

Document properties:

**Prepared by:** Vijay Sharma

**Version:** 0.11

**Publish Date:** 08/04/2020

**Filename:** EDL Ingestion Design - IDH Data warehouse.docx

Template properties:

**Prepared by:** Riaan Jacobs

**Template Version:** 1.2

**Solution Design:**

DataPhile Data Ingestion

***Disclaimer:***

*The information contained in this document and any attachments is strictly confidential and is for internal use only.*

Table of Content:

[1. Introduction 4](#_Toc469578164)

[1.1. Purpose 4](#_Toc469578165)

[1.2. Intended Audience 4](#_Toc469578166)

[1.3. Definitions, Acronyms and Abbreviations 4](#_Toc469578167)

[1.4. Scope 4](#_Toc469578168)

[2. Development Environment and Coding Standards 5](#_Toc469578169)

[2.1. Development Toolset and Environment 5](#_Toc469578170)

[2.2. Coding Standards 5](#_Toc469578171)

[3. Solution Architect Document 6](#_Toc469578172)

[4. System Domain Design 7](#_Toc469578173)

[4.1. System Domain Chart 7](#_Toc469578174)

[4.2. System Domains 7](#_Toc469578175)

[5. Data Design 8](#_Toc469578176)

[5.1. Source Data Description 8](#_Toc469578177)

[5.2. Transformation of Data 8](#_Toc469578178)

[5.3. Data Security 8](#_Toc469578179)

[5.3.1. Internal Controls 8](#_Toc469578180)

[5.3.2. Data Retention 8](#_Toc469578181)

[6. Process Design 9](#_Toc469578182)

[6.1. Processes Design Overview 9](#_Toc469578183)

[6.2. Process Flow 9](#_Toc469578184)

[6.3. Processes 9](#_Toc469578185)

[6.4. Error and Exception Handling 10](#_Toc469578186)

[7. Services Design 11](#_Toc469578187)

[7.1. Service 11](#_Toc469578188)

[7.1.1. Service Methods 11](#_Toc469578189)

[7.1.2. Service Components 11](#_Toc469578190)

[7.1.3. Service Security 12](#_Toc469578191)

[7.1.4. Identification and Authentication 12](#_Toc469578192)

[7.2. Error and Exception Handling 12](#_Toc469578193)

[7.3. Deployment Information 12](#_Toc469578194)

[7.3.1. Development Environment 12](#_Toc469578195)

[7.3.2. Test Environment 12](#_Toc469578196)

[7.3.3. UAT Environment 12](#_Toc469578197)

[7.3.4. Production Environment 12](#_Toc469578198)

[8. Logical Structure 13](#_Toc469578199)

[9. Capacity and Performance Design 14](#_Toc469578200)

[10. Test Strategy 15](#_Toc469578201)

[11. References 16](#_Toc469578202)

[12. Appendix 17](#_Toc469578203)

[Appendix A. Test Cases 17](#_Toc469578204)

Document history

Notation legend:

<…> - mandatory

[…] - optional

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version Number | Revised by | Document Change Details |
| 04/08/2020 | 0.1 | Vijay Sharma | Initial version |
| 04/14/2020 | 0.11 | Vijay Sharma | Modified scope, process design overview. Added Typed hql, sqoop example, |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this project is to provide end users a holistic view of multiple lines of business under one central location, Investment Division Data Lake (IDDL). This design document is focusing on data required from DataPhile database into IDDL ingestion layer.

## Intended Audience

Intended audience for this document includes members of the AUM program, including project management, technical architects, developers and quality assurance.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **ABBREVIATION** | **DESCRIPTION** |
| AUM | Assets Under Management |
| EDL | Enterprise Data Lake |
| NiFi | Hadoop file management toolset |
| OOZIE | Workflow scheduler for Hadoop jobs |
| Atlas | Hadoop metadata management tool |
| JHI | John Hancock Investment |
| IDH | Investment Data Hub |
| HDFS | Hadoop Distributed File System |
| HWK | Hortonworks |

## Scope

The scope of DataPhile data ingestion is to import/load data from DataPhile-Progress DB into IDDL raw zone. The process imports data from 16 of the DataPhile tables.

Ingestion of 2 of years of DataPhile history for tables required for AUM with no Personally identifiable information (PII) that could be sourced from Operational Progress database.

Process extract deltas from DataPhile as soon as daily refresh is completed and data are available for consumption at 5 am each weekday. Monday To Friday.

subsequent daily loads of new data as well as medications to previous days

after receiving a trigger file dpl\_daily\_trigger.dat from DataPhile. This design does not include any curation processing that might copy data out of the Raw Zone.

# Development Environment and Coding Standards

## Development Toolset and Environment

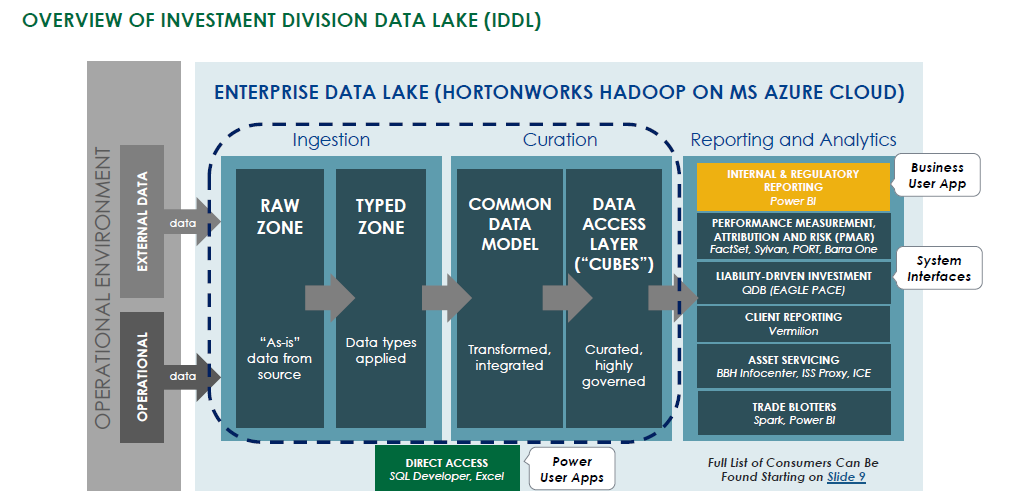
Server and connection information is provided in the [EDL User Guide](http://mfcglobalinv/pm/PMO/GoDataStreamProgram/GoDtaStrmDtaWreHouse/Design%20and%20Development/Enterprise%20Data%20Lake/edl-user-guide_v0.1.docx).

Additional Development environment information is provided in the [Manulife Enterprise Data Lake - Workstation Environment](http://mfcglobalinv/pm/PMO/GoDataStreamProgram/GoDtaStrmDtaWreHouse/Design%20and%20Development/Enterprise%20Data%20Lake/Manulife%20Enterprise%20Data%20Lake%20-%20Workstation%20Environment.docx) guide.

## Coding Standards

Coding Standards for the EDL are addressed in the <[Coding Standards Document](https://mfc.sharepoint.com/:w:/r/sites/GWAMIDDLProjects/FAUM/Shared%20Documents/DataPhile/IDDH%20Best%20Practices%20for%20Curation%20Coding.docx?d=w0b612738f26c4c26a8dd4d80e3bac6ac&csf=1&web=1&e=kiX7Vj)>

# Solution Architect Document



# System Domain Design

## System Domain Chart



## System Domains

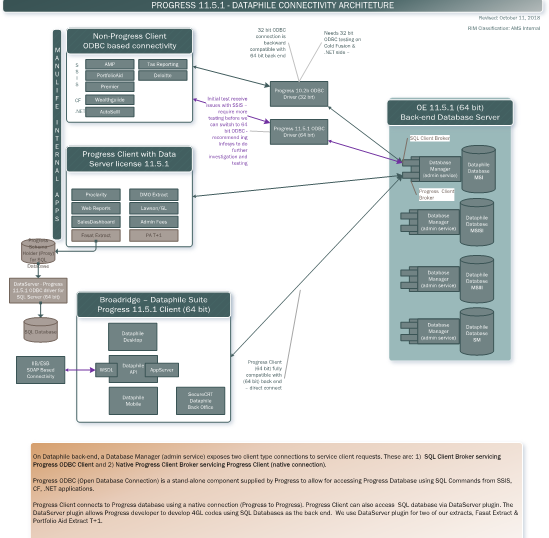
* Ingestion refers to the process of landing and persisting data within the EDL’s ‘Raw Zone’.
* Curation refers to the process of validating, transforming, and canonicalizing data within the EDL’s ‘Curated Zone’.
* The Data Warehouse represents the curated data within the EDL that is available for Consumers.
* Data Consumption and Access is provided to various Consumers who wish to utilize EDL data for reporting, analytics and import into downstream applications.

# Data Design

## Source Data Description

This component handles the importation and persistence of data from DataPhile Database into IDDL. At present, we have 16 tables to load from Progress DB to IDDL. The tables to import from DataPhile to IDDL is specified by using a configuration file (dpl.conf) which lists out the tables and columns of each table that needs to be imported to IDDL. Depending on the delivery of a definitive data dictionary, these tables are enumerated in [BRD DATAPHILE](https://mfc.sharepoint.com/:w:/r/sites/GWAMIDDLProjects/FAUM/Shared%20Documents/DataPhile/BRD-Dataphile%20to%20EDL%20v.2.docx?d=w73b20e7368e342238e224b0e48440edf&csf=1&web=1&e=H9y4fC).

[DataPhile](https://mfc.sharepoint.com/:u:/r/sites/GWAMIDDLProjects/FAUM/Shared%20Documents/DataPhile/Dataphile%20Connectivity%20Architecture.vsdx?d=wff8083941b9441e692347d780c0830d3&csf=1&web=1&e=Ga5QKL) connectivity architecture.



## Transformation of Data

A true copy of the original source data will be persisted in the Raw Zone as it is received. The data will be stored in the Raw Zone under the HDFS folder /apps/inv/aum/raw/dpl/<table\_name>/<process\_timestamp=yyyymmdd\_hhmmss>. An external Hive table will be created for each domain of the file received. These tables will reside in the inv\_aum\_raw schema and have same name as in DataPhile

source Database with an ‘dpl\_’ prefix to signify the source (<dpl>\_<table\_name>), where all the data will be typed as string.

In addition, the data will be parsed and imported to a managed Hive tables persisted in the inv\_aum\_typed schema. These tables will follow a different naming convention than the inv\_aum\_raw schema and data will be typed properly. Also, this will be an append operation which will combine tables from various brokers/ports into one. The typed table name will be prefixed with <source> (idh) along with the table\_name where the ‘t\_ods’ will be trimmed out. (E.g. typed table name, idh\_dpl\_ gnexch). The typed tables are partitioned on process\_date column.

## Data Security

Data will be secured through the use of an ACL.

### Internal Controls

Access to data will be limited to users who are members of the ACL group. Members of the group will be allowed to access data residing in the Hive tables in the inv\_aum\_typed schema.

An ACL for use by the EDL internal processes (in dev this is inv\_aum\_dev) will have access to the /apps/inv/aum/raw data as well.

### Data Retention

Data ingested into the EDL will be retained indefinitely. The data in HDFS is not encrypted.

# Process Design

The business process model for each of the Topics is represented in BRD. The BRD is listed here.

[BRD DATAPHILE](https://mfc.sharepoint.com/:w:/r/sites/GWAMIDDLProjects/FAUM/Shared%20Documents/DataPhile/BRD-Dataphile%20to%20EDL%20v.2.docx?d=w73b20e7368e342238e224b0e48440edf&csf=1&web=1&e=H9y4fC)

## Processes Design Overview

IDDL DPL Ingestion imports data from Progress Server to IDDL daily from Monday to Friday (All Days?) after receiving a 'load done trigger file' (IDH\_Load\_<yyyymmdd>.txt) from DPL. All the data will be imported from DPL to IDDL on an initial load (Day0) and Day 1 onwards, only the delta will be imported to IDDL.

IDDL DPL Ingestion consist of 2 steps, an IDH NiFi Process and an oozie process. The IDH NiFi process looks for a specific trigger file from DPL DWH in the landing directory and start nifi process as soon as the trigger file is received in the landing. Once the trigger file is in landing folder, the NiFi process reads a configuration file 'dpl.conf' from hdfs and creates an oozie workflow 'wf\_idh.xml' and creates a trigger file (idh\_trigger.dat) which acts as an input trigger for oozie workflow. The NiFi process checks whether the workflow is created and skips the creation of ‘wf\_idh.xml’ if the workflow already exists in hdfs.

The next step in DPL Ingestion is an Oozie process which checks for a trigger file 'idh\_trigger.dat' in a specific location in hdfs and imports all the tables mentioned in the dpl.conf file from JHI Sql Server Database to IDDL. Once the ingestion process is done, the oozie workflow logs an entry in the typed registry table and sends a Success email notification to the respective support/business group.

NIFI Process group is trigerred by the existing mechanism ie CA Scheduler.

## Process Flow

A close up of a map

Description automatically generated

## Processes

| Process Name | Service/UI – URL | If Success | If Fails | Description |
| --- | --- | --- | --- | --- |
| Investments  /IDH Ingestion | NiFi Process Group | * Oozie workflow is created by reading the dpl.conf configuration file * Trigger file is created in a specified directory in hdfs to trigger Oozie workflow. | * Error occurred will be logged into Error Log table in inv\_typed schema. * Email is sent to [support group] | This process runs on the NiFi server and creates a file to kick off Oozie based on availability of new/updated data in JHI SQL Server. |
| Investments /IDH Ingestion | Oozie workflow | * Imported data is exposed in an external Hive table in the inv\_dpl\_raw database. * Data is exposed in a managed Hive table in the inv\_dpl\_typed database. * A SUCCESS registry entry is made into Registry Table in inv\_typed. | * Error occurred will be logged into Error Log table in inv\_typed schema. * Email is sent to [support group] | This workflow runs on the Oozie server. |

## Error and Exception Handling

NiFi processes log errors to the nifi-app.log located in

/data-01/LatestHDF/HDF-3.0.0/nifi-1.2.0.3.0.0.0-453/logs on the NiFi server.

Errors occurred in NiFi process will be moved to an error handling process group, where all the errors occurred is logged into an error\_log table in inv\_typed schema with type, process\_group, process\_name,jdbc string, message, source, topic, filename, user\_name, comments, process\_timestamp and error\_timestamp as the columns. Also, an email will be sent to the respective [support group] email address to notify that an error had occurred.

Error occurred while running oozie workflows are also logged into the error\_log table in inv\_typed schema with type, process\_group, process\_name, message, source, topic, filename, user\_name, comments, process\_timestamp and error\_timestamp as the columns.

Error messages are sent to [support group] from the workflow upon encountering unexpected errors.

# Services Design

The process will be implemented on the Hortonworks Data Platform (HDP). The Database ingestion pattern will be initiated in NiFi and implemented in an Oozie workflow.

List of the services:

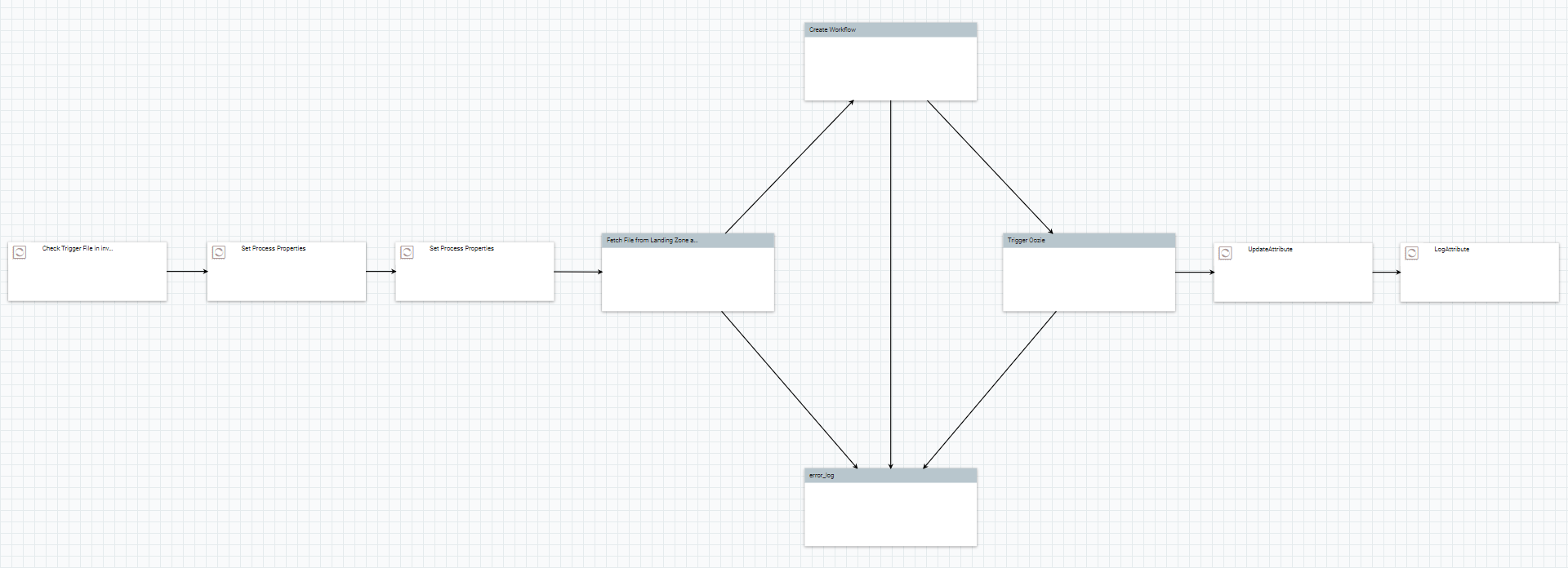
| Service | Service Type | Description |
| --- | --- | --- |
| Investments Division/DPL | NiFi process | This process runs on the NiFi server and creates an oozie workflow and a trigger file in a specified directory to trigger oozie. |
| Investments Division/DPL | Oozie workflow | This process runs on the Oozie server and loads the data into the IDDL. |

## NiFi process

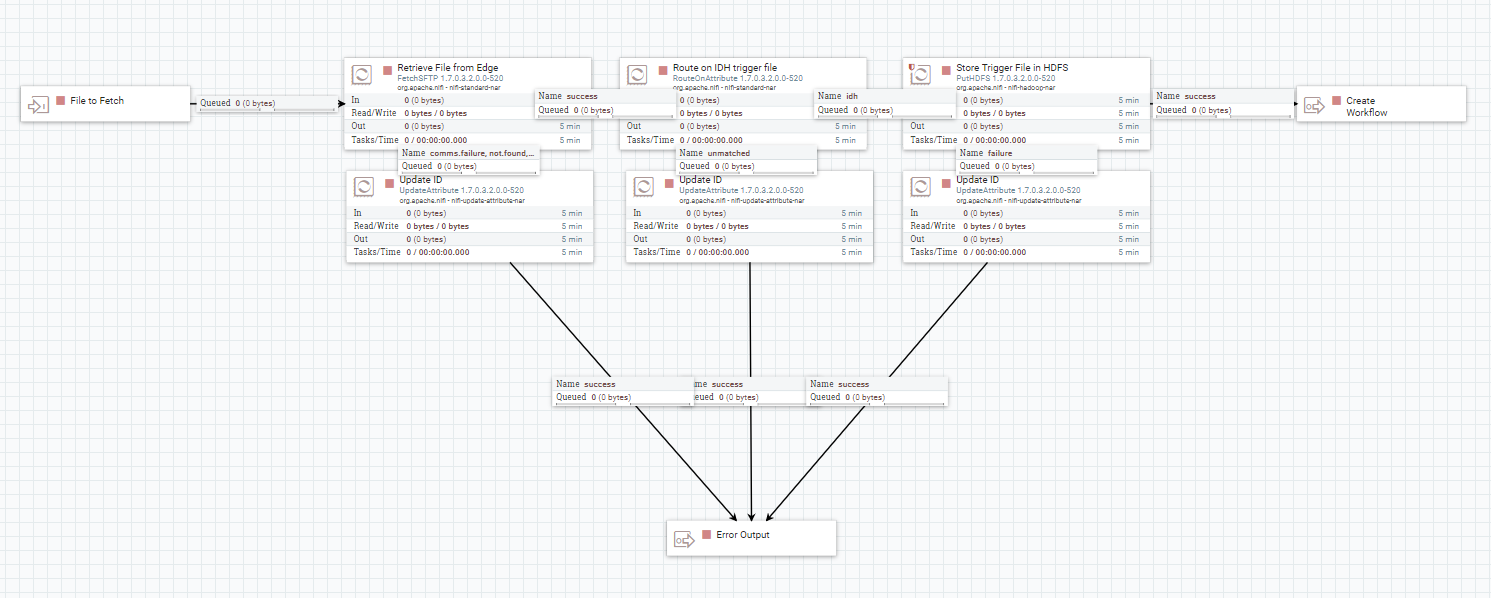
The HDF instance in the Manulife EDL is shared by different groups within Manulife. All of the Investments Division Data Flows are contained within one Investments Division Process Group. Within Investments process group, there are different nested Process Groups, each representing an ingestion of a source. The Data Flow implementing this design is in the IDH NiFi Process Group.

The IDH ingestion Process Group is based on the Database Ingestion Pattern outlined in the [EDL Ingestion Patterns](https://mfc.sharepoint.com/:w:/r/sites/pm/PMO/GoDataStreamProgram/GoDtaStrmDtaWreHouse/Design%20and%20Development/Enterprise%20Data%20Lake/EDL%20Ingestion%20Patterns.docx?d=w694e8e8d31d24f91bb5831d71e3a6708&csf=1&e=MusaSr) document.

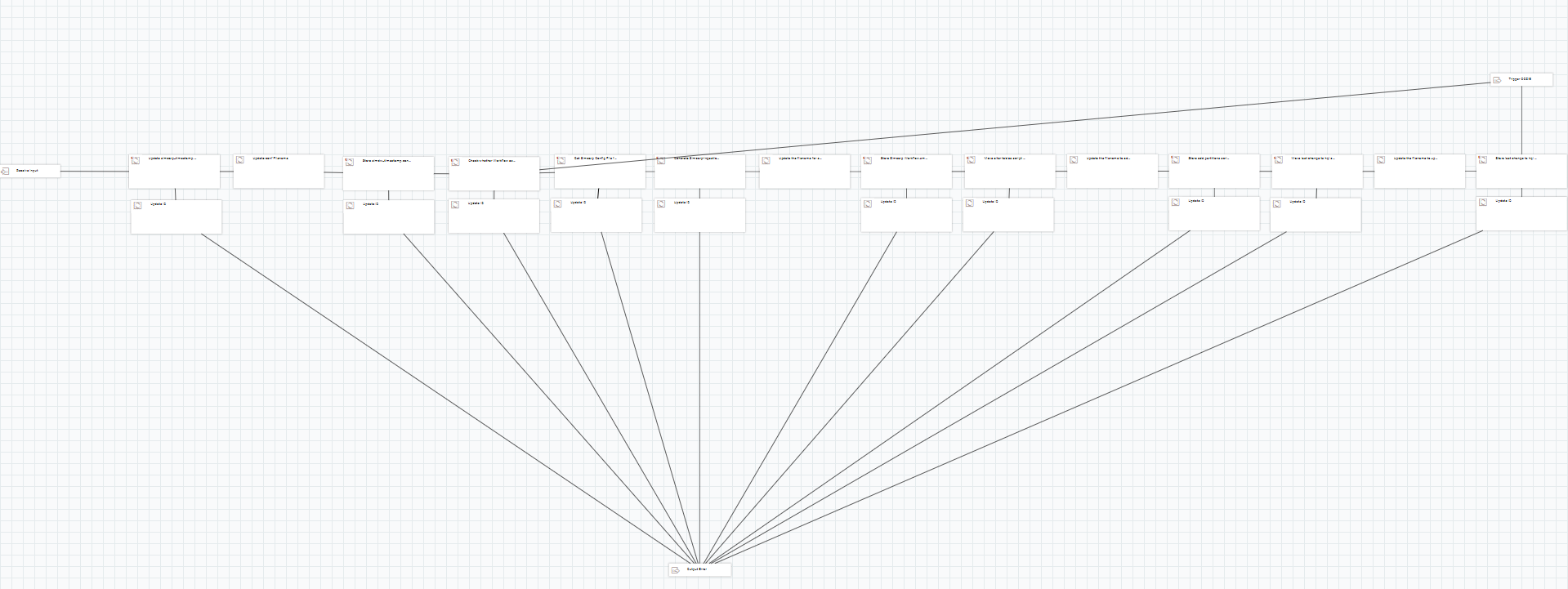
A picture of the IDH ingestion process is shown below.



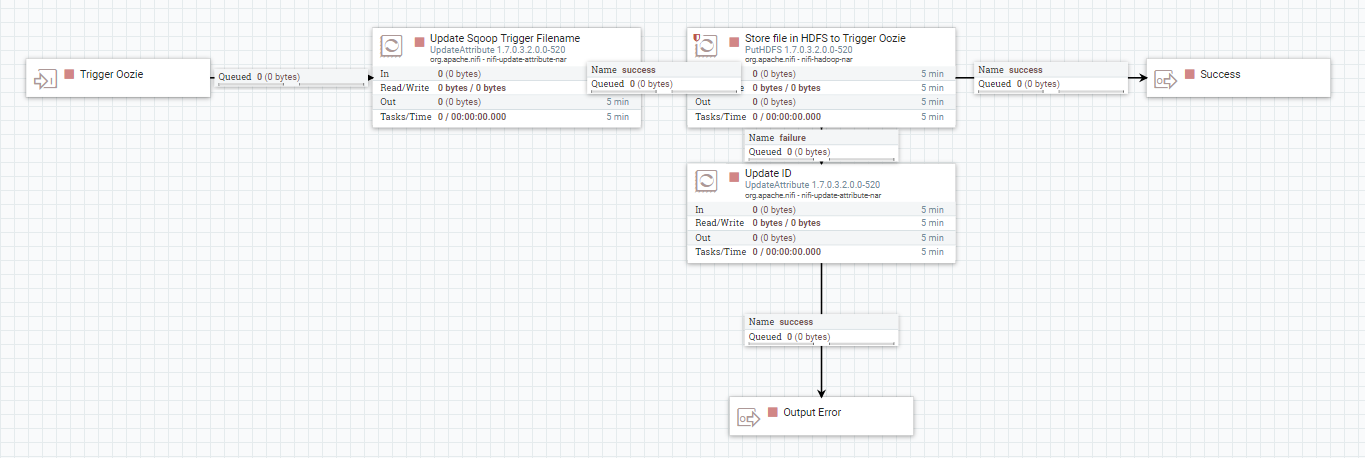
A detailed picture of “Fetch File from Landing Zone” process group is shown below.



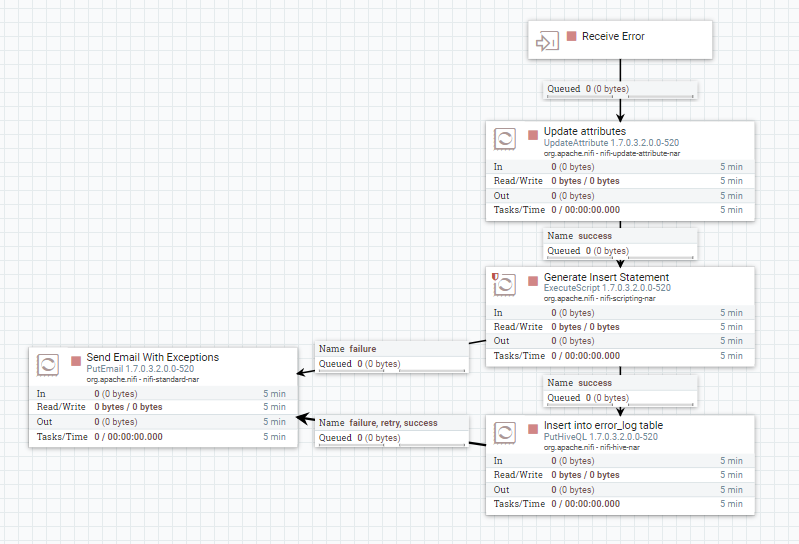
A detailed picture of “Create Workflow” process group is shown below.



A detailed picture of “Trigger Oozie” process group is shown below.



A detailed picture of “error\_log” process group is shown below.



### Process Group Processes

| Process | Type | Description |
| --- | --- | --- |
| Check Trigger file in inv\_landing | ListSFTP | Check whether the trigger file is there in inv\_landing zone. |
| Set Process Properties | UpdateAttribute | Update the attributes of the FlowFile. |
| “Fetch File from Landing Zone” Process Group | | |
| File to Fetch | InputPort | An Input Port to transfer the FlowFile from Update Attribute to “Fetch File from Landing Zone” Process Group. |
| Retrieve File from Edge | FetchSFTP | Fetch the trigger file from inv\_landing zone |
| Route on Attribute | RouteOnAttribute | Route the flow based on the trigger file |
| Create Workflow | OutputPort | Output Port to pass the FlowFile from “Fetch File from Landing Zone” to “Get List of Tables” process group. |
| “Create Workflow” Process Group | | |
| Receive Input | InputPort | An Input Port to transfer the FlowFile from “Fetch File from Landing Zone” process group to “Get List of Tables” Process Group. |
| Update idh\_timestamp.conf File | ExecuteScript | Groovy Script to update idh\_timestamp.conf config file |
| Store idh\_timestamp.conf File in HDFS | PutHDFS | Stores the idh\_timestamp config file to HDFS |
| Check whether Workflow exists in HDFS | FetchHDFS | FetchHDFS processor to check whether oozie ingestion WF exists |
| Get Idh Config File from HDFS | FetchHDFS | Fetch dpl.conf file to create Idh ingestion Ooziw workflow |
| Generate Idh-Ingestion-Workflow.xml | ExecuteScript | Groovy Script to create Idh ingestion workflow based on the config file |
| Update the filename for storing in HDFS | UpdateAttribute | An Update Attribute processor to update the workflow\_idh.xml |
| Store Workflow.xml in HDFS | PutHDFS | Store Ingestion Workflow.xml in HDFS |
| Move alter tables script from attributes into flowfile | ExecuteScript | Groovy Script to move alter table statements from attribute to flow-file content |
| Update the filename for storing in HDFS | UpdateAttribute | An Update Attribute processor to update the alter table statement |
| Store add partitions script in HDFS | PutHDFS | Store partition scripts in HDFS |
| Move last\_change\_ts script from attributes into flowfile | ExecuteScript | Groovy Script to move last\_change\_ts scripts from attribute to flow-file content |
| Update the filename for storing in HDFS | UpdateAttribute | An Update Attribute processor to update the last\_change\_ts script |
| Store last change ts hql in HDFS | PutHDFS | Store last\_change\_ts scripts in HDFS |
| Trigger OOZIE | OutputPort | Output Port pass the flowfile to Trigger oozie process |
| Update ID | UpdateAttribute | Update Attribute processor to update the attributes required for error log table |
| Output Error | OutputPort | An Output Port to transfer the Error FlowFile to an Error Log Process Group. |
| “Trigger Oozie” Process Group | | |
| Trigger Oozie | InputPort | Input Port to receive FLowfile to “Trigger Oozie” Process Group. |
| Update Sqoop Trigger Filename | UpdateAttribute | Update attribute to update the trigger filename |
| Store file in HDFS to Trigger Oozie | PutHDFS | Store trigger file in HDFS |
| “Error Log” Process Group” | | |
| Receive Error | InputPort | Input port to receive errored out flow files |
| Update Attributes | UpdateAttribute | An update processor to update the attributes of the failed flow-files |
| Generate Insert Statement | ExecuteScript | A groovy script to create the HQL to insert records into error log table. |
| Insert into error\_log table | PutHiveQL | Processor to execute the Insert statement |
| Send Email with Exceptions | OutputPort | An Output Port to transfer the Error FlowFile to an Error Log Process Group. |

#### Oozie workflow

A close up of a map

Description automatically generated

#### Oozie workflow actions

| Action Name | Type | Description |
| --- | --- | --- |
| Oozie Sqoop action | Sqoop import | Connects to the source DPL IDH Progress Server database and imports data into HDFS based on each table’s last modified timestamp. |
| Oozie Hive action | Hive Queries | Insert data from IDDL inv\_aum\_raw schema to inv\_aum\_typed schema. |

##### Oozie Sqoop Actions

The DPL ingestion will get triggered by an dpl\_trigger.dat file which gets created in HDFS under /apps/inv/aum/code/ingestion/oozie/dpl/trigger/ after successful execution of IDH NiFi process.

The Sqoop action will import data from the DPL IDH SQL Server database into IDDL’s Raw Zone in HDFS under the folder /apps/inv/aum/raw/dpl/<tablename> /<process\_timestamp=yyyyMMdd\_HHmmss>. Sqoop import the data in parallel by running as map-only job. IDH ingestion does an incremental load of data for all the tables unless specified in the dpl.conf configuration file.

For initial run, sqoop imports all the data in the mentioned tables in DPL Data warehouse into the target location in HDFS. From second run onwards, only the records that got added/changed after the last sqoop import, will be imported to the target location in HDFS by checking the table’s DATA\_COUMN1 and DATE\_COLUMN2 columns. Those records whose last\_change\_ts/ dwh\_load\_ts greater than the max of ODS\_INS\_DT / ODS\_UPD\_DT in IDDL tables will be loaded to IDDL during the sqoop import.

###### Sqoop command

sqoop import \

-Dmapreduce.job.queuename=<queueName> \

-Dhadoop.security.credential.provider.path=jceks://hdfs/<hdfs\_path>/ <sqoop\_password\_filename>.password.jceks \

--connect <sql server db connection> \

--username <username> \

--password-alias <idh\_sqoop>\_password.alias \

--target-dir /apps/inv/aum/raw/dpl/<table\_name>/process\_timestamp=yyyyMMdd\_HHmmss \

--query "SELECT

col1, col2, ..., coln

FROM PUB.<tablename>

WHERE ODS\_INS\_DT > TO\_DATE (MAX (PROC\_DT from IDDL))

AND $CONDITIONS

UNION

SELECT

col1, col2, ... coln

FROM PUB.<tablename>

WHERE ODS\_UPD\_DT > TO\_DATE (MAX (PROC\_DT from IDDL))

AND $CONDITIONS \

--hive-drop-import-delims \

--fields-terminated-by '\001' \

--null-string \\\\N \

--null-non-string \\\\N \

--temporary-rootdir /apps/inv/aum/raw/dpl/tmp \

-m 1 \

--append

##### Oozie Hive Actions

Hive Action will copy data from inv\_aum\_raw schema tables into the managed tables in the inv\_aum\_typed schema by incrementally appending the partitioned data by timestamp.

### Service Components

*N/A*

### Service Security

Access to the Oozie server is only available to developers and administrators who are members of the inv\_aum\_dev ACL group

### Identification and Authentication

This is a server process. There are no external users.

Enterprise Data Lake infrastructure (specifically Hortonworks Data Platform) is secured by Kerberos (MIT KDC on Linux) to support its distributed computing architecture. At the same time, EDL has been integrated with Manulife global Active Directory (MFCGD.COM) for user authentication. The only users are developers and administrators.

## Error and Exception Handling

*The following table describes how errors and exceptions will be handled.*

*Roll Back and Error- 4 different nifi processes.*

| Action | Type | Description |
| --- | --- | --- |
| Log Oozie errors to an error\_log table (inv\_typed)  Send Error Notifications  (Oozie) | Create Table  uri:oozie:email-action | Logs the error into the error\_log table in inv\_typed schema when an error occurs.  Error\_log table  Sends a plain text email to the operations group e-mail (<email\_address>) with the subject “IDH ingestion Exception Occurred”. The body of the message to include "An error occurred” along with the job ID. |
| Log NiFi errors to an error\_log table (inv\_typed) | Create Table | Inserts the error occurred into an error log table if an error occurs while running NiFi. |
| Update Processor name | UpdateAttribute | Updates the unique Process name for each processor in NiFi. |
| Error table | Output port | Port to pass the flow file to the Error log process group |

## Deployment Information

There are 4 categories of servers assumed for this design and shared among the other ingestion designs.

* The KDC server runs the MIT Kerberos Key Distribution Center used for authentication.
* The Oozie server is the server that contains the workflow information.
* The HDF server is the server that runs Apache NiFi.
* The Hadoop Cluster servers include a cluster management node, 3 master nodes, and multiple data nodes. These servers run the Hortonworks Data Platform Hadoop components.

### Development Environment

|  |  |  |
| --- | --- | --- |
| KDC Realm name | DEV.EDL.COM |  |
| AD domain name | MFCGD.COM |  |
|  |  |  |
| Edge Node | azcedledged001.mfcgd.com | 10.236.248.6 |
| HDF node | azcedlnifid001.mfcgd.com | 10.236.248.15 |
|  |  |  |
| **KDC node** | azcedlkdcd001.mfcgd.com | 10.236.248.5 |
| **Cluster Management Node** | azcedlmgtd001.mfcgd.com | 10.236.248.7 |
| **Master1** | azcedlmstd001.mfcgd.com | 10.236.248.8 |
| **Master2** | azcedlmstd002.mfcgd.com | 10.236.248.9 |
| **Master3** | azcedlmstd003.mfcgd.com | 10.236.248.10 |
| **Data node** | azcedlwrkd###.mfcgd.com | 10.236.248.[11-32] |

### Test/QA Environment

|  |  |  |
| --- | --- | --- |
| KDC Realm name | QA.EDL.COM |  |
| AD domain name | MFCGD.COM |  |
|  |  |  |
| Edge Node | azcedledgev001.v01caedl.manulife.com | 10.237.20.18 |
| HDF node | azcedlnifiv002.v01caedl.manulife.com azcedlnifiv003.v01caedl.manulife.com azcedlnifiv004.v01caedl.manulife.com |  |
|  |  |  |
| **KDC node** | azcedlkdcv001.v01caedl.manulife.com | 10.237.248.5 |
| **Cluster Management Node** | azcedlmgtv001.v01caedl.manulife.com | 10.237.248.7 |
| **Master1** | azcedlmstv001.v01caedl.manulife.com | 10.237.248.8 |
| **Master2** | azcedlmstv002.v01caedl.manulife.com | 10.237.248.9 |
| **Master3** | azcedlmstv003.v01caedl.manulife.com | 10.237.248.10 |
| **Data node** | azcedlwrkv###.v01caedl.manulife.com | 10.237.248.[11-32] |

### UAT Environment

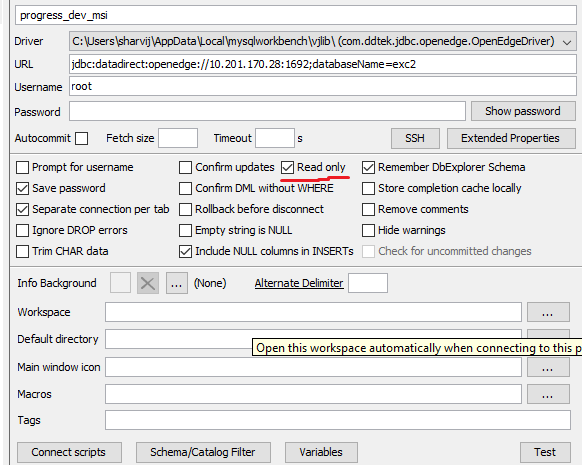
|  |  |  |
| --- | --- | --- |
| KDC Realm name | UAT.EDL.COM |  |
| AD domain name | MFCGD.COM |  |
|  |  |  |
| Edge Node | azcedledgev001.v01caedl.manulife.com | 10.237.20.18 |
| HDF node | azcedlnifiv002.v01caedl.manulife.com azcedlnifiv003.v01caedl.manulife.com azcedlnifiv004.v01caedl.manulife.com |  |
|  |  |  |
| **KDC node** | azcedlkdcv001.v01caedl.manulife.com | 10.237.248.5 |
| **Cluster Management Node** | azcedlmgtv001.v01caedl.manulife.com | 10.237.248.7 |
| **Master1** | azcedlmstv001.v01caedl.manulife.com | 10.237.248.8 |
| **Master2** | azcedlmstv002.v01caedl.manulife.com | 10.237.248.9 |
| **Master3** | azcedlmstv003.v01caedl.manulife.com | 10.237.248.10 |
| **Data node** | azcedlwrkv###.v01caedl.manulife.com | 10.237.248.[11-32] |

### Production Environment

|  |  |  |
| --- | --- | --- |
| KDC Realm name | PROD.EDL.COM |  |
| AD domain name | MFCGD.COM |  |
|  |  |  |
| Edge Node | azcedledgop01.op01caedl.manulife.com | 10.237.18.26 |
| HDF node | azcedlnifiop001.op01caedl.manulife.com azcedlnifiop002.op01caedl.manulife.com azcedlnifiop003.op01caedl.manulife.com | 10.237.18.4  10.237.18.5  10.237.18.6 |
|  |  |  |
| **KDC node** | azcedlkdcop001.op01caedl.manulife.com | 10.237.18.5 |
| **Cluster Management Node** | azcedlmgtop001.op01caedl.manulife.com | 10.237.18.7 |
| **Master1** | azcedlmstop001.op01caedl.manulife.com | 10.237.18.8 |
| **Master2** | azcedlmstop002.op01caedl.manulife.com | 10.237.18.9 |
| **Master3** | azcedlmstop003.op01caedl.manulife.com | 10.237.18.10 |
| **Data node** | azcedlwrkop###.op01caedl.manulife.com | 10.237.18.[11—33] |

Progress DB Connection Details.

**The connection needs to be read only.**



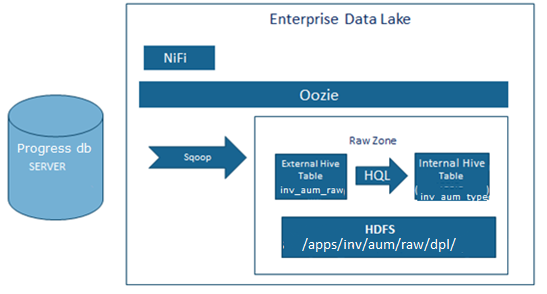
|  |  |  |  |
| --- | --- | --- | --- |
| DB\_IP | connPort | User | jdbc url |
| 10.101.96.30 | 1892 is MSISI | root | jdbc:datadirect:openedge://10.201.170.28:1892;databaseName=exc2 |
| 10.101.96.30 | 1692 is MSI | root | jdbc:datadirect:openedge://10.201.170.28:1692;databaseName=exc2 |
| 10.101.96.30 | 1693 is SM uat | progress | jdbc:datadirect:openedge://10.201.170.28:1693;databaseName=exc2sm |
|  |  |  |  |

Always use --relaxed-isolation on SQOOP

sqoop-import --connect 'jdbc:datadirect:openedge:// 10.101.96.30:1892;databaseName=exc2' --username root -P --driver 'com.ddtek.jdbc.openedge.OpenEdgeDriver' --query "SELECT a.\*,'MSI' as DEALER\_CODE FROM PUB.gngnco a WHERE \$CONDITIONS" --split-by 'fst\_bal' --enclosed-by '"' --target-dir '/tmp/gngnco/' --relaxed-isolation

Typed zone ddl can be found [here](https://git.platform.manulife.io/sharvij/aum_dpl/blob/Development/dpl_all_ddl/typed_ddl.hql) on gitlab

# Logical Structure



# Capacity and Performance Design

The projected size of the datasets is listed below:

* <dataset1> xxx MB/GB
* <dataset2> xxx MB/GB

# Test Strategy

Testing of this feed will be accomplished by acquiring representative files of test data from the source system and loading them into the EDL, where they will be verified.

Exception processing will be verified by manipulating the input files to trigger exception conditions and the associated error handling.

# References

*Provide a list of all documents and other sources of information referenced in the System Design Description and utilized in developing the System Design Description. Include for each the document number, title, date, and author.*

| Document No. | Document Title | Date | Author |
| --- | --- | --- | --- |
| 1 | [EDL Ingestion Patterns](https://mfc.sharepoint.com/:w:/r/sites/pm/PMO/GoDataStreamProgram/GoDtaStrmDtaWreHouse/Design%20and%20Development/Enterprise%20Data%20Lake/EDL%20Ingestion%20Patterns.docx?d=w694e8e8d31d24f91bb5831d71e3a6708&csf=1&e=9jC3hs) | 1/9/2017 | Nocera, Rob |
| 2 | [EDL-GO Data Warehouse - Solution Architecture](https://mfc.sharepoint.com/:w:/r/sites/pm/PMO/GoDataStreamProgram/GoDtaStrmDtaWreHouse/Design%20and%20Development/Enterprise%20Data%20Lake/Investment%20Division%20Data%20Lake%20-%20Technology%20Solution%20Architecture.docx?d=w9f297d0a7fab4d5ea3346d3ae46dbeea&csf=1&e=KhYs1p) | 12/07/2016 | Dascalu, Alexandra |
| 3 | [Edl-user-guide](http://mfcglobalinv/pm/PMO/GoDataStreamProgram/GoDtaStrmDtaWreHouse/Design%20and%20Development/Enterprise%20Data%20Lake/edl-user-guide.docx) | 11/10/2016 | Kang, James |
| 4 | [Manulife Enterprise Data Lake - Workstation Environment](http://mfcglobalinv/pm/PMO/GoDataStreamProgram/GoDtaStrmDtaWreHouse/Design%20and%20Development/Enterprise%20Data%20Lake/Manulife%20Enterprise%20Data%20Lake%20-%20Workstation%20Environment.docx) | 11/10/2016 | Nocera, Rob |