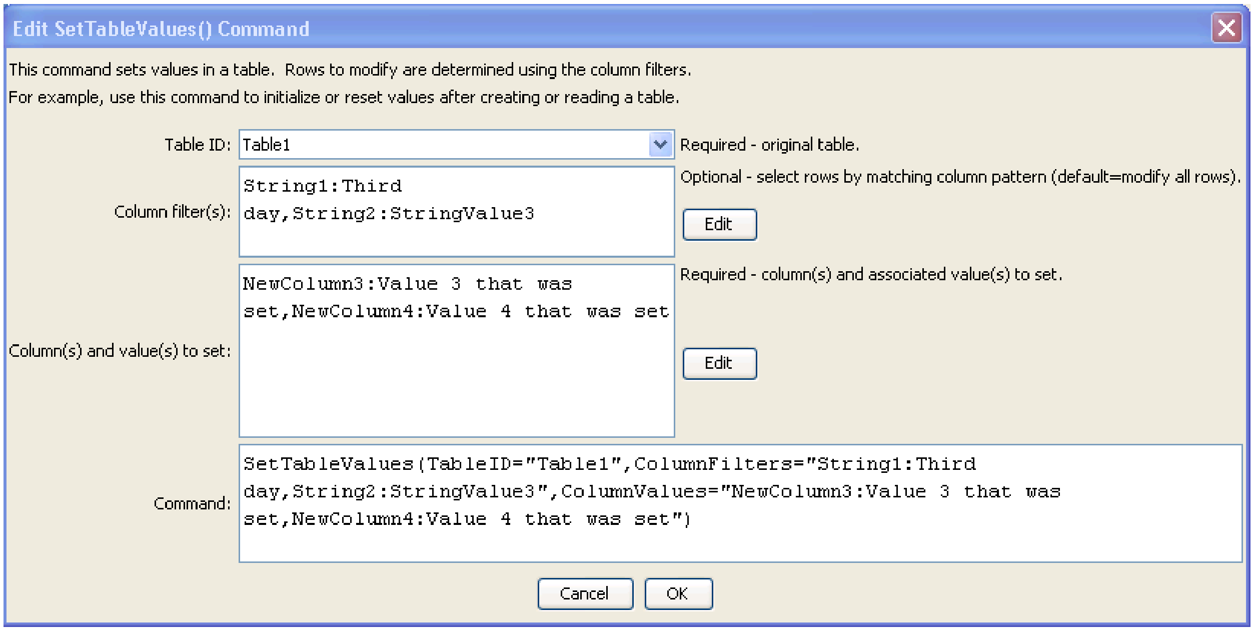
Command Reference: SetTableValues()

Set values in a table by first matching specific rows

Version 10.22.00, 2013-08-77

The `SetTableValues()` command sets values in a table. It does so by first optionally matching rows by filtering on column values (see `ColumnFilters` parameter), and then setting values in specified columns (see `ColumnValues` parameter). For example, this command can be used to set or override table values that were not included in the original data. See also the `FormatTableString()` and `ManipulateTableString()` commands.

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



Edit SetTableValues() Command

This command sets values in a table. Rows to modify are determined using the column filters.
For example, use this command to initialize or reset values after creating or reading a table.

Table ID: Table1 — Required - original table.

Column filter(s): String1:Third day,String2:StringValue3 — Optional - select rows by matching column pattern (default=modify all rows).

Column(s) and value(s) to set: NewColumn3:Value 3 that was set,NewColumn4:Value 4 that was set — Required - column(s) and associated value(s) to set.

Command:

```
SetTableValues(TableID="Table1",ColumnFilters="String1:Third
day,String2:StringValue3",ColumnValues="NewColumn3:Value 3 that was
set,NewColumn4:Value 4 that was set")
```

SetTableValues

SetTableValues() Command Editor

The command syntax is as follows:

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

Command Parameters

Parameter	Description	Default
`TableID`	The identifier for the table being modified.	None – must be specified.
`ColumnFilters`	Filters that limit the number of rows being modified, using the syntax: `FilterColumn1:FilterPattern1, FilterColumn2:FilterPattern2` Patterns can use `*` to indicate wildcards for matches. Only string values can be checked (other data types are converted to strings for comparison). Comparisons are case-independent. All patterns must be matched in order to match the row.	No filtering.
`ColumnMap`	The new names for the output columns, using syntax: `OriginalColumn1:NewColumn1, OriginalColumn2:NewColumn2`	The column names in the copy will be the same as in the original table.