Create Custom Data Types Using RegEx

Contents:

As needed, you can deploy custom data types into the Trifacta® platform, in which type validation is performed against regular expressions that you specify. This method is most useful for validating against patterns, as opposed to specific values.

> **NOTE:** Creation of more than 25 custom data types is not supported.

If your custom data type contains a pre-defined set of values, you can create the custom type using a dictionary file for validation. See *Create Custom Data Types*.

Custom Types Location

On the server hosting the Trifacta platform, type definitions such as dictionaries and custom data types are stored in the following directory:

```
/opt/trifacta/js-data/type-packs/trifacta
```

This directory is referenced as `$CUSTOM_TYPE_DIR` in the steps below.

> **Before you begin creating custom data types, you should backup the `type-packs/trifacta` directory to a location outside of your Trifacta deployment.**

> **NOTE:** The `trifacta-extras` directory in the `type-packs` directory contains experimental custom data types. These data types are not officially supported. Please use with caution.

Directory contents:

- The `dictionaries` sub-directory contains user-defined dictionaries.

  > **NOTE:** Please use the user interface to interact with your dictionaries. See *Custom Type Dialog*.

- The `types` sub-directory contains individual custom data type definitions, each in a separate file.
- The `manifest.json` file contains a JSON manifest of all of the custom dictionaries and types in the system.

Examples

Each custom data type is created and stored in a separate file. The following example file contains a regular expression method for validating data against the set of days of the week:

```
{
  "name": "DayOfWeek",
  "prettyName": "Day of Week",
  "category" : "Date/Time",
  "defaultProbability": 1E-15,
  "testCase": {
    "stripWhitespace": true,
    "regexes": [
      "^(monday|tuesday|wednesday|thursday|friday|saturday|sunday)$",
      "^(mon|tue|wed|thu|fri|sat|sun)$"
    ],
    "probability": 0.001
  }
}
```

Parameters:

Parameter Name	Description
`name`	Internal identifier for the custom type. Must be unique across all standard types and custom types. **NOTE:** You should verify that your data type's `name` value does not conflict with other custom data type names.
`prettyName`	Display name for the custom type.
`category`	The category to assign to the type. The current categories are displayed within the data type drop-down for each column.
`defaultProbability`	Assign a default probability for the custom type. See below.
`testCase`	This block contains the regular expression specification to be applied to the column values.
`stripWhitespace`	When set to `true`, whitespace is removed from any value prior for purposes of validation. The original value is untouched.
`regexes`	This array contains a set of regular expressions that are used to validate the column values. For a regex type, the column value must match with at least one value among the set of expressions. **NOTE:** All match types must be double-escaped in the regex expression. For example, to replicate the `\d` pattern, you must enter: `\\d`. Trifacta Wrangler Enterprise implements a version of regular expressions based off of *RE2* and *PCRE* regular expressions.
`probability`	(optional) Assign an incremental change to the probability when a match is found between a value and one of the regular expressions. See *Defining probabilities* below.

Tip: In the `types` sub-directory, you can review the regex-based types that are provided with the Trifacta platform. While you should not edit these files directly, they may provide some guidance and some regex tips on how to configure your own custom data types.

Defining probabilities

For your custom type, the probability values are used to determine the likelihood that matching values indicate that the entire column is of the custom data type.

- The `defaultProbability` value specifies the baseline probability that a match between a value and one of the regular expressions indicates that the column is the specified type. On a logarithmic scale, values are typically 1E-15 to 1E-20.
- When a value is matched to one of the regular expressions, the `probability` value is used to increment the baseline probability that the next matching value is of the specified type. This value should also be expressed on a logarithmic scale (e.g. `0.001`).
- In this manner, a higher number of matching values increases the probability that the type is also a match to the custom type.

Probabilities become important primarily if you are creating a custom type that is a subset of an existing type. For example, the Email Address custom type is a subset of String type. So, matches for the patterns expressed in the Email Address definition should register a higher `probability` value than the same incremental for the String type definition.

> **Tip:** For custom types that are subsets of other, non-String types, you should lower the `defaultProbability` of the baseline type by a factor of 10 (e.g. 1E-15 to 1E-16) and raise the same probability in the custom type by a factor of 10 (e.g. 1E-14). In this manner, you can give higher probability of matching to these subset types.

Add custom types to manifest

To the `$CUSTOM_TYPE_DIR/manifest.json` file, you must add the filenames of any custom types that you have created and stored in the `types` directory:

```
{
  "types": ["bodies-of-water.json", "dayofweek.json"],
  "dictionaries": ["oceans", "seas"]
}
```

Enable custom types

To enable use of your custom data types in the Trifacta platform, locate and edit `enabledSemanticTypes` property.

You can apply this change through the *Admin Settings Page* (recommended) or

`trifacta-conf.json`

. For more information, see *Platform Configuration Methods*.

> **NOTE:** Add your entries to the items that are already present in `enabledSemanticTypes`. Do not delete and replace entries.

```
"webapp.enabledSemanticTypes": [
    "<CustomTypeName1>",
    "<CustomTypeName2>",
    "<CustomTypeNameN>"
]
```

where:

- `<CustomTypeName1>` corresponds to the internal `name` value for your custom data type.

Register your custom types

To add your custom types to the Trifacta platform, run the following command from the js-data directory:

```
node bin/load-types --manifest ${PATH_TO_MANIFEST_FILE}
```

Restart platform

Restart services. See *Start and Stop the Platform*.

Check for the availability of your types in the column drop-down. See *Create Custom Data Types*.