Command Reference: ManipulateTableString()

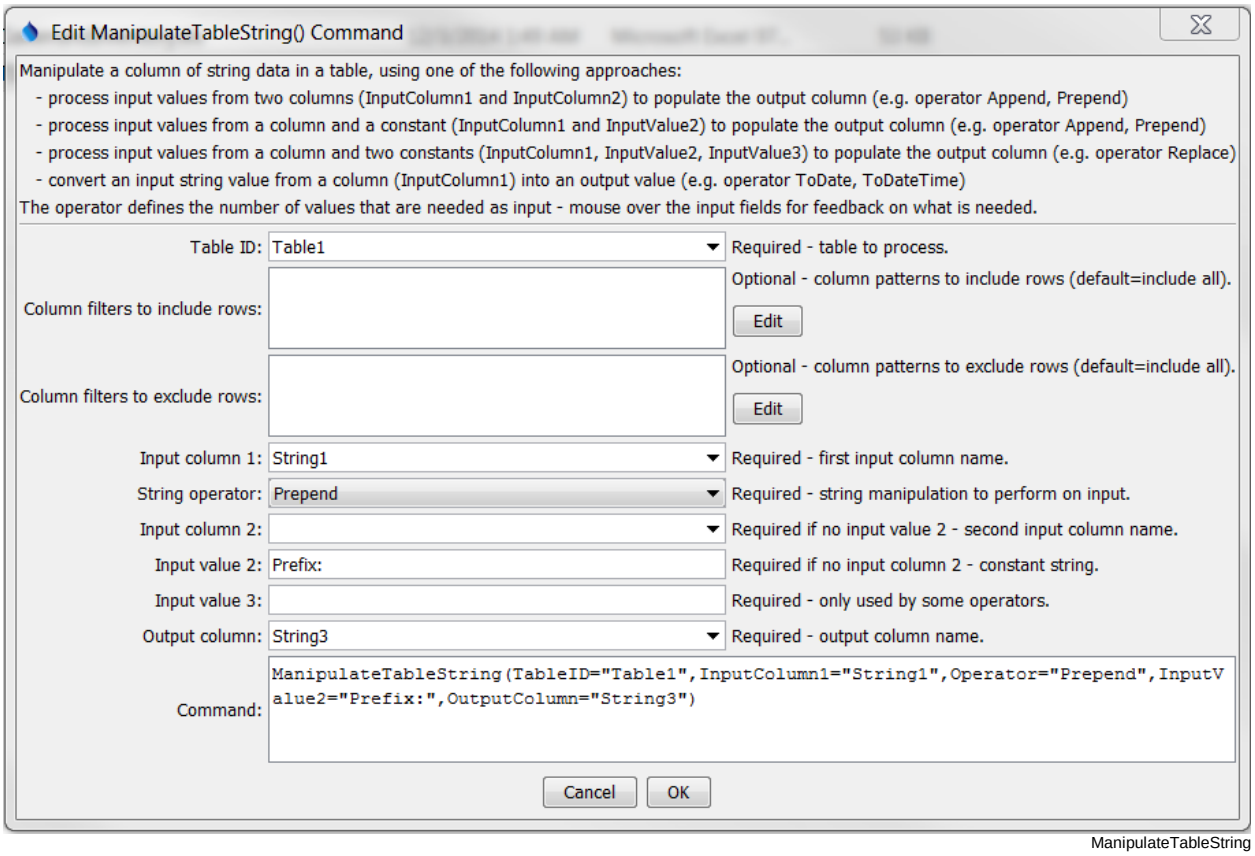
Manipulate string a string column in a table

Version 11.11.00, 2016-05-11

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.



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 `${Property}` notation.	None – must be specified.
`Column Include Filters`	Specify values to match to include rows using syntax: `Column1:Value1,Column2,Value2`, where values can use `*` for wildcard. All values must be matched to include a row. Can be specified with `${Property}` notation.	Include all rows.
`Column Exclude Filters`	Specify values to match exclude rows using syntax: `Column1:Value1,Column2,Value2`, where values can use `*` for wildcard. All values must be matched to exclude a row. Can be specified with `${Property}` notation.	Include all rows.
`Input Column1`	The name of a column containing strings, as the first input. Can be specified with `${Property}` notation.	None – must be specified.
`Operator`	The operation to perform on the input strings: • `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) • `Remove` – start with the first input, remove the substring indicated by the second input (requires 2 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 `${Property}` notation.	Required if a 2nd input value is needed no `InputValue2`.
`Input Value2`	A string constant, as the second input. Can be specified with `${Property}` notation. For `Replace` operator, use `^` to indicate start of line, `$` to indicate end of line and `\s` to indicate space.	Required if a 2nd input value is needed and no `InputColumn2`.
`Input Value3`	A string constant, as the third input. Can be specified with `${Property}` notation. See note for `InputValue2` for `Replace` operator.	Required if a 3rd input value is needed.
`Output Column`	The name of a column to receive the output. Can be specified with `${Property}` notation.	None – must be specified.