# Display_Ads_Finance_Stacked_URR

**Contents**

# 1 Overview

Stacked URR is a consolidated dataset that provides a holistic view of all Display Data. It is composed of the following data sources:

| Source | Details |
|---|---|
| URR | Booked Data from Operative for Managed Display |
| Salesforce | Pipeline Data from Salesforce (these are campaigns that are not yet contracted) |

| | |
|---|---|
| DSP-SS | Delivered data on DSP-Self Service Product. |
| Salesforce-DSP-SS | Pipeline Data pertain to DSP-SS Campaigns |
| ABNB Adjustment | Campaigns that were missed in our regular billing cycle (BAR) |
| SDP | Single Display Product Revenue |
| Cognos Adjustment | Allocation that ties the data to Cognos |

The Stacked URR is available on a restricted basis (via Hoot). Please connect with Jeff Lu (jefferlu@) for subscribing to this table.An extension of the Stacked URR is called the Total Ads Snapshot which contains "AMS" and "AMS-SELLER" as the data source for Sponsored Ad. To enable ease of reporting and support visualization tools, we have both tables saved in two formats (normalized vs denormalized). Stacked Transform jobs denormalized all source data in one uniform format.

Normalized Table Set (All normalized dataset resides in Report schema under Disad Cluster). The intend of these tables are for Visualization such as Tableau or Quicksight.

1. report.d_urr_stacked_normalized_snapshot
2. report.d_urr_stacked_product_normalized_snapshot
3. report.total_ads_normalized_snapshot

Marketplace Coverage

Currently URR/Total Ads covered Marketplaces as below

- US - 1
- UK - 3
- DE - 4
- FR - 5
- JP - 6
- CA - 7
- IT - 35691
- ES - 44551
- IN - 44571
- MX - 771770
- AU - 111172

**For any new request please create a SIM using link**

# 2 High Level Flowchart

# 3 URR Booked Revenue (Booked Revenue)

Revenue comes in as booked and delivered. For booked revenue, the upstream query in ADW prorated each line proposal based on proposal start date and end date so revenue can be aggregated into a monthly format.

Booked revenue is derived from Operative contracted amount. Each line proposal has reservation and approval status that are used as include and exclude status for the revenue line. When the month ends and the billable revenue is available in ADW fact_monthly_ad_reconciled_revenue table , the transform will flip the revenue from booked to billed. This process is driven by an internal URR system table with pre-defined month end date (m_urr_snapshot_filters). In addition to reporting booked and billed revenue, the URR process incorporates a revenue smoothing process by calculating remaining contracted amount based on what has been billed and smooths it evenly for the remaining months.

Tranform Job – O_URR_SNAPSHOT, REVENUE_SMOOTHING

## 3.1 Property mapping

Finance uses property name and rollup as one of the key dimensions to measure financial performance based on products. The property information is derived from Yield Team Product Mapping table. The BI data engineer team maintain an internal mapping table that uses platform, product mapping, site and program from yield product mapping for property name and rollup.  This process replaces the original manual mapping process which is error prone. The new process allows complete automation and unblock daily URR revenue report. Finance URR process utilizes revenue share of each product to segregate revenue between two different properties. This process applies a revenue split in the o_urr_snapshot process, the very first transform job that includes property names mapping.

# 4 DSP-SS (Delivered Revenue)

Display Finance data engineer team brought in DSP-SS delivered revenue as a daily grain. The daily data is brought in via ADW Query Tool export into S3. The delivered revenue is then aggregated into a monthly grain during URR transform process. The daily grain is available for users to do deep dive when necessary.

### 4.1 *DSP-SS Advertiser Id -> SF Account Mapping*

A mapping table M_ESS_MAPPING provides advertiser Ids to sf account id mapping on DSP-SS dataset. This allows DSP-SS data to have similar format as URR booked revenue and also all salesforce attributes.

Starting July 2018, Display Finance team started providing daily display managed + DSP-SS revenue to CP tool team. China and India delivered Revenue have been included in the new data infrastructure due to CP requirement.

Transform Job - DSP_SS_INFO

# 5 Salesforce Pipeline

Salesforce campaign information such as total price, probability, campaign start & end date and property names got loaded into Disad cluster on a daily basis. The URR process first mapped the SF Property names to Finance property so data can be rollup with the same grain as URR booked. In addition to SF property name mappings.

## 5.1 Exception Process

The process prorated total price into daily SF pipeline after the internal exception process. The exception process joins URR booked revenue to check if any 100% probability campaigns are not in URR. If they don't exist, the process will update the probability to 99% which allows the SF revenue to be remained in URR reporting. If it already existed in URR, the pipeline will not be included. In addition to URR book revenue matching, there are total 19 different exception conditions that the process handled and it updates the probability of the campaign accordingly based on each scenario. Please see attached exception logic for detail information. The updated probability is baked in the SF daily revenue proration to ensure all SF pipelines are evaluated before any reporting.

Transform Job – O_SF_PIPELINE_SNAPSHOT, DAILY_SF_ALLOCATION, DAILY_EXCEPTIONS

Pipeline Logic copy.pdf

# 6 SDP Revenue

Clicks, impressions, revenue and attributed OPS for advertisers are brought in to Finance cluster through ACH cluster. The SDP revenue is expected to be part of Stacked URR by end of Jan and early Feb 2019. The initial data are brought in as normalized format and then transformed to match Stacked URR format.

Transform Job – OFF_SITE_PERFORMANCE, OFFSITE_PERFORMANCE_PAID, ONSITE_PERFORMANCE, SYNDICATION_BETA

## 6.1 SDP Seller Id -> SF Account

SDP sellers' account id  maps to sf_sellid_staging table sourced from Hercules. This allow additional sf account attributes when appends to Stacked and Total Ads snapshot

# 7 SSPA Revenue

To provide a complete view of total advertising, AMS vendor information has been sourced from Spektr provider. The information is at a daily grain and aggregated to monthly revenue

Transform Job – TOTAL_AMS

## 7.1 SSPA -> SF Account Id mapping

Hercules team has a table that updates daily to map vendor code to SF Account Id. This data has been brought in to Finance cluster daily to map AMS revenue to SF account information

# 8 ABNB Adjustment & COGNOS Adjustment

Each month end, Geo Finance lead uploads SIM billing adjustment to Finance cluster. This revenue is incorporated into Stacked URR as ABNB. The billing adjustment has proposal ids for URR and order Id for DSP-SS. The Stacked URR process allocates these SIM billing adjustments with respect to the weighted distribution of the original revenue by proposal id and order id. It appends the ABNB adjustment after the prorated distribution. This allocation occurs daily and will pick the very last upload file. In addition to ABNB, Stacked URR calculates the difference between the Finance URR revenue and COGNOS actuals. It spreads the differences by marketplace id, sales vertical & primary goal and appends them to Stacked URR so end user can identify the delta between COGNOS and Finance URR data.

Transform Job – ABNB PRODUCT STACKED

# 9 Total Ads Snapshot vs Product Stacked vs Stacked URR

Stacked URR - DISAD_DDL.D_URR_STACKED_SNAPSHOT

Product Stacked URR - DISAD_DDL.D_URR_STACKED_PRODUCT_SNAPSHOT

TOTAL_AD_SNAPSHOT - DISAD_DDL.TOTAL_ADS_SNAPSHOT

Below chart illustrated the difference of the core table existed in DISAD cluster to report revenue

| | Product Id | Product Grains | Vendor Code | Display Revenue | DSP Revenue | Salesforce Pipeline | AMS Revenue |
|---|---|---|---|---|---|---|---|
| Stacked URR | | | | X | X | X | |
| Product Stacked URR | X | X | | X | X | X | |
| TOTAL_AD_SNAPSHOT | | | X | X | X | X | X |

*Display Finance Bi Team has incorporated SSPA Revenue and Display Revenue in one table (total ads snapshot).*

*This report updates daily and provide a complete glance of total advertising financial performance in one single format. Below are data sources*

- *URR - Display Booked Revenue*

- *URR-ABNB - Display Booked Revenue Accrued but Not Booked*
- *AAP-SS - DSP SS Delivered Revenue*
- *AAP-SS-ABNB DSP-SS Accrued but Not Booked*
- *AMS - SSPA Vendor Revenue*
- *AMS - SSPA Seller Revenue*

Transform Job -AMS, TOTAL ADS

# 10 List of jobs used in Stacked URR and other areas

| **Area** | **Job Name** | **Schedule** | **ADW Jobs** | **Transform** | **Load** | **What does this query do?** | **Data Owner** |
|---|---|---|---|---|---|---|---|
| **URR** | w_urr_snapshot_xsm_cy_edx | Daily | | | 415021 | Raw manage display data from ADW for contracted and billable | ADW |
| URR_Nightly | w_urr_snapshot_xsm_cy | Daily | w_urr_snapshot_xsm_cy | 6242528 | 437616 | Main job, if doesn't run , run the hammer stone URR_Nightly | Yield |
| | w_urr_snapshot_xsm_ny | New Year Only | w_urr_snapshot_xsm_ny | 6939078 | 426431 | Raw manage display data from ADW for contracted and billable for 2019 data | Yield |
| | s_urr_snapshot_xsm_cy | Daily | | 6360455 | 427094 | De normalizing w_urr_snapshot_xsm | Disad |
| | s_urr_snapshot_xsm_ny | New Year Only | | 6939091 | 477034 | De normalizing w_urr_snapshot_xsm | Disad |
| | D_URR_EXCAHNGE_RATES | Daily | | 5913530 | 385741 | | BDT |
| | o_proposal_currency_xsm | Daily | | 6373689 | 427698 | Currency for each proposal id for CY and NY | Disad |
| | o_urr_snapshot | Daily | | 6391676 | 435208 | | Disad |
| | w_non_dfp_allocation | Daily | | 6436462 | 435223 | | Disad |
| | | | | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **SALESFORCE** | o_salesforce_opportunity | Daily | | 6434975 | 433141 | Raw data from sale force | Yield |
| | o_sf_pipeline_snapshot | Daily | | 6368115 | 427562 | snapshots of o_salesforce_opp | Disad |
| | o_sf_pipeline_exceptions | Daily | | 6474250 | 436542 | | Disad |
| | o_sf_pipeline_dly_alloc_exc | Daily | | 6460021 | 435190 | | Disad |
| | sf_account_staging | Daily | | | 446736 | | Hercules |
| | SF_BILLING_ENTITY__C | Daily | | | 382697 | | Hercules |
| | | | | | | | |
| | | | | | | | |
| **AAP-SS** | AAP_SS_INFO | Daily | AAP_SS_SNAPSHOT Query from ADW | 6094603 | 13071170 | | ADW |
| | | | | | | | |
| **Brand Performance** | D_PROP_OPTIMIZE_TYPE | Daily | | 5986624 | 392106 | 2018 Process | ADW |
| | DIM_PRODUCT_GOAL_MAPPING | Daily | dim_product_goal_mapping from ADW | 6972007 | 486259 | 2019 Process | ADW |
| | | | | | | | |
| | | | | | | | |
| **Endemic Non Endemic** | D_SF_ENDEMIC | Weekly | | 6324322 | 419925 | | Sales Analytics |
| | | | | | | | |
| | | | | | | | |
| **Stacked-URR** | d_urr_stacked_snapshot_fact | Daily | | 6460299 | | | Disad |
| | d_urr_stacked_depend | Daily | | | 437970 | | Disad |
| Retired- No need to run | d_urr_stacked_snapshot | Daily | | 6474562 | 436560 | | Disad |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Retired- No need to run | d_urr_stacked_snapshot with ABNB | Daily | | 6565960 | 442768 | | Disad |
| | | | | | | | |
| | | | | | | | |
| **Product Stacked** | d_urr_stacked_product_snapshot | Daily | | 6512875 | 417144 | | Disad |
| | d_urr_stacked_product_snapshot with ABNB | Daily | | 7123810 | 526064 | | Disad |
| | d_urr_stacked_snapshot | | | 7200110 | 442768 | | Disad |
| | | | | | | | |
| **DIM Vendor/ Seller** | asat_dw_v_dim_seller | Daily | | 6958170 | 480513 | | Hercules |
| | asat_dw_v_dim_vendor | Daily | | 6173548 | 409263 | | Hercules |
| | | | | | | | |
| | | | | | | | |
| **Total Ads Snapshot** | Total AMS Snapshot | Daily | | 6183067 | 410214 | | Hercules |
| | Total Ads Snapshot | Daily | | 6183167 | 408019 | total_ads_snapshot doesn't have URR.RETAIL_PRODUCT_LINE for aap-ss | Hercules |
| | | | | | | | |
| **CP Tool Allocation** | cp_tool_revenue_mcogs_alloc (Redshift) | | | 6338513 | 425305 | | Disad |
| | cp_tool_revenue_mcogs_alloc (Andes) | | | 6338513 | Andes | | Disad |
| | | | | | | | |
| **WBR Queries** | | | | | | | |
| | d_opportunity_campaign_fact | Weekly | | 6436558 | 433411 | | Disad |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | WBR Summary Before Adjustments | Weekly | | | 6436852 | | Disad |
| | WBR Summary | Weekly | | | 6517292 | | Disad |
| | | | | | | | |
| | | | | | | | |
| **SDP current tables** | advertiser_off_site_perf | Daily | | 7158646 | 538244 | | ACH |
| | advertiser_off_site_perf_paid | Daily | | 7157476 | 537916 | | ACH |
| | advertiser_on_site_perf | Daily | | 7157707 | 537997 | | ACH |
| | advertiser_syndication_beta | Daily | | 7156947 | 537852 | | ACH |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| **Head CT Data Jobs** | ww_ads_bis | Daily | | 7100297 | 524157 | | HRBI |
| | ww_ads_pending_starts | Daily | | 7120196 | 529419 | | HRBI |
| | ww_ads_open_reqs | Daily | | 7127664 | 529421 | | HRBI |
| | dim_job_hr | Daily | | 7191791 | 551009 | | MOSIAC |
| | direct_indirect_plan | Daily | | 7191804 | 551011 | | MOSIAC |
| | vw_dac_headcount | Daily | | 7191820 | 551010 | | MOSIAC |

# 11 Additional Processes

## 11.1 FX Rates (Prerequisites Phase - Step 1)

The FX rates process sources rate from O_FPA_EXCHANGE_RATES and O_EXCHANGE_RATES tables by calendar month. It utilizes the URR snapshot filters table (an internal table keep track of defaults and data parameters) to derive current report year by report type. For year 2017, the snapshot filters have the following year types and report years.

| REPORT_YEAR_TYPE | REPORT_YEAR |
|---|---|
| CY | 2018 |

| | |
|---|---|
| NY | 2019 |

The snapshot filter also specifics the following required currency that will be included in the exchange rate process.

| CURRENCY_CODE | EXCHANGE_RATE |
|---|---|
| CAD | Canadian Dollars |
| USD | US Dollars |
| EUR | Euros |
| JPY | Japanese Yen |
| GBP | Great British Pounds |
| INR | India Rupees |

The exchange rate process gets monthly rate from table O_FPA_EXCHANGE_RATES (monthly rate table). If the rates do not exist in this table, it will use the rate from O_EXCHANGE_RATES table. Since O_EXCHANGE_RATES is a daily rate table, average calculations are used to get the monthly rates. All calculations on O_EXCHANGE_RATES table are based on transaction with EXCHANGE_TYPE = ‘C’. The following logics have been used

- Calculate an average monthly rate by using sum of daily rates divided by the number of days for each month
- Include exchange rate = 1 when currency codes from both sides are equal
- Calculate forward rate by averaging the daily rates for the next 30 days (1-month period) and apply on months that have no rates from O_FPA_EXCHANGE_RATES or O_EXCHANGE_RATES.

Calculation used:

1. This process obtains rates from two different table. If the first table (O_FPA_EXCHANGE_RATES) does not carry the rate, it will use the rate from second table (O_EXCHANGE_RATES) and they will be averaged out by month.
2. When O_EXCHANGE_RATES (daily rates) are used, average monthly rates are calculated by sum of daily exchange rate/number of days for the month

| | Descriptions |
|---|---|
| Purposes | This process obtains exchange rates from exchange rate tables and put them into Finance data warehouse tables |
| Key Attributes | FROM CURRENCY CODE, TO CURRENCY CODE, FX RATES BY MONTH |
| Key Measures | FX Rates |
| Source Tables | M_URR_SNAPSHOT_FILTERS, O_PROPOSAL_CURRENCY_XSM, O_EXCHANGE_RATES, O_REPORTING_DAYS, O_FPA_EXCHANGE_RATES |
| Destination Table | D_URR_EXCHANGE_RATES |

| | |
|---|---|
| Calculations/Logic | Rates will be derived from O_EXCHANGE_RATES table if they do not exist in O_FPA_EXCHANGE_RATES |
| Schedule | SUN, THURS, SAT |
| Dependencies | See source tables |

Jeff Lu: The current query is taking a long time to run due to the calculation of average rate. Can this process run once a month?

# 12 URR Raw Data Generation (Data Transformation Phase)

## 12.1 URR Raw Data File Staging (Data Transformation Phase – Step 2)

This is a lift and shift process. Raw data is not transformed or re-calculated when importing from RedShift. All attributes remain the same and data is at the most granular level by product Id. No mapping has occurred in this process and measures are stored at a vertical level (by report year & report month). Most of the metadata (Advertiser info, agency, order, product name …etc.)  for the transactions are already available in the dataset. Thus, minimum lookup will be required downstream to get this information.  The current daily schedule only contains one snapshot in the table. There are no historic raw data at the table level but Datanet carries previous extract files up to certain date.

| | Descriptions |
|---|---|
| Purposes | Working table to house RedShift data in normalized format |
| Key Attributes | SNAPSHOT DATE, AD ID, ADVERTISER ID, AGENCY ID, XSM OR DSM, PRODUCT ID, LINE ITEM ID, REPORT YEAR, REPORT MONTH, SALESFORCE ID, CURRENCY CODE |
| Key Measures | CONTRACTED UNITS, CONTRACTED REVENUE, BILLABLE REVENUE |
| Source Tables | RedShift |
| Destination Table | W_URR_SNAPSHOT_XSM_CY,<br>W_URR_SNAPSHOT_XSM_NY |
| Calculations/Logic | N/A |
| Schedule | DAILY |
| Dependencies | RedShift EDX |

## 12.2 URR Raw Data Transformation “De-normalization” (Data Transformation Phase – Step 3)

This is a raw data transformation process ***without*** business rules applied from RedShift. The raw data from previous process W_URR_SNAPSHOT_XSM_CSY has vertical month column (normalized data). This process de-normalized the monthly data by flatten the source data into months across the table. It also excludes any line items that are considered archived by joining the table M_XSM_ARCHIVED _LINES. The monthly measures are as follow

- CONTRACTED_UNITS

- CONTRACTED_REVENUE
- ORIGINAL_RECORDED_UNITS
- FORECASTED_UNITS
- BILLABLE_UNITS
- BILLABLE_REVENUE
- CONTRACTED_REVENUE_GROSS

The table also included grand total of the year for the above measures. The data are being stored at a detail level by product Id and DSM or XSM so property mapping can be applied at the downstream level.

Logics Used: This query will check M_XSM_ARCHIVED_LINES to exclude Proposal Line Item that has a status of "Archived" (line 287 – 290)

| | Descriptions |
|---|---|
| Purposes | Staging table with aggregated data by months |
| Key Attributes | SNAPSHOT DATE, DSM OR XSM, ORDER NAME PROPOSAL NAME, ADVERTISER NAME, PRODUCT ID, AD ID, SALES PERSON, PROPOSAL ID, AD UNIT NAME, AGENCY NAME |
| Key Measures | CONTRACTED UNITS, CONTRACTED REVENUE, BILLABLE REVENUE BY MONTH |
| Source Tables | W_URR_SNAPSHOT_XSM_CY, W_URRSNAP_XSM_NY, M_XSM_ARCHIVED_LINES |
| Destination Table | S_URR_SNAPSHOT_XSM_CY, S S_URR_SNAPSHOT_XSM_NY |
| Calculations/Logic | Exclude archived line item |
| Schedule | DAILY |
| Dependencies | W_URR_SNAPSHOT_XSM_CY, W_URR_SNAPSHOT_XSM_NY |

# 12.3 Currency Proposal Mapping (Data Transformation Phase – Step 4)

The currency proposal mapping identifies the distinct currency code used for a particular XSM_PROPOSAL_ID or DSM_PROPOSAL_ID. Any DSM Proposal Id that ***contains*** non-numeric values will be discarded (null values are used) in this process. Only DSM Proposal Ids with 100% numeric value are retained.  XSM Proposal Ids are not being evaluated and are retained 100% of the time. The DataMart captures proposal Ids (XSM, DSM), currency code and snapshot date to represent the currency code associates to a particular proposal.

Calculation:  DSM Proposal Id evaluation

- Contains only numeric values -  DSM Proposal Id will be kept
- Contains alphanumeric values – DSM Proposal Id will be discarded. Blank value will be used
- Blank – Blank will be used

| | Descriptions |
|---|---|
| Purposes | Define XSM PROPOSAL or DSM PROPOSAL default currency code |
| Key Attributes | SNAPSHOT DATE, XSM PROPOSAL ID, DSM PROPOSAL ID CURRENCY CODE |
| Key Measures | N/A |
| Source Tables | W_URR_SNAPSHOT_XSM_CY ,<br>W_URR_SNAPSHOT_XSM_NY |
| Destination Table | O_PROPOSAL_CURRENCY_XSM |
| Calculations/Logic | N/A |
| Schedule | DAILY |
| Dependencies | See source tables |

| | Descriptions |
|---|---|
| Purposes | Upload property mapping into data warehouse |
| Key Attributes | Product Id |
| Key Measures | N/A |
| Source Tables | MUTP_YYYYMMDD.txt,<br>M_URRPRODUCTID_TO_PROPERTY_XSM |
| Destination Table | M_URRPRODUCTID_TO_PROPERTY_XSM |
| Calculations/Logic | N/A |
| Schedule | THURS, MONDAY (not automated) |
| Dependencies | None |

## 12.4 O_URR_SNAPSHOT

The first major transform that converts raw URR data with business logics and mappings. The below illustrations contains all the business logics involved in this transform.

The query includes the following logics for data transformation

Highlights

1. Apply Property Mapping to Raw URR dataset (URR Sub-Query)
    a. Raw data mapping joined by product Id and DSM or XSM field to obtain product mapping attributes such as product, country rollup, is AAP Mobile included and sf property name

2. Apply URR data filter to identify agencies that are applicable for commissions percentage. For non-agency related URR transactions, 100% will be applied (URR Sub-Query)
    a. Apply agency percentage to overall contracted/billable revenue when there is a country roll up matched

3. Apply URR data filter to identify product Id with billable = contract Amount for billable transactions (URR Sub-Query)
    a. For product Ids that are part of this filter, contracted amount will be used instead of billable amount for billable unit/revenue

4. Calculation of Mobile Delivered Mix based on Ad Id and Ad Proposal Start Date (Final Query)
    a. For all mapped URR transactions that have is aap mobile included = Y, a percentage breakout based on W_AAP_MOBILE_ALLOCATION will be used to segregate AAP mobile portion and non AAP mobile portion when Proposal Start Date + 14 days is less than or equal to URR generate date

5. Calculation of Opportunity Mix based on Ad Campaign and Ad Proposal Start Date (Final Query)
    a. For all mapped URR transactions that have is aap mobile included = Y, a percentage breakout based on O_SF_PIPELINE_SNAPSHOT will be used to segregate AAP mobile portion and non AAP mobile portion when Proposal Start Date + 14 days is greater than URR generate date

| | Descriptions |
|---|---|
| Purposes | Transforms raw URR data with property mapping, mobile mix, agency percentage, opportunity mix and billable amount as contract amount for certain product Ids |
| Key Attributes | PRODUCT ID, DSM OR XSM, AD ID, SALESFORCE ID, REPORT YEAR |
| Key Measures | CONTRACTED UNITS, CONTRACTED REVENUE, BILLABLE UNITS, BILLABLE REVENUE |
| Source Tables | S_URR_SNAPSHOT_XSM_NY,<br>S_URR_SNAPSHOT_XSM_CY,<br>M_URRPRODUCTID_TO_PROPERTY_XSM,<br>M_URR_SNAPSHOT_FILTERS,<br>W_AAP_MOBILE_ALLOCATION,<br>O_SF_PIPELINE_SNAPSHOT |
| Destination Table | O_URR_SNAPSHOT |
| Calculations/Logic | Refer to highlights of this section |

| | |
|---|---|
| Schedule | SUN, TUES, THURS, SAT |
| Dependencies | See source tables |

Exclusion

Snapshot Filters (URR_SNAPSHOT_FILTERS)

- Order name proposal name to exclude filter type = TEST_ORDER_EXCLUSION
- Advertiser name to exclude filter type = ADVERTISER_EXCLUSION
- Product Id to exclude filter type = INVALID_PRODUCT
- Ad Id to exclude filter type = INVALID_LINEITEM
- Sales person to exclude AE_LIST and EXCLUDE

# 13 Transform Mapped URR Data with DFP Allocation & Pipelines

## 13.1 The non DFP allocation uses URR snapshot filters to derive reporting parameter such as report year, report month year, recon date and month key

It only includes product with 'Include-SignedRevenue" status. The query also has the following filter from property mapping table.

| | VALUES |
|---|---|
| FLAT FEE | Yes |
| IS Double Click Service | No |
| Property Name Wildcard | Mobile<br>Ext<br>AAP<br>CPC<br>Kindle |
| Cost Type | CPC |

The main purpose of this query is to apply Year to Date delivery differences (Contracted Revenue – Billable Revenue) and divided it by the remaining calendar months (included the current month) of the year and spread it across by the same number of months. The differences (smooth revenue) will start on February (month key 03) and January differences will be used for the remaining 11 months. December will be the last month with smooth revenue effect by using YTD November numbers. 2 additional months (month key 01 and month key 14) will be included in the query without any smooth revenue effect. The query will only return data associated to a specific month key from the URR snapshot filter mentioned in the beginning of this section.

Examples of Ad ID 100001, Calendar Month = Feb, Month Key = 03

January Contracted Revenue = 1,000

January Billable Revenue = 500

February Contracted Revenue = 2,000

February Billable Revenue = 1,000

YTD Smooth Revenue: (1,000 – 500) + (2000 – 1000) = 1,500

Monthly Smooth Revenue: 1,500/11 = 136.36

| | Descriptions |
|---|---|
| Purposes | Applies smooth revenue (prorated revenue) at a campaign level based on billed and contracted revenue amount |
| Key Attributes | AD_ID, MONTH_KEY, REPORT_YEAR |
| Key Measures | CONTRACTED_REVENUE, BILLED_REVENUE |
| Source Tables | M_URR_SNAPSHOT_FILTERS, M_URRPRODUCTID_TO_PROPERTY_XSM, M_URR_REVENUE_SIGNED, O_URR_SNAPSHOT |
| Destination Table | W_NON_DFP_ALLOCATION_XSM |
| Calculations/Logic | Please refer to above logic |
| Schedule | MON, TUES, THURS, SAT |
| Dependencies | See source tables |

## 13.2 DPAM Property Mapping

The current property mapping is a manual process which is error prone and time consuming. The proposed plan is to use yield team’s DIM_PRODUCTID_ALLOCATION_MAP table which has already been mapped/cleansed and does not require financial analyst’s manual intervention. Although yield team’s product Id mapping table show a lot of similarities with Display Ad’s, there are differences between these tables. They are:

| | MUTP | DPAM |
|---|---|---|
| Number of rows | 17k+ | 5k+ |
| Report Group attribute | Yes | No |
| Is Double Click attribute | Yes | No |
| Product attribute | Yes | No |
| Placement name attribute | No | Yes |

| | | |
|---|---|---|
| Product classification attribute | No | Yes |
| Product Mapping attribute | No | Yes |
| Revenue Share attribute | No | Yes |
| Impression Share attribute | No | Yes |

To ensure there will be no downstream impact if new proposed property mappings are used, the following section will provide comparison between the two using URR raw data with legacy mappings (MUTP) and yield team’s mapping (DPAM) to find out the differences.