Command Reference: For()

Start a block of commands as part of a "for" loop

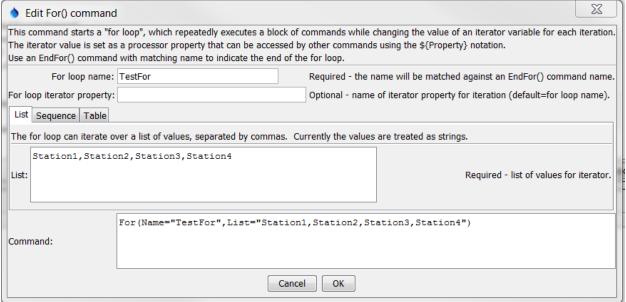
Version 12.03.00, 2016-06-28

The For () command iterates through a block of commands between For () and matching EndFor () commands. A processor property is set to the value of the iteration property and can be used by other commands that support properties, using the \${Property} notation. This command is an alternative to implementing loops in templates (see ExpandTemplateFile()), in particular for straightforward command logic. For () commands can iterate over:

- a list of supplied values
- a sequence of integers or floating-point double precision numbers specified with start, end, and increment
- values from a table column

For () commands can be nested. Status messages for run mode are accumulated in each command (this update is occurring over time – status messages for some commands may be cleared out each iteration). A limitation of using For () with properties is that command when edited may show time series identifiers and other command parameters as \${Property} values, rather than actual data, because the values get expanded at run-time. This provides increased processing power but errors may not be evident until commands re run.

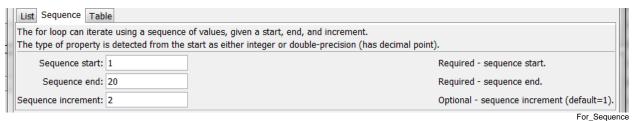
The following dialog is used to edit this command and illustrates the command syntax.



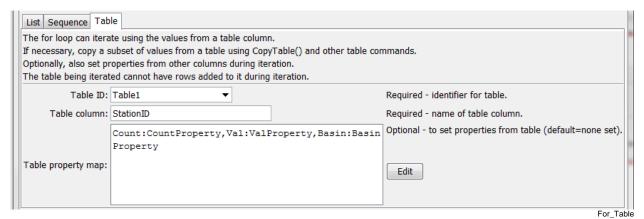
For() Command Editor Illustrating Using a List for Iteration Values

For_List

For() Command TSTool Documentation



For() Command Editor Illustrating Using a Sequence of Integers for Iteration Values



For() Command Editor Illustrating Using a Table for Iteration Values

The command syntax is as follows:

For (Parameter=Value,...)

Command Parameters

Parameter	Description	Default
Name	The name of the "for" loop, which will	None – must be specified.
	be matched with the name of an	
	EndFor() command to indicate the	
	block of commands in the loop.	
IteratorProperty	The processor property that will be set	Same as Name.
	to the iterator property. The object type	
	will depend on that used to provide the	
	iteration property list. For example, if a	
	column of strings from a table is used	
	for iteration, the property will contain a	
	string.	
List	A list of comma-separated values to be	None if list is used – must
	used as variables for the iteration.	specified a list of values.
SequenceStart	Starting value when a sequence is	None if sequence is used.
	specified for iteration, an integer or	
	floating-point number (with decimal).	
	Can be specified with \${Property}.	
SequenceEnd	Ending value for sequence. Can be	None – must be specified if
	specified with \${Property}.	sequence is used.

TSTool Documentation For() Command

Parameter	Description	Default
SequenceIncrement	Increment for sequence iterator.	1 or 1.0 depending on
		SequenceStart type.
TableID	The table identifier, when specifying the	None if table is used – must
	iterator as a column from a table. Can	specify the table ID.
	be specified with processor	
	\${Property}.	
TableColumn	The table column name, when	None – must be specified if
	specifying the iterator as a column from	table is used.
	a table.	
TablePropertyMap	Specify the names of column names and	None – only the iterator column
	corresponding processor property names	value will be set as a property
	to set. This allows other commands to	using IteratorProperty.
	access the values of those properties	
	using \${Property} notation. Specify	
	using format:	
	ColumnName1:PropertyName1,	
	ColumnName2:PropertyName2	

The following example illustrates a simple For () and EndFor () usage. In this example the StationID column in the input table is used to provide the list of values to iterate over. The following input table is a delimited file but could come from another source:

```
# Test table data for For() command tests
"Count","Val","StationID","Basin"
1,1.0,Station1,Basin1
2,2.0,Station2,Basin2
3,3.0,Station3,Basin3
4,4.0,Station4,Basin4
```

The following command file reads the above input table, iterates over the StationID column, and creates a simple output file:

```
ReadTableFromDelimitedFile(TableID="Table1",InputFile="Data\testtable.csv")

RemoveFile(InputFile="Results/Test_For_TableString_out.txt",IfNotFound=Ignore)

For(Name="TestFor",TableID="Table1",TableColumn="StationID")

WritePropertiesToFile(OutputFile="Results/Test_For_TableString_out.txt",

IncludeProperty="TestFor",WriteMode=Append,FileFormat=NameTypeValue)

EndFor(Name="TestFor")
```

The resulting output file is as follows:

```
TestFor="Station1"
TestFor="Station2"
TestFor="Station3"
TestFor="Station4"
```

For() Command TSTool Documentation

This page is intentionally blank.