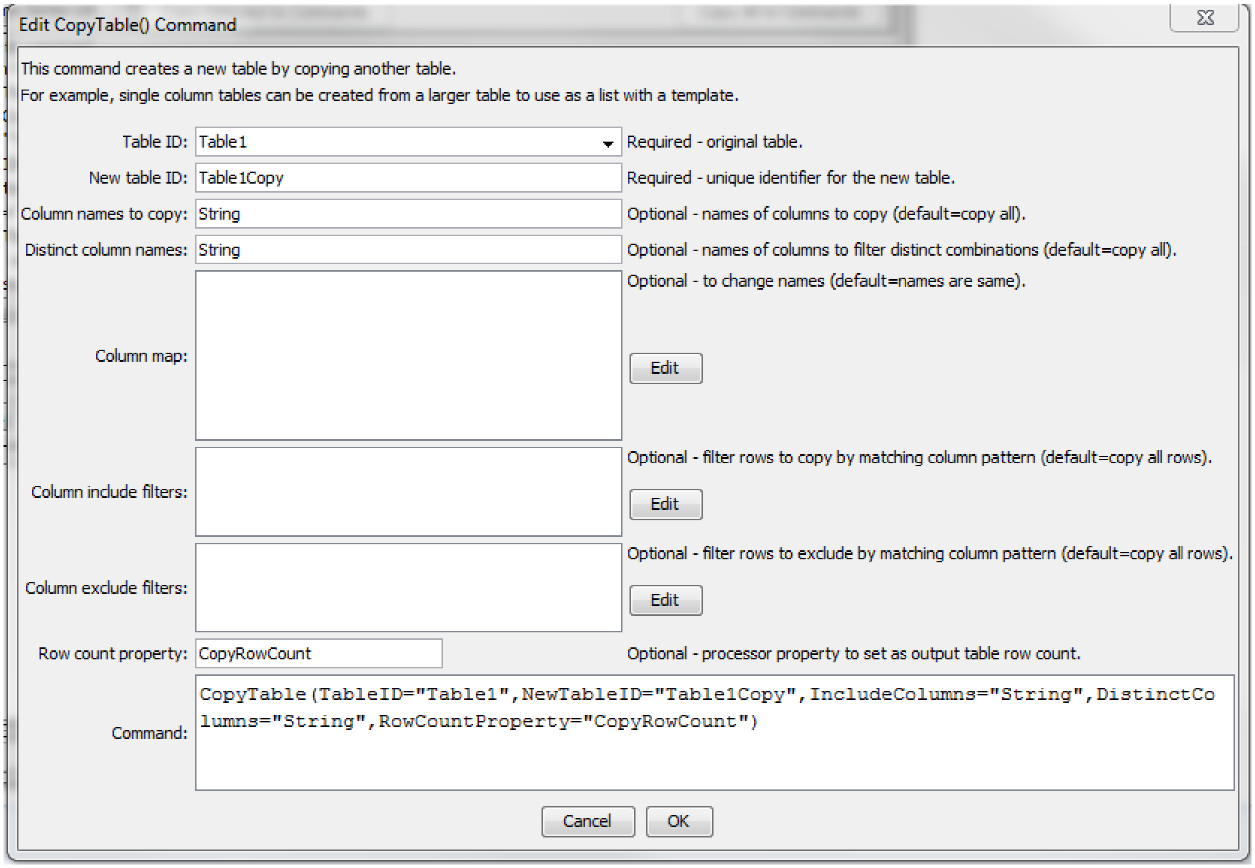
Command Reference: CopyTable()

Create a table as a full or partial copy of a table

Version 11.03.00, 2015-06-01

The `CopyTable()` command copies all or a subset of the columns and rows from one table to create a new table. For example, this is useful to create one-column lists that can be used to expand template files with the `ExpandTemplateFile()` command, or to create a subset of a table to output to a file or write to a database (for example filter a large table by geographic area or other criteria). The following dialog is used to edit the command and illustrates the syntax of the command (in this case illustrating how values in a column named `LocationID` are copied to a new table).



This command creates a new table by copying another table.
For example, single column tables can be created from a larger table to use as a list with a template.

Table ID:	Table1	Required - original table.
New table ID:	Table1Copy	Required - unique identifier for the new table.
Column names to copy:	String	Optional - names of columns to copy (default=copy all).
Distinct column names:	String	Optional - names of columns to filter distinct combinations (default=copy all).
Column map:		Optional - to change names (default=names are same).
Column include filters:		Optional - filter rows to copy by matching column pattern (default=copy all rows).
Column exclude filters:		Optional - filter rows to exclude by matching column pattern (default=copy all rows).
Row count property:	CopyRowCount	Optional - processor property to set as output table row count.

Command:

```
CopyTable(TableID="Table1",NewTableID="Table1Copy",IncludeColumns="String",DistinctColumns="String",RowCountProperty="CopyRowCount")
```

CopyTable

CopyTable() Command Editor

The command syntax is as follows:

```
CopyTable(Parameter=Value,…)
```

Command Parameters

Parameter	Description	Default
`TableID`	The identifier for the original table. Can be specified using processor `${Property}`.	None – must be specified.
`NewTableID`	The identifier for the new table. Can be specified using processor `${Property}`.	None – must be specified.

Parameter	Description	Default
`IncludeColumns`	The names of columns to copy, separated by commas.	Copy all of the columns.
`DistinctColumns`	The names of columns to copy, separated by commas. Only distinct values from the specified column(s) will be copied. For example, if column `A` contains strings `X`, `Y`, `Z`, `Y`, `C`, the resulting distinct value column will have rows with `X`, `Y`, `Z`, `C`. The following behavior is implemented: • If multiple column names are specified, the unique combination of the values will be checked. • If only the `DistinctColumns` are to be in the copy, specify the columns with `IncludeColumns`. • If columns other than `DistinctColumns` are in `IncludeColumns`, then the distinct row will be output. • Prior to TSTool 10.26.00, `DistinctColumns` would override `IncludeColumns`. • Column values null, and blank strings are not considered in distinct comparisons.	Don't do a distinct comparison.

`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.

`ColumnFilters`	Filters that limit the number of rows being copied, 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 copy the row. Can be specified using processor `${Property}`.
		No filtering.

`ColumnExclude Filters`	Filters that exclude rows being copied, by matching column values:
	`ColumnExcludeFilter1:FilterPattern1,`
	`ColumnExcludeFilter2:FilterPattern2`
	Patterns can use * to indicate wildcards for matches. Only string values can be checked (other data types are converted to strings for comparison). All patterns must be matched to exclude the row.	

`RowCount Property`	The row count for the copy will be set as a processor property. This is useful for error-checks (e.g., check that number of time series read with `ReadTimeSeriesList()` matches the expected count. Can be specified using processor `${Property}`.	