International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

Data Warehouse Extracts for Connected Corporate Banking

Jagadish Sri Kumar K[1], Alok Kumar[2], Lalitha V P[3], Mahesh A[4]Fo

[1]*Bachelor Student, Department of Computer Science and Engineering, R V College of Engineering, Bengaluru- 59*
[2] *Bachelor Student, Department of electronics and communication engineering, R V College of Engineering Bengaluru-59*
[3]*Assistant Professor, Department of Computer Science and Engineering, RV College of Engineering, Bengaluru – 59*
[4]*Associate Professor, Department of Electronics and Communication Engineering, RV College of Engineering, Bengaluru – 59*

Abstract - *The Data Warehouse Extracts is a solution that uses the evergreening approach for the Finastra hosted solution. The customer can send the extracts to its data warehouse system for various purposes such as reporting, reconciliation. The attributes of the extract solution can be customized according to customer implementation. If any changes are applied to a product, the Data Warehouse Extracts solution continues to support the extract as it is. A product can have multiple business objects (BOs) at the database level or at the logical level. For each business object a set of attributes is associated, and extracts are generated. Only one extract file is applied to a business object.*

***Key Words*: *Data Warehouse Extracts, Streamset, Pipelines*, Manifest, Data Dictionary.**

1. INTRODUCTION

The extract files are sent from Finastra subscribers through the ETL/API (extract transform and load) to the Data Landing Area. The ETL used for the solution is StreamSets.

The Data Landing Area uses the evergreening pipelines to transform the extracts into output-extract format. Where needed, certain rules are defined to support the transformations in the evergreening pipelines. The extract files that are processed successfully by the evergreening pipelines are sent to the Processed Area. In case an error appears when processing them, then they are sent to the Error Area.

The extracts are sent to the Data Extraction Area and, when ready to be consumed, a notification message is sent through the Transport Layer Security (TLS) or through the Secure Sockets Layer (SSL) to the Notification Handler. When the notification is received, the bank pulls the extract files through SFTP.

2. SYSTEM DEVELOPMENT

2.1 Methodology adopted

2.1.1 Receiving Extracts from Finastra Subscribers: -

The following artifacts are sent from the Finastra subscribers to the Data Landing Area.

• Data dictionary – indicates the metadata of the extract.

• Data extract – indicates the ETL extract of business objects.

• Manifest file – indicates when the ETL has finished processing.

2.1.2 About Data Dictionary:

The artifact represents the metadata of the extract and it is required to understand and process an extract. The data dictionary is defined for each business object.

The evergreening solution uses data dictionaries to populate control files for the extract.

It is recommended to use the JSON format from each product. For more information, see RFC 7159, The JavaScript Object Notation (JSON) Data Interchange Format. In addition, it is recommended to use a separate data dictionary JSON file for each business object.

The length of data types is present in the data dictionary of the business object. Certain data types have the length defined in the relational data model, but others can be determined based on the conversion used. For example, all timestamps are extracted as ISO 8601 string meaning that the length is determined based on it. In addition, if a decimal number is encountered, the scale is also required.

2.1.3 Data Dictionary Structure:

The data dictionaries for each of the product extracts are stored in two folders:

- ReleaseDD
- COMRLENDING
- COREBANKING
- FSSO
- MPM
- BaselineDD
- COMRLENDING
- COREBANKING
- FSSO
- MPM

The latest data dictionaries are placed in the ReleaseDD folder.

The baseline version of the product (for example, the one the Data Warehouse Extracts folder is expected to be as the evergreened version) contains the working version of the data dictionary.

The baseline versions are determining the version of the data dictionaries that the bank needs to evergreen to as a baseline.

The release versions are determining the latest versions of the data dictionaries.

Note: The same folder structure must be followed when configuring the DDComparator. For more information, see the Starting the Data Dictionary Comparator Tool section.

2.1.4 About Data Extract:

The artifact contains a snapshot of the actual data generated for each business object. There are two types of data extracts generated from each Finastra product:

• Daily incremental or delta extract – the extraction process selects all the records that have been created and/or updated between last successful run date and current date for a business object. The last successful run date is the date when previous daily incremental extraction process has been successfully executed.

• Full-snapshot extract – can be generated at any time and it provides a complete list of records as they are present at the time of the generation.

It is recommended to use the CSV format for input data extracts. For more information, see RFC 4180, Common Format and MIME Type for Comma-Separated Values (CSV) Files. Certain special characters are supported in the CSV format. For more information on the list of characters, see RFC 2234, Augmented BNF for Syntax Specifications: ABNF.

It is important that the first row is the header row.

The DATETIME and TIMESTAMP data types must be written in the ISO 8601 format. For example, YYYY-MM-DD HH:MM:SS.

The DATE data types must be written in the ISO 8601 format. For example, YYYY-MM-DD.

2.1.5 About the Manifest File:

The artifact indicates the ETL completion for an upstream product. When the manifest file is displayed, it triggers the input data extract to be processed through the Evergreening Pipeline. For more information, see the Triggering the Evergreening Pipelines section.

The manifest file must contain the information below:

• The name of the product and the version.

• The type of extract.

• The status.

• The selection criteria, if any is used for the extract.

2.1.6 Manifest File Structure:

The structure of the manifest file is the following:

product: {product_name}

version: {releaseVersion}

extractDate: {YYYY-MM-DD hh:mm:ss}

typeOfExtract: {Extract_type}

status: {status_of_process e.g. SUCCESS or, FAILURE}

locationOfExtract:
{folder_location_where_extracts_are_created}

NumberOfFilesCreated:
{total_count_of_extract_files_created}

selectionCriteria1: {name}::{value}

selectionCriteria2: {name}::{value}

.

.

.

selectionCriteriaN: {name}::{value}

Note: Extracts from each upstream product need to be regenerated in case any errors are run into.

2.1.7 Successful Status for the Manifest File:

• The comma-delimited extract files and the manifest file are copied in the {processedArea.root}/{product-name}/{zone}/{unique date-time} folder.

• The files remain in the landing area until the JMS message is sent to FFC notification. After that, the landing area is cleaned up and the files are moved to the processed area.

2.1.8 Failure Status for the Manifest File:

The comma-delimited extract files and the manifest file are copied in the {errorArea.root}/{product-name}/{zone}/{unique date-time} folder.

Note: The files are not removed for Fusion Essence core and Fusion Essence transaction extracts when the manifest file status is displayed as failure.

If there is inconsistency on the product version, then the comma-delimited extract files and the manifest file are copied in the {errorArea.root}/{product-name}/{zone}/{unique date-time} folder.

2.2 Experimental Details

2.2.1 High Level Design:

There are 4 steps involved in this project:

1. Pipeline Generator
2. Extract Listener
3. DD Comparator
4. Notification

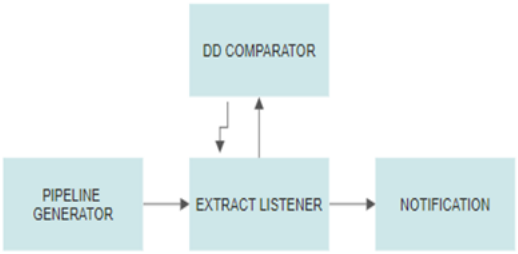

Fig -1: Block Diagram of Data Warehouse Extracts System

Each step plays a major role in this project, the roles played by each of the steps is discussed in low level design below.

2.2.2 Low Level Design:

I. Pipeline Generator:

The main objective of the pipeline generator is to create the suitable pipelines for the files. This is achieved by using the REST services provided by the Streamsets i.e., we have to create a JSON file which has the details of the pipeline which should be created for a particular extract and give this JSON file to the Streamsets so that it automatically creates the pipeline as mentioned in the JSON file. The pipeline generator should run even when the status in the manifest file is 'Failure' or 'Success'.

II. Extract Listener:

The main work of the extract listener is to check whether any new files have come in landing area (i.e. the input location) or not and if new files have landed it should be fed to the suitable pipeline and the resulting file should be sanitized and stored in the processed area. The extract listener should identify the extracts from which product it has been generated and It also checks whether the product has been evergreened or not. If the product has not been evergreened it will call the DD comparator.

III. DD Comparator:

The job of the DD comparator is to check whether the latest released version of the product is compatible with the old product's pipeline or not. It will compare the Data Dictionary of the both the latest and old product and tell the user to make the required changes in the pipeline to make it compatible with the latest product files.

IV. Notification:

The job of the notification is to notify the end users (Banks) that the extracts have been processed and has been received. The notification also zips all the extract of each products and encrypts using asymmetric key algorithms such as AES, Two Fish, Triple DES.

3. CONCLUSION

In this paper we have demonstrated how our organization have implemented the data warehousing system. The Data Warehouse Extracts is a Back-end project which serves to automate the process of Data warehousing of the data/extracts related to corporate banking. The Streamset tool provides various operations that can be performed on the data. The data could be of any type (i.e., .csv, .txt, whole file etc.). The Job of pipeline generator is to generate pipleines in Streamset for each extract according to the configurations made in the database. Then the extract listener watches over the directory for each of the extracts (Landing Area), if it encounters any new extracts it will trigger the running of pipeline for that extract in streamset through RESTful services. The main job of Notification is to Notify the end users and to send the extracts in an encrypted zip file.

REFERENCES

[1] Ballard Chuck et al. Data Modeling Techniques for Data Warehousing. - San Jose: International Business Machines Corporation, 1998.

[2] Chaudhuri, S. and Dayal, U. 1997. An overview of data warehousing and OLAP technology. SIGMOD Records, 26, 1, pp.65-74, Mar. 1997.

[3] Golfarelli, M., Maio, D., and Rizzi, S. 1998. Conceptual Design of Data Warehouses from E/R Schema. Proceedings of the Annual Hawaii international Conference on System Sciences- January 1998. HICSS.

[4] Greenfield Larry The Data Warehousing Information Center. LGI Systems Incorporated, 2006. www.dwinfocenter.org/

[5] Jukic, N. Modeling Strategies and Alternatives for Data Warehousing Projects. Communications of the ACM 49, pp.83-88. 4, 2006.

[6] Kimball Ralph [et al.] The Data Warehouse Lifecycle Toolkit, New York: Wiley Computer Publishing, 1998.

[7] Kimball Ralph and Caserta Joe The Data Warehouse ETL Toolkit- Indianapolis: Wiley Publishing, Inc., 2004.

[8] Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., and Paraboschi, S., Designing data marts for data

warehouses. ACM transactions on software engineering and methodology. 10(4), October 2001.

[9] Gardner, S., Building the data warehouse. Communications of the ACM. 41(9), September 1998.

[10] Wang, Richard Y., and Strong, Diane M. Beyond accuracy: What data quality means to data consumers, Journal of Management Information Systems; Armonk; Spring 1996, 12 (4), pp. 5-34.

[11] Jarke, M., M. Lenzerini, Y. Vaasssiliou, P.Vassiliadis, "Fundamentals of Data Warehouse," 2000.

[12] Y. L. and Zhang, X. M., "A reliable strategy and design of architecture of ETL in data warehouse", Computer Engineering and Applications, Vol. 10, 172_174, 2005.

[13] B. K. Seah, "An application of a healthcare data warehouse system", Innovative Computing Technology (INTECH), 2013, pp. 269 - 273.