MITS Processing (Incoming Feeds)

Most providers, provides us (RentPath) data in MITS xml format.MITS is the standard format we use for any feed that updates an Apartment Guide or Rent.com listings.There are a few different versions of the format, but each version is only slightly different. The 4.1 version is the latest. We receive properties and listings data through providers in the following different way.

1) RentPath picks up the file from provider's FTP site.
2) Provider drops off the file on Rentpath FTP site.
3) RentPath receives the file via HTTP.

We can divide full process of loading data from providers into APTG in main two processing layers as below;
Processing Layers

1) Python Feeds Process
2) MITS Load Process into APTG

1. **Python Feeds Process (Upstream)**

The Python Feeds Processing Layer, preprocesses 47 different providers XML (MITS) files into four MITS standards. Basically it does data quality check and cleans up files.

Once the files are preprocessed by Python Feeds process, they are available for MITS Load process on an internal web server. The 4 files as below for each provider.

1) XML file
2) Date file
3) Listing file
4) Image file

The url to check if the exists are as below
Example: For AVALON feed
feeds-01.atl.primedia.com:5000/listing/avalon
feeds-01.atl.primedia.com:5000/image/avalon
feeds-01.atl.primedia.com:5000/date/avalon
feeds-01.atl.primedia.com:5000/xml/avalon

These files are available on the feeds box at the following location
Feeds box : feeds-01.atl.primedia.com
Location : /common/feeds/avalon

```
feeds@feeds-01 /common/feeds/avalon [qa.atl] $ ls -ltr
total 1076
-rw-r--r-- 1 feeds feeds 518232 May 25 07:01 avalon.xml.gz
-rw-r--r-- 1 feeds feeds 517869 May 25 07:01 avalon.listing.gz
-rw-r--r-- 1 feeds feeds     52 May 25 07:01 avalon.date.gz
-rw-r--r-- 1 feeds feeds  54253 May 25 07:03 avalon.image.gz
```

2. MITS Load Process into APTG

MITS Load Processing is done using shell/perl scripts which can be divided into few steps

1) Main call : mits_load.sh based on provider's schedule time.
2) Data load to staging tables using shell script call : data_load.sh
3) MITS data file load based on their MITS xml version using following perl scripts
   a) mits.pl
   b) mits2.pl
   c) mits3.pl
   d) mits41.pl

MITS data load processing which involved loading feeds file from provider into APTG takes place by calling script on

Batch server: **Db-batch-01-atl.primedia.com**

Script Name: mits_load.sh

Location: **/home/oracle/scripts/mits_load.sh**

Parameter: Needs to pass following 3 parameters:

ORACLE_SID=$1 (APTG)

ENVTYPE=$2 (PROD)

PROVIDER=$3 (EREI2)

Example:

```
db-batch-01.atl.primedia.com:/home/oracle/scripts>ps -ef | grep /bin/ksh | grep mits_load.sh
batch    19180 19179  0 09:45 ?        00:00:00 /bin/ksh /home/oracle/scripts/mits_load.sh aptg prod EREI2
```

Purpose: To process MITS xml files into APTG database for listings.

Process run Check:

The script does following to make sure only one process for each provider is running and loading data into APTG**.**

1) Check if another MITS load process for the same provider is running or not. If another process for the same provider is running then abort the existing process with message of “PROVIDER already running” and exit the script.
2) Check if check process ran fine or not. If check process exit status is not zero (0) It will remove any stale queue file for the provider if any exists.
3) Check process to run only 3 concurrent MITS process at a time.

Once above checks are done, Script data_load.sh is called with parameter to process MITS data,date and image files.

Process Load MITS data and MITS Image files.

Batch server: **Db-batch-01-atl.primedia.com**

Script Name: data_load.sh

Location: /home/oracle/scripts/data_load.sh

$SCRIPTDIR/data_load.sh $ORACLE_SID $ENVTYPE mits $PROVIDER | tee $TMPLOGFILE

Example:

$ORACLE_SID: APTG

$ENVTYPE: PROD

$LOADTYPE: mits

$PROVIDER: PSOLUTN

The script data_load.sh loads data into staging table using Oracle SQL Loader utility.The tables affected are properties. mits_data, properties. mits_image and properties. mits_refresh

The script does following task for each provider run.

1) Build control file on fly which is useful information about data load, format and rejection etc.
2) For each provider previously loaded data is wiped out and reloaded with current data.

Follow are the subroutines used to create control and load scripts

1) buildControlFile
2) getNumCols

Once data_load.sh is done, the parent script: mits_load.sh checks for any stale data from provider and deletes it.

Also following checks are done using procedure call for sanity check

1) **Procedure**: properties.p_smitschkfloorplanids to prevent a provider from running if all their floorplans ID'S changed. Since that wipes out any linked images for them.

2) **Procedure**: properties.p_mits_contactinfo_changes to ignore incoming null values for property. Due to the issues of properties constantly dropping in/out from the mits files.

3) Procedure: properties.p_smitsdeleteavailability to delete any application/availability urls for properties that aren't in the latest xml file and rebuild their property service entries to get rid of the check availability functionality for any mapped listings. If the properties that aren't in latest xml file for provider (the property used to be in xml but now isn't there) it's not going to be processed by the mits.pl scripts. By deleting them we are turning of their check availability functionality for those properties.

Once all sanity check done for data following Perl scripts are run to parse xml file and load data into APTG schema.

Script Name: mits.pl
Location: /home/oracle/scripts/mits.pl
Purpose: Loads MITS version 1 xml file for provider

Script Name: mits2.pl
Location: /home/oracle/scripts/mits2.pl
Purpose: Loads MITS version 2 xml file data for provider.
Script Name: mits3.pl
Location: /home/oracle/scripts/mits3.pl
Purpose: Loads MITS version 3 xml file data for provider.
Script Name: mits41.pl
Location: /home/oracle/scripts/mits41.pl

Purpose: Loads MITS version 4 xml file data for provider.

Above Perl scripts takes xml file, breaks each property node out using Twig, extracts the primary id field from the xml, queries the ag database for property whose external_id matches the id. If ag property with external_id exists, retrieves the current ag xml for that property and compares it to the mits xml for the property. Updates values in many areas based on the flag setting for these fields in the listing_details_source table: