed design.

## Approach

The approach followed in order to build the interface for LOG-INT-18 Oracle to Logfire is:

* Integration in ICS is created and scheduled near real time.
* Integration will invoke DBCS procedure to get query based on the last run details.
* Query is based on the conditions specified in the mapping document.
* ICS integration invokes a BI report with parameters as request to fetch the data.
* Fetched encoded records are written in ICS stage file using opaque schema.
* File is read in chunk and data inserted into the staging table - SHAW\_OM\_POS\_LOGFIRE\_STG and an update is performed on SHAW\_INTERFACE\_RUN\_DETAILS DBCS to track processed data.
* Another integration batches and sub-batches the data based on creation date
* Post Validation and transformation, valid transaction records are fetched from the staging table and mapped to Logfire api end point for processing.
* WSDL for Logfire API endpoint is invoked with mapped data and status of the same is captured in staging table SHAW\_OM\_POS\_LOGFIRE\_STG
* User is notified in case of any error while processing

## Object List

| OBJECT NAME | DESCRIPTION | OBJECT TYPE | PURPOSE | NEW(Y/N) |
| --- | --- | --- | --- | --- |
| LOG-INT-18 SHAW OM PointOfSales From Cloud to DBCS (1.0) | ICS Integration | ICS Interface | This interface gets data from erp oracle cloud and inserts into DBCS staging table. | Y |
| LOG-INT-18 SHAW OM PointOfSales From DBCS to LGF (1.0) | ICS Integration | ICS Interface | This interface gets data from DBCS staging table and interfaces it to logfire. | Y |
| SHAW\_OM\_POS\_LGF\_PKG | Package | PL/SQL package | It inserts into staging table, updates errors. | Y |
| SHAW\_OM\_POS\_LOGFIRE\_STG\_S | Sequence | Sequence | Generates Batch ID Sequence | Y |
| SHAW\_OM\_POS\_LOGFIRE\_STG | Table | Table | It is a custom table and stores the staging data. | Y |
| SHAW\_OM\_POS\_INS\_REC | Type | DB Object Type | It is a custom object type with the columns of the staging table shaw\_om\_pos\_logfire\_stg. | Y |
| SHAW\_OM\_POS\_INS\_TAB | Type | DB Table Type | It is a custom table type for object shaw\_om\_pos\_ins\_rec and used as a parameter in ics. | Y |

## Interface Architecture

Below is the process description for Point Of Sales (Oracle to DBCS):



| Process Description |
| --- |
| 1. The ICS integration process is scheduled to every 60 mins. |
| 1. Fetch query based on the last run details of the integration. |
| 1. Invoke bi report to fetch data from cloud erp and decode it. |
| 1. Insert the records into the staging tables shaw\_om\_pos\_logfire\_stg. |
| 1. In case of any error send notification |

Below is the process description for Point Of Sales (DBCS to Logfire):



| Process Description |
| --- |
| 1. The ICS integration process is scheduled to run every 15 mins. |
| 1. Fetch Shaw Project Issue , Shaw Inv Adj Issue data from staging table shaw\_inv\_transactions\_stg |
| 1. Insert Shaw Project Issue, Shaw Inv Adj Issue data into staging table shaw\_om\_pos\_logfire\_stg |
| 1. Extract the POS records from staging tables |
| 1. Invoke the rest adapter to import data to logfire stage and interface into logfire. |
| 1. Update the staging tables with the errors received from the call back response. |
| 1. Send a notification to the team for any error or completion. |

## Data Object Definitions

NA

### Data Source and Data File layout

|  |  |  |  |
| --- | --- | --- | --- |
| **Logfire Column** | **Oracle table** | **Field Name** | **Comments** |
| FACILITY\_CODE | INV\_ORG\_PARAMETERS | ORGANIZATION\_CODE |  |
| COMPANY\_CODE | HARDCODED TO ‘SHAW’ |  | DVM Lookup |
| TRANSACTION\_TYPE |  |  | SALE/RETURN |
| TRANSACTION\_ID | INV\_MATERIAL\_TXNS | TRANSACTION\_ID | A generic seq will be used in case of Shaw Project Issue, Shaw Inv Adj Issue |
| SEQ\_NBR | BLANK |  | Row Number based on Transaction ID |
| REF\_TRANSACTION\_ID | BLANK |  | DVM LookUp for RETURN Transactions |
| REF\_SEQ\_NBR | BLANK |  | DVM LookUp for RETURN Transactions |
| LOCATION | INV\_MATERIAL\_TXNS | WKITTING001/S002/ATTRIBUTE1 | In case of Shaw Project Issue, shaw inv adj issue Attribute8 from shaw\_inv\_transaction\_stg |
| ITEM\_ALTERNATE\_CODE |  |  |  |
| ITEM\_PART\_A | EGP\_SYSTEM\_ITEMS\_B | ITEM\_NUMBER |  |
| ITEM\_PART\_B | BLANK |  |  |
| ITEM\_PART\_C | BLANK |  |  |
| ITEM\_PART\_D | BLANK |  |  |
| ITEM\_PART\_E | BLANK |  |  |
| ITEM\_PART\_F | BLANK |  |  |
| INVN\_ATTR\_A | BLANK |  |  |
| INVN\_ATTR\_B | BLANK |  |  |
| INVN\_ATTR\_C | BLANK |  |  |
| EXPIRY\_DATE | INV\_MATERIAL\_TXNS | AGING\_EXPIRATION\_DATE |  |
| BATCH\_NBR | BLANK |  |  |
| SERIAL\_NBR | INV\_UNIT\_TRANSACTIONS | SERIAL\_NUMBER |  |
| QUANTITY | INV\_MATERIAL\_TXNS | PRIMARY\_QUANTITY | QTY from shw\_inv\_trsnaction\_stg in case of Shaw Project Issue, shaw inv adj issue |
| POS\_USER | BLANK |  |  |
| INVN\_ATTR\_D | BLANK |  |  |
| INVN\_ATTR\_E | BLANK |  |  |
| INVN\_ATTR\_F | BLANK |  |  |
| INVN\_ATTR\_G | BLANK |  |  |

### Tables and View Usage

DBCS Tables

| Table Name | Select | Insert | Update | Delete |
| --- | --- | --- | --- | --- |
| SHAW\_OM\_POS\_LOGFIRE\_STG | X | X | X | X |
| shaw\_inv\_transactions\_stg | X |  |  |  |

DBCS Tables

| Table Name | Select | Insert | Update | Delete |
| --- | --- | --- | --- | --- |
| INV\_MATERIAL\_TXNS | X |  |  |  |
| INV\_UNIT\_TRANSACTIONS | x |  |  |  |
| INV\_ORG\_PARAMETERS | X |  |  |  |
| INV\_TRANSACTION\_TYPES\_VL | X |  |  |  |

### Data Mapping/Translations

ICS Lookup Mapping:

| Lookup Name | Attribute | Value | Description |
| --- | --- | --- | --- |
| Shaw\_logfire\_common\_lookup | REF\_TXN\_ID\_18 | POS001 | Reference Number being used by Integration Process. |
| Shaw\_logfire\_common\_lookup | REF\_SEQ\_NBR\_18 | 1 | Reference Sequence Number being used by Integration process. |
| SHAW\_INTERFACE\_NOTIFICATION\_MAILERS | All\_Inbounds\_lgf\_From | <Fromemail>@sjrb.ca | From Email Id for all notifications |
| SHAW\_INTERFACE\_NOTIFICATION\_MAILERS | LOG-INT-18\_To | <Toemail>@sjrb.ca | To email Ids for sending notifications |

### Custom Table/View Definition

SHAW\_OM\_POS\_LOGFIRE\_STG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Required** | **Data Type** | **Size** | **Description** |
| FACILITY\_CODE | NO | VARCHAR2 | 2000 | FACILITY\_CODE |
| COMPANY\_CODE | NO | VARCHAR2 | 2000 | COMPANY\_CODE |
| TRANSACTION\_TYPE | NO | VARCHAR2 | 200 | TRANSACTION\_TYPE |
| TRANSACTION\_ID | NO | VARCHAR2 | 200 | TRANSACTION\_ID |
| TRANSACTION\_NBR | NO | VARCHAR2 | 200 | NULL |
| VENDOR\_CODE | NO | VARCHAR2 | 2000 | NULL |
| ACTION\_CODE | NO | VARCHAR2 | 2000 | NULL |
| DELIVERY\_DATE | NO | DATE |  | NULL |
| SHIP\_DATE | NO | DATE |  | NULL |
| CANCEL\_DATE | NO | DATE |  | NULL |
| PRE\_PACK\_CODE | NO | VARCHAR2 | 2000 | NULL |
| PRE\_PACK\_RATIO | NO | VARCHAR2 | 2000 | NULL |
| PRE\_PACK\_TOTAL\_UNITS | NO | VARCHAR2 | 2000 | NULL |
| UNIT\_COST | NO | NUMBER | 30 | NULL |
| UNIT\_RETAIL | NO | NUMBER | 30 | NULL |
| SEQ\_NBR | NO | NUMBER | 30 | SEQ\_NBR |
| REF\_TRANSACTION\_ID | NO | VARCHAR2 | 2000 | REF\_TRANSACTION\_ID |
| REF\_SEQ\_NBR | NO | VARCHAR2 | 2000 | REF\_SEQ\_NBR |
| LOCATION | NO | VARCHAR2 | 2000 | LOCATION |
| ITEM\_ALTERNATE\_CODE | NO | VARCHAR2 | 2000 | ITEM\_ALTERNATE\_CODE |
| ITEM\_PART\_A | NO | VARCHAR2 | 2000 | ITEM\_PART\_A |
| ITEM\_PART\_B | NO | VARCHAR2 | 2000 | NULL |
| ITEM\_PART\_C | NO | VARCHAR2 | 2000 | NULL |
| ITEM\_PART\_D | NO | VARCHAR2 | 2000 | NULL |
| ITEM\_PART\_E | NO | VARCHAR2 | 2000 | NULL |
| ITEM\_PART\_F | NO | VARCHAR2 | 2000 | NULL |
| INVN\_ATTR\_A | NO | VARCHAR2 | 2000 | NULL |
| INVN\_ATTR\_B | NO | VARCHAR2 | 2000 | NULL |
| INVN\_ATTR\_C | NO | VARCHAR2 | 2000 | NULL |
| EXPIRY\_DATE | NO | DATE |  | EXPIRY\_DATE |
| BATCH\_NBR | NO | VARCHAR2 | 2000 | NULL |
| SERIAL\_NBR | NO | VARCHAR2 | 2000 | SERIAL\_NBR |
| SOURCE\_RECORD\_ID | NO | NUMBER | 30 | SOURCE\_RECORD\_ID |
| ERP\_TRANSACTION\_ID | NO | NUMBER | 30 | ERP\_TRANSACTION\_ID |
| QUANTITY | NO | NUMBER | 30 | QUANTITY |
| POS\_USER | NO | VARCHAR2 | 240 | NULL |
| INVN\_ATTR\_D | NO | VARCHAR2 | 2000 | NULL |
| INVN\_ATTR\_E | NO | VARCHAR2 | 2000 | NULL |
| INVN\_ATTR\_F | NO | VARCHAR2 | 2000 | NULL |
| INVN\_ATTR\_G | NO | VARCHAR2 | 2000 | NULL |
| STATUS | NO | VARCHAR2 | 50 | NEW/ERROR/SUCCESS |
| RECORD\_ID | NO | NUMBER | 30 | RECORD\_ID |
| BATCH\_ID | NO | NUMBER | 30 | BATCH\_ID |
| ERROR\_CODE | NO | VARCHAR2 | 1000 | ERROR\_CODE |
| ERROR\_MESSAGE | NO | VARCHAR2 | 4000 | ERROR\_MESSAGE |
| CREATED\_BY | NO | VARCHAR2 | 240 | CREATED\_BY |
| CREATION\_DATE | NO | TIMESTAMP |  | SYSDATE |
| LAST\_UPDATE\_DATE | NO | TIMESTAMP |  | SYSDATE |
| LAST\_UPDATED\_BY | NO | VARCHAR2 | 240 | SYSDATE |
| LAST\_UPDATE\_LOGIN | NO | VARCHAR2 | 240 | LAST\_UPDATE\_LOGIN |

**Primary Key :** TRANSACTION\_ID

**Foreign Key :** NA

**Constraints :** NA

**Index :** TRANSACTION\_TYPE, BATCH\_ID, TRANSACTION\_ID, SEQ\_NBR, TRANSACTION\_NBR

**Schema Name:** XXSHAW

### Procedures and Functions

|  |  |  |  |
| --- | --- | --- | --- |
| Package Name | SHAW\_OM\_POS\_LGF\_PKG | | |
| Owner | XXSHAW | | |
| Sub Unit/Object Type | Sub Unit/Object Name | Global/Private | Description |
| PROCEDURE | geTLASTRUNDATE | PRIVATE |  |
| PROCEDURE | INSERTPOSDATA | PRIVATE |  |
| PROCEDURE | INSERTPROJECTDATA | PRIVATE |  |
| PROCEDURE | updateBATCHID | PRIVATE |  |
| PROCEDURE | update\_TRNxs | PRIVATE |  |
| PROCEDURE | update\_Errors | PRIVATE |  |
| PROCEDURE | DeLETE\_RECORDS | PRIVATE |  |

Table 7 - Procedures and Functions

### Other Objects

|  |  |  |  |
| --- | --- | --- | --- |
| Sub Unit/Object Type | Sub Unit/Object Name | Global/Private | Description |
| DB Object Type | SHAW\_OM\_POS\_INS\_REC | PRIVATE |  |
| DB Table Type | SHAW\_OM\_POS\_INS\_TAB | PRIVATE |  |
| DB Object Type | SHAW\_OM\_PROJ\_INS\_REC | PRIVATE |  |
| DB Table Type | SHAW\_OM\_PROJ\_INS\_TAB | PRIVATE |  |

## Program Logic

The Oracle PointOfSales is an outbound integration. This interfaces are near real time interface. Once the 1st process runs, it get the Query based on the last run details and invokes a BI report. The report returns base64 encoded data which then saved in csv format with help of opaque schema. Records from csv file are inserted into the staging table SHAW\_OM\_POS\_LOGFIRE\_STG in chunks. An update on the last\_run\_date is performed in the table SHAW\_INTERFACE\_RUN\_DETAILS. The 2nd process fetches the Shaw Project Issue, Shaw Inv Adj Issue data from shaw\_inv\_transaction\_stg and inserts into SHAW\_OM\_POS\_LOGFIRE\_STG.Valid records for PointOfSales are fetched and mapped to Oracle WMS. These transactions are sent to Oracle WMS by invoking the WMS API. For all processed records staging table gets updated accordingly.

### Input Parameters

p\_last\_successful\_run: Stores the last successful run date

P\_CUR\_POS\_TYPE: Stores the response status of logfire

### Data Sorting, Sequencing and Selection

NA

### Starting Event(s) for the interface

This interface is scheduled.

### Validation and Error Handling

**Validations**

NA

**Error Handling**

1. When Cloud ERP System is down and the ERP Services are down

* ICS Integration process stops after sending an email notification to the specified email id.
* During the next scheduled integration process run, the Pending Point of Sales will also be picked up for processing.

1. When DBCS is down

* ICS Integration process stops after sending an email notification to the specified email id.
* During the next scheduled integration process run, the Pending Point of Sales will also be picked up for processing.

1. When the records don’t get imported successfully

* The error messages and error codes are updated in the staging table for each record.
* Any failed data has to be corrected again in cloud. For immediate reprocessing, fire an adhoc run by clicking on Submit Now option in ICS Console.Else wait for next scheduled run of the interface so that this file will be processed.

1. If invocation of LogFire REST- Processor service fails, an email notification is sent and Interface status is updated to "Error" in SHAW\_OM\_POS\_LOGFIRE\_STG.

### Pseudo Logic

The steps followed in Point of Sales Load from Oracle Cloud to LogFire are as follows:

1. Start the Orchestration integration.
2. Call a DB adapter to get batch id.
3. Get the last successful run date from the interface lookup table ‘shaw\_interface\_run\_details’
4. Call the Cloud service to get the report incremental data for Point Of Sales.
5. Update the current date in the interface lookup table ‘shaw\_interface\_run\_details’ and get the batch id which is a sequence and assign it to a variable.
6. Write the report data as opaque element in stage file.
7. Read the opaque element with the new schema for Point Of Sales.
8. Call a DB adapter to insert the Point of Sales details into staging table SHAW\_OM\_POS\_LOGFIRE\_STG.
9. Insert Shaw Project Issue, Shaw Inv Adj Issue data into SHAW\_OM\_POS\_LOGFIRE\_STG
10. Extract the POS data from the staging tables for the particular batch using the Database adapter.
11. Call LogFire API.
12. Call a DB adapter to update the response into the staging table.
13. Send a notification to the team about the errors and completion.
14. End the Intergration.

### SQL Statements

|  |  |
| --- | --- |
| SQL# | Query |
| SQL1 | select  FACILITY\_CODE,  COMPANY\_CODE,  TRANASCTION\_TYPE,  TRANSACTION\_ID,  '2'||ROW\_NUMBER() OVER(  PARTITION BY ITEM\_ALTERNATE\_CODE,SERIAL\_NUMBER  ORDER BY  creation\_date  ) seq\_nbr,  REF\_TRANSACTION\_ID,  REF\_SEQ\_NBR,  LOCATION,  ITEM\_ALTERNATE\_CODE,  EXPIRY\_DATE,  SERIAL\_NUMBER,  PRIMARY\_QUANTITY  from(  SELECT  org.organization\_code facility\_code,  'SHAW' company\_code,  CASE  WHEN typ.transaction\_type\_name IN (  'Shaw WIP Component Issue',  --'Shaw Inv Adj Issue', commented as part of 1.1  'Whse to Float'  ) THEN  'SALE'  ELSE  'RETURN'  END tranasction\_type,  inv.transaction\_id,  ROW\_NUMBER() OVER(  PARTITION BY inv.transaction\_id  ORDER BY  inv.transaction\_id  ) seq\_nbr,  '' ref\_transaction\_id,  '' ref\_seq\_nbr,  CASE  WHEN typ.transaction\_type\_name = 'Shaw WIP Component Issue' THEN  'WKITTING001'  WHEN typ.transaction\_type\_name = 'Float to Whse' THEN  'RT002'  /\* WHEN typ.transaction\_type\_name IN (  'Shaw Inv Adj Issue'  ) THEN  inv.attribute1 commented as part of 1.1 \*/  WHEN typ.transaction\_type\_name IN (  'Shaw Project Receipt'  ) THEN  'REC002'  WHEN typ.transaction\_type\_name ='Whse to Float' THEN  ''  WHEN typ.transaction\_type\_name = 'Transfer Order Interorganization Transfer' THEN  'S002'  END location,  (  SELECT  itm.item\_number  FROM  egp\_system\_items\_b itm  WHERE  itm.inventory\_item\_id = inv.inventory\_item\_id  AND itm.organization\_id = inv.organization\_id  ) ITEM\_ALTERNATE\_CODE,  inv.aging\_expiration\_date expiry\_date,  txn.serial\_number,  CASE  WHEN serial\_number IS NULL THEN  abs(inv.primary\_quantity)  ELSE  1  END primary\_quantity,  inv.creation\_date  FROM  inv\_material\_txns inv,  inv\_unit\_transactions txn,  inv\_org\_parameters org,  inv\_transaction\_types\_vl typ  WHERE  inv.transaction\_id = txn.transaction\_id (+)  AND inv.organization\_id = org.organization\_id  AND inv.transaction\_type\_id = typ.transaction\_type\_id  AND typ.transaction\_type\_name IN (  'Shaw WIP Component Issue',  --'Shaw Inv Adj Issue', commented as part of 1.1  'Whse to Float',  'Float to Whse',  'Transfer Order Interorganization Transfer',  'Shaw Project Receipt'  )  /\* AND ( ( typ.transaction\_type\_name IN (  'Shaw Inv Adj Issue'  )  AND inv.attribute1 IS NOT NULL )  OR ( typ.transaction\_type\_name NOT IN (  'Shaw Inv Adj Issue'  ) ) ) commented as part of 1.1\*/  AND ( ( typ.transaction\_type\_name = 'Transfer Order Interorganization Transfer'  AND org.attribute4 = 'MINOR' ) OR ( typ.transaction\_type\_name != 'Transfer Order Interorganization Transfer' ) )  AND org.organization\_code NOT IN (  'FLD',  'FLT',  'DC1'  )  AND (INV.CREATION\_DATE > TO\_TIMESTAMP(:P\_last\_successful\_run\_date, 'YYYY-MM-DD HH24:MI:SS.FF')  AND  INV.CREATION\_DATE <= TO\_TIMESTAMP(:P\_current\_run\_date, 'YYYY-MM-DD HH24:MI:SS.FF')  )) |

# Other Considerations

## Restart Strategy

1. When Cloud ERP System is down and the ERP Services are down

* ICS Integration process stops after sending an email notification to the specified email id.
* During the next scheduled integration process run, the Pending Point of Sales will also be picked up for processing.

1. When DBCS is down

* ICS Integration process stops after sending an email notification to the specified email id.
* During the next scheduled integration process run, the Pending Point of Sales will also be picked up for processing.

1. When the records don’t get imported successfully

* The error messages and error codes are updated in the staging table for each record.
* Any failed data has to be corrected again in cloud. For immediate reprocessing, fire an adhoc run by clicking on Submit Now option in ICS Console.Else wait for next scheduled run of the interface so that this file will be processed.

## Crash Recovery

Any failed records will be in error status in the staging table and a notification will be sent to the team. The records updated in oracle will be picked and updated with the new values in the next run.

# Open and Closed Issues

## Open Issues

| Issue ID | Description | Opened by | Responsible | Due Date |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Closed Issues

| Issue ID | Description | Resolution | Signoff | Closed Date |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |