Power BI setup guide

Connect Avistrata to Power BI and Excel

For organization admins and report builders · Avistrata 2.2

Avistrata gives your organization **report-ready data** for Power BI and Excel: aircraft and availability, maintenance due, squawks, work orders, labor, flights, invoices, stock, and **programs with their contract line items (CLINs) and costs**. A **reporting key** lets your reports refresh on their own, on your schedule, with no database access and no gateway. A reporting key can only **read**; it can never change anything in Avistrata.

Setup takes about 15 minutes. An organization admin creates the key once. Whoever builds the reports then adds the tables in Power BI Desktop and publishes them.

ORG ADMIN · ONCE
1. Create a reporting key
In Avistrata: Admin → Users → API Partners.

REPORT BUILDER
2. Add the tables
In Power BI Desktop: one small query per table.

REPORT BUILDER
3. Publish and refresh
In the Power BI service: Anonymous credentials and a refresh schedule.

OPTIONAL
4. Excel
The same queries in Excel's Power Query.

Before you start

- **An organization admin in Avistrata** to create the key (step 1).
- **Power BI Desktop** (free, Windows) to build the reports, and a Power BI license that lets you publish and share them (usually Pro) for step 3.
- **What the key sees.** The key reads your organization and its sub-organizations, and sees exactly what the admin who set it up can see in Avistrata. For program costs, that admin must also be **finance or a director** in Program Management.

1 Create a reporting key ORG ADMIN

1. Sign in to Avistrata at `https://login.mrotracker.com`.
2. Open **Admin → Users → API Partners**. Organization admins go straight to **Reporting keys**. Super admins click **Reporting keys (Power BI)** at the top of API Partners.
3. If you're admin of more than one organization, pick the organization.
4. Type what the key is for, for example *Power BI – fleet dashboard*, and click **Create reporting key**.
5. Click **Copy** and keep the key somewhere safe, such as your password manager. **It's shown only once.** If you lose it, create a new one and revoke the old one.

> **Treat the key like a password.** Anyone who has it can read this organization's report data. Give it only to the people who build the reports. To stop a key, click **Revoke** next to it; reports using it stop refreshing within about a minute. Every key you create or revoke goes to the Audit Log.

What a key can read

Table (resource)	What's in it
FLEET AND MAINTENANCE	
`rpt_aircraft`	One row per aircraft: make, model, serial, base, owner, status, availability (Available / Down / AOG), hours, landings, in service.
`rpt_maintenance_due`	Active maintenance items: type, intervals, last done, next due date / hours / cycles, remaining, status (OK / Coming due / Overdue / Not set).
`rpt_squawks`	Squawks and discrepancies: reported on and by, grounding, deferred, status (Open / Deferred / Closed), closed on, days open.
`rpt_work_orders`	Work orders: aircraft, customer, status, opened / closed, days open, clocked hours, labor / parts / total invoiced in your base currency.
`rpt_labor`	Clock-ins on task cards: date, person (name only), work order, task card, ATA, hours, customer work (yes/no).
`rpt_flights`	Flight log, plus bookings without a flight log entry: PIC, rule, customer, planned and actual times, block and flight hours, landings, from / to.
`rpt_invoices`	Invoices: customer, source, issue and due dates, status, totals, paid, balance, base-currency totals, labor / parts, days overdue, aging bucket.
`rpt_stock`	Stock: part, condition, serial / batch, location, quantity, unit cost, value, certificate, shelf life, status.
PROGRAMS AND CONTRACTS	
`rpt_programs`	One row per program: number, title, customer, prime or sub, contract and task order, contract type, period of performance, ceiling, funded, status, health (overall / cost / schedule), marking, aircraft and work orders.
`rpt_clins`	Contract line items (CLINs and SLINs): type, option and when exercised, quantity, unit price, ceiling, funded, period of performance; **cost to date**: labor hours, labor, material, other direct, indirect at provisional rates, total; and **% of funding used**.
`rpt_program_costs`	Every direct cost line on a program, by CLIN: date, type (Labor / Material / Subcontract / Travel / Other direct cost), person (name only), labor category, hours, cost rate, amount, indirect cost, total, approved, unallowable, work order, charge code.

People appear by name only, never with an email or phone number. No payment links, signatures or file locations are included. Stock of export-controlled parts (ITAR / EAR) and programs marked **ITAR or CUI** are left out unless an admin ticks *Include export-controlled parts* in the Reporting keys window. Program cost figures are filled in only when the key's admin is finance or a director in Program Management.

2 Add the tables in Power BI Desktop REPORT BUILDER

a. Add the key as a parameter

1. In Power BI Desktop, click **Home → Transform data** to open Power Query.
2. Click **Manage Parameters → New Parameter**. Name it `ApiKey`, set Type to **Text**, paste your key as the Current Value, and click **OK**.

b. Add the reader function (once)

1. Click **New Source → Blank Query**, then **Advanced Editor**.

2. Replace everything with the code below and click **Done**.
3. Rename the query to `MroTable`. It reads every page of a table, 500 rows at a time.

```
(resource as text, optional filters as record) as table =>
let
    BaseUrl = "https://login.mrotracker.com/api/v1",
    Extra   = if filters = null then [] else filters,
    GetPage = (offset as number) =>
        Json.Document(Web.Contents(BaseUrl, [
            RelativePath = resource,
            Query   = Record.Combine({[limit = "500", offset = Text.From(offset)], Extra}),
            Headers = [#"X-API-Key" = ApiKey]])),
    Pages = List.Generate(
        () => GetPage(0),
        each _ <> null,
        each if _[meta][next_offset] = null then null else GetPage(_[meta][next_offset]),
        each _[data]),
    Rows = List.Combine(Pages)
in
    Table.FromRecords(Rows, null, MissingField.UseNull)
```

c. Add one query per table

1. Click **New Source → Blank Query**. In the formula bar, type `= MroTable("rpt_aircraft")` and press Enter.
2. Rename the query, for example to *Aircraft*, and set the column types (dates, numbers) as usual.
3. Repeat for each table you need, for example `= MroTable("rpt_work_orders")`.
4. The first time, Power BI asks how to connect to `login.mrotracker.com`. Choose **Anonymous** at the level `https://login.mrotracker.com/api/v1`, then click **Connect**. The key travels in the `X-API-Key` header, so no sign-in is needed. If it asks for a privacy level, choose **Organizational**.
5. Click **Close & Apply**.

Only the history you need. Large tables refresh faster with a filter, given as a record:
`= MroTable("rpt_flights", [flight_date = "gte:2025-01-01"])`
`= MroTable("rpt_program_costs", [line_date = "gte:2026-01-01"])`
Filters take `field = "op:value"`, where `op` is one of `eq, neq, gt, gte, lt, lte, like, in, is`. `updated_since` works on every table except `rpt_aircraft` and `rpt_program_costs`.

d. Relationships

From	To	Column
Maintenance due, Squawks, Work orders, Labor, Flights, Invoices	Aircraft	**`aircraft_id`**
Labor, Invoices, Program costs	Work orders	**`work_order_id`**
Work orders, Labor, Flights, Invoices, Programs	A Customer table (optional)	**`customer_id`**
CLINs, Program costs	Programs	**`program_id`**
Program costs	CLINs	**`clin_id`**

Add a Date table, mark it as the date table, and relate it to the date you report on, such as *opened_on*, *work_date*, *flight_date*, *issue_date* or *line_date*.

P Report page: Programs and CLINs REPORT BUILDER

For government and contract work: one page that shows, for every program and CLIN, what's been funded, what's been spent and how much is left, with the burn by month. It uses the three program tables.

Queries

```
Programs     = MroTable("rpt_programs")
CLINs        = MroTable("rpt_clins")
ProgramCosts = MroTable("rpt_program_costs", [line_date = "gte:2025-01-01"])
```

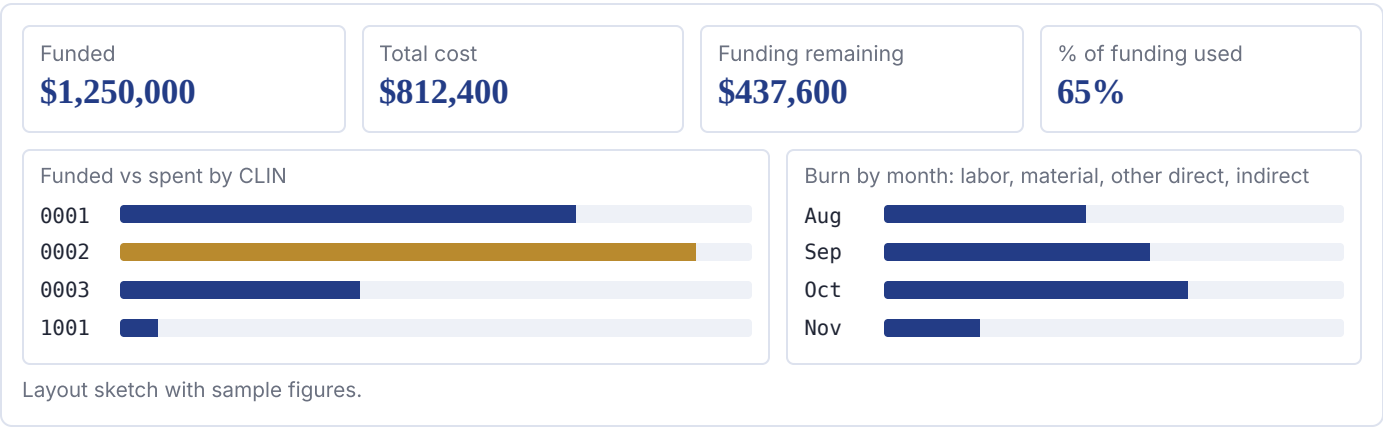
Set the types: *line_date*, *pop_start*, *pop_end* as Date; *ceiling*, *funded*, *amount*, *indirect_cost*, *total_cost* and the *..._to_date* columns as Fixed decimal number; *hours* as Decimal number. Relate them as in the table above (*program_id*, *clin_id*), and ProgramCosts[line_date] to your Date table.

Measures (DAX)

```
Funded            = SUM ( CLINs[funded] )
Ceiling           = SUM ( CLINs[ceiling] )
Direct cost       = CALCULATE ( SUM ( ProgramCosts[amount] ), ProgramCosts[unallowable] = FALSE () )
Indirect cost     = CALCULATE ( SUM ( ProgramCosts[indirect_cost] ), ProgramCosts[unallowable] = FALSE
() )
Total cost        = [Direct cost] + [Indirect cost]
Labor hours       = SUM ( ProgramCosts[hours] )
Material cost     = CALCULATE ( [Direct cost], ProgramCosts[cost_type] = "Material" )
Funding remaining = [Funded] - [Total cost]
% of funding used = DIVIDE ( [Total cost], [Funded] )
Unapproved labor  = CALCULATE ( SUM ( ProgramCosts[amount] ), ProgramCosts[cost_type] = "Labor",
ProgramCosts[approved] = FALSE () )
Unallowable cost  = CALCULATE ( SUM ( ProgramCosts[amount] ), ProgramCosts[unallowable] = TRUE () )
```

Visuals

- **Cards:** Funded, Total cost, Funding remaining, % of funding used.
- **Bar chart, funded vs spent by CLIN:** axis CLINs[clin], values [Funded] and [Total cost]. Add conditional formatting on [% of funding used] (amber from 75%, red from 90%) to show which line items are close to their funding notice point.
- **Stacked column, burn by month:** axis Date[Month], values [Total cost], legend ProgramCosts[cost_type].
- **Matrix:** rows Programs[program_number] → CLINs[clin], columns [Labor hours], [Direct cost], [Indirect cost], [Total cost], [Funded], [% of funding used].
- **Slicers:** Programs[program_number], Programs[status], Date.



Layout sketch with sample figures.

How the costs are worked out

- **Labor:** approved and unapproved timesheet hours on the program's charge codes, at each person's labor cost rate (Accounting setup). Use *approved* to report only approved time.
- **Material:** parts issued from stores to the program's work orders, at their stock value. A return takes the cost back off. Purchases charged straight to a program come in as other direct costs.
- **Indirect cost:** each line's share of your indirect pools (for example fringe, overhead, G&A) at the fiscal year's **provisional** rates, applied in the same order as Avistrata's program Costs tab.
- **Total cost** is before fee. Unallowable costs (FAR 31.205) are shown separately and left out of totals and % of funding used.

> **The incurred cost submission and vouchers stay in Avistrata.** Power BI is for watching programs. The figures DCAA sees come from Program Management → Cost accounting, which also handles final rates and closed months.

3 Publish and refresh on a schedule REPORT BUILDER

1. In Power BI Desktop, click **Publish** and pick the workspace.
2. In the Power BI service, open the workspace and find the semantic model (dataset). Click **⋯ → Settings**.
3. Under **Data source credentials**, click **Edit credentials** for `login.mrotracker.com`. Choose **Anonymous**, set the privacy level to **Organizational**, tick **Skip test connection**, and click **Sign in**.
4. Under **Refresh**, turn on scheduled refresh, choose how often and when, and click **Apply**. No gateway is needed.
5. Click **Refresh now** once to check it works.

> **Why "Skip test connection"?** The test calls the address without your key, and Avistrata refuses calls without a key. The real refresh sends the key and works. Also keep the key in the `X-API-Key` header as in the code above, not an `Authorization` header.

4 Excel (optional)

Excel for Windows uses the same Power Query:

1. Click **Data → Get Data → From Other Sources → Blank Query**.
2. Add the `ApiKey` parameter, the `MroTable` function and one query per table, as in step 2.
3. Click **Close & Load**. To update the data, click **Data → Refresh All**.

If something goes wrong

What you see	What to do
401 "Unknown API key" or "has been revoked"	Check the `ApiKey` parameter has the whole key, starting with `mrot_`. If the key was revoked, create a new one.
403 "This key may not read ..."	Use one of the eleven `rpt_` tables. Reporting keys can't read anything else.
400 "Can't filter ... by ..."	That column can't be used as a filter. Filter in Power Query after loading instead.
429 "More than 120 requests this minute"	Too many refreshes at once. Wait a minute and refresh again, or stagger refresh times.
Refresh fails in the service, but works in Desktop	Set the credentials to **Anonymous** with **Skip test connection** ticked (step 3).
A table is empty or missing records	The key sees what the admin who set it up sees. Ask an admin to check that person's access, or which organization the key belongs to.
Program cost columns are empty	The key's admin isn't finance or a director in Program Management, or the program is marked ITAR / CUI and export-controlled records are off.
Labor shows hours but no cost	No labor cost rate is set. An admin sets one for the organization, or per person, in Accounting setup.

Security and data

- **Read-only.** A reporting key can only read the eleven report tables, never change or delete anything.
- **Your organization only.** Every request is limited to your organization and its sub-organizations. Avistrata's own access rules apply exactly as in the app.
- **Logged and limited.** Every request is logged, and each organization can make up to 120 requests a minute.
- **Sharing files.** The key is saved in the Power BI file. Share the published report, not the .pbix file, with people who shouldn't have the key. Program cost reports carry labor cost rates, so share them only with finance and program leadership.

Need help? Email **support@avistrata.com** and we'll help you connect. · www.mrotracker.com

© 2026 Leesburg Aviation LLC. Power BI and Excel are trademarks of Microsoft Corporation; Avistrata is not affiliated with Microsoft.