Android Partner Insights New Dashboard Creation

! Unreviewed · Owned by rahulkale · Edited 2022-10-10

go/ape-insights-new-dashboard

The document includes checklists and steps to follow while designing and integrating a new dashboard into Android Partner Insights. For more information about Android Partner Insights, visit go/android-partner-insights. Keep in mind that this page is designed for teams that want to share their data to external partners. Each point denotes the key step towards a successful implementation of the dashboard with consistent look and feel to the internal end user and external partners of Google.

Guidelines overview

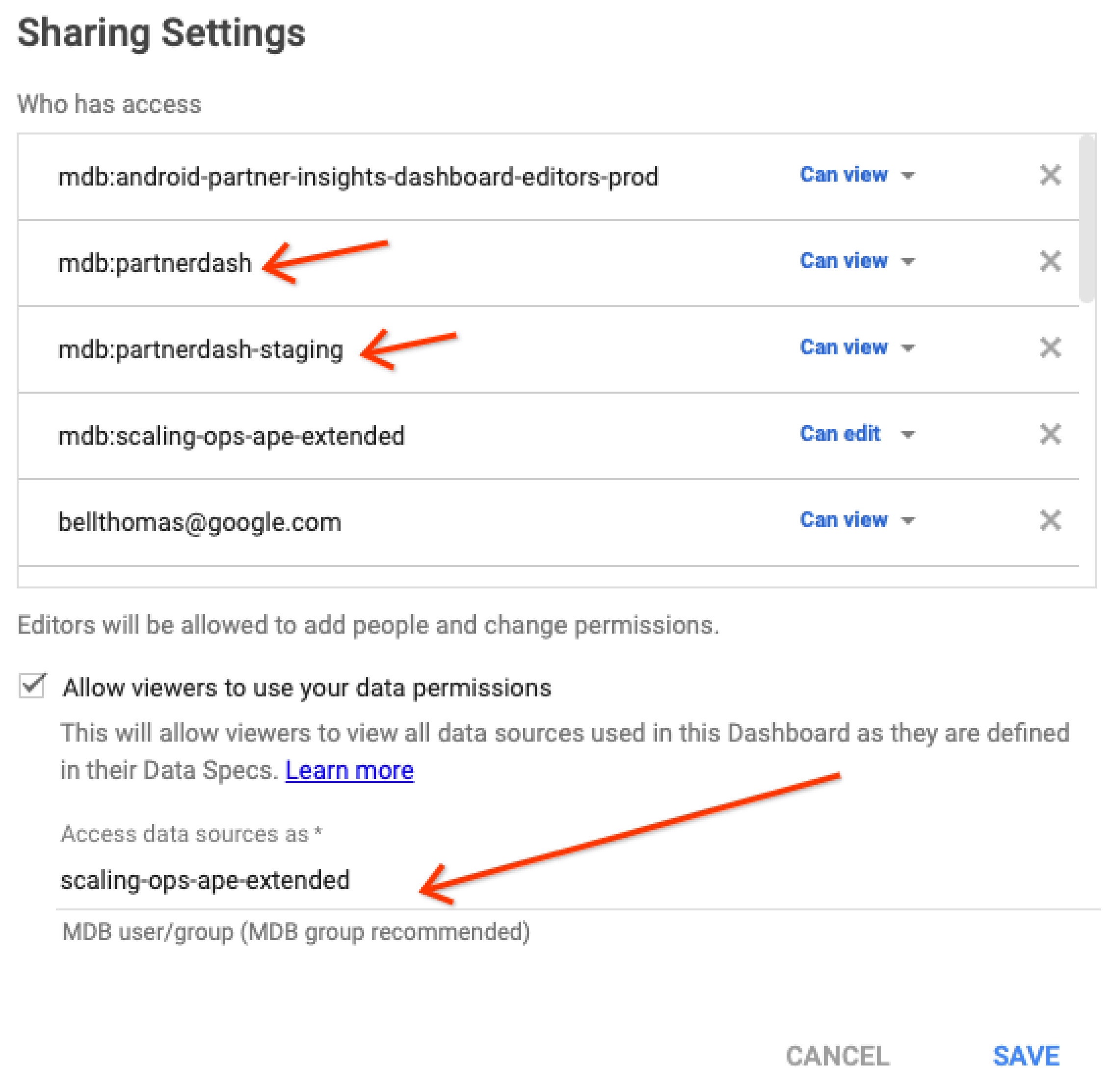
If you are creating a new dashboard that you want to eventually surface to partners via the Insights portal, you should understand the deployment architecture. Here is a summary of items you should keep in mind:

1. Create a staging and prod version of tables that you use in the dashboards. Although not necessary (for example you could use the same table for both), this approach is highly recommended.
2. You only need to create a staging dashboard because Partner Insights publishes the prod version. We copy the spec of your staging dashboard and check it in using this script. The script uses string substitution during checking in the spec of staging dashboard. This basically means that if the staging source tables have `_stg` in their name, it will get changed to `_prod`. The prod dashboard is published by the Partner Insights team from this checked in spec. The prod dashboard has the same url as staging, suffixed with `_prod`. Because it's a

Piper published dashboard, you cannot edit it directly. This allows us to rollback any breaking changes.

3. OneStack adds dashboards to the Partner Insights portal, so you need to follow their requirements as well. Your dashboard needs to grant view permission to two mdb groups: `mdb:partnerdash-staging` and `mdb:partnerdash`.

4. Your dashboard needs to enable delegated access as shown in the following image:



Sharing Settings

Who has access

mdb:android-partner-insights-dashboard-editors-prod	Can view ▾	✕
mdb:partnerdash	Can view ▾	✕
mdb:partnerdash-staging	Can view ▾	✕
mdb:scaling-ops-ape-extended	Can edit ▾	✕
bellthomas@google.com	Can view ▾	✕

Editors will be allowed to add people and change permissions.

☑ Allow viewers to use your data permissions

This will allow viewers to view all data sources used in this Dashboard as they are defined in their Data Specs. Learn more

Access data sources as *

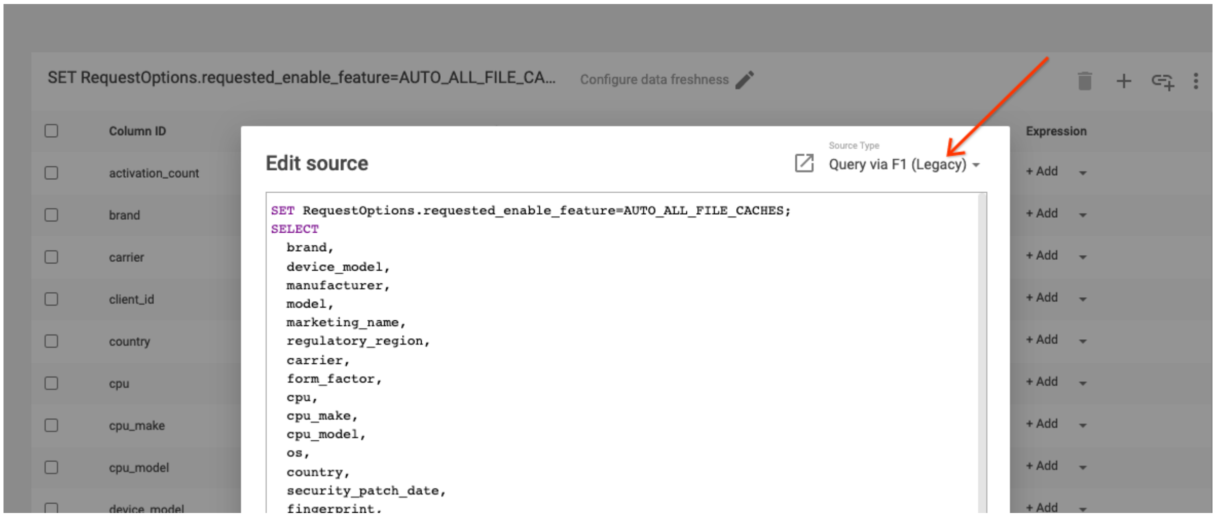
scaling-ops-ape-extended

MDB user/group (MDB group recommended)

You can use your own group for sharing, but `mdb:android-partner-insights-eng` needs to be member of the group you decide to use in delegated access or it can't be published. You should also share view access with this same group of your choice. Partner Insights won't edit your dashboard, but might need edit access to be able to publish the prod version.

5. Data is filtered at the partner level, so each partner only sees their own data. Partner Insights uses a hashed version of `dmt_partner_id` for this. Hash your `dmt_partner_id` as shown in this sample script. Use the macro we have published. To obfuscating dmt, use this macro. Each data source you use in the dashboard needs to include this hashed id. Otherwise the data won't flow through.

6. Onestack does not support v2 data sources, so make sure you use `legacy` data sources when creating your dashboard as shown in the following image:



SET RequestOptions.requested_enable_feature=AUTO_ALL_FILE_CA...

	Column ID	Expression
□	activation_count	+ Add
□	brand	+ Add
□	carrier	+ Add
□	client_id	+ Add
□	country	+ Add
□	cpu	+ Add
□	cpu_make	+ Add
□	cpu_model	+ Add
□	device_model	+ Add

Edit source

Source Type: Query via F1 (Legacy)

```
SET RequestOptions.requested_enable_feature=AUTO_ALL_FILE_CACHES;
SELECT
  brand,
  device_model,
  manufacturer,
  model,
  marketing_name,
  regulatory_region,
  carrier,
  form_factor,
  cpu,
  cpu_make,
  cpu_model,
  os,
  country,
  security_patch_date,
  fingerprint,
```

7. Through trial and error, Partner Insights has determined that F1 materialized tables make dashboards faster. You don't have to follow this suggestion, but if you find your dashboard's performance is lacking, then you might want to try this. Here's a doc that goes into how to make plx dashboards faster.

Guideline checklists

This section contains guideline checklists you can follow when developing your dashboards.

Process checklist

1. Schedule an initial consultation with the Partner Insights team to determine eligibility and fit of your proposed dashboard.

2. Share the project overview documentation (one pager, BRD, and so on) with the Partner insights group.
3. Get necessary approvals from the Privacy Working Group.
4. Ensure the data used for reporting is following go/udap and anonymization guidelines.
5. File a bug with the Insights team (Partner Insights component) to create a tab and integrate into Alkali.

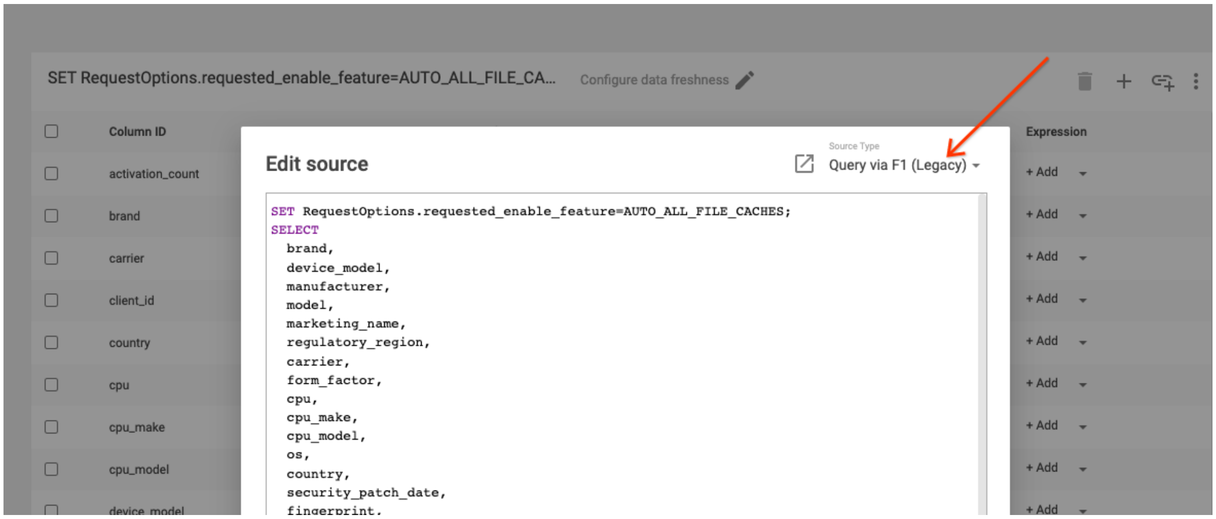
Development checklist

1. Create the stage table scripts and workflow and chart scripts.
2. Design the staging dashboard.
3. Review the necessary share access required for the tables, workflows, and dashboard.
4. Request a review with the Partner Insights team.
5. Deploy prod table scripts, workflow, and tables. Then share access to ACLs.
6. Handover staging dashboard for Alkali deployment to APE Eng team.

Technical checklist

1. Integration into Partner Insights - go/android-partner-insights.

2. Be sure to use legacy data sources while creating dashboards (limitation of Alkali).



```
SET RequestOptions.requested_enable_feature=AUTO_ALL_FILE_CACHES;
SELECT
  brand,
  device_model,
  manufacturer,
  model,
  marketing_name,
  regulatory_region,
  carrier,
  form_factor,
  cpu,
  cpu_make,
  cpu_model,
  os,
  country,
  security_patch_date,
  fingerprint,
```

Detailed design guidelines

This section contains detailed design guidelines for you to follow when building your dashboards.

Table and build design guidelines

Table script

- Use macros to streamline the code for staging and prod tables.
- You can create the stage and prod tables using a `${ENV}` parameter passed at the table level.

Chart script

- Create stage and prod tables and pull data into charts scripts dynamically from each by using the `${ENV}` variable. For more information, see this reference.

DMT Partner Id

- Source table query should contain the hashed version of `dmt_partner_id` to populate the data on the new chart, because OneStack doesn't support v2 data sources and data will not show up in the OneStack app.

- Include the macro path in the table script to ensure the DMT partner id gets hashed.

- Include the `dmt_partner_id` in the table. For more information, see this reference.

Workflow

- Ensure the workflow has proper naming, has relevant mdb groups, and ldaps to notification alerts.

- Ensure right id, labels, and run_script_nodes.

Build file

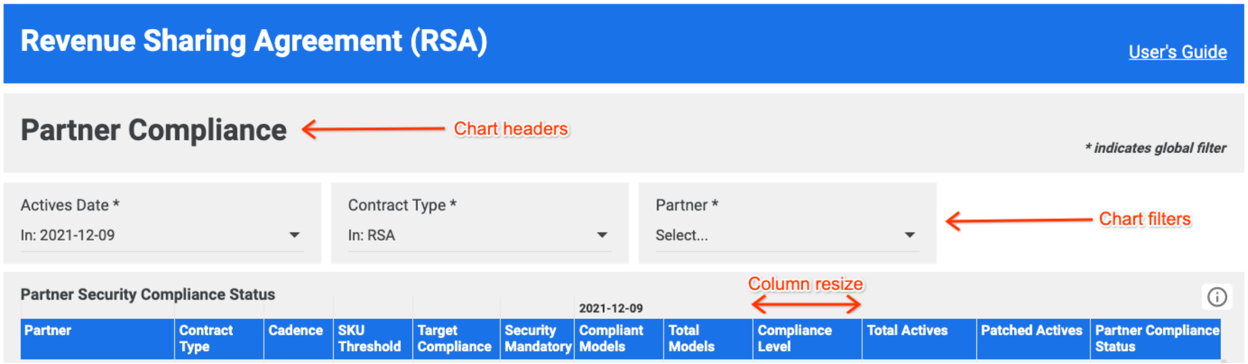
- Ensure that you used the correct script name, display name, params, and query engine.
- Create relevant staging and prod deployment in the build file.

Table element naming conventions

- Field names should all be in lowercase, such as firstname and not FirstName.
- Field names should start with a letter and not a numeric character.
- Underscores can be used, such as first_name, user_stats.
- No other special characters can be used in field naming, such as ! or . or [].
- Field names should be meaningful. For example, do not use temp1 as a field name.

Dashboard Design

- Alkali supports only Plx 1.0 version dashboards. Dashboard using Plx 2.0 version might not support all the features once deployed.
- Preferably use standard design templates for dashboard development.
- Create a copy of the dashboard template and make any necessary updates.
- Follow standard naming conventions for design elements.
- All dashboard names should start with capital letters, such as titles, filters,and so on.
- Resize columns in a table for easy readability.



Revenue Sharing Agreement (RSA)

Partner Compliance

Chart headers

indicates global filter

Actives Date * In: 2021-12-09

Contract Type * In: RSA

Partner * Select...

Chart filters

Partner Security Compliance Status

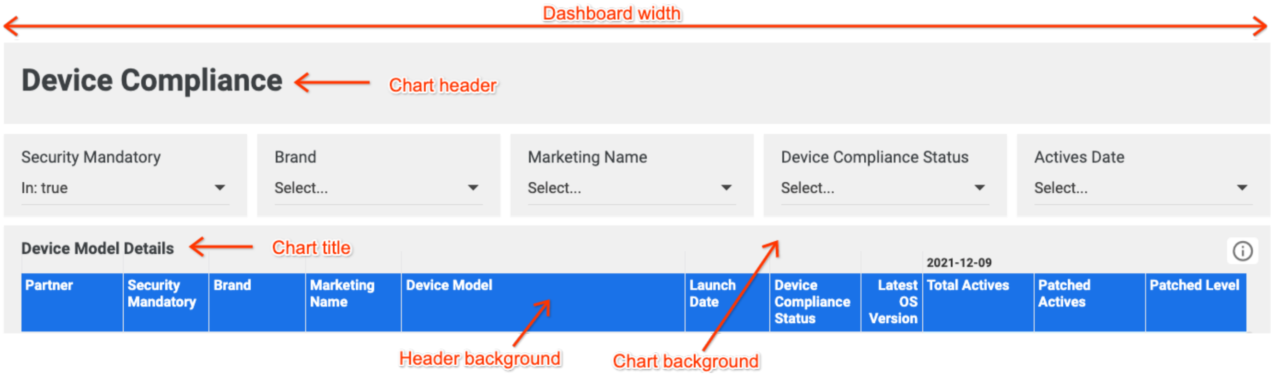
Column resize

2021-12-09

Partner	Contract Type	Cadence	SKU Threshold	Target Compliance	Security Mandatory	Compliant Models	Total Models	Compliance Level	Total Actives	Patched Actives	Partner Compliance Status

Design style preferred

- Dashboard Font: **Google Sans**
- Dashboard width: **1250px**
- All Paddings: **16px**
- Chart header/title font color: **Black**
- Header background color: **Blue #1A73E8**
- Chart, filters background color: **Gray #F0F0F0**
- Border and Shadow: **Transparent**



Dashboard width

Device Compliance

Chart header

Security Mandatory In: true

Brand Select...

Marketing Name Select...

Device Compliance Status Select...

Actives Date Select...

Device Model Details

Chart title

2021-12-09

Partner	Security Mandatory	Brand	Marketing Name	Device Model	Launch Date	Device Compliance Status	Latest OS Version	Total Actives	Patched Actives	Patched Level

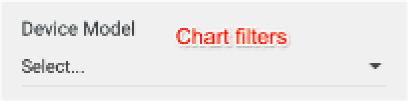
Header background

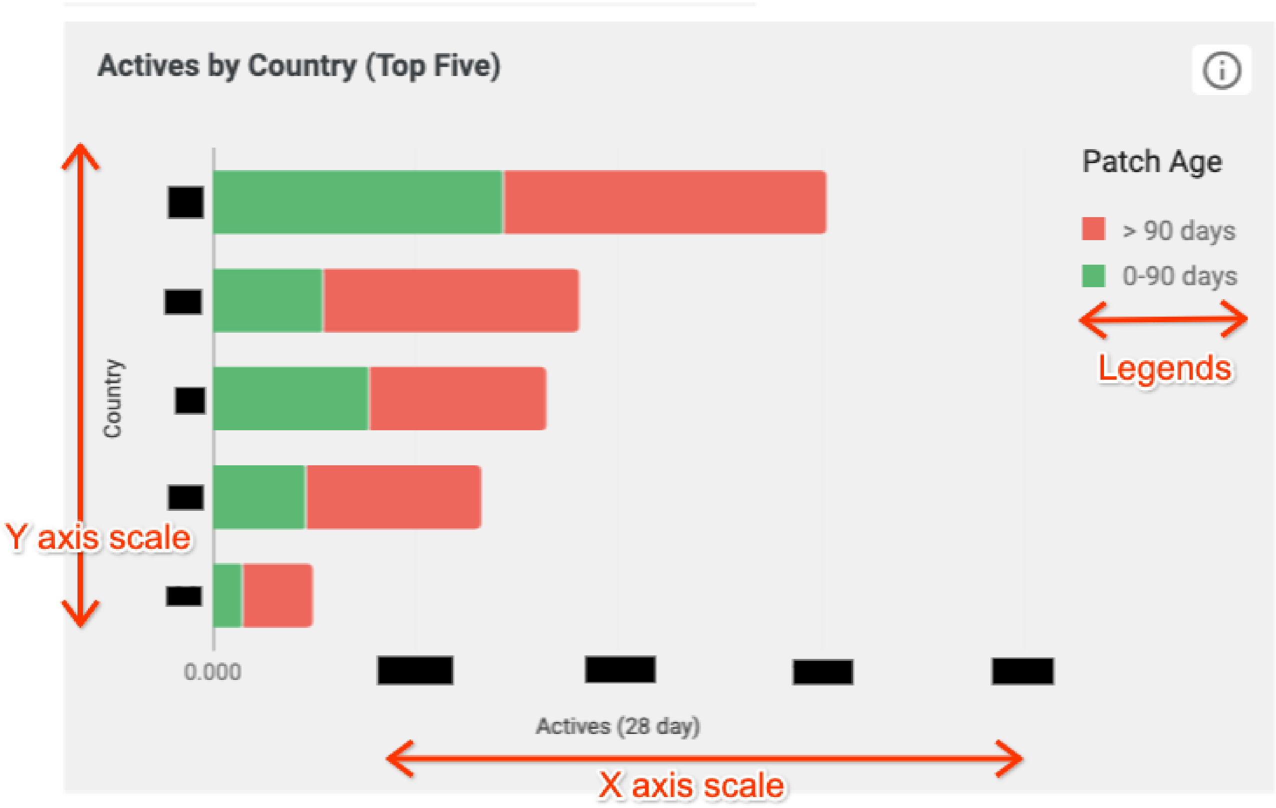
Chart background

Chart design

- Chart should have appropriate headers explaining the functionality.
- Chart filters should be on top of each chart.
- Bar graph, line graph, pie charts should display scale in use.

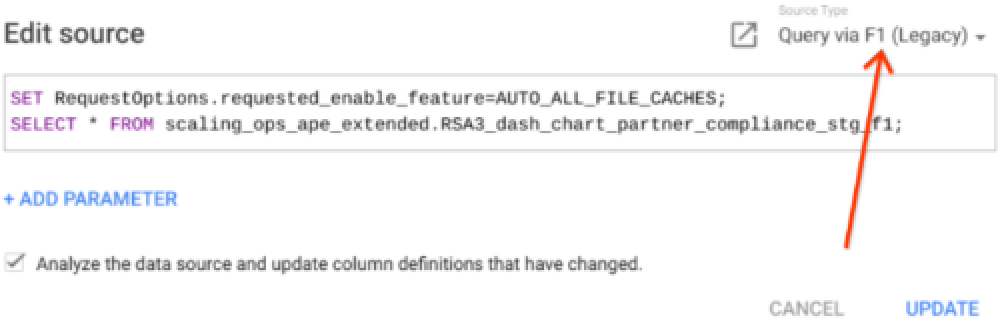
- Each chart should have an X axis and Y axis scale.
- Expose global filter for complete dashboard (if required).
- Legends should be displayed wherever possible.





Query should be set to legacy data sources, given the limitations of OneStack

- When selecting of data source for populating the data on a new chart, select the source type: **Query via F1(Legacy)**.


Edit source
Source Type
Query via F1 (Legacy)

```
SET RequestOptions.requested_enable_feature=AUTO_ALL_FILE_CACHES;
SELECT * FROM scaling_ops_ape_extended.RSA3_dash_chart_partner_compliance_stg_f1;
```

+ ADD PARAMETER
✓ Analyze the data source and update column definitions that have changed.
CANCEL
UPDATE

Access to dashboard and data

- Include the below two groups to tables with share edit access, and dashboard access in order to have it visible to the APE members:
  - Android-partner-insights-eng
  - ape-analytics
- Access is required to publish the prod dashboards and sharing with Alkali group, and to view the backend development as needed.

Point of contact

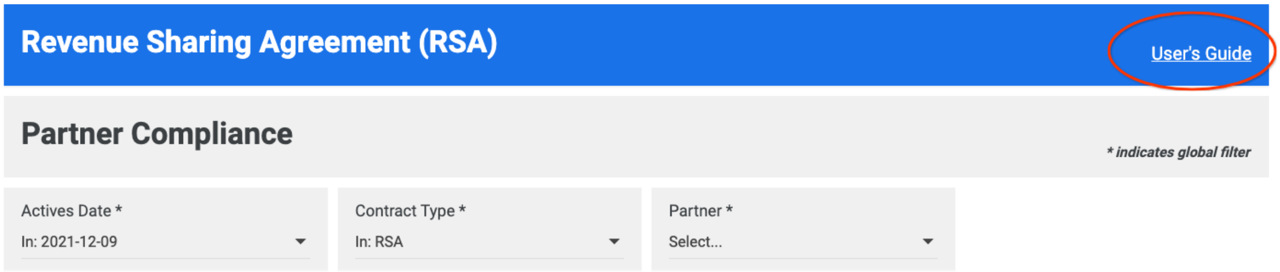
- Share the member list on the business team used as a point of contact for queries , issues and updates regarding the dashboard.

Maintenance

- Expected future feature requests.
- Scope of maintenance for the dashboard.

Example RSA user guide

- Document explaining the tables used in the dashboard.
- List of columns and column definition.



Revenue Sharing Agreement (RSA)

Partner Compliance

** indicates global filter*

Actives Date *
In: 2021-12-09

Contract Type *
In: RSA

Partner *
Select...

Other recommendations

- Try using the Plx Dashboard inbuilt color schemes for charts.
- Place universal filters for the dashboard.
- Maintain adequate vertical and horizontal spacing/padding between viz.
- Place local filters for the charts just above the specific viz they control.
- Sort table data based on an element value in numeric or alphabetical order.
- Avoid too many colors in single graph.
- Avoid heavy gridlines, thick borders, and dark labels.
- Update column formatting for date columns as YYYY-MM-DD.
- Format numbers with commas.
- Accessibility guidelines/requirements reference:
  - General guidelines
  - GAR self-assessment template
  - Guidelines regarding elements in the DOM follow the correct order so users can access and cycle through everything using a screen \

Comments

Before you can create or read g3doc comments, you need to grant g3doc access to Buganizer and Google people information. Learn more

Get access to g3doc comments

View all unresolved or resolved comments