syncsort

DMX-h ETL Use Case Accelerator

Fraud Detection using Apache Kafka

Table of Contents

1 Introduction

One use case for Apache Kafka message queues is point-of-sale debit card fraud detection, which involves determining whether a debit card is being used to make unauthorized purchases or cash withdrawals based on transaction history and cardholder information. This use case accelerator implements fraud detection in DMX-h ETL by extracting debit card transaction messages from an Apache Kafka topic and joining them with the associated cardholder information. Potentially fraudulent transactions are loaded into another Kafka topic for further inspection, and non-fraudulent transactions are loaded into a Hive table.

DMX-h ETL use case accelerators are developed as standard ETL jobs and can be run on an edge node, single cluster node, or in a Spark/MapReduce cluster without making any changes to the application.

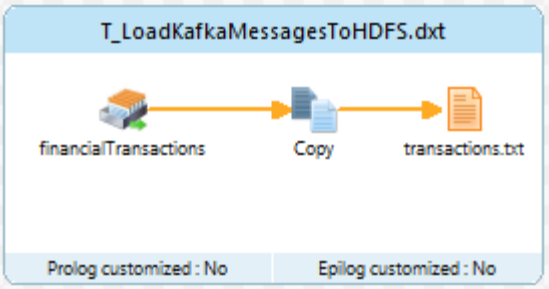
For guidance on setting up and running this and other use case accelerators, see the Guide to DMX-h ETL Use Case Accelerators. Note that the Guide details additional Windows setup that must be done prior to working with this UCA in DMExpress.

2 Fraud Detection with DMX-h IX

The Fraud Detection solution in DMX-h ETL consists of a job J_FraudDetectionDemo.dxj, containing two subjobs. The first subjob uses a copy task to ingest Kafka messages (debit card transactions) into HDFS, and the second subjob flags the transaction as either valid or potentially fraudulent.

2.1 J_LoadKafkaMessagesToHDFS.dxj Subjob

This subjob consists of one copy task.

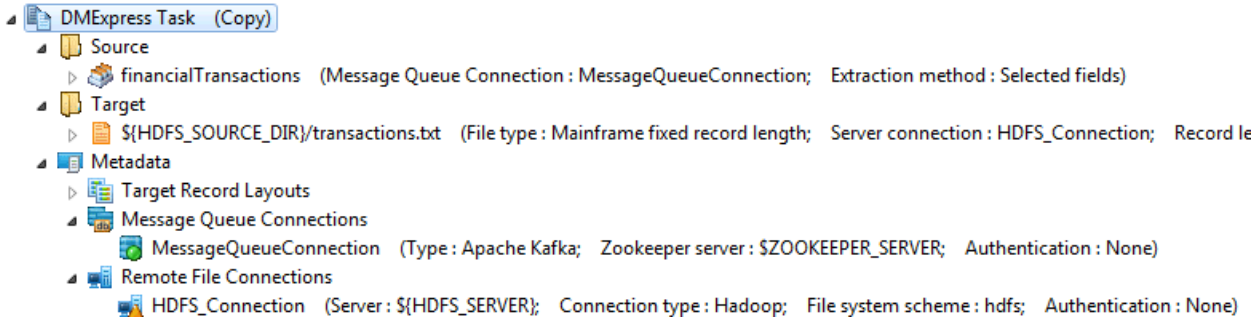


2.1.1 T_LoadKafkaMessagesToHDFS.dxt Task

This copy task reads debit card transactions from the Kafka topic “financialTransactions” and loads them into HDFS.

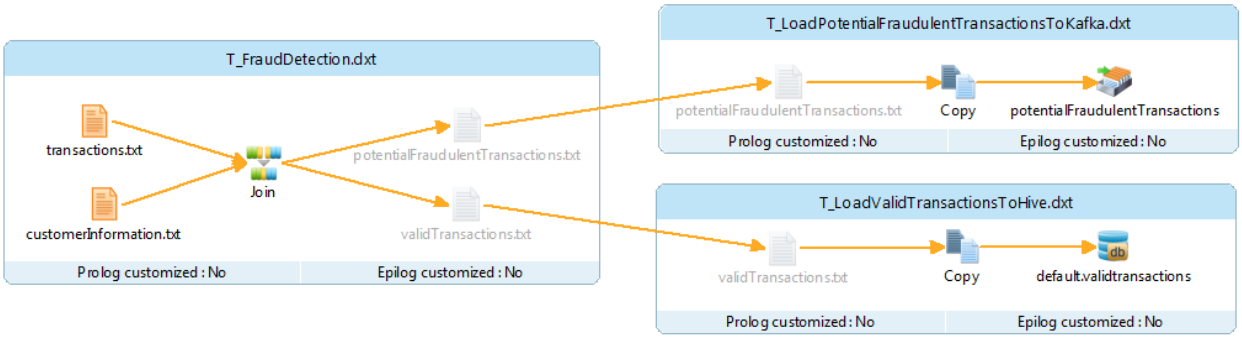
The batch size is 100 messages, and the group ID is set to allow Kafka to recognize which messages were already read by this consumer. If there are fewer than 100 messages, the job will wait until the queue has at least 100 messages. The financialTransactions topic has exactly 100 messages, so after running the job once, the specified group ID will have consumed all the messages and the queue will be empty. If you want to run the job again, you will need to repopulate the queue by re-running the prep script as described in the Guide to DMX-h ETL Use Case Accelerators.

The Kafka source is defined with a message queue connection, and the HDFS target file is defined with a remote file connection to the Hadoop cluster.



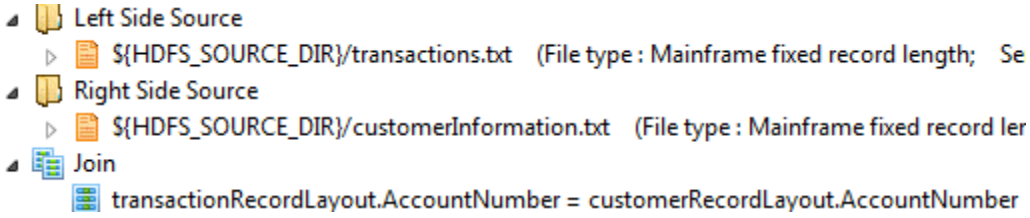
2.2 J_FraudDetection.dxj Subjob

This DMX-h IX subjob consists of one join task and two copy tasks. Direct data flow is defined between the join task targets and the two copy task sources.

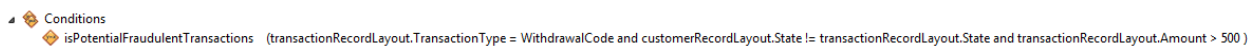


2.2.1 T_FraudDetection.dxt Task

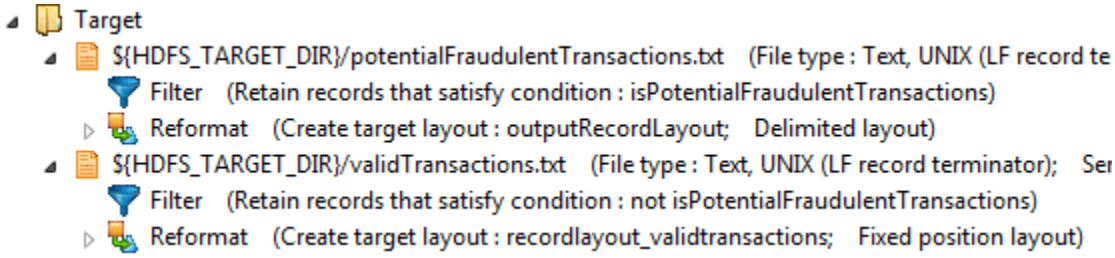
In this join task, debit card transactions in HDFS are joined with customer information on the “AccountNumber” field. Both the transaction and the customer information are in Mainframe fixed format. All the source and target files in this task are in HDFS.



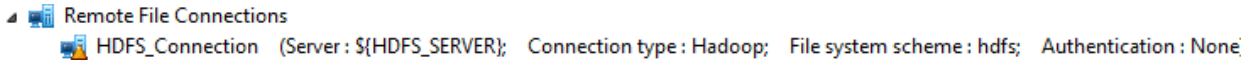
A transaction is flagged as potentially fraudulent if the amount withdrawn is more than $500 and the transaction is made in a different state from where the customer resides.



The potentially fraudulent transactions are output into one target, while the valid transactions are output into another target.

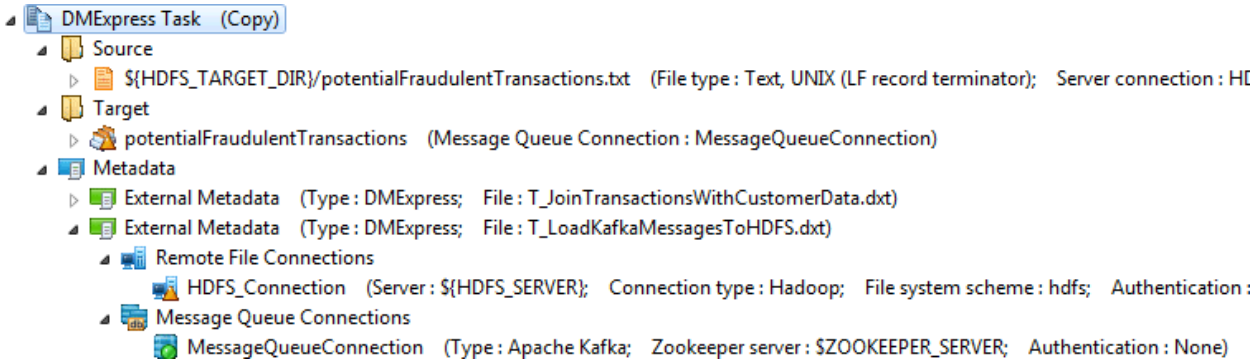


The HDFS source and target files are defined with a remote file connection to the Hadoop cluster.



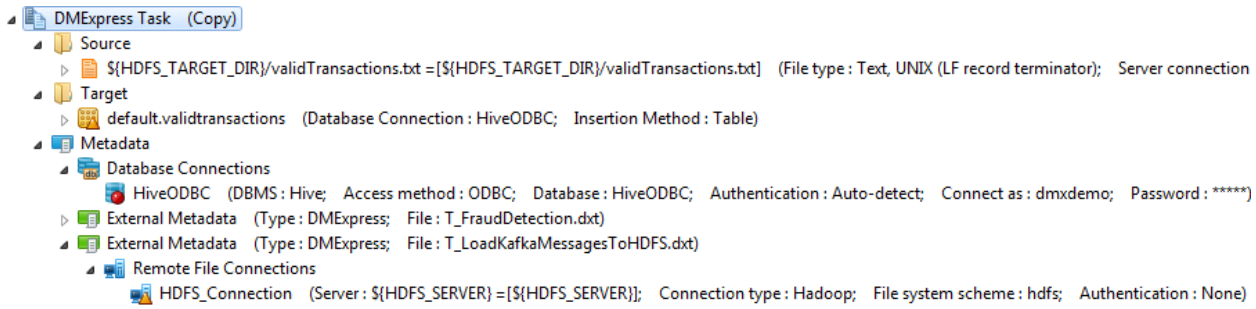
2.2.2 T_LoadPotentialFraudulentTransactionsToKafka.dxt Task

This copy task loads potentially fraudulent transactions into a Kafka topic. The HDFS source file is defined with a remote file connection to the Hadoop cluster and the Kafka target is defined with a message queue connection.



2.2.3 T_LoadValidTransactionsToHive.dxt Task

This copy task loads valid transactions into a Hive table. The HDFS source file is defined with a remote file connection to the Hadoop cluster and the Hive target is defined with a Hive ODBC connection.


DMExpress Task (Copy)
Source
${HDFS_TARGET_DIR}/validTransactions.txt =[${HDFS_TARGET_DIR}/validTransactions.txt] (File type : Text, UNIX (LF record terminator); Server connection
Target
default.validtransactions (Database Connection : HiveODBC; Insertion Method : Table)
Metadata
Database Connections
HiveODBC (DBMS : Hive; Access method : ODBC; Database : HiveODBC; Authentication : Auto-detect; Connect as : dmxdemo; Password : *****)
External Metadata (Type : DMExpress; File : T_FraudDetection.dxt)
External Metadata (Type : DMExpress; File : T_LoadKafkaMessagesToHDFS.dxt)
Remote File Connections
HDFS_Connection (Server : ${HDFS_SERVER} =[${HDFS_SERVER}]; Connection type : Hadoop; File system scheme : hdfs; Authentication : None)

About Syncsort

Syncsort provides enterprise software that allows organizations to collect, integrate, sort, and distribute more data in less time, with fewer resources and lower costs. Thousands of customers in more than 85 countries, including 87 of the Fortune 100 companies, use our fast and secure software to optimize and offload data processing workloads. Powering over 50% of the world's mainframes, Syncsort software provides specialized solutions spanning "Big Iron to Big Data", including next gen analytical platforms such as Hadoop, cloud, and Splunk. For more than 40 years, customers have turned to Syncsort's software and expertise to dramatically improve performance of their data processing environments, while reducing hardware and labor costs. Experience Syncsort at www.syncsort.com.

Syncsort Inc. ***50 Tice Boulevard, Suite 250, Woodcliff Lake, NJ 07677*** ***201.930.8200***