Overview of RapidTarget

Contents:

In Trifacta®, a **target** is the set of columns, their order, and their formats to which you are attempting to wrangle your dataset. This target can be defined through imported or created datasets and must be assigned to an existing recipe. After it is assigned to a recipe, a target appears in the Transformer page to assist in your wrangling efforts. You can also apply changes to selected columns based on the target.

- This feature was formerly known as, "target matching."

Overview

In general, a target consists of the set of information required to define the expected data in a dataset. Often referred to as a "schema," this target schema information can include:

- Names of columns
- Order of columns
- Column data types
- Data type format
- Example rows of data

A dataset associated with a target is expected to conform to the requirements of the schema. Where there are differences between target schema and dataset schema, a validation indicator (or **schema tag**) is displayed.

Targets in the platform

In Trifacta, a target is created from the information in a dataset and can be applied to a recipe in a flow. When you are working with the flow, the target information is available for your wrangling activities, so that you can match up columns in your dataset (source) with their corresponding columns in the target. As you make changes in your recipe through the Transformer page, the target schema is available as a reference to see if your latest changes get you closer to matching the dataset to the target.

Known Limitations

- Targets are applied only after initial type inferencing has been applied to the loaded dataset.

  > **Tip:** As needed, you can disable initial type inferencing when data is imported into the product. See *Import Data Page*.

- Type-based matching applies a `settype` transform to any selected column. No pattern matching or standardization is applied. For more information, see *Overview of Pattern Matching*.

- Changes to the underlying objects of a target schema are not reflected in the schema. A target schema is a snapshot of the object at the time of its creation. To update, delete the target and create a new one.

  > **Tip:** If your target schema source is a recipe, then you can modify the recipe as needed and use it as your target again.

Creating Targets

Sources for creating a target

The schema used to define a target can be imported and assigned from any of the following objects, including:

- Output of a recipe in the same flow
- A reference dataset from another flow
- An imported dataset

Ideally, the source of the target schema should come from the publishing target. If you are publishing to a pre-existing target, you can create do one of the following:

- **Reference the target:** If the schema is represented in a dataset to which you have access in Trifacta, you can use it as your target schema.
- **Import the target:** Import the target table or schematized source into Trifacta as an imported dataset. Then, it can be selected as the target schema for any recipe to which you have access. See *Import Data Page*.
- **Extract target to a supported format:** If you cannot import the target directly into Trifacta, you could create an extract of a few rows, including the header, for the target into one of the formats supported for import. For more information, see *Supported File Formats*.

Creating a target for a recipe

You can create a target through one of the following mechanisms:

- **Flow View:** Select a recipe. From the context menu in the right panel, select **Assign Target to Recipe**. See *Flow View Page*.
- **Transformer Page:** Above the data grid, click the Target icon and select **Attach a new Target**.
  - See *Transformer Toolbar*.
  - You can do the same thing in the Column Browser panel. See *Column Browser Panel*.
- **Job Details Page:** After you have successfully run a job, you can create a new dataset from the Output Destinations tab. Through Flow View, this imported dataset can be used as the schema for wrangling. See *Job Details Page*.

For more information, see *Create Target*.

Using a target

After a target has been attached to a recipe, the target schema appears in a toolbar above the data grid along with a preview of the data. You can then make modifications to the data so that each column matches the definition for the corresponding column in the schema. See *Data Grid Panel*.

Through the data grid and the Column Browser, you can perform operations on selected columns in your dataset to align them with the target schema. For more information, see *Column Browser Panel*.

Running jobs on recipes with assigned targets

> **NOTE:** You can run a job even if there are differences between the schema and your dataset. In Trifacta, no error checking is performed between schema and data prior to job execution. If you are publishing to a target that has a predefined schema, a publication error may be generated.

Configure

Configure fuzzy matching threshold

You can experiment with fuzzy matching thresholds to ensure that matches are occurring properly. This parameter applies a specific threshold value when two values are compared for matching. Lower values increase the probability of a match.

Steps:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Adjust the value between 0.00 and 0.99 for the following parameter:

```
"feature.targetMatching.fuzzyMatchingThreshold": 0.30;
```

3. Save your changes and restart the platform.

Disable

If you prefer to disable this feature, please complete the following steps.

> **NOTE:** If you are experiencing performance issues with target matching, you can first try to disable fuzzy matching, which can be resource-intensive.

> **Tip:** If there is no schema associated with a recipe, then the target schema matching features are not displayed.

Steps:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Set the following parameters to `false`:

```
"feature.targetMatching.enabled" : false,
"feature.targetMatching.fuzzyMatchingEnabled" : false,
```

3. Save your changes and restart the platform.