Analytics for LS Central Fabric Data Agent

Notebook Deployment Guide

Use this guide to import and run any Fabric Data Agent notebook in your Microsoft Fabric workspace using Analytics for LS Central Power BI semantic models. The agents are based on the Power BI report templates available at our Analytics for LS Central product package.
Follow the steps in order: import the notebook first, then personalise the two variable values you will find inside it and finally run all cells.

PRE Prerequisites

- Microsoft Fabric environment with capacity F64 or higher
- The following tenant switches enabled — contact your Fabric admin if unsure:
  - AI skill
  - Copilot
  - Cross-geo processing and cross-geo storage
- The target semantic model from Analytics for LS Central Power BI templates has already been published and is visible in your workspace
- Note the exact display name of the semantic model (visible in your workspace item list — it is case-sensitive)

> **Note:** Have the exact semantic model display name to hand before you start — you will need it when personalising the notebook in Section 02.

01 Import the Notebook into Your Workspace

- In your Fabric workspace, open the **workspace menu** and select **Import > Notebook**.
- Upload the provided .ipynb file.
- Once the import completes, the notebook appears in your workspace item list. Open it.

> **Checkpoint:** The notebook should open showing all 7 steps in sequence. If you see an error on import, ensure the file has the .ipynb extension and was not renamed.

02 Personalise the Notebook

With the notebook open, locate the two cells below and update the highlighted line in each one before executing anything.

Step 3 Edit **`data_agent_name`** to be the name you want your agent to have e.g. data_agent_supply_chain

```
→ data_agent_name = "data_agent_supply_chain"
data_agent = create_data_agent(data_agent_name)
```

Choose a unique name with no spaces (underscores are fine). This becomes the item name visible in your workspace.

Step 5 Edit **`semanticmodel_name`** to match the semantic model name at your environment

```
→ semanticmodel_name = "Supply Chain Analytics"
data_agent.add_datasource(semanticmodel_name, type="semanticmodel")
```

Must match the semantic model's display name in your workspace exactly — including capitalisation and spaces.

03 Run All Cells — Top to Bottom

After updating the two values in Section 02, click “Run All” (at the top-left) to run every cell in sequence from top to bottom. Do not skip cells.

Step	Action	What to confirm
1	Install SDK	No pip errors — if the kernel just started, restart and re-run
2	Import classes	No import errors
3 ✎	Create agent	Edit data_agent_name first — agent object returned without errors
4	Configure instructions	get_configuration() returns the full instructions block
5 ✎	Add data source	Edit semanticmodel_name first — model appears in get_datasources()
6	Select tables	Tables marked with * in pretty_print() output
7	Publish agent	Agent is visible in your workspace item list

> **Tip:** Steps marked ✎ require you to have updated the variable values in Section 02. If you see a NameError, this is the cause — return to Section 02 and update the values before re-running.

04 Verify the Agent

- Refresh your workspace — the agent appears as a Data Agent item with the name you defined.
- Open the agent and confirm:
  - The correct semantic model is listed as a data source
  - The expected tables are selected (visible in the agent configuration)
  - The agent description matches what was published in Step 7
- Send a test question in the agent chat to confirm it responds using your data. For example:

```
"What is the total inventory quantity by location for the last month?"
```

05 Common Issues

Issue	Resolution
`NameError: name not defined`	You ran a cell before updating the variable values. Return to Section 02, edit both values, then re-run from Step 3.
`add_datasource() error`	Model name mismatch. Re-check the exact display name in your workspace item list — it is case-sensitive.
`No tables visible in Step 6`	Model is not fully published, or you do not have Contributor or higher access to the workspace.
`pip install fails in Step 1`	Restart the Spark session and re-run. If it persists, confirm outbound internet access is enabled on your Fabric capacity.
`Tenant switch errors`	Contact your Fabric admin — AI skill and Copilot must be enabled at tenant level by an administrator.
`Agent not visible after Step 7`	Refresh the workspace. If still absent, re-run Step 7 and confirm the publish() call completed without errors.

06 Cleanup (Optional)

To remove the agent — for example, during testing or to recreate it with different settings — uncomment and run the last cell in the notebook:

```
# delete_data_agent(data_agent_name)
```

> **Warning:** Deleting the agent removes all its configuration, instructions, and published state. This action cannot be undone. You can recreate it by re-running the notebook from Step 3.

Disclaimer

These agents are provided "as-is" and are not yet part of the officially supported Analytics for LS Central product package. They are intended to support exploration, learning, and innovation.

We encourage you to try them out and explore their potential. At this stage, they are offered without warranty, formal support commitments, and features and availability may evolve over time.

We see this as an opportunity for you to experiment, test ideas, and discover new ways to leverage Analytics for LS Central. We’re excited to learn how you use them.