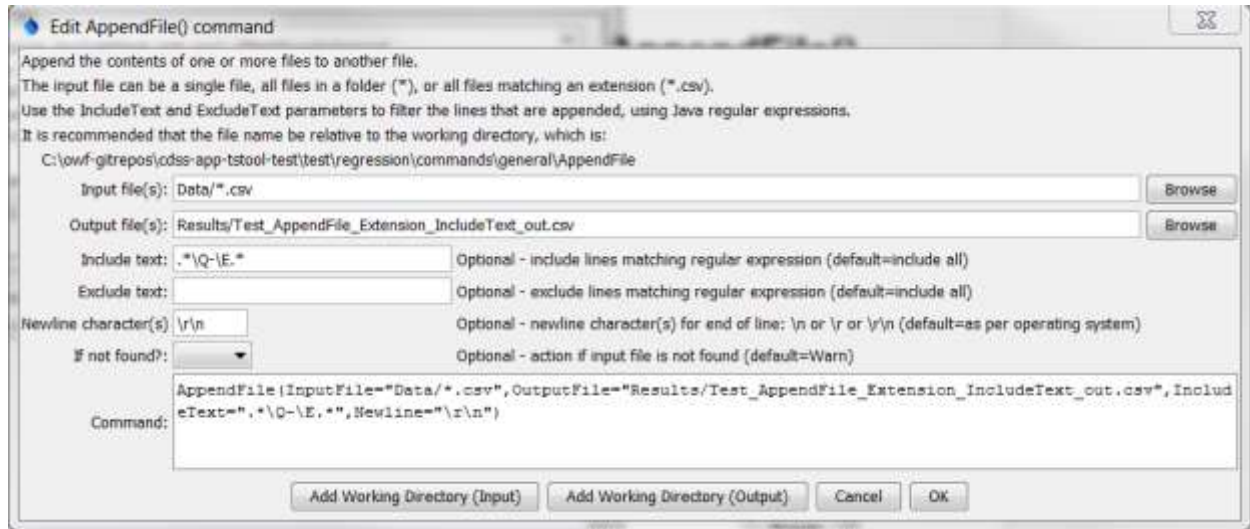# Command Reference: AppendFile()

Append 1+ files to another file

Version 11.02.00, 2015-04-09

The `AppendFile()` command appends one or more files to another file. All or only matching lines from input files can be transferred. This command is useful for appending multiple data files into a single file that can be read by TSTool.

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

AppendFile() Command Editor

The command syntax is as follows:

```
AppendFile(Parameter=Value,…)
```

Command Parameters

Parameter	Description	Default
InputFile	The name of one or more files to delete, using the following conventions: • No `*` in name – match one file. • Filename of `*`– match all files in input directory (working directory by default). • Filename of `*.ext` – match all files with extension More options may be supported in the future when TSTool is updated to use Java 1.7+.	None – must be specified.
OutputFile	The output file that will be appended to. The file is created if it does not exist. Use the `RemoveFile()` command to remove the old file.	None – must be specified.
IncludeText	A regular expression pattern to include text. Only the matching lines will be included. This uses the Java regular expressions syntax (see http://en.wikipedia.org/wiki/Regular_expression).	Append all lines.
ExcludeText	A regular expression pattern to exclude text. Matching lines will be excluded from the append.	Append all lines.
Newline	Indicate the newline character to use at the end of lines. Normally the operating system default is ok (`\n` for Linux, `\r` for Mac, `\r\n` for Windows), but the newline may need to be specified to ensure cross-platform compatibility.	Operating system default.
IfNotFound	Indicate action if the file is not found, one of: • `Ignore` – ignore the missing file (do not warn). • `Warn` – generate a warning (use this if the file truly is expected and a missing file is a cause for concern). • `Fail` – generate a failure (use this if the file truly is expected and a missing file is a cause for concern).	Warn

The following table lists regular expression examples:

InputText Regular Expression	Description
`.*\Q-\E.*`	Match lines that start with any character, end with any character, and contain a dash. The `\Q` and `\E` characters are special characters to start and end a quoted character, and are necessary because the dash has special meaning in a regular expression.