Prepared by: Buzz IT Company Limited

Author: Steven Chen

Date: 20/01/2017

Version: 0.9

Maxim’s POS Polling Enterprise Service Bus Implementation Service

**EL-FY16-902**

**Sales/Master/Pricing Data Exchange**

System Design Specification

# Document Control

## Document History

| Version | Date | Author | Revision Remark |
| --- | --- | --- | --- |
| 0.9 | 20/01/2017 | Steven Chen | 1st draft |

## Document/Design Owner

| Name | Title |
| --- | --- |
| Steven Chen | System Analyst |
| Edward Leung | System Analyst |
| Tommy Leung | System Analyst |

## Key Comments

| Name/Title |  | |
| --- | --- | --- |
| # | Comments |
| Comment | 1 |  |
| Response | 1 |  |

# Table of Content

[Document Control 2](#_Toc471318764)

[Document History 2](#_Toc471318765)

[Document/Design Owner 2](#_Toc471318766)

[Key Comments 2](#_Toc471318767)

[Table of Content 3](#_Toc471318768)

[1 Background 5](#_Toc471318769)

[1.1 Document Purpose 5](#_Toc471318770)

[1.2 Document Scope 5](#_Toc471318771)

[1.3 Document Audience 5](#_Toc471318772)

[1.4 Terms & Abbreviations 6](#_Toc471318773)

[1.5 Reference Materials 6](#_Toc471318774)

[2 Executive Summary 7](#_Toc471318775)

[3 Architecture Design 8](#_Toc471318776)

[3.1 Overall Architecture 8](#_Toc471318777)

[4 Data Model Design 8](#_Toc471318778)

[4.1 POS Client Data Source Data Model 8](#_Toc471318779)

[4.2 EDW Data Model 8](#_Toc471318780)

[4.3 Pricing/Master Data Model 8](#_Toc471318781)

[4.4 Staging Table Data Model 8](#_Toc471318782)

[4.5 Polling Application Work Tables 9](#_Toc471318783)

[5 High Level Interface Design 10](#_Toc471318784)

[6 Function Design 11](#_Toc471318785)

[7 Report Design 12](#_Toc471318786)

[8 Interface Specification 13](#_Toc471318787)

[9 Interface Specification 13](#_Toc471318788)

# Background

## Document Purpose

The purpose of the System Design Specification (SDS) is to describe the detailed system design specification for a project and its main aim is to provide system design context for the project and its objectives. It will provide the input for high-level development activities.

The System Design Specification is part of the deliverables in the Business Case Development phase of Project Delivery Lifecycle.

## Document Scope

The scope of the System Design Specification (SDS) is to describe the architectural view of the system. It has section such as architecture design, data model design, high-level interface design, report design, etc. The technical designs and specifications of the impacted applications are not included in this document.

## Document Audience

The audience of this System Design Specification (SDS) is the technical staff of the IT department of the project owner.

## Terms & Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| ESB | Enterprise Service Bus |
| API | Application Programming Interface |
| EDW | Enterprise Data Warehouse |
| EOD | End of Day |
| POS Client | One Database Owner on the Sales side, provided by the POS machine vendor/manufacturer |
| DB | Database |
| POS | Point of Sales |
| Staging DB | The service bus database to stage the polling data |

## Reference Materials

| Document Names |
| --- |
| Maxim’s POS Polling ESB Implementation Service Proposal EL-FY16-902-v3.docx |
| POS Polling User Requirement Confirmation-20161121-Discussion Note.xlsx |

# Executive Summary

The aim of the ESB project is to migrate three data processing flows in Maxim’s current enterprise architecture using database provided technologies (linked server & stored procedure) to a new platform using Oracle Enterprise Services Bus technology. In the new ESB polling system (ESB system), three data processing flows will be implemented:

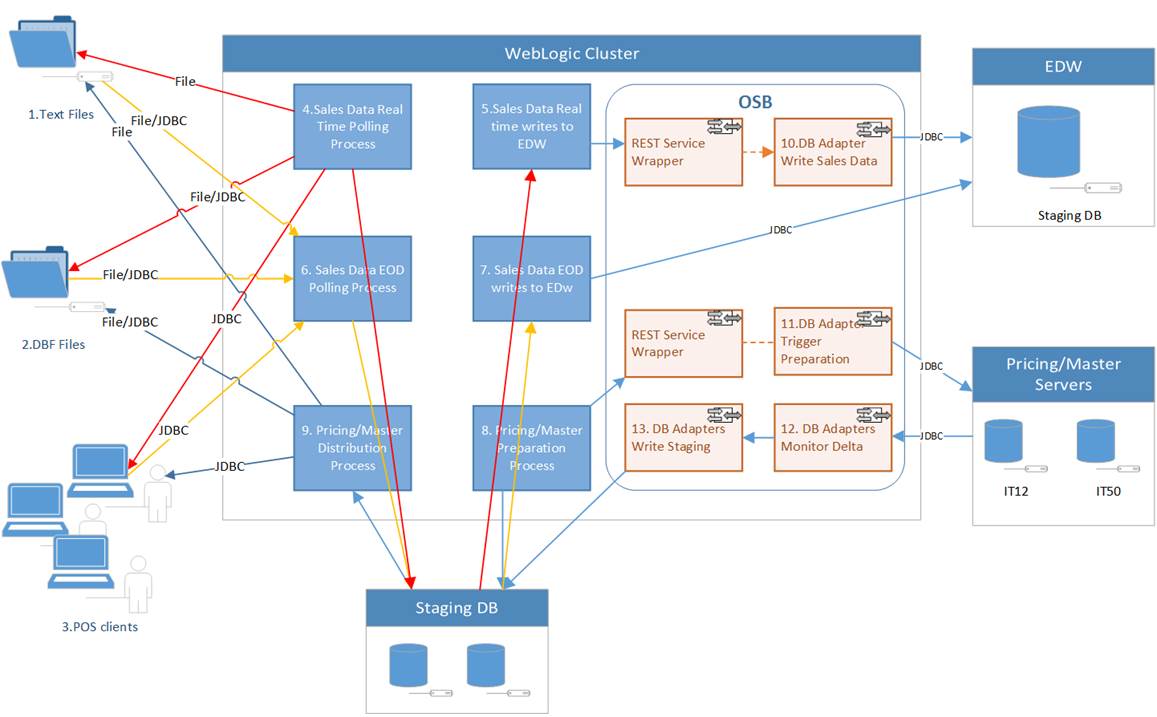
* Sales data real time processing to EDW
* Sales data EOD processing to EDW
* Synchronize master data to POS clients (e.g. Pricing/Master)

The ESB system will use JDBC to connect to the databases of existing POS clients in outlets by pre-configured connection information in order to collect sales data from the existing POS systems and update the pricing/master data back to the existing POS clients. Referring to the polling logic found in the production IT51 server, for any connection error, the ESB system will log down the error and retry data synchronization. After reaching maximum retry count, the ESB system will halt the synchronization for that particular POS client and generate alert to related parties for follow up.

All POS clients’ connection settings are configurable and maintainable by Maxim’s IT. It makes the adding of new POS client easy and no alteration of programming code is required. The ESB system can invoke several configurable concurrent threads to poll/push the data to/from POS clients concurrently for maximizing system performance.

# Architecture Design

## Overall Architecture



Key Data Sources

1. Text File
2. DBF files
3. POS Clients SQL Server

Key Process

1. Sale Data Real Time Polling (POS -> Staging)
2. Sales Data Real Time to EDW (Staging -> EDW, depend on #4 completion)
3. Sales Data EOD (POS -> Staging)
4. Sales Data EOD to EDW (Staging –> EDW, depend on #6 completion)
5. Pricing/Master Generation/Preparation (Pricing/Master servers -> Staging)
6. Pricing/Master Distribution (Staging –> POS clients, depending on #8 completion)

Key OSB components

1. Write EDW tables service (Virtual Branch Sales Inbound tables)
2. Pricing generation service (trigger SP udsp\_gen\_pricing\_group)
3. DB Adapter monitoring pricing/master tables delta
4. DB Adapter write Pricing/Master data to staging (triggered by #12)

# Data Model Design

## POS Client Data Source Data Model

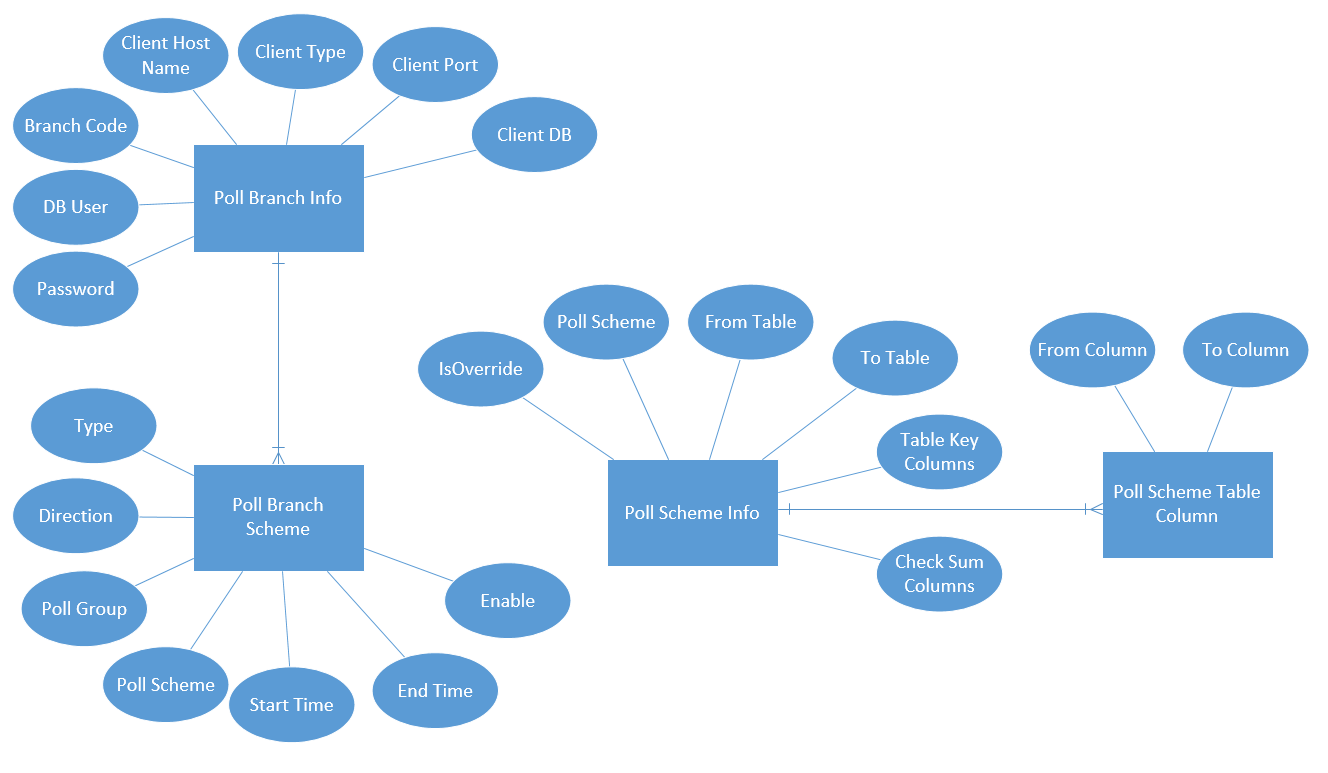
## EDW Data Model

## Pricing/Master Data Model

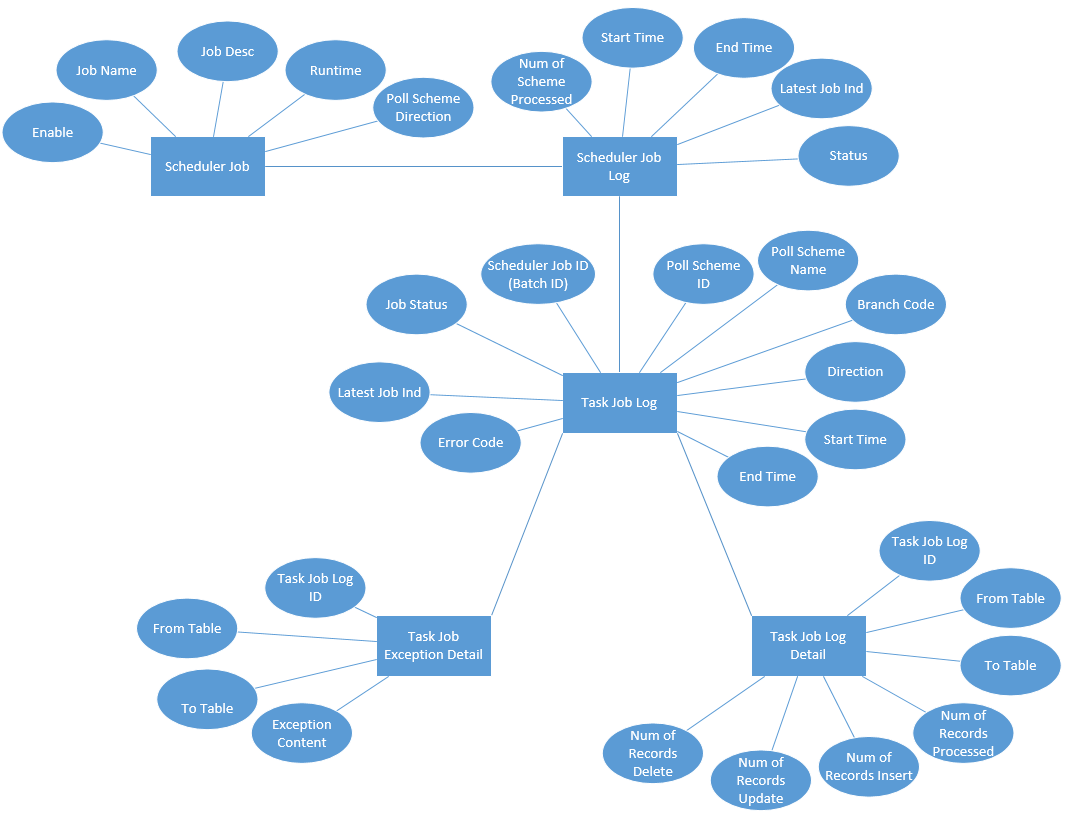
## Staging Table Data Model

## Polling Application Work Tables

Polling Schemes



Job Control (Job Logs, Exceptional Data)



Reporting Data (Optional)

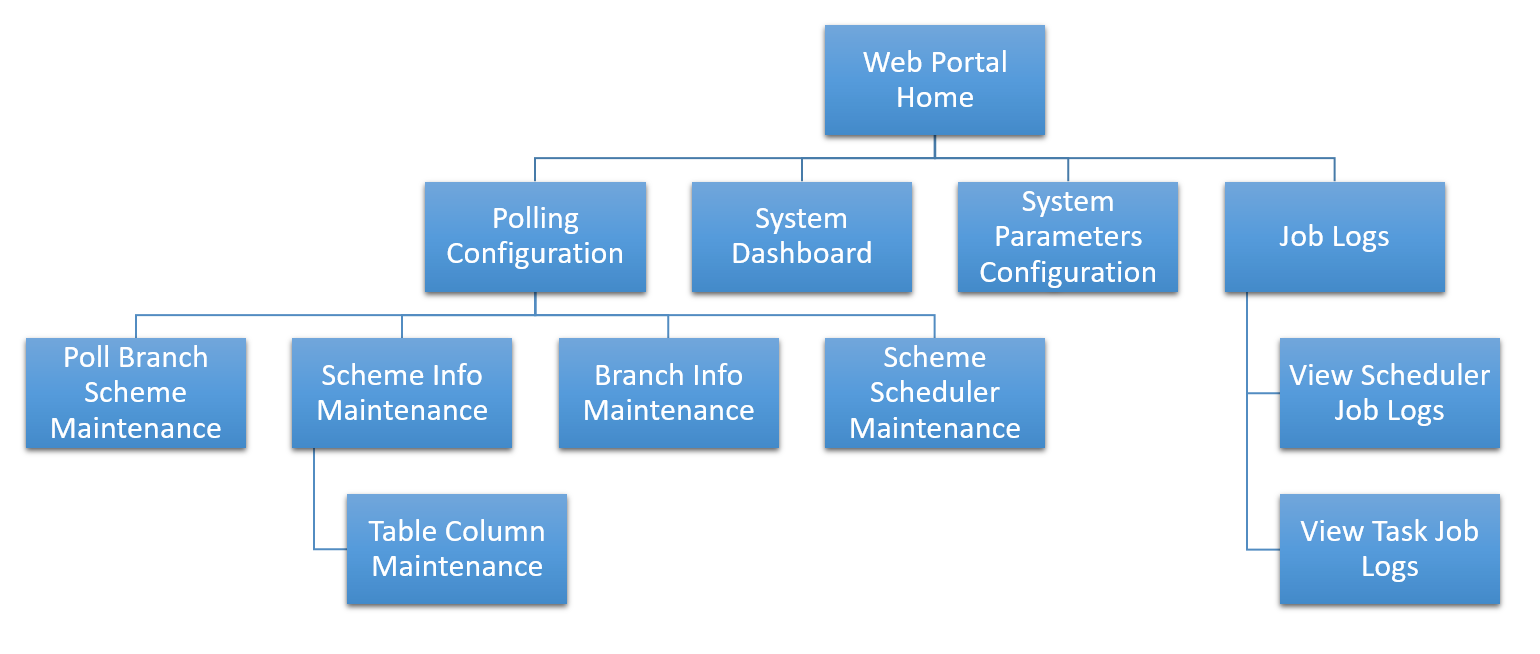
# High Level Interface Design

# Functional Design

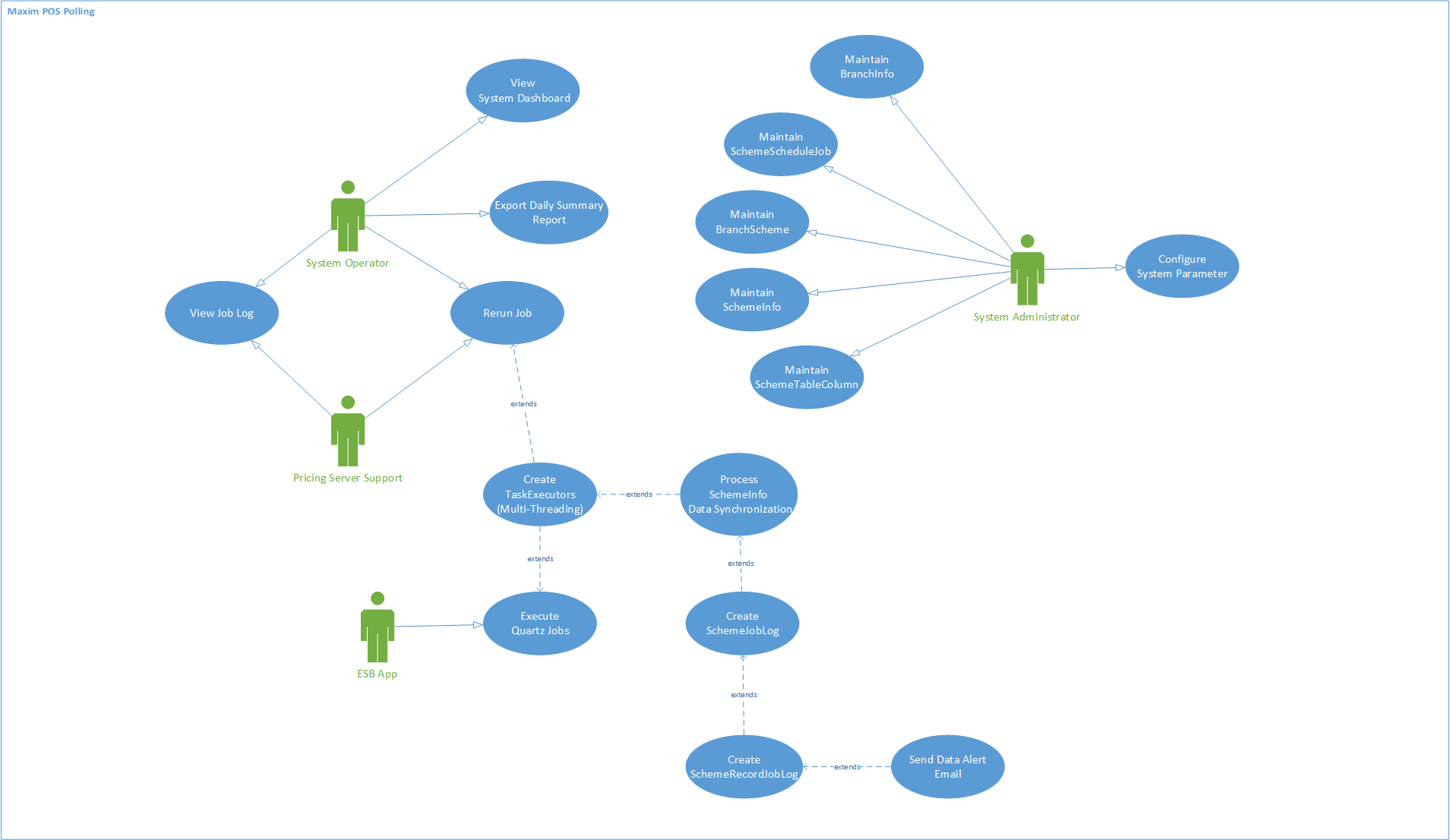
## System Functional Overview

### Application Context

### Site Map



### Use Case



### Interfaces

## Branch Info Maintenance

### Description

The “Branch Info Maintenance” function allows user’s manipulation on POS client data source definition data. The data should be initially input on the basis of current polling server. The view of these data are available for the access from both Service Bus User and POS User, but Service Bus user are allowed to use operations including Create, Edit and Delete. On the Add or Modification of the branch info data, the data source connectivity will be validated upon saving.

Use Case functions

# Maintain Branch Info (List)

# Maintain Branch Info (Create/Edit/Delete)

### Input

N/A

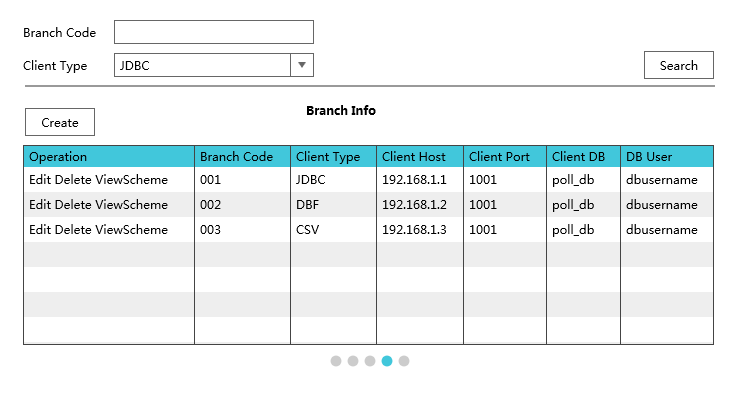
### Output

N/A

### Maintain Branch Info (List)

#### Process/Work Flow

#### Screen



#### Data Fields & Presentation Logic

N/A

#### Screen Objects & Action

N/A

#### User/Security Group

N/A

#### Assumptions/Constraints

N/A

#### Error & Exception List

N/A

### Maintain Branch Info (Create/Edit/Delete)

#### Process/Work Flow

#### Screen

#### Data Fields & Presentation Logic

N/A

#### Screen Objects & Action

N/A

#### User/Security Group

N/A

#### Data Fields & Presentation Logic

N/A

#### Assumptions/Constraints

N/A

#### Error & Exception List

N/A

## Scheduler Jobs Maintenance

### Description

### Input

### Output

## Branch Scheme Maintenance

### Description

The “Branch Scheme Maintenance” function allows user’s manipulation on POS client data processing logic data. The data is input/configured by service bus administrator according to the poll brank scheme data in current polling server. The view of these data are available for the access from both Service Bus, but Service, bus Admin user are allowed to use operations including Create, Edit and Delete.

Use Case functions

# Maintain Branch Scheme (List)

# Maintain Branch Scheme (Create/Edit/Delete)

# Maintain Scheme Info (List)

# Maintain Scheme Info (Create/Edit/Delete)

# Maintain Table Column (List)

# Maintain Table Column (Create/Edit/Delete)

### Input

N/A

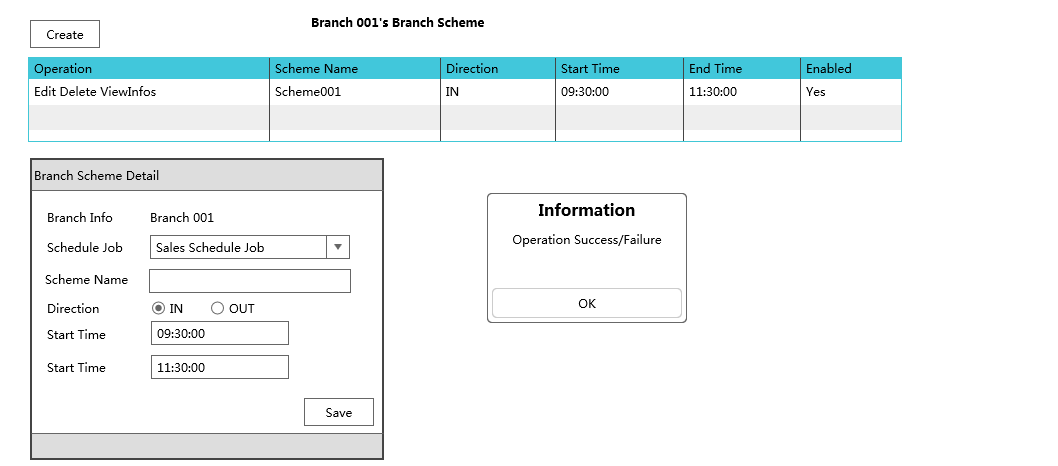
### Output

N/A

### Maintain Branch Scheme (List)

#### Process/Work Flow

#### Screen



#### Data Fields & Presentation Logic

N/A

#### Screen Objects & Action

N/A

#### User/Security Group

N/A

#### Data Fields & Presentation Logic

N/A

#### Assumptions/Constraints

N/A

#### Error & Exception List

N/A

### Maintain Branch Scheme (Create/Edit/Delete)

#### Process/Work Flow

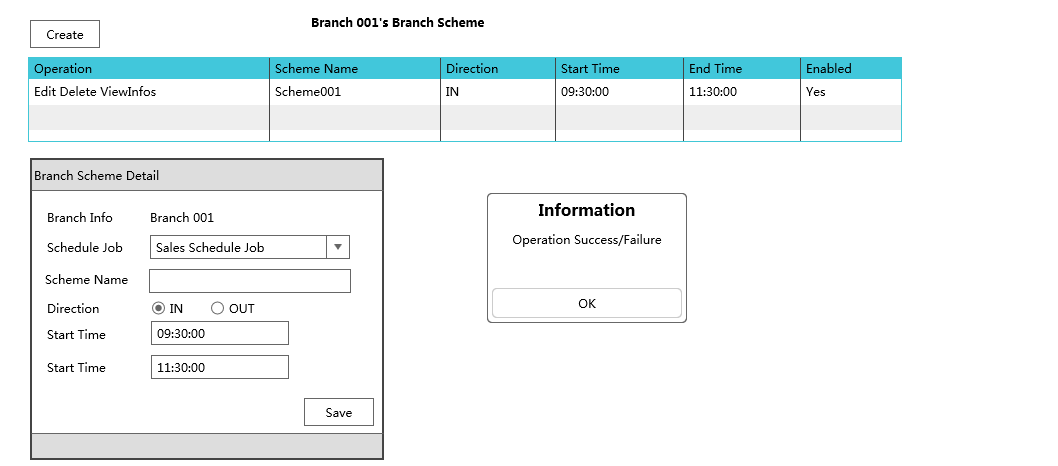
1. Create Event Home > Polling Configuration > Branch Scheme > Create

2. Edit Event Home > Polling Configuration > Branch Scheme > Search Record > Edit

3. Delete Event Home > Polling Configuration > Branch Scheme > Search Record > Delete

4. Quit: Click **Cancel** Button

#### Screen



#### Data Fields & Presentation Logic

N/A

#### Screen Objects & Action

N/A

#### User/Security Group

N/A

#### Data Fields & Presentation Logic

N/A

#### Assumptions/Constraints

N/A

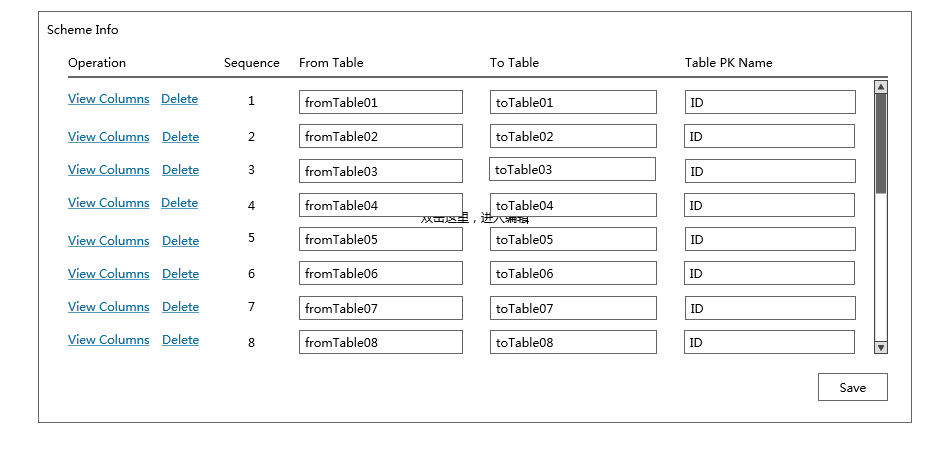
#### Error & Exception List

N/A

### Maintain Scheme Info (List)

#### Process/Work Flow

#### Screen



#### Data Fields & Presentation Logic

N/A

#### Screen Objects & Action

N/A

#### User/Security Group

N/A

#### Data Fields & Presentation Logic

N/A

#### Assumptions/Constraints

N/A

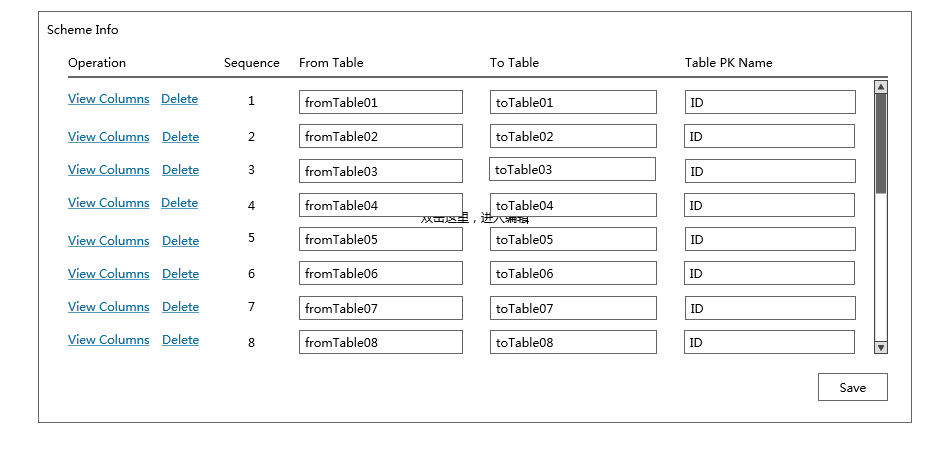
#### Error & Exception List

N/A

### Maintain Scheme Info (Create/Edit/Delete)

#### Process/Work Flow

#### Screen



#### Data Fields & Presentation Logic

N/A

#### Screen Objects & Action

N/A

#### User/Security Group

N/A

#### Data Fields & Presentation Logic

N/A

#### Assumptions/Constraints

N/A

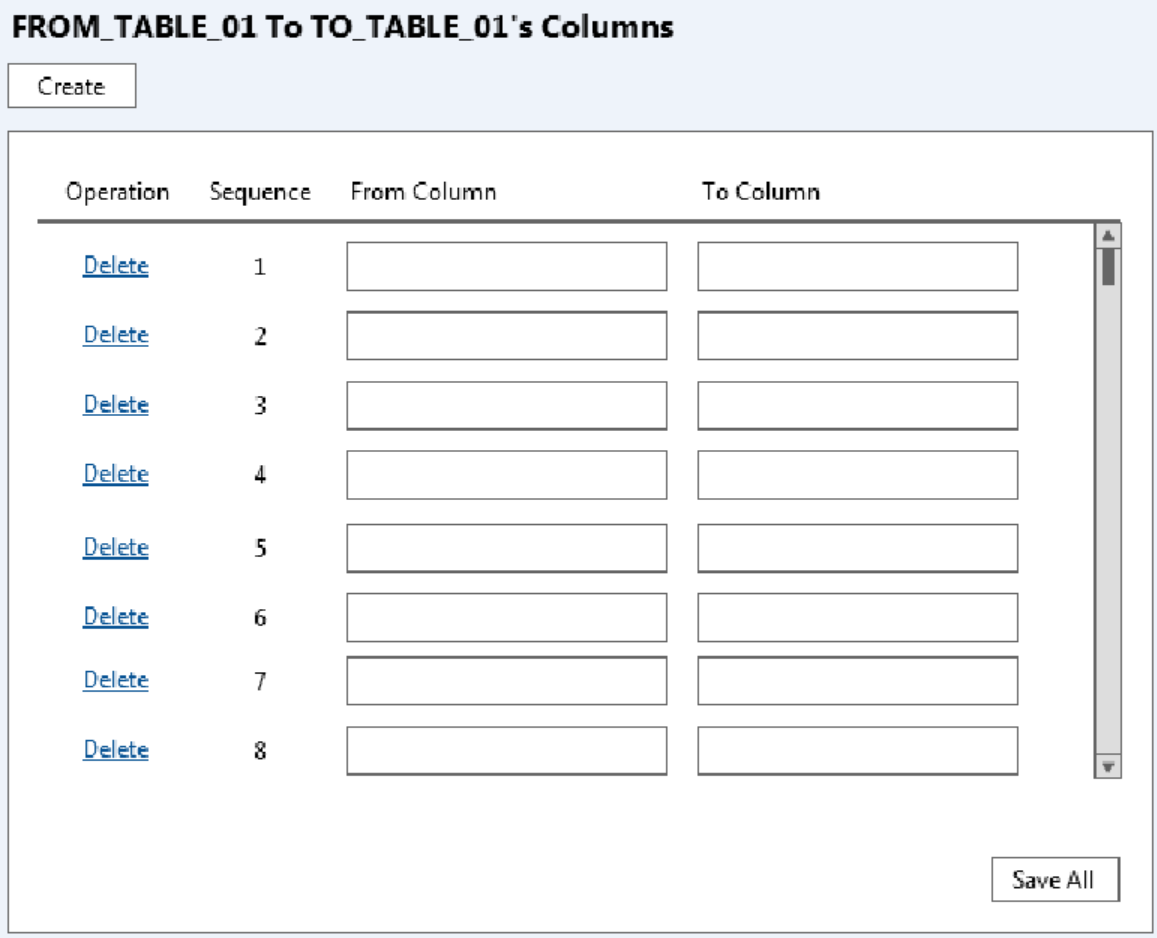
#### Error & Exception List

N/A

### Maintain Table Column Info (List)

#### Process/Work Flow

#### Screen



#### Data Fields & Presentation Logic

N/A

#### Screen Objects & Action

N/A

#### User/Security Group

N/A

#### Data Fields & Presentation Logic

N/A

#### Assumptions/Constraints

N/A

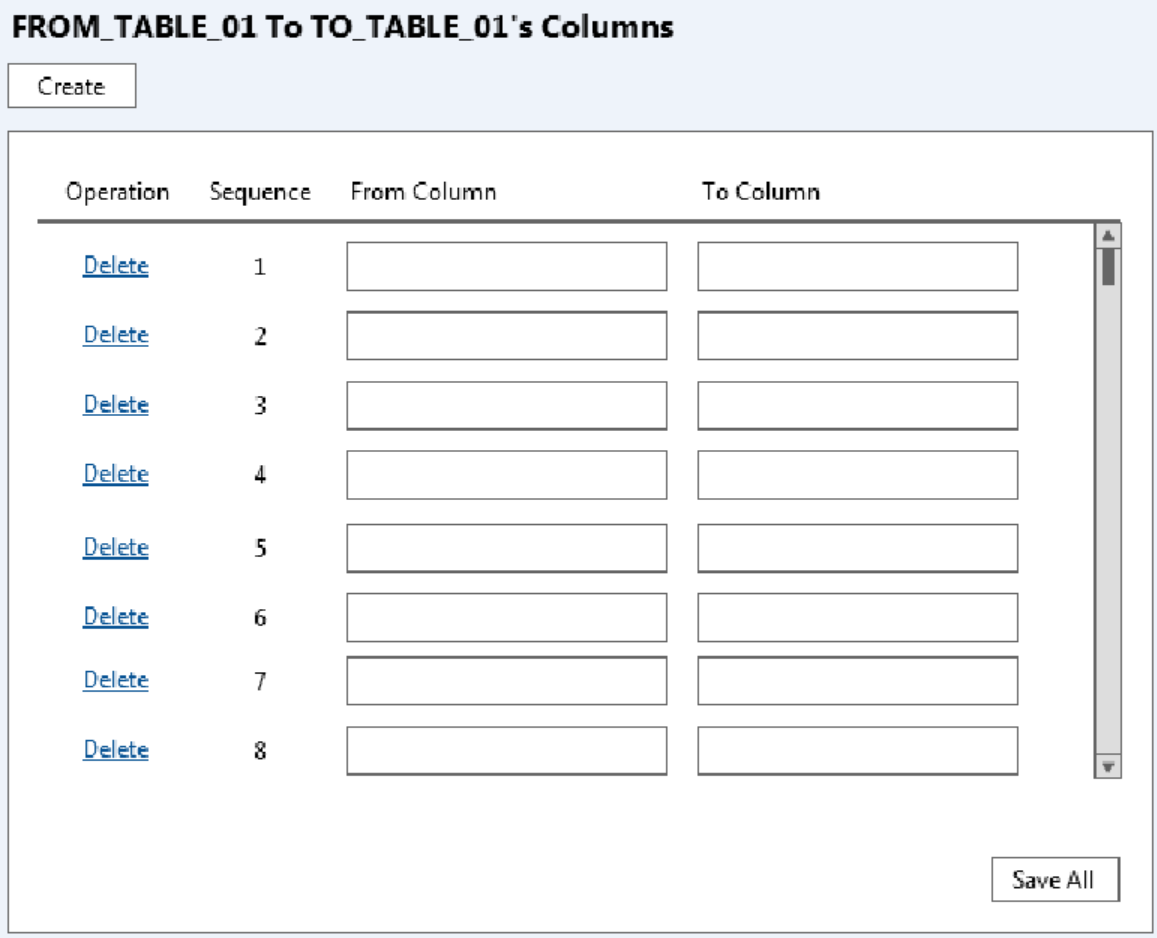
#### Error & Exception List

N/A

### Maintain Table Column Info (Create/Edit/Delete)

#### Process/Work Flow

#### Screen



#### Data Fields & Presentation Logic

N/A

#### Screen Objects & Action

N/A

#### User/Security Group

N/A

#### Data Fields & Presentation Logic

N/A

#### Assumptions/Constraints

N/A

#### Error & Exception List

N/A

## Job Logs View

## Sales Data Real Time Polling Batch Job

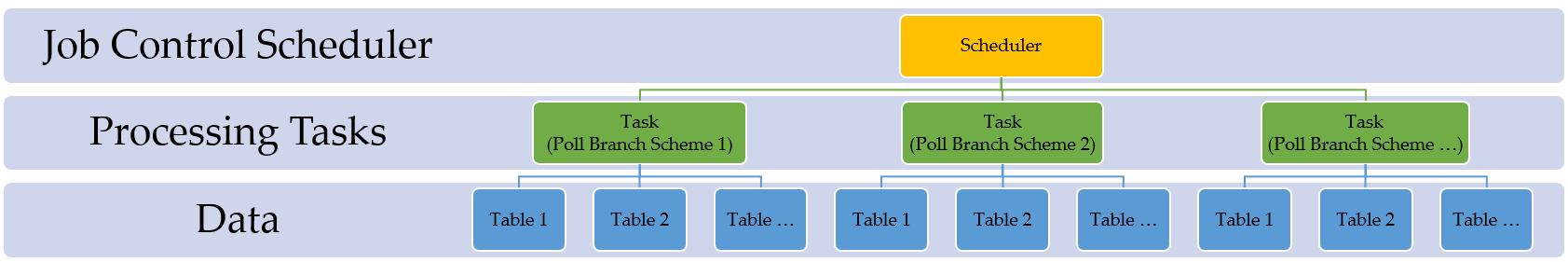
### Description

The scheduled batch job runs in the configurable time interval (e.g. 15 mins) to download POS client sales data from all active POS clients into the staging database. This is the pre-requisite batch job of the other batch which push real time sales data from staging database into EDW.

The job is triggered by a scheduler and it works as a task controller, submitting separated standalone tasks. Each task handles one available branch scheme. The job will handle the data in the pre-defined tables under the scheme according to the scheme configuration (e.g. direction, override logic, and validation).

In below session there will be the details descriptions about

1. Sales Data Real Time (POS - Staging) Scheduler
2. Sales Data Real Time (POS - Staging) Task



### Input

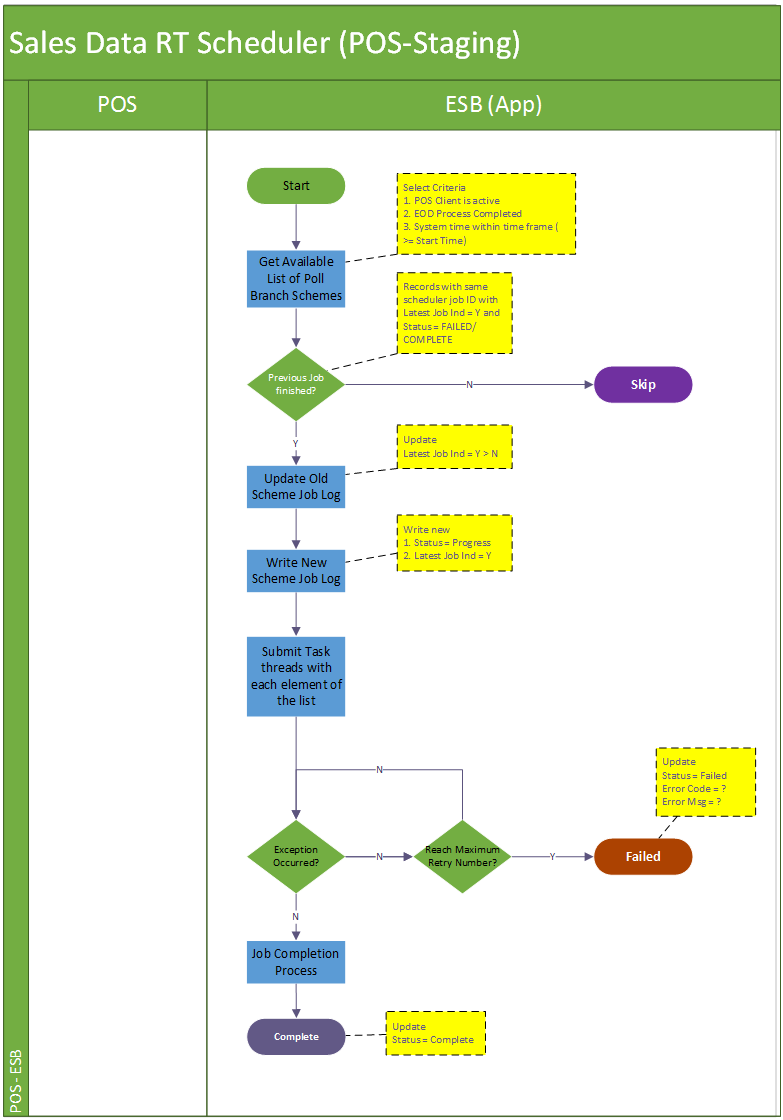
POS Client Sales Data (Real time)

### Output

Staging Sales Data (Real time)

### Sales Data Real Time (POS - Staging) Scheduler

#### Process/Work Flow



**Job Details**

Step1. This batch job will be triggered every 15 mins (configured). It would iterate all available data accumulated in poll branch scheme table (with condition POS client is enabled, the POS client has not yet conducted EOD process and the current time is greater or equal to the start time)

Step 2. Update scheduler job log. Firstly update the previous job log’s latest job indicator from ‘Y’ to ‘N’. Onwards newly create scheduler job log is having the status “PROGRESS”. The latest batch indicator for the new record will be marked as “Y”, and that of the current batch will be marked as “P”.

Step 3. Submit processing tasks for all eligible poll branch schemes, which are selected in step 1.

Step 4. Upon any errors in the task submission and cannot be resolved (e.g. by retry). The job will abort and keep the job status to reflect the problematic stage “FAILED”.

Step 5. If all tasks of the poll branch schemes are successfully submitted, the scheduler is regarded as complete and update the job status to “COMPLETE” with a record number of schemes processed

|  |  |  |
| --- | --- | --- |
| **Job Name** | **Status** | **Description** |
| Sales Real Time Polling Scheduler | “PROGRESS”  “FAILED”  “COMPLETE” | “PROGRESS” when scheduler is invoked and started  “FAILED” when encounter errors reading poll scheme data from database or errors submitting tasks.  “COMPLETE” when scheduler job iterates all available poll schemes and submit related tasks of them and all these tasks are either completed or failed. |

**Logging**

Log directory: /repos/esb/polling/log/

Upon each poll branch scheme task submission: the scheme info will be written into follow format:

[INFO] [Timestamp] [Scheme name]-[branch code]-[scheme type]-[direction]

**Entry & Exit Criteria**

The processing job starts once the schedule job invokes.

In the process of the scheduler job, it will be regarded as job failure and send alert email to IT support according to the error’s severity

Error occurs when error reading poll branch scheme data from staging database.

Error occurs when submitting poll branch scheme tasks into subsystem.

#### Screen

N/A

#### Data Fields & Presentation Logic

N/A

#### Screen Objects & Action

N/A

#### User/Security Group

N/A

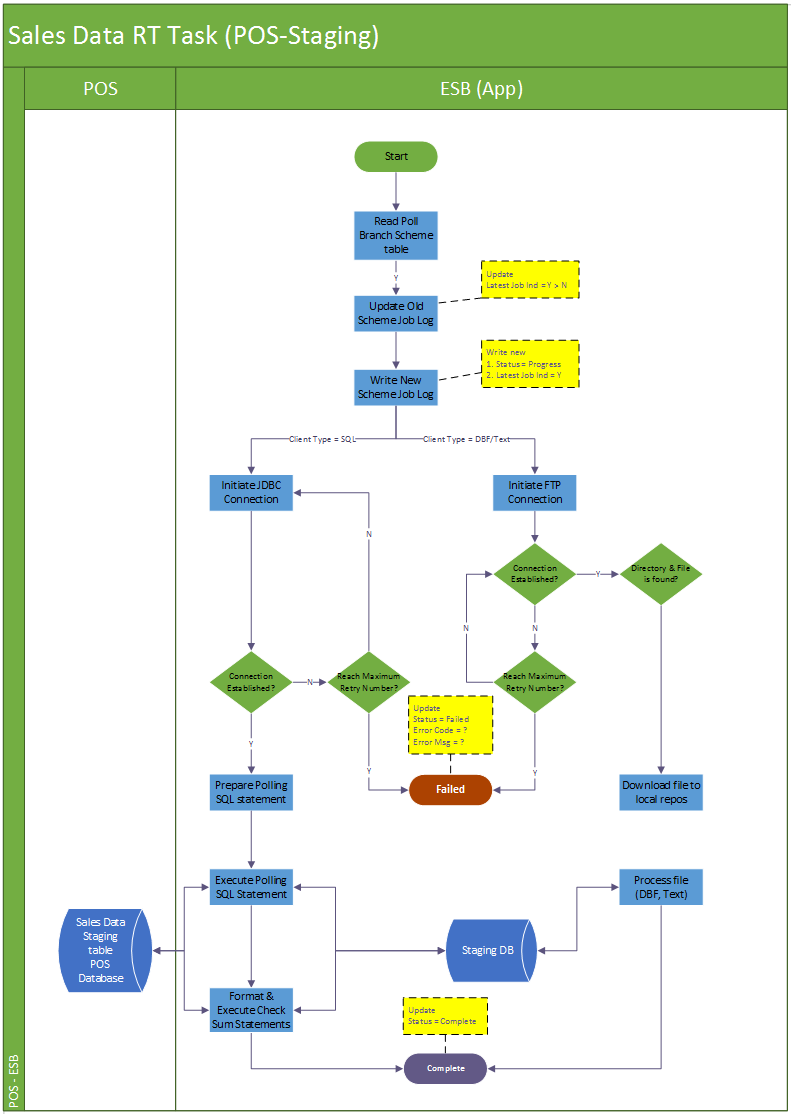
#### Assumptions/Constraints

#### Error & Exception List

|  |  |  |
| --- | --- | --- |
| **Error Code** | **Severity** | **Error Message** |
|  |  | Failed to establish FTP connection |
|  |  | Target directory or file not found |
|  |  | Failed to download target file |
|  |  | Failed to process records into staging database |

### Sales Data Real Time (POS - Staging) Task

#### Process/Work Flow



**Job Details**

**Logging**

**Entry & Exit Criteria**

#### Screen

N/A

##### Data Fields & Presentation Logic

N/A

##### Screen Objects & Action

N/A

##### User/Security Group

N/A

##### Data Fields & Presentation Logic

N/A

##### Assumptions/Constraints

##### Error & Exception List

|  |  |  |
| --- | --- | --- |
| **Error Code** | **Severity** | **Error Message** |
|  |  | Failed to establish FTP connection |
|  |  | Target directory or file not found |
|  |  | Failed to download target file |
|  |  | Failed to process records into staging database |

## Sales Data EOD Polling Batch Job (POS-Staging)

### Description

The scheduled batch job runs in the configurable time interval (e.g. 5 mins) to monitor EOD indication data table (e.g. HIST\_POSSYSTEM). This is the pre-requisite batch job of the other batch which push real time sales data from staging database into EDW.

The job is triggered by a scheduler and it works as a task controller, submitting separated standalone tasks. Each task handles one available branch scheme. The job will handle the data in the pre-defined tables under the scheme according to the scheme configuration (e.g. direction, override logic, and validation).

In below session there will be the details descriptions about

1. Sales Data Real Time (POS - Staging) Scheduler
2. Sales Data Real Time (POS - Staging) Task

### Input

EOD Sales Data in POS client database

### Output

EOD Sales Data in staging database

### Sales Data EOD (POS - Staging) Scheduler

#### Process/Work Flow

**Job Details**

**Logging**

**Entry & Exit Criteria**

#### Screen

N/A

##### Data Fields & Presentation Logic

N/A

##### Screen Objects & Action

N/A

##### User/Security Group

N/A

##### Data Fields & Presentation Logic

N/A

##### Assumptions/Constraints

##### Error & Exception List

### Sales Data EOD (POS - Staging) Task

#### Process/Work Flow

**Job Details**

**Logging**

**Entry & Exit Criteria**

#### Screen

N/A

##### Data Fields & Presentation Logic

##### Screen Objects & Action

##### User/Security Group

##### Data Fields & Presentation Logic

##### Assumptions/Constraints

##### Error & Exception List

|  |  |  |
| --- | --- | --- |
| **Error Code** | **Severity** | **Error Message** |
|  |  | Failed to establish FTP connection |
|  |  | Target directory or file not found |
|  |  | Failed to download target file |
|  |  | Failed to process records into staging database |

## Pricing/Master Data Generation Batch Job (Pricing Server - Staging)

### Description

### Input

Pricing/master data in pricing server

### Output

Pricing/master data in staging database

### Pricing/Master Data (EDW - Staging) Scheduler

#### Process/Work Flow

Flow Summary

Job Status

Logging

Entry & Exit Criteria

#### Screen

N/A

##### Data Fields & Presentation Logic

N/A

##### Screen Objects & Action

N/A

##### User/Security Group

N/A

##### Data Fields & Presentation Logic

N/A

##### Assumptions/Constraints

##### Error & Exception List

|  |  |  |
| --- | --- | --- |
| **Error Code** | **Severity** | **Error Message** |
|  |  | Failed to establish FTP connection |
|  |  | Target directory or file not found |
|  |  | Failed to download target file |
|  |  | Failed to process records into staging database |

### Pricing/Master Data (Staging-POS) Scheduler

#### Process/Work Flow

Flow Summary

Job Status

Logging

Entry & Exit Criteria

#### Screen

N/A

##### Data Fields & Presentation Logic

##### Screen Objects & Action

##### User/Security Group

##### Data Fields & Presentation Logic

##### Assumptions/Constraints

##### Error & Exception List

## Pricing/Master Data Distribution Batch Job (Staging-POS)

### Description

### Input

POS Client Sales Data

### Output

### Pricing/Master Data Distribution Scheduler (Staging-POS)

#### Process/Work Flow

Flow Summary

Job Status

Logging

Entry & Exit Criteria

#### Screen

N/A

##### Data Fields & Presentation Logic

##### Screen Objects & Action

##### User/Security Group

##### Data Fields & Presentation Logic

##### Assumptions/Constraints

##### Error & Exception List

### Pricing/Master Data Distribution Task (Staging-POS)

#### Process/Work Flow

Flow Summary

Job Status

Logging

Entry & Exit Criteria

#### Screen

N/A

##### Data Fields & Presentation Logic

##### Screen Objects & Action

##### User/Security Group

##### Data Fields & Presentation Logic

##### Assumptions/Constraints

##### Error & Exception List

|  |  |  |
| --- | --- | --- |
| **Error Code** | **Severity** | **Error Message** |
|  |  | Failed to establish FTP connection |
|  |  | Target directory or file not found |
|  |  | Failed to download target file |
|  |  | Failed to process records into staging database |

# Non-Functional Design

# Interface Design

- End -