Command Reference: ExpandTemplateFile()

Process a template file to create the fully-expanded file

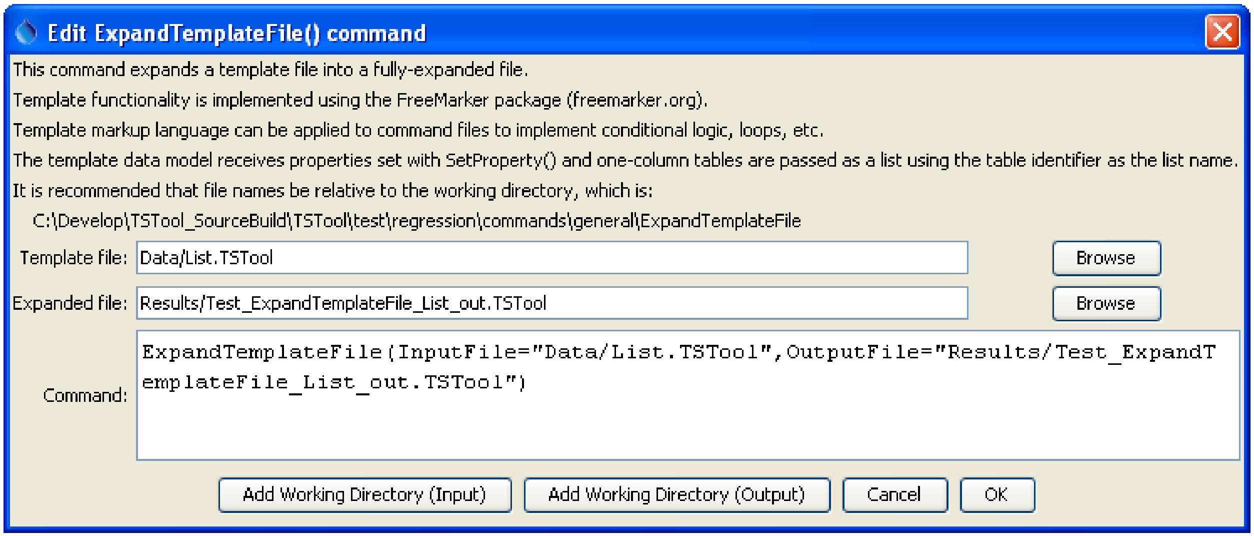
Version 09.09.00, 2010-09-23

The `ExpandTemplateFile()` command processes a template file (typically a command file or time series product file but can be any text file) to create a fully-expanded file. Templates facilitate utilizing conditional logic, loops, and other dynamic processing functionality. For example, a template can be used to repeat commands for multiple location identifiers. Templates can also be applied to other text files.

The FreeMarker software (http://freemarker.org) is used to implement templates. Refer to the online documentation for information about the markup language used to create templates. The built-in `normalizeNewlines` user directive is automatically used to ensure that expanded files use newline characters appropriate for the operating system – this leads to extra first and last lines in the template during processing.

Properties set with the `SetProperty()` command are passed to the template tool. One-column tables are also passed as lists, using the table identifier as the property name. For example, use the `CopyTable()` command to create a one-column table that can be used as a list for template expansion.

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


Edit ExpandTemplateFile() command
Browse
Browse
Add Working Directory (Input)
Add Working Directory (Output)
Cancel
OK

This command expands a template file into a fully-expanded file.
Template functionality is implemented using the FreeMarker package (freemarker.org).
Template markup language can be applied to command files to implement conditional logic, loops, etc.
The template data model receives properties set with SetProperty() and one-column tables are passed as a list using the table identifier as the list name.
It is recommended that file names be relative to the working directory, which is:
C:\Develop\TSTool_SourceBuild\TSTool\test\regression\commands\general\ExpandTemplateFile

Template file: Data/List.TSTool

Expanded file: Results/Test_ExpandTemplateFile_List_out.TSTool

Command:

```
ExpandTemplateFile(InputFile="Data/List.TSTool",OutputFile="Results/Test_ExpandT
emplateFile_List_out.TSTool")
```

ExpandTemplateFile

ExpandTemplateFile() Command Editor

The command syntax is as follows:

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

Command Parameters

Parameter	Description	Default
`InputFile`	The name of the template file to process.	None – must be specified.
`OutputFile`	The name of the expanded output file.	None – must be specified.

The following example illustrates a simple template command file and expanded result:

```
# Simple test to expand a text file using FreeMarker
<#assign message="Hello World">
${message}
```

```
# Simple test to expand a text file using FreeMarker
Hello World
```

The following example illustrates a template command file and expanded result to repeat a command for a list of location identifiers. A block of multiple commands can be repeated, as appropriate. Long lines are indented for illustration but would exist on a single line without indentation in the template file.

```
# Simple template to illustrate how to repeat commands with a list of
location identifiers
# Create a time series for each location
# The following ensures that the created template is read-only, so users
# modify the template instead:
#@readOnly
<#assign setStart = "2000-01-01">
<#assign setEnd = "2000-03-15">
<#assign units = "CFS">
<#assign locList = ["loc1", "loc2", "loc3", "loc4"]>
<#list locList as loc>
TS ${loc} =
NewPatternTimeSeries(NewTSID="${loc}..Streamflow.Day",SetStart="${setStart}",
  SetEnd="${setEnd}",Units="${units}",PatternValues="${loc_index + 1},0")
</#list>
```

```
# Simple template to illustrate how to repeat commands with a list of location identifiers
# Create a time series for each location
# The following ensures that the created template is read-only, so users
# modify the template instead:
#@readOnly
TS loc1 = NewPatternTimeSeries(NewTSID="loc1..Streamflow.Day",
    SetStart="2000-01-01",SetEnd="2000-03-15",
    Units="CFS",PatternValues="1,0")
TS loc2 = NewPatternTimeSeries(NewTSID="loc2..Streamflow.Day",
    SetStart="2000-01-01",SetEnd="2000-03-15",
    Units="CFS",PatternValues="2,0")
TS loc3 = NewPatternTimeSeries(NewTSID="loc3..Streamflow.Day",
    SetStart="2000-01-01",SetEnd="2000-03-15",
    Units="CFS",PatternValues="3,0")
TS loc4 = NewPatternTimeSeries(NewTSID="loc4..Streamflow.Day",
    SetStart="2000-01-01",SetEnd="2000-03-15",
    Units="CFS",PatternValues="4,0")
```