**LOAD PATTERNS – INFORMATICA WORKLFOWS**

[**DB2JMS**]

SEFS Interface Related Tables -> JMS Consumer

**Load Pattern**:

1. Source Table -> Filter (DTL\_\_CAPXACTION<>'D') -> Mapplet\* -> XML Generator -> Java Transformation -> Target Table [SEFS\_JMS]

Mapplet: Input -> Seq Gen -> LKP\_TIMEZONE\_OFFSET -> Expression -> Output

1. Source Table -> Filter (DTL\_\_CAPXACTION<>'D') -> Seq Gen -> Expression (Translate Timestamp) -> XML Generator -> Java Transformation -> Target Table [SEFS\_JMS]

Within each mapping many tables are being loaded, these are based on logical grouping.

Below are the workflows that are being executed:

wkf\_db2\_SEFS\_other\_to\_JMS

wkf\_db2\_SEFS\_ship\_base\_to\_JMS

wkf\_db2\_SEFS\_shipment\_to\_JMS

parameter file is used for setting either L2, L3, L4 etc.

Unix Server:

http://prh10071.freight.fedex.com (domain102\_prh10071 - PROD)

http://vrh10075.freight.fedex.com (QA102\_RAC\_02)

command: /home/ilogin/lbin/[fxfdw\_infa\_workflow\_monitor\_status\_html.sh](https://nam03.safelinks.protection.outlook.com/?url=https%3A%2F%2Furldefense.com%2Fv3%2F__http%3A%2F%2Ffxfdw_infa_workflow_monitor_status_html.sh__%3B!!IQRisXbPUUyI!t1vd-Kg4bh9Jsyt0fT-QJA9QRzhN4eR9iN9lF4-mo28yBhywChNtFvvAePmItsOd3AAFIw%24&data=02%7C01%7Cmohit.mathur.osv%40fedex.com%7C285578ba166940461d1f08d8427fc07b%7Cb945c813dce641f884575a12c2fe15bf%7C1%7C0%7C637332463739276773&sdata=eMcgmU4bkpFgeDUdb1j3KMl1W8kO31A9%2B4WrQ%2B7Kaw8%3D&reserved=0) -f <foldername>

The workflows are scheduled as Continuous.

--------------------------------------

**DB2STAGE**:

**Load Pattern**: (mpg\_driver)

Source Table -> Expression (for datatype conversions) -> Update Strategy -> Target (same as source)

mpg\_driver\_delay\_pay

Source (DRIVER\_DELAY\_PAY\_CDC) -> Sort -> Aggregate \* -> Update Strategy -> Target (DRIVER\_DELAY\_PAY)

Aggregator fetches the last row based on timestamp and does group by key columns.

mpg\_cust\_prof\_sl\_update\_from\_CDC

Source (CUST\_PROF\_SL\_CDC) -> Sort -> Aggregate \* -> Update Strategy -> Target (CUST\_PROF\_SL)

mpg\_ps\_aff\_eff\_driver

Compare DB2 table (PS\_AF\_EFF\_DRIVER) with Oracle Table-> Sort -> Expression (datatype conversion) -> Joiner -> Router -> Update Strategy -> Update/Ins/Delete on Target (PS\_AF\_EFF\_DRIVER)

[**DB2TAB**] - CDC process

wkf\_CDC\_DB2\_DSN\_LH

wkf\_CDC\_DB2\_DSN\_LH\_db2ffSL3

wkf\_CDC\_DB2CDC\_AR

wkf\_CDC\_DB2CDC\_BOR

Source: DB2 Table (CDC Real-Time Reader for DB2zOS)

Target: \_CDC Table (Oracle)

**Load Pattern**:

Source -> Expression (timestamp conversion) –> Seq gen -> Update Strategy -> Target

DB2 table is registered on PowerExchange. DBA turn on the CDC flag on IBM DB2 database.

Informatica is able to extract only the changed records.

Notes: *Payroll data is sensitive. Nobody is given access to CDC data.*

*Not sure what qualifies is sensitive. Could be personal data, driver data maybe sensitive or safety regulation*

[**ADS**]

Overall flow: cdc table -> work table -> final

naming of mapping as mpg\_<target\_table>. Example: lh\_co\_sched

**Part A)** mpg\_lh\_co\_sched\_work

Source: \_CDC Table

Target: \_Work Table

**Load Pattern**:

source table -> sort -> Expression -> Aggregator\* -> use mapplet lkp for emp\_nbr -> tgt

Aggregator fetches the last row based on timestamp and does group by key columns.

mpg\_ar\_gl\_detail\_work

source table -> Filter -> Expression -> sort -> Aggregator\* -> tgt

**Part B)** mpg\_lh\_co\_sched (Work to main table)

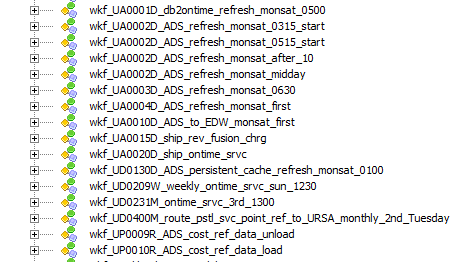
**Load Pattern**:

Source -> sort -> Joiner (old to new) -> router -> updates and insert on lh\_co\_sched Target table

mpg\_refresh\_context\_cdc\_XXX\_target\_XXX: This looks like a mapping to load audit table called REFRESH\_CONTEXT to track the session wise count of records being inserted, updated and deleted.

wkf\_UA0003D\_ADS\_refresh\_monsat\_0630

Workflow that run on continuous basis:



ADS small batch - Real time like every few minutes

ADS large batch - some hourly, once every 2 hours, once a day, twice a day (aggregations)

L0,L1,L2 --> Dev

L3,L4,L5,L6 --> QA

Prod\_DB --> Prod

Prod data brought to lower environments

Prod has 6 to 7 years of data (72-84 months)

QA has 1 year of data

Dev has 3 months to 1 year