Type Conversions

Contents:

Import

When data is imported:

- Supported data types from the source are converted to corresponding data types supported by the application, based upon the conversions listed in this section.
- Types that are not supported but are recognized by the application are converted to String types.
- Data for types that cannot be read from the source due to technical reasons are converted to null values on import.

Type Inference

By default, the Trifacta application applies type inference for imported data. The application attempts to infer a column's appropriate data type in the application based on a review of the first lines in the sample.

> **NOTE:** Mapping source data types to Trifacta data types depends on a sufficient number of values that match the criteria of the internal data type. The mapping of import types to internal data types depends on the data.

- Type inference needs a minimum of 25 rows of data in a column to work consistently.
- If your dataset has fewer than 20 rows, type inference may not have sufficient data to properly infer the column type.

In some datasets, the first 25 rows may be of a data type that is a subset of the best matching type. For example, if the first 25 rows in the initial same match the Integer data type, the column may be typed as Integer, even if the other 2,000 rows match for the Decimal data type. If the column data type is unmodified:

- The data is written out from Trifacta Self-Managed Enterprise Edition as Integer data type. This works for the first 25 rows.
- The other 2,000 rows are written out as null values, since they do not match the Integer data type. If the source data used decimal notation (e.g. `3.0` in the source), then those values are written out as null values, too.

In this case, it may be easier to disable type inference for this dataset. See below.

> **Tip:** If you are having trouble getting your imported dataset to map to expected data types, you can disable type inference for the individual dataset. For more information, see *Import Data Page*.

Strong typecasting:

After data is imported, Trifacta application provides some mechanisms for applying stronger typecasting to the data. Example:

- If all input values are double-quoted, then Trifacta Self-Managed Enterprise Edition evaluates all columns as String type. As a result, type inference cannot be applied.
- Since non-String data types cannot be inferred, then the first row cannot be detected as anomalous against the inferred type (String). Column headers cannot be automatically detected from double-quoted source files.

Solutions:

- After data has been imported, you can remap individual column types through recipe steps. For more information, see *Change Column Data Type*.
  In the preceding example, you can apply functions to the column data to parse the values as specific data types. You can parse String values as other data types using the following functions:

Function	Description
PARSEINT Function	Evaluates a String input against the Integer datatype. If the input matches, the function outputs an Integer value. Input can be a literal, a column of values, or a function returning String values.
PARSEFLOAT Function	Evaluates a String input against the Decimal datatype. If the input matches, the function outputs a Decimal value. Input can be a literal, a column of values, or a function returning String values.
PARSEBOOL Function	Evaluates a String input against the Boolean datatype. If the input matches, the function outputs a Boolean value. Input can be a literal, a column of values, or a function returning String values.
PARSEDATE Function	Evaluates an input against the default input formats or (if specified) an array of Datetime format strings in their listed order. If the input matches one of the formats, the function outputs a Datetime value.

Export

On export from the Trifacta application:

- The application maps the internal Trifacta data type to the explicit type listed in the appropriate page in this section.
- Unmapped types are converted to the equivalent of strings.

Tip: You can import a target schema to assist in lining up your columns with the expected target. For more information, see *Overview of RapidTarget*.

Supported Data Types

Item	Description
String Data Type	Any non-null value can be typed as String. A String can be anything. See *String Data Type*.
Integer Data Type	The Integer data type applies to positive and negative numeric values that have no decimal point. See *Integer Data Type*.
Decimal Data Type	Decimal data type applies to floating points up to 15 digits in length. • In the Trifacta application, this data type is referenced as `Decimal`. • In storage, this data type is written as `Double`. See *Decimal Data Type*.
Boolean Data Type	The Boolean data type expresses true or false values. See *Boolean Data Type*.
Social Security Number Data Type	This data type is applied to numeric data following the pattern for United States Social Security numbers. See *Social Security Number Data Type*.

Phone Number Data Type	This data type is applied to numeric data following common patterns that express telephone numbers. See *Phone Number Data Type*.
Email Address Data Type	This data type matches text values that are properly formatted email addresses. See *Email Address Data Type*.
Credit Card Data Type	Credit card numbers are numeric data that follow the 14-digit or 16-digit patterns for credit cards. See *Credit Card Data Type*.
Gender Data Type	This data type matches a variety of text patterns for expressing male/female distinctions. See *Gender Data Type*.
Zip Code Data Type	This data type matches five- and nine-digit U.S. zipcode patterns. See *Zip Code Data Type*.
State Data Type	State data type is applied to data that uses the full names or the two-letter abbreviations for states in the United States. See *State Data Type*.
Object Data Type	An **Object** data type is a method for encoding key-value pairs. A single field value may contain one or more sets of key-value pairs. See *Object Data Type*.
Array Data Type	An **array** is a list of values grouped into a single value. An array may be of variable length; in one record the array field may contain two elements, while in the next record, it contains six elements. See *Array Data Type*.
IP Address Data Type	The IP Address data type supports IPv4 address. See *IP Address Data Type*.
URL Data Type	URL data type is applied to data that follows generalized patterns of URLs. See *URL Data Type*.
HTTP Code Data Type	Values of these data types are three-digit numeric values, which correspond to recognized HTTP Status Codes. See *HTTP Code Data Type*.
Datetime Data Type	Trifacta® Self-Managed Enterprise Edition supports a variety of Datetime formats, each of which has additional variations to it. See *Datetime Data Type*.

For more information on the data types that are supported within the Trifacta application, see *Supported Data Types*.