

Command Reference: ManipulateTableString()

Manipulate string a string column in a table

Version 11.08.00, 2016-02-03

The `ManipulateTableString()` command manipulates a string column in a table. For example, it may be necessary to manipulate strings in a table in order to match time series identifier parts, so that lookups can occur. The input is specified by:

- a table column name (`InputColumn1`)
- optionally, either a second input column name (`InputColumn2`) or a constant string value (`InputValue2`), depending on operator
- optionally, some operators require an additional input value (`InputValue3`)

The result is placed in the output column (`OutputColumn`). Missing/blank input will be considered as empty strings when formatting the output. The output column can be the same as an existing table column. The following dialog is used to edit the command and illustrates the syntax of the command (in this case illustrating how the contents of column `String2` are prepended to the contents of a column named `String1` and placed in the output column `String3`). Mouse over the various parameter fields to see information about whether the parameters are required for the selected operator.

Edit ManipulateTableString() Command

Manipulate a column of string data in a table, using one of the following approaches:

- process input values from two columns (`InputColumn1` and `InputColumn2`) to populate the output column (e.g. operator `Append`, `Prepend`)
- process input values from a column and a constant (`InputColumn1` and `InputValue2`) to populate the output column (e.g. operator `Append`, `Prepend`)
- process input values from a column and two constants (`InputColumn1`, `InputValue2`, `InputValue3`) to populate the output column (e.g. operator `Replace`)
- convert an input string value from a column (`InputColumn1`) into an output value (e.g. operator `ToDate`, `ToDateTime`)

The operator defines the number of values that are needed as input - mouse over the input fields for feedback on what is needed.

Table ID:	Table1	Required - table to process.
Column filters to include rows:		Optional - column patterns to include rows (default=include all). <input type="button" value="Edit"/>
Column filters to exclude rows:		Optional - column patterns to exclude rows (default=include all). <input type="button" value="Edit"/>
Input column 1:	String1	Required - first input column name.
String operator:	Prepend	Required - string manipulation to perform on input.
Input column 2:		Required if no input value 2 - second input column name.
Input value 2:	Prefix:	Required if no input column 2 - constant string.
Input value 3:		Required - only used by some operators.
Output column:	String3	Required - output column name.

Command:

```
ManipulateTableString (TableID="Table1", InputColumn1="String1", Operator="Prepend", InputValue2="Prefix:", OutputColumn="String3")
```

ManipulateTableString

ManipulateTableString() Command Editor

The command syntax is as follows:

```
ManipulateTableString (Parameter=Value,...)
```

Command Parameters

Parameter	Description	Default
TableID	The identifier for the table to process. Can be specified with <code>\${Property}</code> notation.	None – must be specified.
Column Include Filters	Specify values to match to include rows using syntax: <code>Column1:Value1,Column2,Value2</code> , where values can use <code>*</code> for wildcard. All values must be matched to include a row. Can be specified with <code>\${Property}</code> notation.	Include all rows.
Column Exclude Filters	Specify values to match exclude rows using syntax: <code>Column1:Value1,Column2,Value2</code> , where values can use <code>*</code> for wildcard. All values must be matched to exclude a row. Can be specified with <code>\${Property}</code> notation.	Include all rows.
Input Column1	The name of a column containing strings, as the first input. Can be specified with <code>\${Property}</code> notation.	None – must be specified.
Operator	The operation to perform on the input strings: <ul style="list-style-type: none"> Append – append the second input to the first input (requires 2 inputs) Prepend – prepend the second input before the first input (requires 2 inputs) Replace – start with the first input, replace the substring indicated by the second input with that of the third input (requires 3 inputs) Substring – split out a substring from the first input, where the second input is the starting character position (1+) and the optional third input is the ending character position (1+) (requires 2 or 3 inputs) ToDate – convert the first input to a DateTime object with date precision ToDateTime – convert the first input to a DateTime object ToDouble – convert the first input to a double precision object ToInteger – convert the first input to an integer object 	None – must be specified.
Input Column2	The name of a column containing strings, as the second input. Can be specified with <code>\${Property}</code> notation.	Required if a 2 nd input value is needed no <code>InputValue2</code> .
Input Value2	A string constant, as the second input. Can be specified with <code>\${Property}</code> notation. For Replace operator, use <code>^</code> to indicate start of line, <code>\$</code> to indicate end of line and <code>\s</code> to indicate space.	Required if a 2 nd input value is needed and no <code>InputColumn2</code> .
Input Value3	A string constant, as the third input. Can be specified with <code>\${Property}</code> notation. See note for <code>InputValue2</code> for Replace operator.	Required if a 3 rd input value is needed.
Output Column	The name of a column to receive the output. Can be specified with <code>\${Property}</code> notation.	None – must be specified.