Command Reference: FormatTableDateTime()

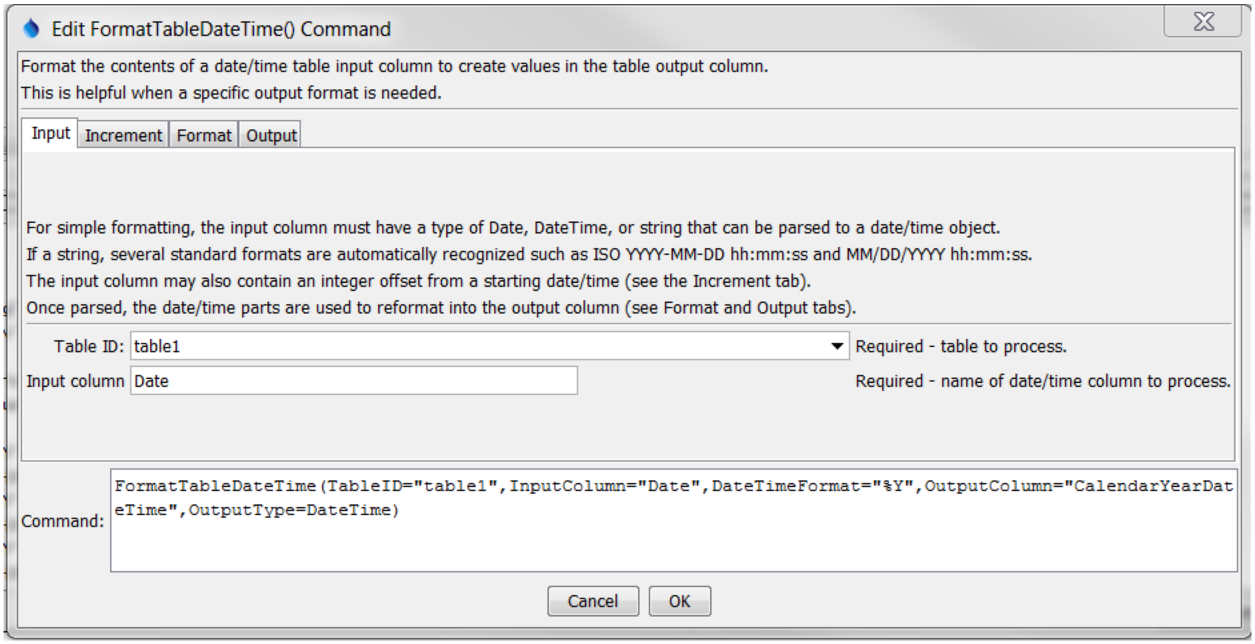
Format a date/time column in a table

Version 11.08.00, 2016-02-01

The `FormatTableDateTime()` command formats a date/time input column from a table to create a table output column. For example, it may be necessary to reformat a date/time column into an object type that is more suitable for reporting, further processing, or export to a spreadsheet. See also the `FormatTableString()` command, which manipulates strings. Formatting occurs as follows:

1. The date/time input column value is parsed into internal date/time object. Currently there is no command parameter to specify the format of the input column and consequently standard formats are expected (ISO `YYYY-MM-DD hh:mm:ss` or `MM/DD/YYYY hh:mm:ss` of varying precision):
   - If the input column is not an increment (***Increment*** tab parameters are blank) then the input column is parsed directly to a date/time object.
   - If the input column is an increment from a starting date/time (***Increment*** tab parameters are not blank), the date/time object is computed as the offset from the starting date/time, for example the number of hours since the start.
2. The date/time object from the previous step is formatted into a string using the format specifier string specified by the `FormatterType` and `DateTimeFormat` parameters. Missing values in input will result in blanks (nulls) in output.
3. The string is converted into the final output column type by specifying the `OutputType` parameter:
   - DateTime output might be used to create date/time objects with less precision that the original input column (for example to truncate `hh:mm:ss` that is superfluous).
   - Integer or double types can be created if the date/time output string from the previous step contains integer or floating-point number, for example `YYYY` or `YYYY.MM`
   - String outputs the string from the previous step.

The following dialog is used to edit the command and illustrates the syntax of the command.



Format the contents of a date/time table input column to create values in the table output column.
This is helpful when a specific output format is needed.

For simple formatting, the input column must have a type of Date, DateTime, or string that can be parsed to a date/time object.
If a string, several standard formats are automatically recognized such as ISO YYYY-MM-DD hh:mm:ss and MM/DD/YYYY hh:mm:ss.
The input column may also contain an integer offset from a starting date/time (see the Increment tab).
Once parsed, the date/time parts are used to reformat into the output column (see Format and Output tabs).

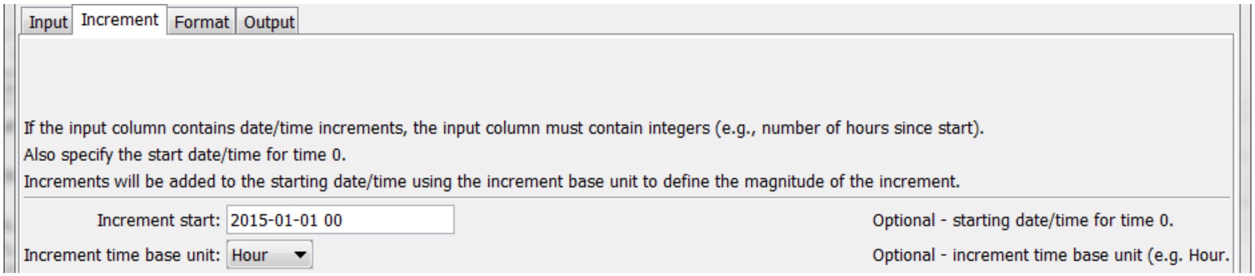
Table ID: table1 — Required - table to process.
Input column: Date — Required - name of date/time column to process.

Command:

```
FormatTableDateTime(TableID="table1",InputColumn="Date",DateTimeFormat="%Y",OutputColumn="CalendarYearDateTime",OutputType=DateTime)
```

FormatTableDateTime

FormatTableDateTime() Command Editor Showing Input Parameters

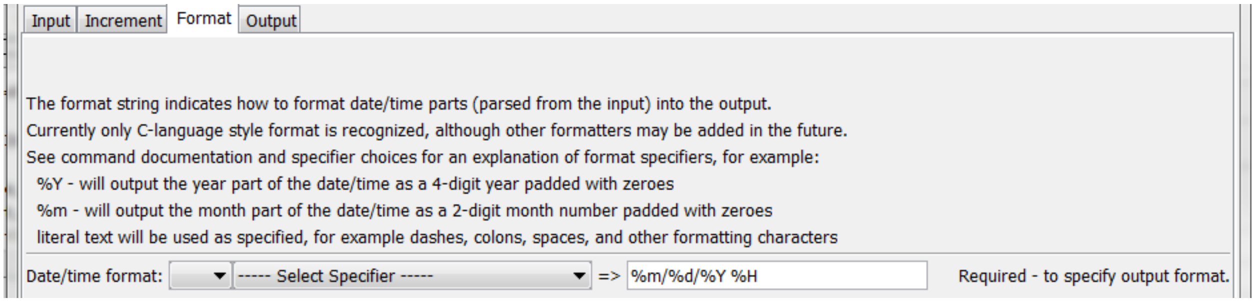


If the input column contains date/time increments, the input column must contain integers (e.g., number of hours since start).
Also specify the start date/time for time 0.
Increments will be added to the starting date/time using the increment base unit to define the magnitude of the increment.

Increment start: 2015-01-01 00 Optional - starting date/time for time 0.
Increment time base unit: Hour Optional - increment time base unit (e.g. Hour.

FormatTableDateTime_Increment

FormatTableDateTime() Command Editor Showing Increment Parameters

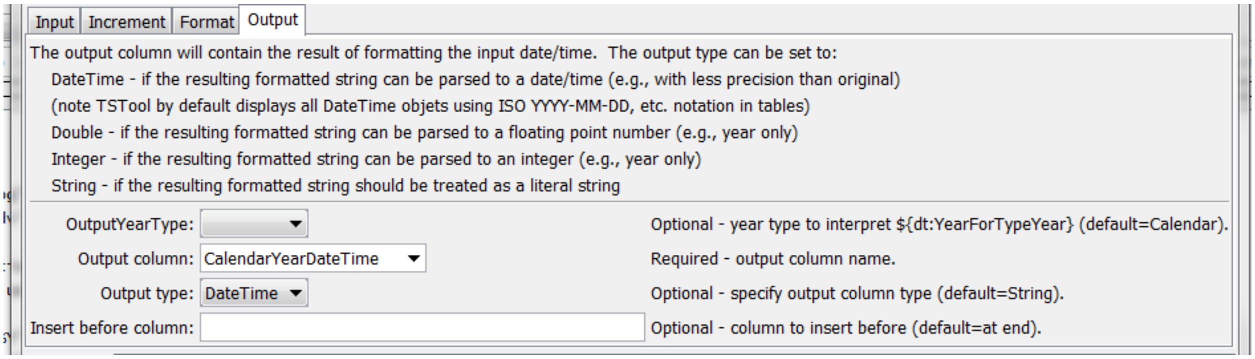


The format string indicates how to format date/time parts (parsed from the input) into the output.
Currently only C-language style format is recognized, although other formatters may be added in the future.
See command documentation and specifier choices for an explanation of format specifiers, for example:
%Y - will output the year part of the date/time as a 4-digit year padded with zeroes
%m - will output the month part of the date/time as a 2-digit month number padded with zeroes
literal text will be used as specified, for example dashes, colons, spaces, and other formatting characters

Date/time format: ----- Select Specifier ----- => %m/%d/%Y %H Required - to specify output format.

FormatTableDateTime_Format

FormatTableDateTime() Command Editor Showing Format Parameters



The output column will contain the result of formatting the input date/time. The output type can be set to:
DateTime - if the resulting formatted string can be parsed to a date/time (e.g., with less precision than original)
(note TSTool by default displays all DateTime objets using ISO YYYY-MM-DD, etc. notation in tables)
Double - if the resulting formatted string can be parsed to a floating point number (e.g., year only)
Integer - if the resulting formatted string can be parsed to an integer (e.g., year only)
String - if the resulting formatted string should be treated as a literal string

OutputYearType: Optional - year type to interpret ${dt:YearForTypeYear} (default=Calendar).
Output column: CalendarYearDateTime Required - output column name.
Output type: DateTime Optional - specify output column type (default=String).
Insert before column: Optional - column to insert before (default=at end).

FormatTableDateTime_Output

FormatTableDateTime() Command Editor Showing Output Parameters

The command syntax is as follows:

```
FormatTableDateTime(Parameter=Value,…)
```

Command Parameters

Parameter	Description	Default
`TableID`	The identifier for the table to process. Can be specified using processor `${Property}`.	None – must be specified.
`InputColumn`	The name of the input date/time column to process. The column can contain date/time objects or strings that can be parsed into date/time objects. If `IncrementStart` is specified, this column should contain integers that indicate the offset from the increment start. Can be specified using processor `${Property}`.	None – must be specified.

Parameter	Description	Default
IncrementStart	When input column is an increasing time increment, specify the starting date/time. Can be specified using processor ${Property}.	Do not use increment.
Increment BaseUnit	When input column is an increasing time increment, specify the base unit for increment values: Minute, Hour, Day, Year.	Do not use increment.
FormatterType	The date/time formatter type that defines DateTimeFormat: • C – the C programming language strftime() function, which has been widely copied (described below). • MS – Microsoft convention (currently not supported but may be added in the future).	C
DateTimeFormat	The format specifier string used to format the date/time values. Specify as many format specifiers as appropriate. All other characters will be transferred to the output string. See the table below for valid specifiers. Can be specified using processor ${Property}.	None – must be specified.
OutputYearType	Indicate the year type used to transform the date/time to an output. For example, specify OutputYearType=Water and DateTimeFormat=${dt:YearForYearType} to output the water year corresponding to the input date/time.	
OutputColumn	The name of the column to receive the output. If the column does not exist in the table it will be created, considering OutputType. Can be specified using processor ${Property}.	None – must be specified.
OutputType	Specify if the output column should be other than a String. Successful conversion to the output type requires that the format string result is consistent with the desired output type.	String
InsertBefore Column	The name of the column before which the output column should be inserted (if the output column needs to be created). Can be specified using processor ${Property}.	Insert at the end of the table.

The following table lists the supported format strings for FormatterType=C:

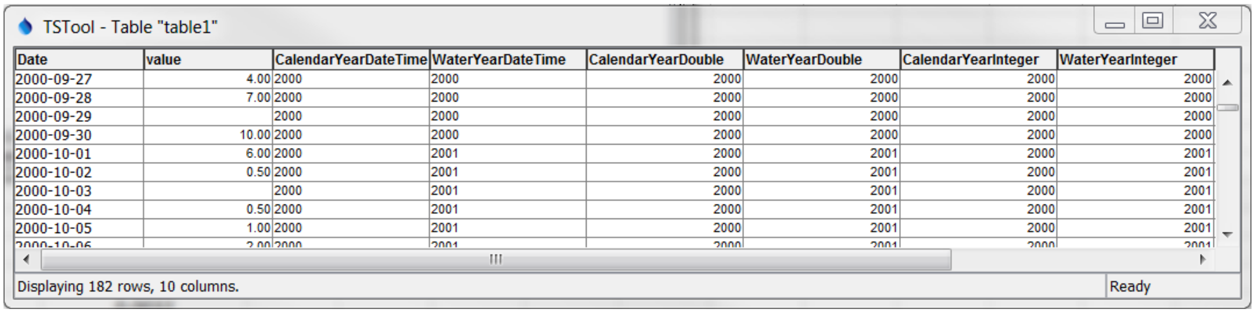
Supported C (Strftime) Format Specifiers

Format Specifier	Description
%a	Weekday abbreviation (e.g., Sun)
%A	Weekday (e.g., Sunday).
%b	Month abbreviation (e.g., Jan).
%B	Month (e.g., January).
%d	Day (01-31).
%H	Hour (00-23).
%I	Hour (01-12).
%j	Day of year (001-366).
%m	Month (01-12).
%M	Minute (00-59).
%p	AM, PM (noon=PM, midnight=AM).
%S	Second (00-59).

Format Specifier	Description
%s	Number of seconds since Jan 1, 1970 00:00:00
%y	Year (00-99).
%Y	Year (0000-9999).
%Z	Time zone (e.g., MST).
${dt:YearForYearType}	4-digit year for the given OutputYearType.

The following example illustrates how to convert an input date/time column into variations of the date/time, with the following input used to generate the WaterYearDateTime column (surrounding quotes will be added automatically by command editor). Note the change in value of the water year.

- InputColumn=Date
- DateTimeFormat=%Y
- OutputYearType=Water
- OutputColumn=WateryearDateTime
- OuputType=DateTime



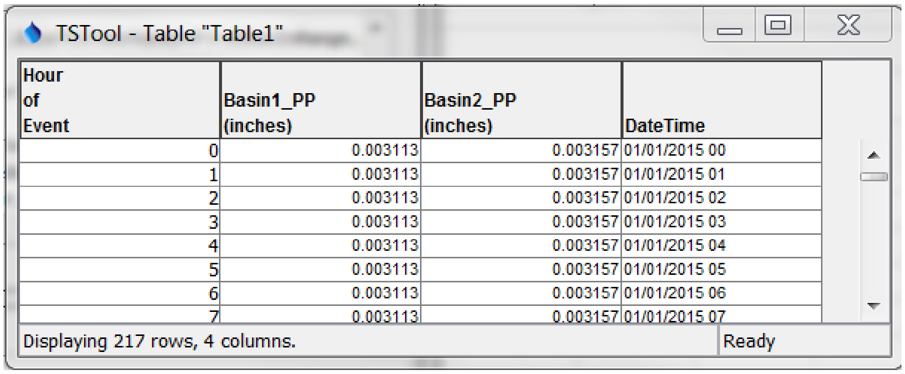
TSTool - Table "table1"

Date	value	CalendarYearDateTime	WaterYearDateTime	CalendarYearDouble	WaterYearDouble	CalendarYearInteger	WaterYearInteger
2000-09-27	4.00	2000	2000	2000	2000	2000	2000
2000-09-28	7.00	2000	2000	2000	2000	2000	2000
2000-09-29		2000	2000	2000	2000	2000	2000
2000-09-30	10.00	2000	2000	2000	2000	2000	2000
2000-10-01	6.00	2000	2001	2000	2001	2000	2001
2000-10-02	0.50	2000	2001	2000	2001	2000	2001
2000-10-03		2000	2001	2000	2001	2000	2001
2000-10-04	0.50	2000	2001	2000	2001	2000	2001
2000-10-05	1.00	2000	2001	2000	2001	2000	2001
2000-10-06	2.00	2000	2001	2000	2001	2000	2001

Displaying 182 rows, 10 columns. Ready

The following example illustrates how to convert increment data into a full date/time column string, with the following input (surrounding quotes will be added automatically by command editor):

- InputColumn=Hour of Event
- IncrementStart=2015-01-01 00
- IncrementBaseUnit=Hour
- DateTimeFormat=%m/%d/%Y %H
- OutputColumn=DateTime



TSTool - Table "Table1"

Hour of Event	Basin1_PP (inches)	Basin2_PP (inches)	DateTime
0	0.003113	0.003157	01/01/2015 00
1	0.003113	0.003157	01/01/2015 01
2	0.003113	0.003157	01/01/2015 02
3	0.003113	0.003157	01/01/2015 03
4	0.003113	0.003157	01/01/2015 04
5	0.003113	0.003157	01/01/2015 05
6	0.003113	0.003157	01/01/2015 06
7	0.003113	0.003157	01/01/2015 07

Displaying 217 rows, 4 columns. Ready