Datetime Data Type

Contents:

Trifacta® Wrangler supports a variety of Datetime formats, each of which has additional variations to it.

Date Range

Supported Date Ranges:

- **Earliest:** January 1, 1400
- **Latest:** December 31, 2599

You can use dates in the Gregorian calendar system only. Dates in the Julian calendar are not supported.

Data Validation

When values are validated against the Datetime data type, the Trifacta application does not compare them to an underlying calendar system. Instead, the application validates the values using regular expressions. This regular expression method checks for general Datetime validation and is fast to evaluate.

However, some values may follow the regular expression validation pattern but are not accurate dates. For example, every four years, February 29 is a valid date. When this date is validated against the Datetime data type, it may be detected as a valid value, while the date is changed in the application to be incremented to a close accurate date, such as March 1 in this example.

Formatting Tokens

You can use the following tokens to change the format of a column of dates:

Letter	Date or Time Component	Presentation	Examples
M	Month in year	Number	1
MM	Month in year	Number	01
MMMM	Month in year	Month	January
MMM	Month in year	Month	Jan
yy	Year	Number	16
yyyy	Year	Number	2016
D	Day in year	Number	352
d	Day in month	Number	9

dd	Day in a month	Number	09
EEE	Day in week (three-letter abbreviation)	Text	Wed
EEEE	Day in week	Text	Wednesday
h	Hour in day (1-12) **NOTE:** Requires an AM/PM indicator (a).	Number	2
hh	Hour in am/pm (01-12) **NOTE:** Requires an AM/PM indicator (a).	Number	02
H	Hour in day (1-12)	Number	2
HH	Hour in day (0-23)	Number	20
m	Minute in an hour	Number	9
mm	Minute in an hour	Number	09
s	Second in a minute	Number	3
ss	Second in a minute	Number	03
SSS	Millisecond	Number	218
X	Time zone	ISO 8601 time zone	`-08:00`
a	AM/PM indicator	String	AM

NOTE: When publishing to relational targets, Datetime values are written as date/time values in newly created tables. If you are appending to a relational table column that is in timestamp format, Datetime values can be written as timestamps.

Tip: If your DateTime column contains data in multiple formats, you must change the format of the DateTime column to one format and then add a transformation to convert that data to the other format. When all formats of your source date values are converted to a single format, the application should infer the appropriate date and time format.

Supported Separators:

- Date separators: blank space, comma, single hyphen, or forward slash
- Time separators: blank space, comma, single hyphen, colon, t or T
- Non-delimited Datetime values are supported. For example, yyyymmdd, yyyymmddThhmmssX.

ISO 8601 Time Zone Notes:

- Support for timezone offset from UTC indicated by +hh:mm, +hhmm, or +hh. For example, the date '2013-11-18 11:55-04:00' is recognized as a DateTime value.
- Datetime part functions (for example, Hour) truncate time zones and return local time.
- If you have a column with multiple time zones, you can convert the column to Unixtime so you can perform Date/Time operations with a standardized time zone using the UNIXTIME function. If you want to work with local times, you can truncate the time zone or use other Datetime functions.

Supported Datetime Formats

For more information on the available formats and examples of each one, see *Datetime Formats (PDF)*.

You can use the DATEFORMAT function to modify the formatting of your Datetime values.

Supported Time Zones

Time zones values (e.g. `UTC-08:00`) are supported.

Job Execution

Datetime data typing involves the basic type definition, plus any supported formatting options. Depending on where the job is executed, there may be variation in how the Datetime data type is interpreted.

- Some running environments may perform additional inference on the typing.

  > **NOTE:** During job execution on Spark, inputs of Datetime data type may result in row values being inferred for data type individually. For example, the String value `01/10/2020` may be inferred by date transformations as `1st Oct, 2020` or `10th Jan, 2020`. Resulting outputs of Datetime values may not be deterministic in this scenario.

- Some formatting options may not be supported.

Differences between Trifacta Photon and Spark running environments

If your Datetime data does not contain time zone information, by default:

- Spark uses the time zone of the Trifacta node for Datetime values.

  > **Tip:** This use of time zone applies to any Spark-based running environment, such as EMR.

- Trifacta Photon uses the UTC time zone for Datetime values.

This difference in how the values are treated can result in differences in Datetime-based calculations, such as the DATEDIF function.

Workarounds:

You can do one of the following:

- Set the time zone for the Trifacta node to be UTC. You must also set the time zone for your Spark running environment to UTC.
- Apply the following Spark execution properties from the Run Job page:

```
"spark":
  "props": {
    ...
    "spark.driver.extraJavaOptions" : "-Duser.timezone=\"UTC\"",
    "spark.executor.extraJavaOptions" : "-Duser.timezone=\"UTC\""
  }
  ...
}
```

Datetime Schema via API

When Datetime data is returned via API calls, the schema for this information is returned as a three-element array. The additional elements to the specific are required to account for formatting options of for Datetime values.

> **Tip:** Schema information for data types is primarily available via API calls. You may find schema information for columns in JSON versions of the visual profile and flow definitions when they are exported.

Example:

```
"end_date": [
            "Datetime",
            "mm-dd-yy",
            "mm*dd*yyyy"
        ]
```

Array Element	Description	Example 1	Example 2
Data type	The internal name for the data type. For Datetime columns, this schema value should always be `Datetime`.	`"Datetime"`	`"Datetime"`
Sub-format	The general format category of the data type	`"mm-dd-yy"`	`"mm-dd-yy"`
Format type	The specific formatting for the data type	`"mm*dd*yyyy"`	`"shortMonth*dd*yy"`