Fix Dependency Issues

Contents:

Where possible, changes made in one dataset or recipe propagate to the datasets that consume it. Datasets that join, union, or lookup against your dataset are likely to be impacted if you delete columns or rows or otherwise change the data. In some cases, the recipes of these dependent datasets can break.

This section describes how to identify these dependency issues and includes general steps for fixing them.

How to Identify

Dependent datasets

When making edits to a recipe, you can verify if your changes potentially impact other recipes or reference datasets that rely on it. In the Transformer page, click the drop-down next to the current dataset's name to open the Recipe Navigator. Select the Flow View tab.

> **Tip:** If your current dataset is connected to datasets to the right of it, those datasets are dependent on the current one. After you make changes to the current one, you should use the Recipe Navigator to open recipes and datasets that are connected to it and to the right of it in flow view.

See *Recipe Navigator*.

Broken data integrations

When you make some changes in an upstream recipe or dataset, the recipes for any downstream datasets can break, such that you cannot generate satisfactory results. In the downstream recipe, you may see errors in the Recipe panel, such as the following:

Figure: Dependency error in the Recipe panel

In the above, the column `Day` does not exist in the current dataset, which is causing problems in the last two recipe steps. These types of errors may be generated when a column in the upstream dataset has been dropped or renamed .

Steps:

1. Open the object where the column was dropped:
    a. If the recipe or dataset is from the same flow, you can use the Recipe Navigator in the Transformer Page. See *Recipe Navigator*.
    b. If the recipe or dataset is in a different flow, use the Flows page to locate it (`REF_CAL.txt` in the above). See *Flows Page*.
2. In the Flow View tab, open the dataset referenced in the error message.
3. In the Recipe panel, locate the step where the column was removed.
4. Fix the issue. Details are below.

Hidden breakages

If you make changes to specific values in a dataset, recipe steps in downstream datasets can break if they rely on detecting specific values. Depending on the usage, the step may not actually be broken, but the generated results are incorrect.

For example, a downstream dataset recipe includes the following step:

Transformation Name	`Filter rows when value is exactly`
Parameter: Condition	`Is exactly`
Parameter: Column	`company_name`
Parameter: Value	`'My Co.'`
Parameter: Action	`Delete matching rows`

If the `company_name` column is sourced from another dataset and the `My Co.` value is changed to `My Company`, the downstream dataset that includes this transform doesn't break in an easily noticeable way. The data is simply not removed from the dataset and any generated results.

Fixing Dependencies

When you locate a dependency issue in the upstream dataset, you can fix it using one of the following methods:

1. Fix the issue in the source dataset. Verify that the change does not impact other datasets.

    > **NOTE:** If you fix the issue in the source dataset, you should verify if any other downstream datasets are impacted by this change.

2. Change the input dataset to use a dataset that is not broken.

    > **Tip:** If you must freeze the data in the dataset that you are using as an input, you can create a copy of the dataset as a snapshot. See *Dataset Details Page*.
    >
    > To use the copy, repair or rebuild the integration using the copied version.

3. Fix the issue in the dataset that depends on it. In this case, you must redefine the transformation that brings in the data.