Overview of TBE

Contents:

Transformation by Example (TBE) enables you to build recipe objects by mapping example output values for source values. Trifacta® then interprets the differences between the inputs and outputs to determine the transformation required to map them.

TBE leverages pattern-based matching and predictive transformation to derive transformations. When you provide explicit mappings of input value to output, the mapping is passed through predictive interaction to determine the best possible matching pattern.

- For more information on patterns, see *Overview of Pattern Matching*.
- **Predictive transformation** is a core component of Trifacta. Based upon user input, the platform provides one or more suggestions of ways in which to transform the data.
  - In TBE, these suggestions are rendered as elements of the transformation in progress.
  - For more information, see *Overview of Predictive Transformation*.

Use cases:

> **Tip:** TBE simplifies the process of defining patterns to match all values in your source column. Since you know and can specify the exact desired output, you can leave the details of defining the pattern or patterns required to match input to output to the product.

Transformation by Example works well in the following use cases:

- You are just getting started with the product and would like to get productive quickly to transform your data into known outputs.
- Your data has groups of values, each of which needs transformation in a different way. In a single recipe step, you can perform these transformations across all groups.
- Your data has special-case exceptions that must be transformed.

> **Tip:** You can use this feature as a final cleanup for other transformations. If you have a transformation that handles 90% of the cases in a column, you can use this transformation to handle the remainder.

Artifacts:

When a TBE step is added to your recipe, the number of individual changes can be many megabytes of data. Instead of storing these objects within the recipe definition, they are stored as a set of artifacts in the artifact storage database and referenced from the recipe.

- These artifacts exist outside the scope of the recipe file.
- These artifacts must be stored in a Trifacta database for the step to be editable and exportable.

> **NOTE:** If the artifact storage service is disabled, this feature is unusable.

- When a flow is exported, an `artifact.data` file is included as part of the export. This file must be imported with the flow definition, or the TBE step in the imported flow is broken. For more information, see *Export Flow*.

Limitations

- TBE works best for inputs that are text-based data types (e.g. String, State, URL, etc.).
  - Non-text inputs are treated as String type and may result in unexpected outputs (Integer, Decimal, etc.).
  - You cannot use multi-value inputs, such as Arrays or Objects, or use the feature to create them.

    > **Tip:** If you have Array or Object input columns, convert them to String type before using TBE.

- TBE bases its transformations on the currently displayed sample.
  - Even if you accurately map all values in your sample, some other values in the full dataset may not be mapped by the transformation.
  - You may need to take additional samples of other parts of the entire dataset to generate a more accurate transformation.
- Arithmetic operations or other numeric functions are not supported.
- You cannot create multiple columns from a single TBE step.

Enable

This feature can be enabled and disabled through the Workspace Settings page.

Steps:

1. Login to the application as an administrator.
2. From the left nav bar, select **User menu > Admin console > Workspace settings**.
3. Locate the following setting: **Create column from examples feature**.
4. Set this value to `Enabled`.
5. Save your changes.

See *Workspace Settings Page*.

This feature uses the Artifact Storage service and related database to store and retrieve historical data on TBEs. This database is installed as part of the normal database install or upgrade processes.

Column by Example

In column-by-example transformations, you create a new column from an existing one by mapping input to output values.

General workflow:

1. Select the column to use as input data.
2. Change the column to String data type, if needed.
3. From the column menu, select **Create column from examples**. See *Transformation by Example Page*.
4. Transform by example:
   a. Locate a row containing an example value to transform.

    b. In the corresponding row in the Preview column, you can enter in the new value to which the input is mapped.
    c. The transformation in development is updated to accurately capture the mapping you just performed. Additional rows in the output column may be accurately mapped, as well.
5. Repeat the above steps until all values in the output column appear to be accurately mapped.
6. When satisfied, add the transformation to your recipe.
7. Change the data type of the target and the source columns, if needed.
8. Remove the source column, if needed.

For more information, see *Create Column by Example*.

CBE for Datetime

Column-by-example also works on Datetime columns. When you use a Datetime column as your input, you specify the output values in the date/time format that you wish to use. That input value and all similarly formatted inputs should be converted to the output format. You can then specify additional example outputs for input values in a different format to standardize all of the values in the output column.

> **NOTE:** For Datetime formatting to work properly, the input column must be specified as Datetime data type.

Alternatives

For string-based inputs, the following options in Wrangle may assist in performing the same functions that you are trying to do in TBE:

Wrangle	**Description**
Extract Transform	You can use the extract transform to retrieve sub-strings from a column and insert into a new column.
String Functions	Wrangle supports a variety of string manipulation functions, which can be used to gather data from a string.