DS.140 Integration Design Specification



Fusion to HIS Outbound GRN Outbound Integration

Author: Nidhi

Creation Date: Jun 15, 2023

Last Updated: Jun 15, 2023

Version: 1.0

**Approvals:**

|  |  |
| --- | --- |
| <Approver 1> | <Rakesh Khanna / Naushad / Amit> |
| <Approver 2> | <Business Owner> |

# Document Control

## Change Record

| Date | Author | Version | Change Reference |
| --- | --- | --- | --- |
| 15-Jun-2023 | Nidhi | 1.0 | No Previous Document, Draft version |
|  |  |  |  |
|  |  |  |  |

2

## Reviewers

| Name | Position |
| --- | --- |
| FHL Business | Business User |
| FHL IT Group | Fortis IT Team |
|  |  |
|  |  |

Contents

1 Document Control ii

1.1 Change Record ii

1.2 Reviewers ii

2 Introduction 5

2.1.1 Scope for this Document 5

2.1.2 Intended Audience 5

3 Overview 6

3.1 Business Objectives 6

3.2 Major Features 6

3.3 Glossary 6

4 High-Level HIS OUTBOUND Workflow 7

5 Oracle Fusion 8

5.1 Data Model 8

5.2 Report 9

6 Oracle PAAS Compute 11

6.1 SFTP Server 11

6.2 Bash Script Processing 11

7 Oracle PAAS database 15

8 HIS System 18

8.1 HIS Services 18

9 Data FIELD Mapping 19

9.1 Field to Field Mapping 19

9.2 Frequency 19

9.3 File Format 19

10 Exception handling 20

10.1 Scenarios 20

11 Assumptions / Considerations 21

12 Open and Closed Issues 22

12.1 Open Issues 22

12.2 Closed Issues 22

# Introduction

### Scope for this Document

This analysis specification document is prepared to layout the Item Outbound design of outbound integration to HIS system.

### Intended Audience

This document is intended for the following groups to provide the action items and consideration that are required to complete the implementation of the various Finance, Supply Chain, HR outbound to HIS system from Oracle Fusion.

#### Fortis Business Users

This document will be helpful for Fortis Business users to understand the technical design and process change with the new structure implemented in Oracle. The business user will validate this document after the verification of the requirement.

#### PwC Technical Team

The PwC technical team will use this document as a source technical design document to develop the technical solution to implement / deploy outbound integration to HIS system from Oracle Fusion.

#### Fortis IT Team

The Fortis IT Team must facilitate the PwC technical team for requisite details and other elements required from Business. The Fortis IT team is also responsible to arrange and provide required technical information wherever is required.

# Overview

Fortis Healthcare Limited (FHL) is a chain of hospitals, headquartered in India. Fortis started its health care operations from Mohali where first Fortis hospital was started. Later, the hospital chain purchased the healthcare branch of the Escorts group and increased its strength in various parts of the country. The Fortis health care also operates its hospital in Vasant Kunj, Faridabad, Gurgaon. The FMRI hospital at Gurgaon is the headquarter of Fortis healthcare with all the major facilities at the hospital.

This specification document is prepared to layout the common design of outbound integration to HIS systems. The outbound integration would from the Oracle cloud to HIS. The data flow would be through Oracle PaaS compute. PaaS system to be used for encryption and transformation.

## Business Objectives

The following are the business objectives.

1. To have a secure, reliable and scalable design flow from Oracle SaaS to HIS.
2. To have a solution providing the monitoring capabilities.

## Major Features

The proposed solution will have the following major features

1. SFTP Delivery
2. Synchronous REST Call with binary attachments.

## Glossary

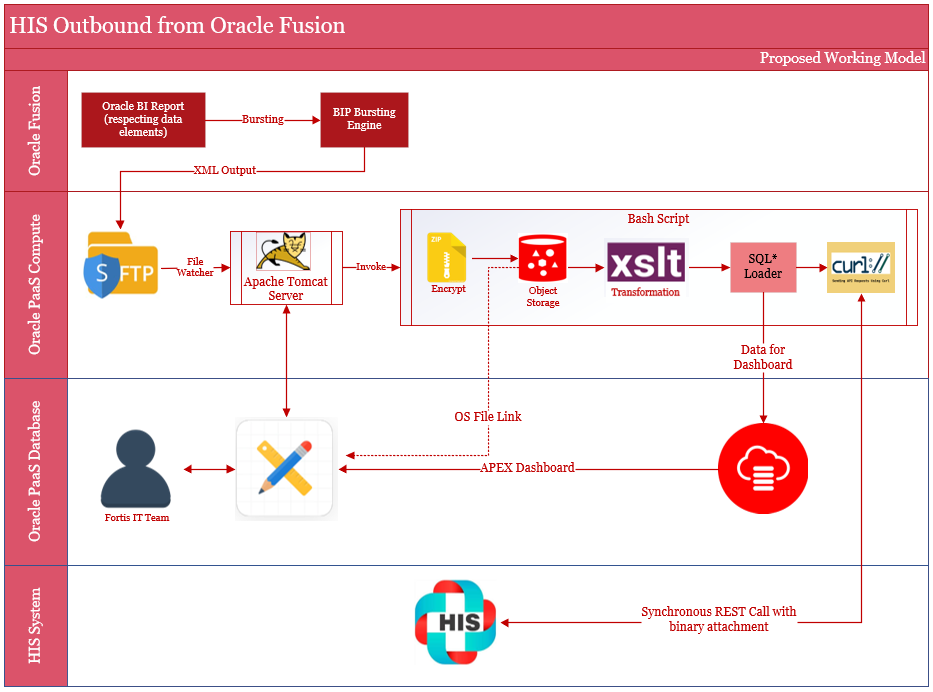
REST

Representational State Transfer.

SFTP

Secure File Transfer Protocol

# High-Level HIS OUTBOUND Workflow



# Oracle Fusion

This section describes the Oracle Fusion component in detail.

Oracle BI Report will be developed and designed in BI Catalogue Shared Folder -> Custom path. The report will be based on data elements as per requirement from HIS.

Reporting bursting capability would be utilized to generate XML Outbound file. The report will be secured using SaaS application BI and Job Roles.

The frequency of report execution would be as per data required from HIS.

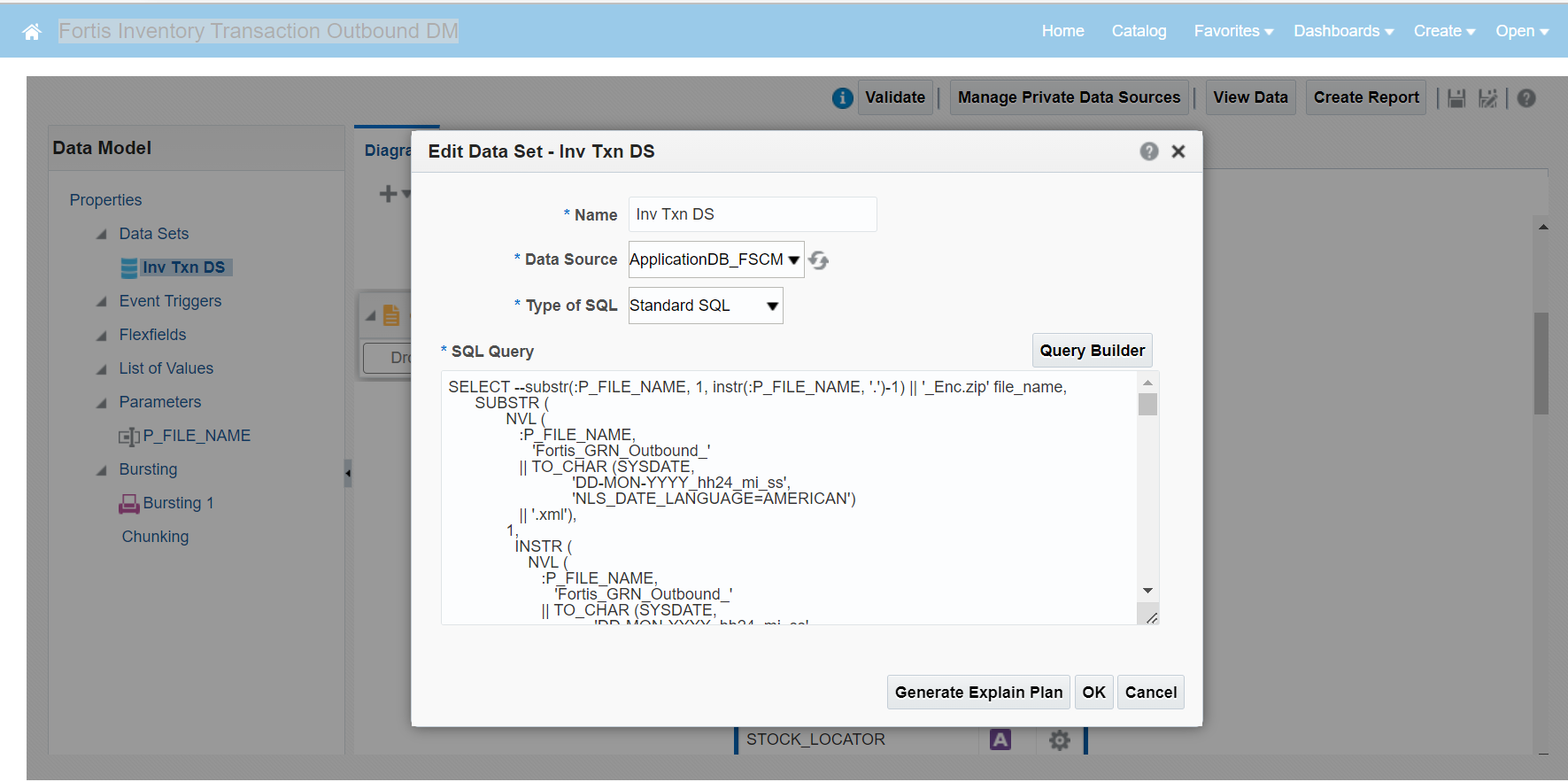
## Data Model

The custom data model would be designed as per data requirement from HIS. The data model provides the capability to write SQL query as per defined criteria. The parameter would be defined as per business requirement. Data model encapsulates the business logic for bursting report XML data.

#### Data Set GRN Outbound

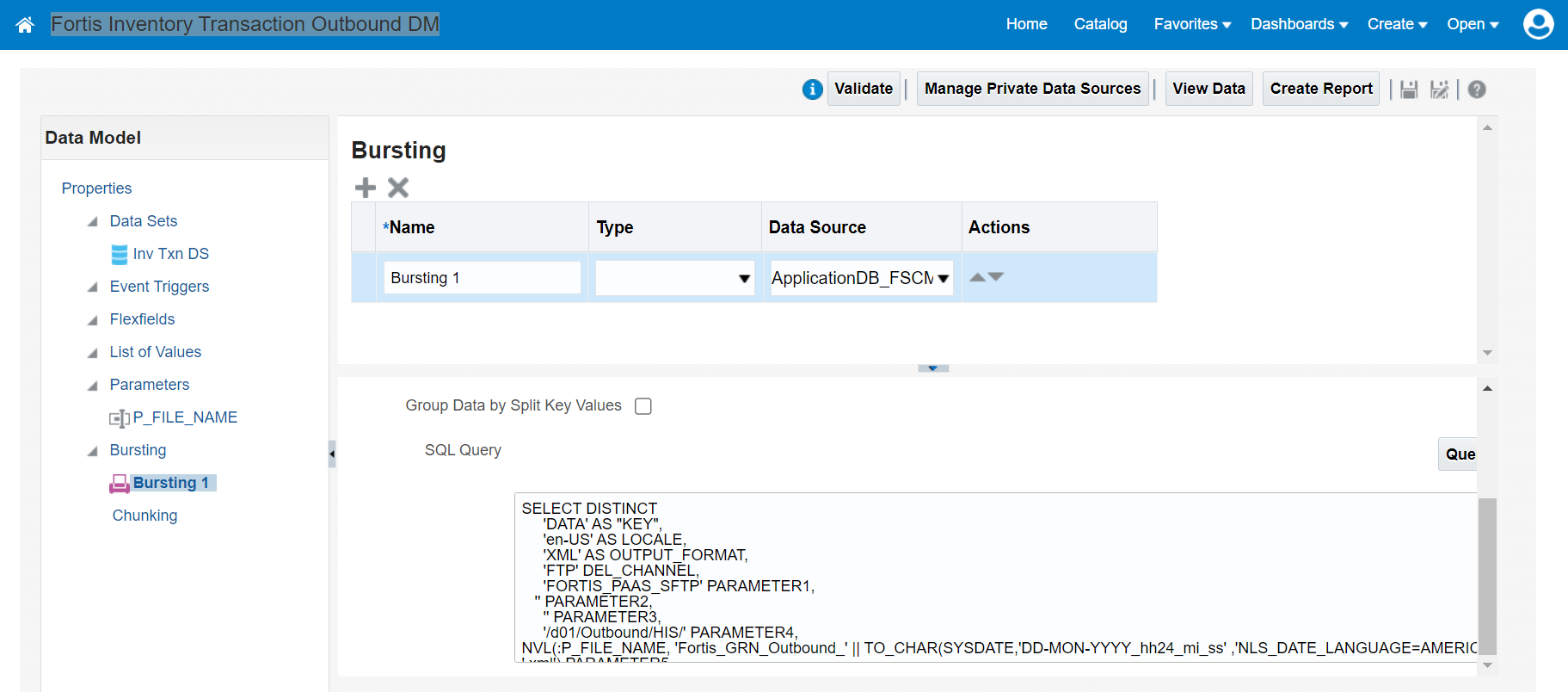
Data sets provides the feature to write standard SQL queries, view structure, data and code. Data tab is for visualization of the sample output, export of sample data and for saving sample data.

|  |  |  |  |
| --- | --- | --- | --- |
| SI No | Interface Name | Data Model Path | Data Model Name |
| 1 | GRN Outbound | /Custom/Interfaces/Integration Reports/Inventory/ | Fortis Inventory Transaction Outbound DM |



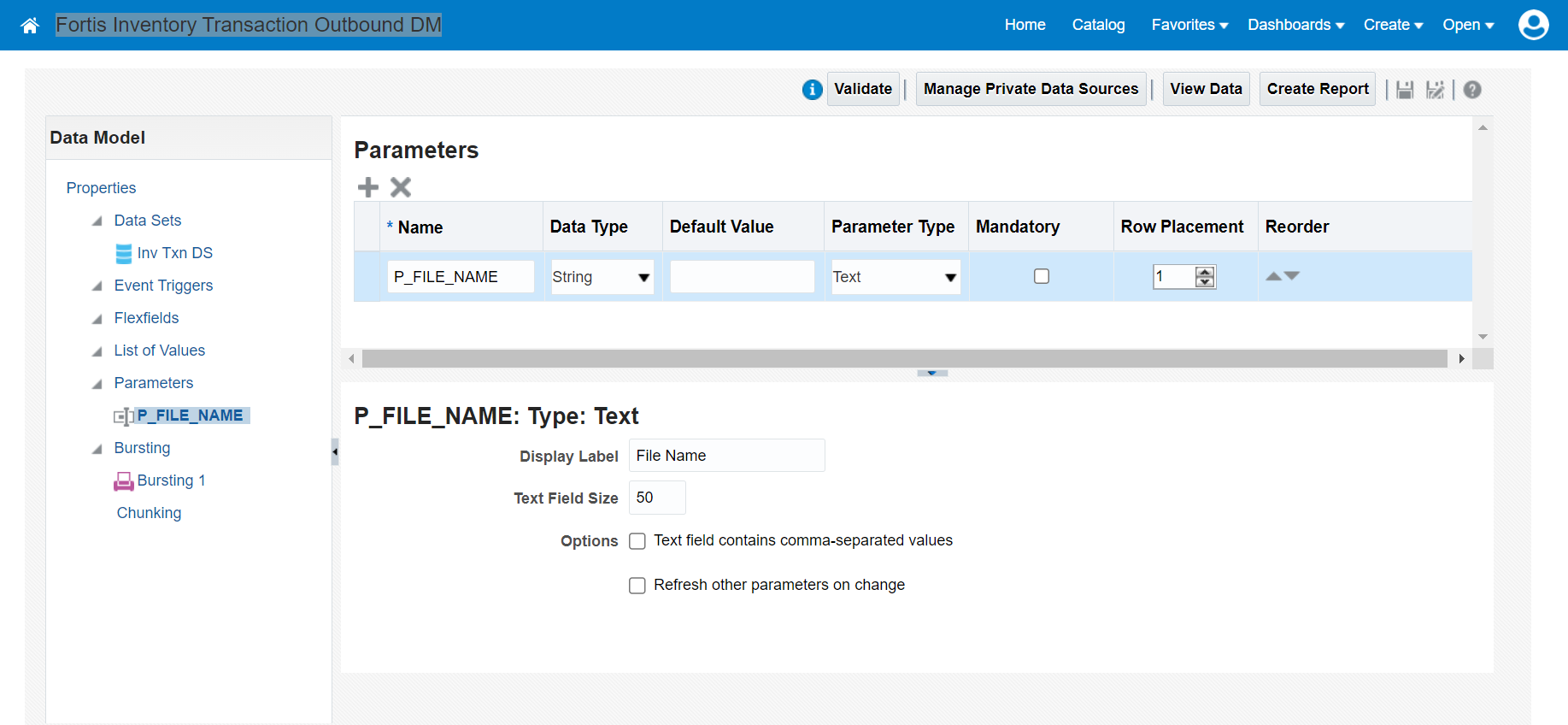
#### Bursting

This section enables you to write the report bursting logic. In this section you can define the deliver by, split by criteria and you’ll get the capability to write the bursting SQL for delivery channel.



#### Parameters

This section enables you to define the report parameters. In this section you can define the parameter name, data type, default values, parameter types, row placement etc. This section also enables you to make parameters as mandatory.



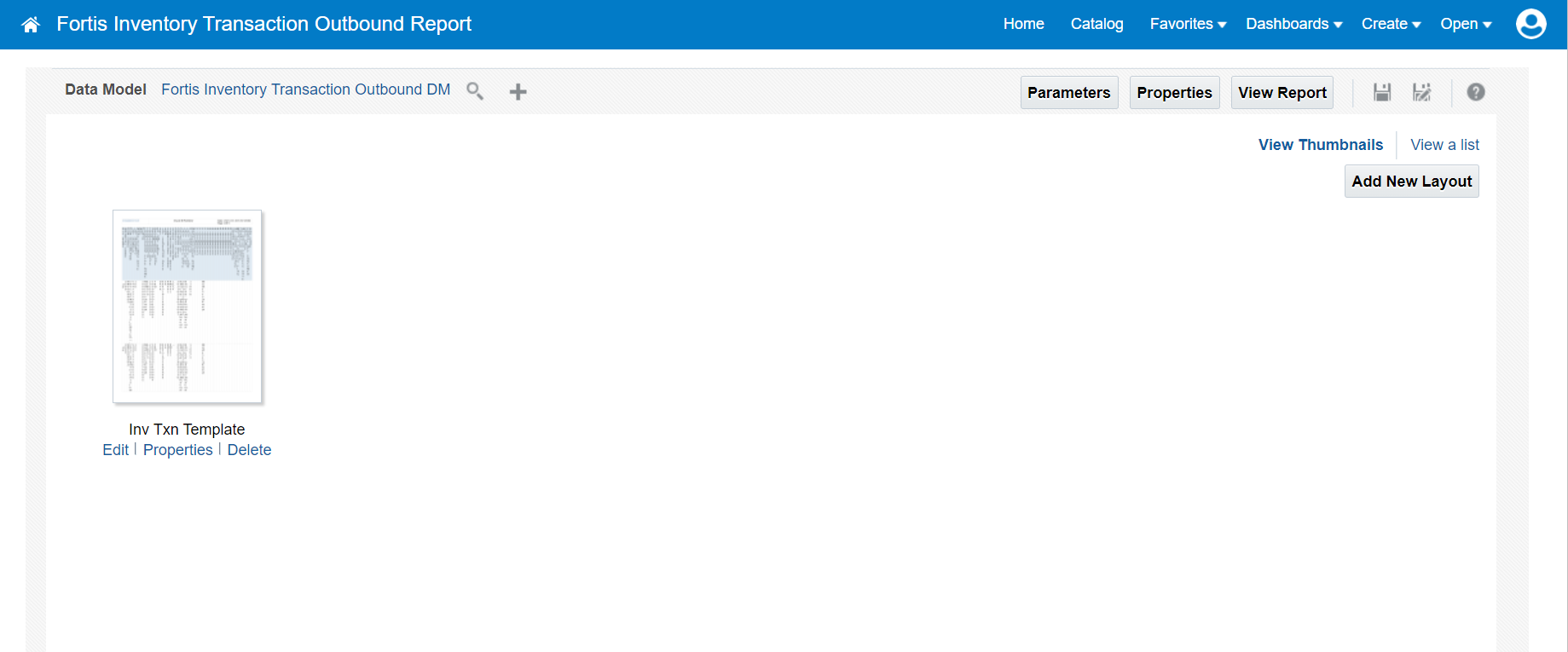
## Report

The report encapsulates the complexity of the data model and enables to define layout for the report output. Reports are secured by setting up the permission.

#### Report Details

This section enables you to enable bursting and select additional properties related to report.

|  |  |  |  |
| --- | --- | --- | --- |
| SI No | Interface Name | Report Path | Report Name |
| 1 | GRN Outbound | /Custom/Interfaces/Integration Reports/ Inventory / | Fortis Inventory Transaction Outbound Report.xdo |



# File Processing

Bursting Logic in the report generates a xml file and places the xml file in the destination path provided in the sql logic as given below:-

SELECT DISTINCT

'DATA' AS "KEY",

'en-US' AS LOCALE,

'XML' AS OUTPUT\_FORMAT,

'FTP' DEL\_CHANNEL,

'FORTIS\_PAAS\_SFTP' PARAMETER1,

'' PARAMETER2,

'' PARAMETER3,

'/d01/Outbound/HIS' PARAMETER4,

NVL(:P\_FILE\_NAME, 'Fortis\_COA\_Outbound\_' || TO\_CHAR(SYSDATE,'DD-MON-YYYY\_hh24\_mi' ,'NLS\_DATE\_LANGUAGE=AMERICAN') || '.xml') PARAMETER5,

'true' PARAMETER6

FROM DUAL

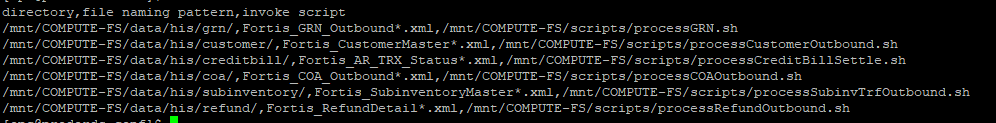
**File Watcher**

The file watcher on the compute node is designed as tomcat servlet (implements ServletContextListener) reading scheduler.conf file under tomcat’s configuration directory (CATALINA\_HOME/conf) to fetch the list of files to be watched for processing. The data format of the file is comma separated fields as below

**Directory:** The directory where a file will arrive

**File naming pattern:** Naming patter of the file for identification, may include wild card character as ‘\*’

**invoke script:** Upon arrival of a file matching with above pattern; then shell script to invoke for processing by passing the file name as parameter



Source code of the watcher java class is a servlet which implements javax.servlet.ServletContextListener is as follow

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.nio.file.FileSystems;

import java.nio.file.FileVisitResult;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.nio.file.SimpleFileVisitor;

import static java.nio.file.StandardWatchEventKinds.ENTRY\_CREATE;

import static java.nio.file.StandardWatchEventKinds.ENTRY\_DELETE;

import static java.nio.file.StandardWatchEventKinds.ENTRY\_MODIFY;

import java.nio.file.WatchEvent;

import java.nio.file.WatchKey;

import java.nio.file.WatchService;

import java.nio.file.attribute.BasicFileAttributes;

import java.util.HashMap;

import java.util.Map;

public class HIS\_JavaFileWatcher {

private final WatchService watcher;

private final Map<WatchKey, Path> keys;

/\*\*

\* Creates a WatchService and registers the given directory

\*/

HIS\_JavaFileWatcher(Path dir) throws IOException {

this.watcher = FileSystems.getDefault().newWatchService();

this.keys = new HashMap<WatchKey, Path>();

walkAndRegisterDirectories(dir);

}

/\*\*

\* Register the given directory with the WatchService; This function will be called by FileVisitor

\*/

private void registerDirectory(Path dir) throws IOException {

WatchKey key = dir.register(watcher, ENTRY\_CREATE, ENTRY\_DELETE, ENTRY\_MODIFY);

keys.put(key, dir);

}

/\*\*

\* Register the given directory, and all its sub-directories, with the WatchService.

\*/

private void walkAndRegisterDirectories(final Path start) throws IOException {

Files.walkFileTree(start, new SimpleFileVisitor<Path>() {

@Override

public FileVisitResult preVisitDirectory(Path dir, BasicFileAttributes attrs) throws IOException {

registerDirectory(dir);

return FileVisitResult.CONTINUE;

}

});

}

/\*\*

\* Process all events for keys queued to the watcher

\*/

void processEvents() {

for (;;) {

WatchKey key;

try {

key = watcher.take();

} catch (InterruptedException x) {

return;

}

Path dir = keys.get(key);

if (dir == null) {

System.err.println("WatchKey not recognized!!");

continue;

}

for (WatchEvent<?> event : key.pollEvents()) {

@SuppressWarnings("rawtypes")

WatchEvent.Kind kind = event.kind();

@SuppressWarnings("unchecked")

Path name = ((WatchEvent<Path>) event).context();

Path child = dir.resolve(name);

String fileNameWithPath = child.toString();

if (kind == ENTRY\_CREATE) {

System.out.format("%s: %s\n", event.kind().name(), child);

int index = fileNameWithPath.lastIndexOf(".");

System.out.println("index =" + index);

if (index > 0) {

String extension = fileNameWithPath.substring(index + 1);

if (extension.equalsIgnoreCase("xml")){

System.out.println("invoke unix script for file ="+fileNameWithPath);

runBashScript(fileNameWithPath);

}

}

}

if (kind == ENTRY\_CREATE) {

try {

if (Files.isDirectory(child)) {

walkAndRegisterDirectories(child);

}

} catch (IOException x) {

}

}

}

boolean valid = key.reset();

if (!valid) {

keys.remove(key);

if (keys.isEmpty()) {

break;

}

}

}

}

public static void main(String[] args) throws IOException {

System.out.println("Inside Main() ");

Path dir = Paths.get("/d01/Outbound/HIS/");

new HIS\_JavaFileWatcher(dir).processEvents();

}

private static void runBashScript(String fileName) {

try {

System.out.println("inside runBashScript for file ="+fileName);

Process proc = Runtime.getRuntime().exec("bash /u02/scripts/XXFH\_HIS\_OUTBOUND\_SHELL\_SCRIPT.sh" + " " + fileName);

BufferedReader read = new BufferedReader(new InputStreamReader(proc.getInputStream()));

try {

proc.waitFor();

} catch (InterruptedException e) {

System.out.println(e.getMessage());

}

while (read.ready()) {

System.out.println(read.readLine());

}

} catch (IOException e) {

System.out.println(e.getMessage());

}

}

}

The MinJob class implements Runnable is invoked by the servlet every minute to watch for a file, which primarily does the following

1. Write all the event details in CATALINA\_HOME/logs/scheduler.log file using java.util.logging.logger utility
2. Reads scheduler.conf and for each line it spawns a thread to match with the naming pattern
3. If matched execute specified shell script by passing the file name as parameter.

**File Details**

|  |  |  |  |
| --- | --- | --- | --- |
| SI No | Interface | File Path | XML File Name |
| 1 | GRN Outbound | /d01/Outbound/HIS/ | Fortis\_GRN\_Outbound\_<DD-MON-YYYY\_HH24\_MI\_SS>.xml |

**File Processing Scripts**

|  |  |
| --- | --- |
| File processing Unix script path | /u02/scripts |

Interface wise script details

|  |  |  |
| --- | --- | --- |
| SI No | Interface | SQL Loader Shell Script |
| 1 | Item Outbound | XXFH\_HIS\_OUTBOUND\_SHELL\_SCRIPT.sh |

**Script File processing Logic-**

1. File watcher will invoke unix script to process GRN Master outbound file using shell script - XXFH\_HIS\_OUTBOUND\_SHELL\_SCRIPT.sh
2. A entry will be created for the file in XXFH\_HIS\_FILE\_DETAILS\_TBL by using the function - XXFH\_HIS\_FILE\_STATUS\_UPD\_PRC from this unix script.
3. Program will zip the xml file, and then encrypt the file using public gpg key of HIS.
4. Post that create a checksum of the zip file, this checksum will be used while sending file to HIS, this is used to make sure at HIS end that file is received from actual source.
5. Now zip file is loaded in objects storage to keep the backup of data file. If there are any errors in uploading the zip file in the object storage, error response will be captured in a log file and same status will be updated in XXFH\_HIS\_FILE\_DETAILS\_TBL
6. Now it will invoke HIS REST API to send the file to HIS

# Oracle PAAS database

Table1: XXFH\_HIS\_FILE\_DETAILS\_TBL

Description: Interface control table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Table Column** | **Data Type** | **Length** | **Remarks** |
| 1 | FILE\_ID | NUMBER | 22 | For Fusion Inbound integrations, this is OIC\_FLOW\_ID of the OIC Integration run. |
| 2 | FILE\_NAME | VARCHAR2 | 240 | File name of Fusion inbound or Outbound integration |
| 3 | INTG\_TYPE | VARCHAR2 | 30 | Inbound/Outbound |
| 4 | MODULE\_NAME | VARCHAR2 | 30 | Used for Fusion Outbound interfaces only, HIS specific field to bifurcate the file as per various Fusion Outbound interfaces |
| 5 | SRC\_CHECKSUM | VARCHAR2 | 240 | Checksum (MD5) calculated of the final file at the source end |
| 6 | FILE\_STATUS | VARCHAR2 | 30 |  |
| 7 | ERROR\_MSG | VARCHAR2 | 4000 |  |
| 8 | ADDITIONAL\_INFO | VARCHAR2 | 4000 | Contains HIS REST API response for Fusion Outbound interfaces. Contains Fusion specific ESS job ids for Fusion Inbound interfaces. |
| 9 | OIC\_FLOW\_ID | NUMBER | 22 |  |
| 10 | SCHEDULER\_JOB\_NAME | VARCHAR2 | 100 |  |
| 11 | PROCESSING\_START\_DATE | DATE | 7 |  |
| 12 | PROCESSING\_END\_DATE | DATE | 7 |  |
| 13 | CREATION\_DATE | DATE | 7 | WHO columns |
| 14 | CREATED\_BY | VARCHAR2 | 30 |
| 15 | LAST\_UPDATE\_DATE | DATE | 7 |
| 16 | LAST\_UPDATED\_BY | VARCHAR2 | 30 |

# HIS System

HIS is hospital information and management system which is used as front desk system through which all the transactions are initiated.

## HIS Services

HIS REST based web-services are used to send the outbound data from Oracle Fusion to HIS, it uses following API’s.

|  |  |
| --- | --- |
| HIS Username | HIS\_Integ\_USER\_PROD |
| HIS Encryption Key | FAC70AF7 |

|  |  |  |  |
| --- | --- | --- | --- |
| SI No | Service Details | Parameters | Description |
| 1 | https://ihisapi.fortishealthcare.com/api/ValidateUser |  | To validate HIS user and get the authentication token |
| 2 | https://ihisapi.fortishealthcare.com/api/FusionPaaS/UploadFile |  | To upload data file in HIS |
| 3 | https://ihisapi.fortishealthcare.com/api/FusionPaaS/errorfiles |  | To upload error files in HIS |

# Data FIELD Mapping

Integration data field mapping from Fusion to HIS format.

## Field to Field Mapping

## Frequency

Repeats every 20 minutes

## File Format



## Code Pack



# Exception handling

Integration data field mapping from Fusion to HIS format.

## Scenarios

The following is the list of exception scenarios –

* If GRN is not updated after the timestamp of previous success run, it won’t be sent to HIS.
* If there are any errors in data processing in HIS, this needs to be checked with the of HIS team.

# Assumptions / Considerations

The Proposed Solution will have the following technical considerations.

In the future, if any product bug arises in functionality then this process needs to be revisited after the bug is fixed.

# Open and Closed Issues

## Open Issues

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

## Closed Issues

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