gTech
<professional services/>

Megalista

User Manual - v1.0

Customer Solutions
ENGINEERING

Customer Solutions
ENGINEERING

Implementation

1. Default implementation [Recommended]

Default implementation consists in creating a template of megalista hosted in a Google Cloud storage bucket, and configuring a Google Cloud Scheduler responsible to initialize Dataflow with the hosted template

Please visit the **code repository** with step-by-step on how to implement megalista

2. Running Apache beam locally

As an alternative, you can also run megalist locally (or in a VM) -- even in that case Google oAuth setup, Google APIs and BigQuery are still required. For that, you will be required to install apache beam, all megalista dependencies and use a local runner to execute the workflow

Customer Solutions
ENGINEERING

Updating

1. Redeploy Pipeline

Redeploy megalista pipeline into Google Cloud Storage according with the documentation

2. Review Cloud Scheduler Parameters

Newer Mega Lista versions might require additional scheduler parameters to support recently added connectors. Please review your current setup and latest supported parameters

Customer Solutions
ENGINEERING

Customer Solutions
ENGINEERING

Runtime Parameters

1. Concept

A few sensitive parameters (i.e. API tokens) need to be passed to Megalista at run-time. This can be accomplished in two ways:

- Defining a JSON payload on Google Cloud scheduler, in case you run megalista in Cloud Dataflow
- Passing them as shell arguments, in case you use apache beam localrun

2. The parameters

Key Name (Cloud scheduler)	Shell Argument (localrun)	Expected value
gcp_project_id	--gcp_project_id	Google Cloud project Id
client_id	--client_id	oAuth2 client Id for Google APIs
client_secret	--client_secret	oAuth2 client secret for Google APIs
refresh_token	--refresh_token	oAuth2 refresh token for Google APIs, obtained through generate_megalist_token.sh
access_token	--access_token	oAuth2 access token for Google APIs, obtained through generate_megalist_token.sh
setup_sheet_id	--setup_sheet_id	Id of Google Spreadsheet that will be used as configuration engine. Copy from here
developer_token	--developer_token	Google Ads developer token
bq_ops_dataset	--bq_ops_dataset	Auxiliary bigquery dataset name used for Megalista operations (dataset needs to exist prior to execution and be hosted in gcp_project_id project)
appsflyer_dev_key	--appsflyer_dev_key	Developer key for AppsFlyer server 2 server API

3. Example config - Cloud Scheduler

Please visit this link for details

Quick tip: if Cloud Scheduler doesn't work with "PERMISSION_DENIED" and Cloud Scheduler was enabled before March, 2019, it might be worth to check this link.

4. Example config - local runner

```
python3 -m main \
--runner DirectRunner \
--direct_num_workers 1 \
--project "-----" \
--client_id "----" \
--client_secret "-----" \
--refresh_token "------" \
--access_token "------" \
[…]
```

Customer Solutions
ENGINEERING

Google Cloud Access Requirements

Minimum Google Cloud Access Requirements

Those are the minimum roles necessary to deploy Megalista:

- OAuth Config Editor
- BigQuery User
- BigQuery Job User
- BigQuery Data Viewer
- Cloud Scheduler Admin
- Storage Admin
- Dataflow Admin
- Service Account Admin
- Logs Viewer
- Service Consumer

Customer Solutions
ENGINEERING

Enabling APIs in Google Cloud

APIs in Google Cloud

Google Sheets (required for any use case)

- https://console.cloud.google.com/apis/library/sheets.googleapis.com

Google Analytics

- https://pantheon.corp.google.com/apis/library/analytics.googleapis.com

Google Analytics Reporting

- https://pantheon.corp.google.com/apis/library/analyticsreporting.googleapis.com

Google Ads

- https://pantheon.corp.google.com/apis/library/googleads.googleapis.com

Campaign Manager

- https://pantheon.corp.google.com/apis/library/dfareporting.googleapis.com

Customer Solutions
ENGINEERING

Customer Solutions
ENGINEERING

Basic Configuration

1. Configuration Spreadsheet

All data upload rules are defined in a single configuration spreadsheet. Template here.

The trix is made of four tabs: Intro, Sources, Destinations, Connector

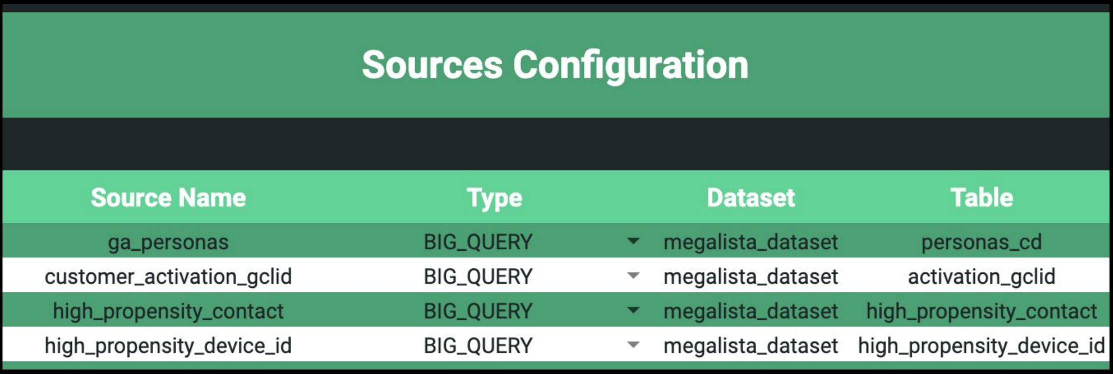
1.1 Intro Tab

Hold basic information common across upload rules

1.2 Sources tab

Holds the bigQuery location of each data source. **Important:** The schema of each bigQuery source table needs to perfectly match the expected schema of the destination type it will be used with:

- Source Name: A friendly name for that source
- Dataset: BigQuery dataset name
- Table: BigQuery table name



Sources Configuration

Source Name	Type	Dataset	Table
ga_personas	BIG_QUERY	megalista_dataset	personas_cd
customer_activation_gclid	BIG_QUERY	megalista_dataset	activation_gclid
high_propensity_contact	BIG_QUERY	megalista_dataset	high_propensity_contact
high_propensity_device_id	BIG_QUERY	megalista_dataset	high_propensity_device_id

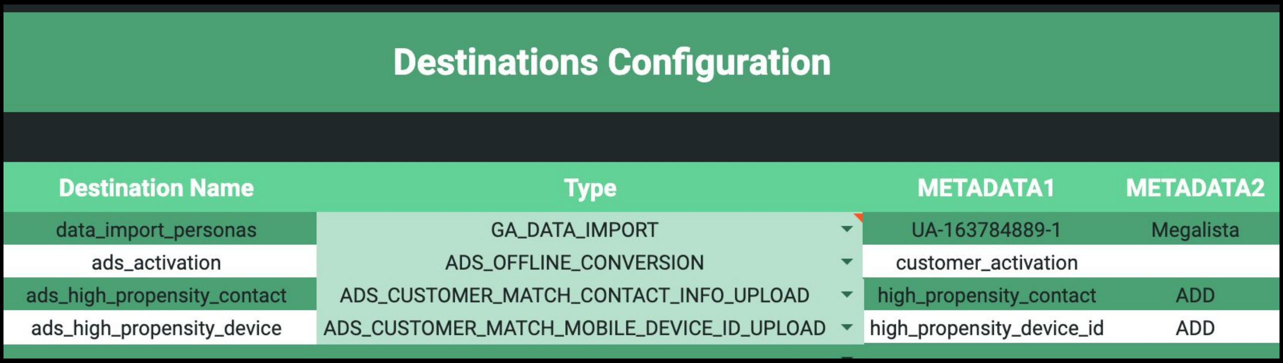
Customer Solutions
ENGINEERING

1. Configuration Spreadsheet

1.3 Destinations Tab

Defines where the data should be send to, according with destination type and expected metadata information

- **Destination Name:** Friendly name of each destination
- **Type:** the type of upload, limited to supported upload destinations of the solution
- **Metadata[1-6]:** Additional required informations expected by each destination type (details on the documentation of corresponding destination type)



Destinations Configuration

Destination Name	Type	METADATA1	METADATA2
data_import_personas	GA_DATA_IMPORT	UA-163784889-1	Megalista
ads_activation	ADS_OFFLINE_CONVERSION	customer_activation	
ads_high_propensity_contact	ADS_CUSTOMER_MATCH_CONTACT_INFO_UPLOAD	high_propensity_contact	ADD
ads_high_propensity_device	ADS_CUSTOMER_MATCH_MOBILE_DEVICE_ID_UPLOAD	high_propensity_device_id	ADD

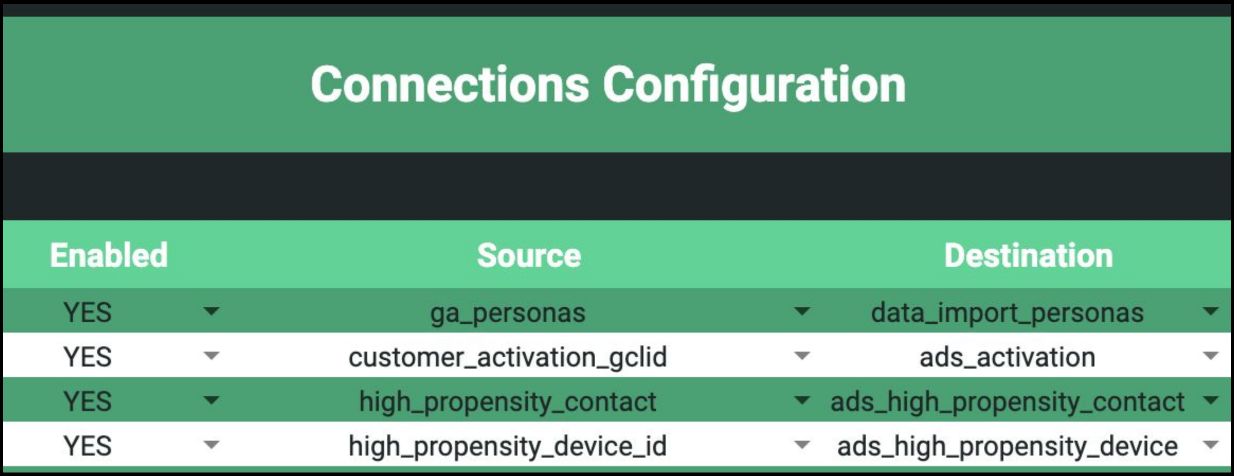
Customer Solutions
ENGINEERING

1. Configuration Spreadsheet

1.4 Connect Tab

Responsible to map a source to a destination

- **Enabled** (Yes/No)**: Disabled lines are ignored at runtime**
- **Source:** The friendly name of a Megalista source
- **Destination:** The friendly name of a Megalista destination



Connections Configuration

Enabled	Source	Destination
YES	ga_personas	data_import_personas
YES	customer_activation_gclid	ads_activation
YES	high_propensity_contact	ads_high_propensity_contact
YES	high_propensity_device_id	ads_high_propensity_device

Customer Solutions
ENGINEERING

Destination Types

Appsflyer S2S API (type: APPSFLYER_S2S_EVENTS)

Send server to server events to appsflyer

Requirements

- Appsflyer dev token configured as a scheduler parameter (appsflyer_dev_key: XXXXXX)
- Auxiliar bigquery dataset defined as a scheduler parameter (bq_ops_dataset: XXXX)

Expected Schema

Column name	type	Description	Requirement
uuid	STRING	Unique event identifier used by megalista to deduplicate events before sending	**required**
appsflyer_id	STRING	Details on Appsflyer S2S documentation	**required**
customer_user_id	STRING		optional
ip	STRING		optional
device_ids_idfa	STRING		optional
device_ids_advertising_id	STRING		optional
device_ids_amazon_aid	STRING		optional
device_ids_oaid	STRING		optional
device_ids_imei	STRING		optional
event_eventName	STRING		**required**
event_eventCurrency	STRING		optional
event_eventTime	DATETIME (UTC)		optional
event_eventValue	STRING		optional

Metadata

- Metadata 1: app_id (i.e. com.mycompany.myapp)

Additional information

At every request successfully sent, Megalista stores its uuid on a temporary table for 15 days. At every new execution, it consults stored uuids and, in case request has already being sent, it prevents it to be sent again. As a result, source table should not maintain events for more than 7 days.

Customer Solutions
ENGINEERING

Customer Match - Contact info
(type: ADS_CUSTOMER_MATCH_CONTACT_INFO_UPLOAD)

Creates (if needed) and send mobile device ids to a Google Ads customer match audience

Requirements

- Google Ads API token configured as a scheduler parameter (developer_token: XXXXXX)
- Google Ads API Enabled in Google Cloud.

Expected Schema

Column name	type	Description	Requirement
email	STRING		optional
phone	STRING	Phone number in E.164 format	optional
mailing_address_first_name	STRING		optional
mailing_address_last_name	STRING		optional
mailing_address_country	STRING	Iso two letter format	optional
mailing_address_zip	STRING		optional

Metadata

- Metadata 1: audience list name (if it does not exist, it will be automatically created)
- Metadata 2: ADD or REMOVE , if ADD, it will append the entries to the audience, if REMOVE, it will exclude them from the audience
- Metadata 3: TRUE or FALSE (DEFAULT=TRUE). Define if you would like megalista to hash the data before sending. Use false if your data is already SHA256 hashed or if you don't mind sending raw identifiers to Google Ads

Additional information

Although all columns are optional, it's required that at least email, phone or all name/address fields are present,

Customer Solutions
ENGINEERING

Customer Match - Device Id
(type: ADS_CUSTOMER_MATCH_MOBILE_DEVICE_ID_UPLOAD)

Creates (if needed) and send mobile device ids to a Google Ads customer match audience

Requirements

- Google Ads API token configured as a scheduler parameter (developer_token: XXXXXX)
- Google Ads API Enabled in Google Cloud.

Expected Schema

Column name	type	Description	Requirement
mobile_device_id	STRING	Mobile device Id identifier (android AdId or IOS IDFA)	**required**

Metadata

- Metadata 1: audience list name (if it does not exist, it will be automatically created)
- Metadata 2: ADD or REMOVE , if ADD, it will append device ids to the audience, if REMOVE, it will exclude them from the audience
- Metadata 3: Can be left empty
- Metadata 4: app_id (i.e. com.mycompany.myapp). If empty, it will use the default app id defined on "Intro" tab

Additional information

XXXXXXX

Customer Solutions
ENGINEERING

Customer Match - User Id
(type: ADS_CUSTOMER_MATCH_USER_ID_UPLOAD)

Creates (if needed) and sends user ids to a Google Ads customer match audience

Requirements

- Google Ads API token configured as a scheduler parameter (developer_token: XXXXXX)
- Google Ads API Enabled in Google Cloud.

Expected Schema

Column name	type	Description	Requirement
user_id	STRING	Google AdsUser Id identifier	**required**

Metadata

- Metadata 1: audience list name (if it does not exist, it will be automatically created)
- Metadata 2: ADD or REMOVE , if ADD, it will append device ids to the audience, if REMOVE, it will exclude them from the audience
- Metadata 3: Can be left empty

Additional information

The field user_id must match de user_id received in the remarketing Ads Tag. It's not the same user_id used in Google Analytics tags.

Customer Solutions
ENGINEERING

GA Measurement Protocol

(type: GA_MEASUREMENT_PROTOCOL)

Sends hits to a GA Property through Measurement Protocol

Requirements

- Google Analytics account id configured in the "Intro" tab.

Expected Schema

Column name	type	Description	Requirement
uuid	STRING	Unique event identifier used by megalista to deduplicate events before sending	**required**
client_id	STRING		**required**
event_category	STRING		**required**
event_action	STRING		**required**
event_label	STRING		optional
event_value	INTEGER		optional
cd1, cd2, cdn...	STRING	Any column starting with "cd" with be sent as a custom dimension. The column represents the CD number.	optional

Metadata

- Metadata 1: GA Property Id
- Metadata 2: "1" if the hit should be non-interactive, "0" otherwise

Additional information

At every request successfully sent, Megalista stores its uuid on a temporary table for 15 days. At every new execution, it consults stored uuids and, in case request has already being sent, it prevents it to be sent again. As a result, source table should not maintain events for more than 15 days.

Customer Solutions
ENGINEERING

GA Data Import (type: GA_DATA_IMPORT)

Uploads Google Analytics Data import to a GA property

Requirements

- Google Analytics account id configured in the "Intro" tab.
- Both Google Analytics and Google Analytics Reporting API enabled in Google Cloud.

Expected Schema

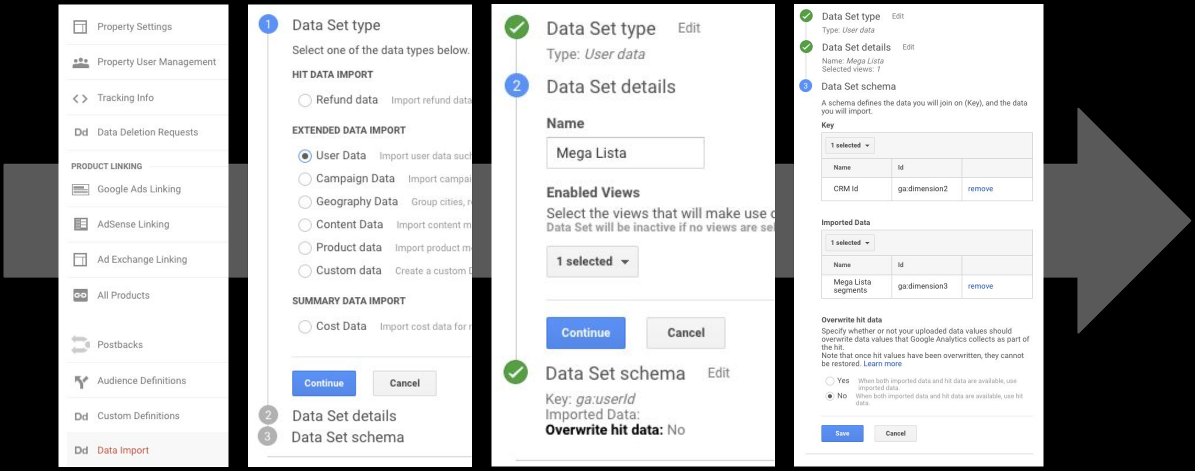
Column name	type	Description	Requirement
dimension1	STRING	Custom dimension (1-250) which will be used as match key for the upload	**required**
dimension2	STRING	Custom dimensions to be uploaded	**optional**
dimension3	STRING		**optional**

Metadata

- Metadata 1: GA Property Id
- Metadata 2: Data Import name

Additional information

- When creating the CSV file for Data Import upload, megalista appends 'ga:' to every bigquery column name so it matches csv expected by GA
- At every execution, the pipeline deletes all previous imported data associated with that data import Name, and then uploads it again to Google Analytics. As a result, we suggest you to use a dedicated data import instance for megalista
- To create a data import instance, follow these steps:



1 Data Set type

Select one of the data types below.

HIT DATA IMPORT

- ○ Refund data — Import refund data
EXTENDED DATA IMPORT
- ◉ User Data — Import user data such
- ○ Campaign Data — Import campai
- ○ Geography Data — Group cities, r
- ○ Content Data — Import content m
- ○ Product data — Import product m
- ○ Custom data — Create a custom D
SUMMARY DATA IMPORT
- ○ Cost Data — Import cost data for r

Continue | Cancel

2 Data Set details
3 Data Set schema

✓ Data Set type — Edit
Type: *User data*
2 Data Set details
Name: Mega Lista
Enabled Views: Select the views that will make use o… Data Set will be inactive if no views are sel…
1 selected
Continue | Cancel
✓ Data Set schema — Edit
Key: *ga:userId*
Imported Data:
Overwrite hit data: No

✓ Data Set type — Edit
Type: *User data*
✓ Data Set details — Edit
Name: *Mega Lista*
Selected views: *1*
3 Data Set schema
A schema defines the data you will join on (Key), and the data you will import.

Key — 1 selected

Name	Id	
CRM Id	ga:dimension2	remove

Imported Data — 1 selected

Name	Id	
Mega Lista segments	ga:dimension3	remove

Overwrite hit data
Specify whether or not your uploaded data values should overwrite data values that Google Analytics collects as part of the hit.
Note that once hit values have been overwritten, they cannot be restored. Learn more

Yes — When both imported data and hit data are available, use imported data. [✓ No] — When both imported data and hit data are available, use hit data.

Save | Cancel

Customer Solutions
ENGINEERING

Google Ads Store Sales Direct
(type: ADS_SSD_UPLOAD)

Uploads store sales direct conversions to Google Ads

Requirements

- Google Ads configured in intro tab
- Store Sales Conversion created in Google Ads
- Google Ads API Enabled in Google Cloud

Expected Schema

Column name	type	Description	Requirement
email	STRING	User email	**required**
time	STRING	2020-04-16T12:00:00.000	**required**
amount	STRING	Value in micros	**required**

Metadata

- **Conversion Name:** Name of the conversion created on Google Ads
- **External Upload ID:** Customer provided identifier for this particular conversion

Additional information

More information on Store Sales direct in the support link

Customer Solutions
ENGINEERING

Google Ads off. conversions
(type: ADS_OFFLINE_CONVERSION)

Uploads gclid based offline conversions to Google Ads

Requirements

- Google Ads configured in intro tab
- Offline Clicks Conversion created in Google Ads
- Google Ads API Enabled in Google Cloud.

Expected Schema

Column name	type	Description	Requirement
gclid	STRING	Gclid of the conversion	**required**
time	STRING	2020-04-16T12:00:00.000	**required**
amount	STRING	Value in micros	**required**

Metadata

- **Conversion name:** Name of the created conversion on google ads

Additional information

Documentation for setting up offline conversions tracking

Customer Solutions
ENGINEERING

CM offline conversions
(type: CM_OFFLINE_CONVERSION)

Uploads Campaign Manager offline conversions to Campaign Manager floodlights (compatible with CM, SA360 and DV360 conversion optimization and CM + DV360 audience creation)

Requirements

- Campaign Manager ID configured in intro
- Floodlight configuration created in CM
- Floodlight activity created in CM
- Campaign Manager API enabled in Google Cloud

Expected Schema

Column name	type	Description	Requirement
uuid	STRING	Unique event identifier used by megalista to deduplicate events before sending	**required**
gclid	STRING	Click id	**required***
mobileDeviceId	STRING	Advertising ID	**required***
encryptedUserId	STRING	Extracted from CM reports	**required***
matchId	STRING	Sent to floodlight tag	**required***

required*: Send at least one of these per row

Metadata

- **Floodlight Activity ID:** Can be obtained in the URL for the activity
- **Floodlight Configuration Id:** Can be obtained in the URL for the activity

Additional information

Documentation for offline conversions in Campaign Manager.

At every request successfully sent, Megalista stores its uuid on a temporary table for 15 days. At every new execution, it consults stored uuids and, in case request has already being sent, it prevents it to be sent again. As a result, source table should not maintain events for more than 15 days.

Customer Solutions
ENGINEERING

Customer Solutions
ENGINEERING

TIPS: Sending data to BigQuery