3PS_ SUPPORT DOCUMENT

1) **Marketplace Creation: -**

Business has to create the marketplace i.e. “eBay” using MC in prepod stg. Prior to creating the marketplace make sure the consumer group and pricelist has been created. Once marketplace has been created successfully then two tables will get reflected in the STG DB schema only.

a. xmarketplace

this table contains the marketplaces name, marketplace id, status etc. For any job market places with status as ACTIVE only will be picked.

b. xmarketplace_extn

it contains username and password for one marketplace.

Both table data has to be available in the live schema as well. This data will be moved to live using the procedure and this procedure will be run after the feed transfer is completed.

2) **PriceLoad changes for Marketplace :-**

Business has to set up the product catalog for the respective marketplaces by adding the eligibility attribute Caas-<marketplacename> EG: Caas-eBay for eBay

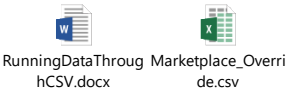
Firstly, price list for the marketplace i.e. eBay_PriceList needs to be created in the MC. When the price load runs, code has been added to check for the active marketplace and add the US general store list price to the respective marketplace pricelist. Only the products which has Caas-<marketplace> attribute will be assigned to the pricelist with the list price.

In the OfferPriceDataReader.java, the changes which are done are as follows:

1. The method getCaasAttrValueList(); is called in order to get all the skus which have attribute mapped for the marketplace.
2. The method getTradePoscnNames(); is called to get the price list created for the market place.
3. The method getPriceDetailsFromSAPSV(); is called to get the price details for HPSAPSV tables for that SKUs mapped to the market place.
4. The queries mentioned in these methods will be available in the pricecatalog.properties.
5. The constants of the query name will be available in the PriceConstants.java.

3) **Price Override: -**

Once the price list is populated by the price job for the respective marketplace, If business wants to override the price for a sku in the marketplace, we can run the marketplace price override job. Which will add price for the sku in the override table. When the market place pst job runs the price in override table will be preferred than the price in the marketplace pricelist. For the override job please create the CSV file, The details about the price override command is mentioned in below document.



Command to trigger priceoverride:
nohup sh dataload.sh /opt/apps/dataload/dataload/MarketPlaceOverRide/wc-dataload.xml -DdataloadParams="|pdhConfigPath=/opt/apps/PDHDataLoad/config/WCAdapter,locale=en_US,jobContext=HP-ROW.NA-G1" -DstoreIdentifier="HP_Global_CAS" -DcatalogIdentifier="HP_Global_CAS" &

4) MarketPlace PST Job: -

Once price load completed successfully then we will run MarketPlacePST Job, using below command:

nohup sh MarketPlacePSTJob.sh HP-ROW.NA-G1 AUTO &
Path:/opt/apps/PDHDataLoad/bin

For every PST job run a new FEED_ID per market place will be generated with the FEED_END_TIME & FEED_START_TIME as null value in "CA_CATALOG_FEED_DATA" table. This feed_id is a auto generated value which is max of feed_id+1. And it will put skus details in "CA_CATALOG_FEED_PROD_DATA" table for each feed_id. So we can check the sku and corresponding feed_id.

Below are the table names which is going to update with this job.

a. CA_CATALOG_FEED_DATA [Stage Database Table]
   This table contains the feed data.
b. CA_CATALOG_FEED_PROD_DATA [Stage Database Table]
   This table contains the sku data like price, eligible promotions and offset.
c. CA_CATALOG_FEED_SHIP_COST [Stage Database Table]
   This table contains the shipping cost for each sku.

Commented [HSP1]: Please explain the logic of feedid and all the matrix you had created what will happen.

After marketplace feed generation job the data will be propagated to LIVE schema. These tables are critical in terms of the latest price and products which will be validated when the order for CA reaches OMS.

5) Marketplace Data Extract and feed generation Job:-

Once Marketplace PST job completed successfully then only we need to run Marketplace Dataextract and feed generation Job, using below command:

nohup sh MarketPlaceDataExtarct.sh HP-ROW.NA-G1 AUTO TRUE TRUE &

Path: /opt/apps/PDHDataLoad/bin/

If MarketPlacePST Job failed then no need to run the Marketplace Data Extract and feed generation Job.

Here, first "TRUE" flag represent marketplace Data Extract job and second "TRUE" flag represent marketplace Feed generation job. If you want to disabled one job then you have to change the command using there two flags (TRUE/FALSE).

3PS_ SUPPORT DOCUMENT

If anytime we need to run the job explicitly / forcefully then we have to run the job using below command:

nohup sh MarketPlaceDataExtarct.sh HP-ROW.NA-G1 MANUAL TRUE TRUE &

Path:/opt/apps/PDHDataLoad/bin

If you want to run Marketplace Data Extract job, existing feed file sftp file transfer, procedure call, and read response:

nohup sh MarketPlaceDataExtarct.sh HP-ROW.NA-G1 MANUAL TRUE FALSE &

Path:/opt/apps/PDHDataLoad/bin

In feed generation we kept load_flag as “full” if anytime there is a chance of change the load_flag we can change it to “delta” we can update It in below file:

/opt/apps/PDHDataLoad/bin/MarketPlaceDataExtarct.sh

```
LOAD_FLAG="full"
```

To

```
LOAD_FLAG="delta"
```

Make sure on the first day it should be full only.

If we want to run the feed generation for particular marketplace then we have to update the file

we can change it to any marketplace_name available, if we want to run for multiple marketplace name then it should be comma separated value,

we can update It in below file:

/opt/apps/PDHDataLoad/bin/MarketPlaceDataExtarct.sh

```
OVERRIDEMARKETPLACENAME=
```

To

```
OVERRIDEMARKETPLACENAME=eBay,Amazon
```

Make sure on the first day it should be blank only.

If you want to run Marketplace Feed Generation job, feed file sftp file transfer, procedure call, and read response:

nohup sh MarketPlaceDataExtarct.sh HP-ROW.NA-G1 MANUAL FALSE TRUE &

Path:/opt/apps/PDHDataLoad/bin

Commented [HSP2]: Add the timestamp logic in the feed file, the timestamp in the response, and the response format. Also add the sftp details as well

Once feed generation completed it will create one file in below location /opt/apps/PDHDataLoad/workingDir/export/marketplace/<marketplaceName>/AddAndUpdate_<marketplaceName>_<LoadFlag>feed.txt

3PS_ SUPPORT DOCUMENT

Once file is created it will copy the file to this folder /opt/apps/data/wcsexport/marketplace/<marketplaceName>/AddAndUpdate_<marketplaceName>_<LoadFlag>feed.txt

Then it will rename the file as

/opt/apps/data/wcsexport/marketplace/<marketplaceName>/AddAndUpdate_<marketplaceName>_<LoadFlag>feed_MMDDYYYY.txt

It will post same file to sftp location and CA will post the response file in same sftp location with there timestamp value.

/accounts/12029961/Inventory/Transform/AddAndUpdate_<marketplaceName>_<LoadFlag>feed_MMDDYYYY_YYYY.MM.DD.HH.MN.SS.SSSS_RESPONSE.txt

SFTP Details:

TEST CA Account:

sftp://sftp.channeladvisor.com:22/
Username: hpetrfeed
Password: hpetrfeed1*M
Path: /accounts/12029961/Inventory/Transform

Production CA Account:

sftp://sftp.channeladvisor.com:22/
Username: hpetrfeed
Password: hpetrfeed1*M
Path: /accounts/12020735/Inventory/Transform

If you want to run sftp file transfer, procedure, and read response :

nohup sh MarketPlaceDataExtarct.sh HP-ROW.NA-G1 MANUAL FALSE FALSE &

Path:/opt/apps/PDHDataLoad/bin

It's just to transfer the existing feed file and call the procedure and read the response.

Firstly, the normal command will trigger the marketplace Data extract job once it's completed successfully then these tables will get reflected in ETL schema.

a. mkplace_export_product

b. mkplace_export_product_attr

c. mkplace_export_product_images

d. mkplace_product_dynatr_data

To check whether job completed successfully or not please check this table "mkplace_product_dynatr_data" in ETL schema. If this table contains any sku that means job Marketplace Data extract job completed successfully if it doesn't contains at least one sku that means marketplace data extract job failed.

3PS_ SUPPORT DOCUMENT

We Can check the logs in below location.

/opt/apps/IBM/WebSphere/CommerceServer80/logs/wc-dataextract.log

/opt/apps/IBM/WebSphere/CommerceServer80/logs/HPMarketPlaceDataExtract.log.<timestamp>.log

We Can check the logs in below location.

/opt/apps/PDHDataLoad/logs/ HPFeedGenerationJob.log

/opt/apps/PDHDataLoad/bin/nohup.out

Once you run the command it will call one procedure. And propagate table to LIVE db.

These two table will get reflected in LIVE db. (it will simply propagate the ETL table to LIVE table)

i. xmarketplace
ii. xmarketplace_extn

Commented [HSP3]: Which two tables

This is default procedure will be called in script so doesn't matter whether we are running data extract or not.

a. **PROPAGATE_XMARKETPLACE**
   iii. xmarketplace@TP_MKT_EDIT_LIVE [Propagate stage db to Live Database Table]
   iv. xmarketplace_extn@TP_MKT_EDIT_LIVE [Propagate stage db to Live Database Table]

Once MarketplaceDataextract job completed it will trigger Marketplace Feed generation automatically (based on command you have given)

Marketplace Feed generation job will create one folder (based on the marketplace name) and one feed file (in created folder) avail in this location /opt/apps/PDHDataLoad/workingDir/export/marketplace/

Once this file has been created it will move this file to /opt/apps/data/wcsexport/ folder and it will update the file with today's date and transfer it to CA Location configured.

Once it is completed it will read the response and send it to id's which is configured in hp_batchjobparams table avail in WCS db for this job. Then it will update in "CA_CATALOG_FEED_DATA" table avail in WCS db.

It will update the starttimestamp of max feed_id and endtimestamp of max-1 feed_id present in this table for active marketplace.

Commented [HSP4]: Put the complete feed id logic from the design doc

For every PST job run a new FEED_ID per market place will be generated with the FEED_END_TIME & FEED_START_TIME as null value in "CA_CATALOG_FEED_DATA" table.

3PS_ SUPPORT DOCUMENT

FEED_START_TIME will be populated after the marketplace catalog feed is successfully transferred

to the CA. FEED_START_TIME will be current feed transfer time + Buffer time.

FEED_END_TIME for the latest FEED_ID will be null.

Ex: FEED2 →FEED_START_TIME=5/3/2017 11.20:33

FEED2 →FEED_END_TIME=null

FEED1→FEED_START_TIME=5/2/2017 10:50:33

FEED1→FEED_END_TIME=5/3/2017 11.20:33

After this it will call two procedures listed below. It will reflect data in STG db, LIVE db and OMS db. The main table which we need to check in oms side is "EXTN_CA_MASK_INV".

All reflected table are mentioned below

Commented [HSP5]: Add the table name here in the oms db and what does it do

b. PROPAGATE_CATALOG_FEED

I. CA_CATALOG_FEED_LOG [Stage Database Table]
This contains logs for steps completed in the procedures.
II. CA_CATALOG_FEED_DATA@TP_MKT_EDIT_LIVE [Live Database Table]
This table contains the feed data.
III. CA_CATALOG_FEED_PROD_DATA@TP_MKT_EDIT_LIVE [Live Database Table]
This table contains the sku data like price, eligible promotions and offset.
IV. CA_CATALOG_FEED_SHIP_COST@TP_MKT_EDIT_LIVE [Live Database Table]
This table contains the shipping cost for each sku.
V. EXTN_CA_MASK_INV@TP_MKT_EDIT_OMS [OMS Database Table]
This table contains the mask value, active_flag etc in OMS db.

c. ARCHIVE_CATALOG_FEED

I. CA_CATALOG_FEED_LOG [Stage Database Table]
This contains logs for steps completed in procedure.
II. CA_CATALOG_FEED_PROD_DATA_ARCH [Stage Database Table]
This is archival table for CA_CATALOG_FEED_PROD
III. CA_CATALOG_FEED_SHIP_COST_ARCH [Stage Database Table]
This is archival table for CA_CATALOG_FEED_SHIP_COST
IV. CA_CATALOG_FEED_DATA_ARCH [Stage Database Table]
This is archival table for CA_CATALOG_FEED_DATA
V. CA_CATALOG_FEED_PROD_DATA [Stage Database Table]
This table contains the sku data like price, eligible promotions and offset.
VI. CA_CATALOG_FEED_SHIP_COST [Stage Database Table]
This table contains the shipping cost for each sku.
VII. CA_CATALOG_FEED_DATA [Stage Database Table]
This table contains the feed data.
VIII. CA_CATALOG_FEED_PROD_DATA@TP_MKT_EDIT_LIVE [Live Database Table]
This table contains the sku data like price, eligible promotions and offset.
IX. CA_CATALOG_FEED_SHIP_COST@TP_MKT_EDIT_LIVE [Live Database Table]
This table contains the shipping cost for each sku.
X. CA_CATALOG_FEED_DATA@TP_MKT_EDIT_LIVE [Live Database Table]

This is archival table for CA_CATALOG_FEED_DATA

6) **ItemLoad Job:-**

No dependency for this job.
This is an existing Job. We are adding “PlantCode” attribute to the existing OMS xml feed. This job generates xml files in /opt/apps/DataExtract/Catalog/XML/Process this files will be pushed to OMS automatically. Below are the commands to run ItemLoad job.

nohup sh hp_deltaextract.sh HP-ROW.NA-G1 MANUAL &

a. Master Load: (In Case of US store): Run the below command for Master Load.
./hp_dataextract.sh /opt/apps/DataExtract/Catalog/DeltaExtract/wc-dataextract-master.xml -DstoreId="US_Store" -DcatalogId="HP_Global_CAS"

b. Delta Load: (In Case of US store): Run the below command for Delta Load.
./hp_deltaextract.sh

Commented [HSP6]: This job is an existing job. We need to mention here what is the delta we did here.