

---

# Command Reference: RunPython()

## Run a Python script

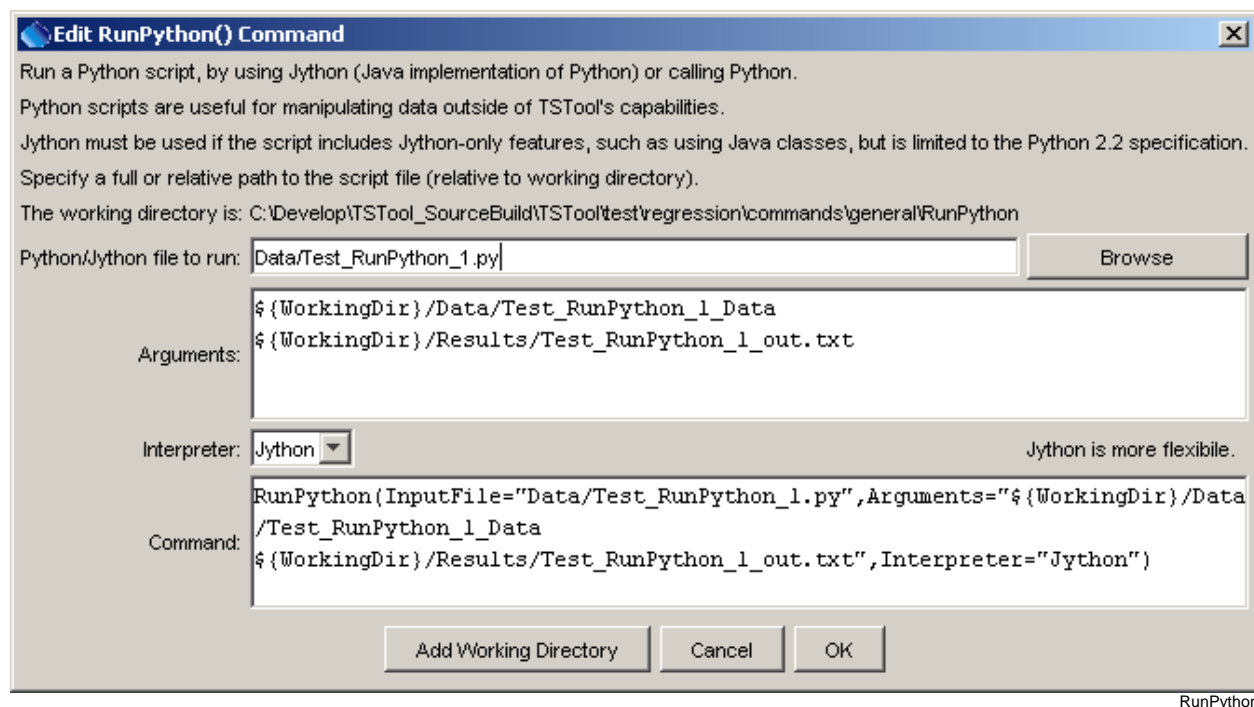
Version 08.16.00, 2008-06-25

The `RunPython()` command runs a Python or Jython script, waiting until the script is finished before processing additional commands. Python is a powerful scripting language that is widely used (see <http://www.python.org>). This command allows either Python or Jython to be used as the interpreter. If Python is used, then Python must be installed on the computer and be in the PATH environment variable. The script is then run by running:

### python Arguments

Jython (see <http://www.jython.org>) is a Java implementation of Python, allowing most Python scripts to be run and also allowing integration with Java software, including using Java classes in scripts. This allows existing Java components to be used and supports more robust integration and error handling. If the Jython interpreter is used, no additional software needs to be installed. Additional features to utilize Jython will be provided in future software releases (e.g., directly passing time series objects to/from python scripts). Currently Jython only supports the Python 2.2 specification; however, future releases are expected to support later specifications.

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



**RunPython() Command Editor**

The command syntax is as follows:

```
RunPython(Parameter=Value,...)
```

#### Command Parameters

Parameter	Description	Default
InputFile	The Python/Jython script to run, specified as an absolute path or relative to the command file.	None – must be specified.
Arguments	Arguments to pass to the script, such as the names of files to process. Use the \${WorkingDir} property to specify the location of the command file. Separate arguments by a space.	None – arguments are optional.
Interpreter	The Python interpreter to run, either Python or Jython.	Jython

The following command example illustrates how to run a Python script.

```
RunPython(InputFile="Data/Test_RunPython_1.py",
Interpreter="Jython",Arguments="${WorkingDir}/Data/Test_RunPython_1_Data
${WorkingDir}/Results/Test_RunPython_1_out.txt")
```

The corresponding Python script is as follows:

```
#
# Test command for running Python script from TSTool
#
import sys
import os
print "start of script"
print 'os.getcwd()="' + os.getcwd() + '"'
infile = None
outfile = None
if ( len(sys.argv) < 3 ):
    print "Error. Expecting input file name as first command line argument,
output file name as second."
    sys.exit(1)
else:
    infile = sys.argv[1]
    outfile = sys.argv[2]
    print 'Input file to process is "' + infile + '"'
    print 'Output file to create is "' + outfile + '"'

inf=open(infile,'r')
outf=open(outfile,'w')
for line in inf:
    outf.write("out: " + line)
inf.close()
outf.close()
print "end of script"
```

The data file is as follows:

```
Line 1 (first line)
Line 2
Line 3
Line 4
Line 5 (last line)
```

The output file is as follows:

```
out: Line 1 (first line)
out: Line 2
out: Line 3
out: Line 4
out: Line 5 (last line)
```

This page is intentionally blank.