Appendix: Running TSTool in Various Modes

2016-02-15

Overview

TSTool is commonly run as an interactive desktop application using the graphical user interface (GUI). However, the TSTool design allows the software to be run in various modes, as described in this document:

1. Desktop application
    a. Interactive graphical user interface
    b. Limited graphical user interface (e.g., no main GUI but graph windows display in support of other applications)
    c. Command line batch mode (e.g., Windows command shell or Linux shell script)
2. Scheduled task (e.g., Windows scheduler or Linux cron)
3. Web server application (e.g., run headless processing in support of website)
4. Batch server to process command files in a hot folder (`-batchServer` mode)
5. REST web server (e.g., run TSTool as a stand-alone REST web server)
6. Web browser applet

The following sections focus on each run mode for various operating systems. In addition to these run modes, the TSTool installation configuration may involve the software being installed on a local computer for one user or a server that provides access to multiple desktop computers and users. It is possible for any of the run modes to be configured on a desktop computer, given the proper supporting software and expertise; however, it is likely that server-oriented run modes will be configured on servers. The following sections describe how to run TSTool in various modes. It is assumed that TSTool has been properly installed and configured based on information in other appendices.

Running TSTool in Interactive Mode

Running TSTool in interactive mode using the graphical user interface (GUI) is the primary run mode for most users, in particular when editing and running command workflows. The **Getting Started** chapter and other chapters in the TSTool documentation focus on this run mode and are not discussed further here.

Often as TSTool processing becomes integrated with an organization’s business processes, it is appropriate to run TSTool command files in batch mode for efficiency or as a scheduled task to support other tasks. The following sections describe how such processing can be implemented. Because the command files are the same regardless of whether TSTool is run in interactive or batch mode, the interactive tool can be used to modify and troubleshoot command files, as appropriate.

Running TSTool in Limited Interactive Mode

It may be useful to run TSTool in support of other applications. For example, TSTool has been integrated with Geographic Information System (GIS) desktop applications where clicking on a map feature results in running TSTool to process data to produce an interactive graph. In this case, TSTool is run in batch mode (see following sections) and the `-nomaingui` command line parameter is used to indicate the special run mode. The user can interact with the displayed visualization window as if the window had been created from the normal GUI, including changing between graph, table, and summary views, changing view properties, and printing.

Potential issues with this approach are:

- Error-handling and feedback to the user are more difficult because TSTool is not fully integrated with the calling software. The `StartLog()` command can be used to ensure that a log file is created in a location that is integrated with the overall system.
- There may be a slight delay displaying the view window because TSTool starts each time. To improve performance, a specific configuration for TSTool may be required (see the `–config` command line parameter). For example, disable all datastores that are not necessary for the application because the time required to connect to databases will slow down TSTool startup. Another option is to point to data products (time series graphs, etc.) provided by a web server. This involves implementing TSTool on a web server, which his discussed in a section below; however, such an approach requires implementing more technology components.
- It can be confusing to specify the location of configuration information and input files. Using absolute paths limits flexibility in changing the configuration. However, relative paths must be "anchored" to a starting location. For example, TSTool assumes an initial working directory depending on software startup and then sets the working directory to the location of the command file. It generally is best to utilize the top-level application to provide the working directory, and then pass the information to TSTool dynamically if possible.

Running TSTool in Command Line Batch Mode

"Command line batch mode" refers to running TSTool from a command prompt, such as the Windows command shell or Linux command line. The TSTool GUI is useful because it provides immediate feedback on whether commands are properly defined and input data are available. However, once a command file has been defined, it may be appropriate to process the commands in an automated "headless mode", without the GUI. To do so, run TSTool on the command line as follows:

```
[PathToTSTool]tstool –commands [PathToCommandFile]commands.TSTool
[TSTool parameters]
```

See the following sections for specific issues that may occur on different operating systems. The following are potential issues on any operating system:

- TSTool relies on a "working directory", which is the root location to which relative paths are referenced. Normally the working directory is the location of the command file once opened. However, when running TSTool in batch mode, there are a number of folders involved: the location of the TSTool executable, the location from which TSTool is run, and the location of the command file. These multiple locations can make it difficult to troubleshoot. One option is to use absolute paths in the TSTool command line for the executable and the command file so it is very clear where the TSTool executable and command file are located. These paths can be coded into a batch file (**.bat* on Windows, or shell script on *NIX).
- TSTool can be run in headless mode and still create graph windows (e.g., when TSTool supports the functionality of another application). Make sure to evaluate whether the `–nomaingui` parameter is needed in addition to `–commands` (see the previous section).
- The TSTool default log file under the logs folder of the installation is used when TSTool first starts up. If a `StartLog()` command is used in the command file, it will be used when the command file is run. Refer to the appropriate log file when troubleshooting.

The following table lists the TSTool command line parameters.

TSTool Command Line Parameters

Command Line Parameter	Description	Default
`-batchServer`	Run TSTool in headless mode to process command files that are copied into the hot folder (`-batchServerHotFolder`). TSTool will exit when a command file named stop is placed in the hot folder.	
`-batchServer HotFolder Folder`	Specify the hot folder to use with the `-batchServer` parameter.	
`-batchTimeout TimeoutSeconds`	Timeout in seconds. If TSTool command processing takes longer, TSTool will exit. This may be appropriate as a final fallback if other timeout controls do not work. Warning – this will leave processing incomplete. **May be removed in future.**	Do not time out.
`-commands CommandFile`	Specify the name of the command file to run in batch mode. TSTool will process the commands and exit.	Start TSTool in interactive mode.
`CommandFile`	The command file can be specified without `–commands` to start up the TSTool GUI and load the command file. This behavior occurs when selecting a `*.TSTool` file in Windows Explorer. See also `-runcommandsonload`.	
`-config ConfigFile`	Start TSTool using the specified configuration file. This is useful in software test environments.	Start TSTool using the *system/TSTool.cfg* file under the software installation.
`-dTerm,Log`	Specify the debug level (0+) for terminal and log file debug messages. This is useful for printing troubleshooting messages at startup. One or both values can be specified.	No debug messages are generated.
`-home InstallFolder`	Specify the install folder. This parameter is used internally by the TSTool launcher (Launch4J) and in the software development environment to specify the TSTool home, in order to locate other files.	Should always be set by the TSTool launcher.
`-httpServer`	Run TSTool in headless mode to act as an HTTP server. Experimental	
`-nodiscovery`	Do not run discovery on commands as they are loaded. This can be used for large command files that will not be edited interactively, in order to decrease	Run discovery on commands as they are loaded.

Command Line Parameter	Description	Default
	load time.	
`-nomaingui`	Do not show the TSTool main GUI when running commands, but do allow graph windows to be displayed. Closing the last visible graph window will close the TSTool application.	If `–commands` is specified, do not show any windows. If `–nomaingui` also is specified, allow graph windows to show.
`-runcommandsonload`	Load and run the commands, used to start the GUI with a specific command file.	
`-server`	Start TSTool as a REST restlet server (under development).	
`-test`	Run TSTool in test mode, for developers.	
`Property=Value`	Define TSTool global properties (similar to the configuration file). These properties typically are used during development to test specific features.	

Running TSTool in Command Line Batch Mode – Windows

To run TSTool in batch mode on Windows, first open a command prompt (e.g., for Windows 7: **Start…All Programs…Accessories…Command Prompt**). Then run the TSTool as shown in the previous section. Issues specific to Windows include:

- If the folder containing the TSTool executable (`tstool.exe`) is in the `PATH` environment variable, then no leading path is needed. However, because different versions of TSTool may be installed on the system, specifying the leading path to the executable may be appropriate.
- Entering the TSTool command line in a Command Prompt window causes the prompt to be immediately returned, even though TSTool is still running. This can be disconcerting in particular because it may be difficult to know when TSTool has finished processing the command file. Placing the command in a batch file (**.bat*) can help.

Running TSTool in Command Line Batch Mode – Linux

To run TSTool in batch mode on Linux, first open a command shell window. Then run the TSTool as shown above. Issues specific to Linux include:

- If the folder containing the TSTool executable (`tstool` script) is in the `PATH` environment variable, then no leading path is needed. However, because different versions of TSTool may be installed on the system, specifying the leading path to the executable may be appropriate. It is helpful on Linux to use symbolic links to provide access to the current version as `tstool`, and other versions (e.g., tstool1022 points to 10.22 version).

Running TSTool in Command Line Batch Mode – Mac

TSTool execution on Mac computers is similar to Linux.

Running TSTool as a Scheduled Task

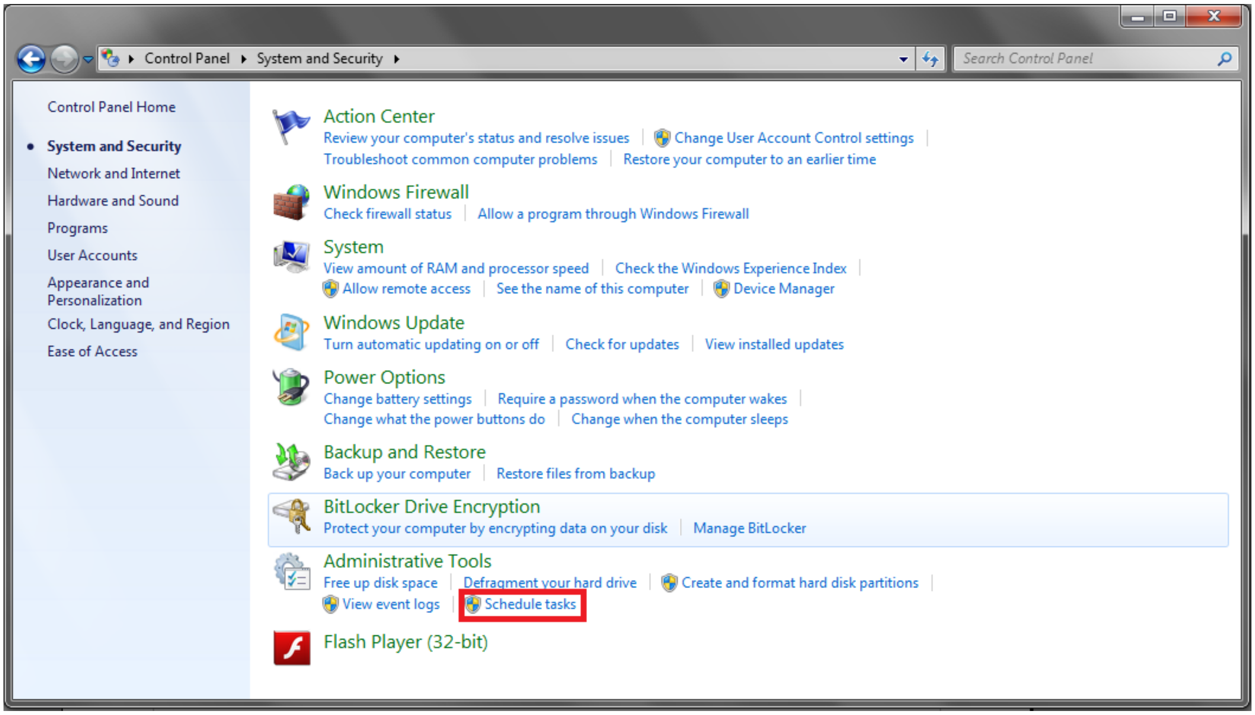
Examples of running TSTool as a scheduled task include automatically processing real-time data, and creating data products prior to viewing (rather than creating upon request). The following sections describe how to configure TSTool to run as a scheduled task.

Running TSTool as a Scheduled Task – Windows

Windows has a task scheduling function, which can be used to run TSTool. These instructions are for Windows 7. However, all versions of Windows have similar features. The following references can be consulted for other Windows versions:

- Windows XP: http://support.microsoft.com/kb/308569
- Windows Vista: http://windows.microsoft.com/en-us/windows-vista/schedule-a-task
- Windows 8: http://technet.microsoft.com/en-us/library/cc725745.aspx.

To define a task, first go to the ***System and Security*** subpanel of the ***Control Panel***, then click on ***Schedule tasks***.



Control Panel ▸ System and Security ▸

Search Control Panel

Control Panel Home

- System and Security
- Network and Internet
- Hardware and Sound
- Programs
- User Accounts
- Appearance and Personalization
- Clock, Language, and Region
- Ease of Access

Action Center
Review your computer's status and resolve issues | Change User Account Control settings | Troubleshoot common computer problems | Restore your computer to an earlier time

Windows Firewall
Check firewall status | Allow a program through Windows Firewall

System
View amount of RAM and processor speed | Check the Windows Experience Index | Allow remote access | See the name of this computer | Device Manager

Windows Update
Turn automatic updating on or off | Check for updates | View installed updates

Power Options
Change battery settings | Require a password when the computer wakes | Change what the power buttons do | Change when the computer sleeps

Backup and Restore
Back up your computer | Restore files from backup

BitLocker Drive Encryption
Protect your computer by encrypting data on your disk | Manage BitLocker

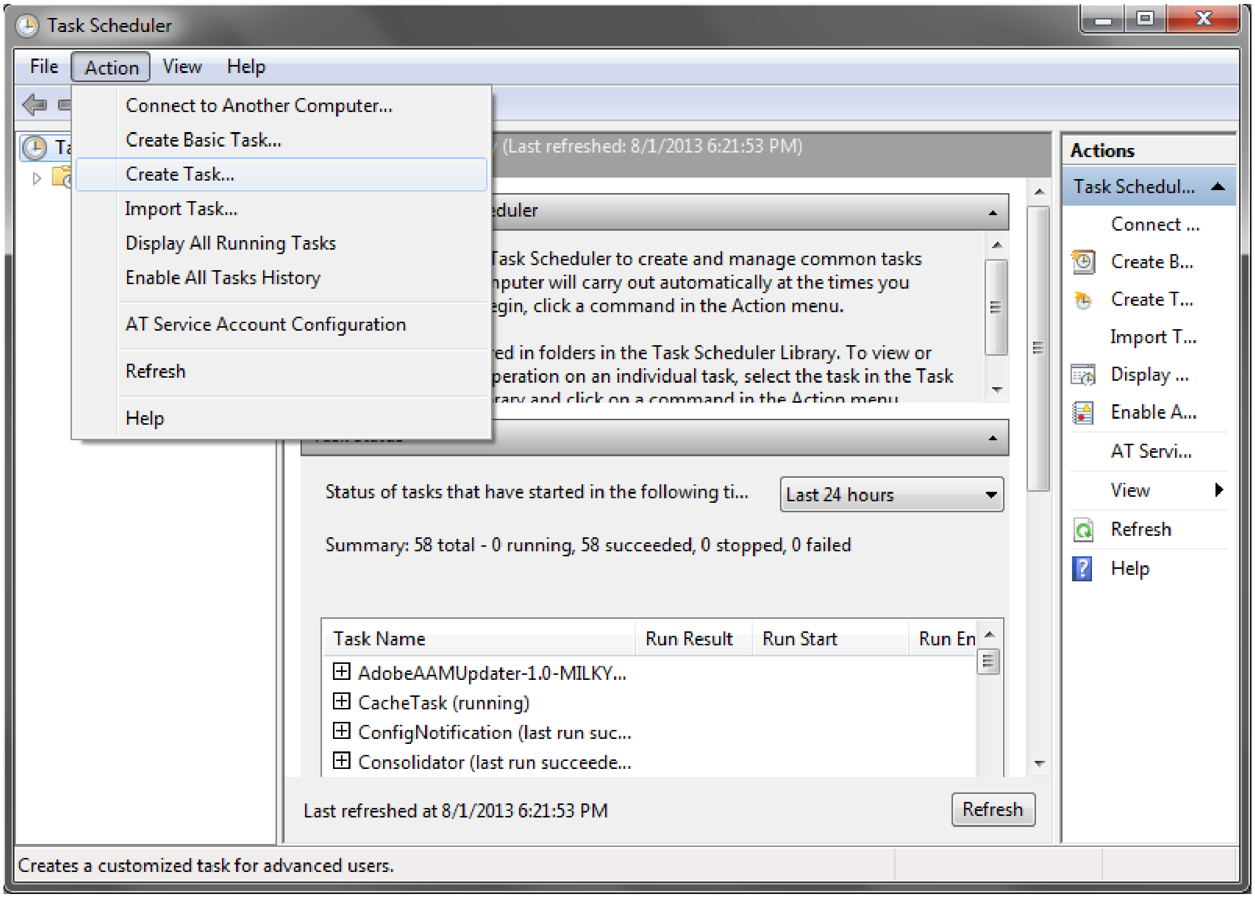
Administrative Tools
Free up disk space | Defragment your hard drive | Create and format hard disk partitions | View event logs | Schedule tasks

Flash Player (32-bit)

RunningTSTool_ScheduleTasks

Running TSTool – Schedule as a Task

A screen showing currently scheduled tasks will appear. Click on the ***Action*** menu in the upper left, and the ***Create Task…*** option.



(Last refreshed: 8/1/2013 6:21:53 PM)

Task Scheduler to create and manage common tasks
nputer will carry out automatically at the times you
egin, click a command in the Action menu.

ed in folders in the Task Scheduler Library. To view or
peration on an individual task, select the task in the Task
rary and click on a command in the Action menu.

Status of tasks that have started in the following ti... Last 24 hours

Summary: 58 total - 0 running, 58 succeeded, 0 stopped, 0 failed

⊞ AdobeAAMUpdater-1.0-MILKY...
⊞ CacheTask (running)
⊞ ConfigNotification (last run suc...
⊞ Consolidator (last run succeede...

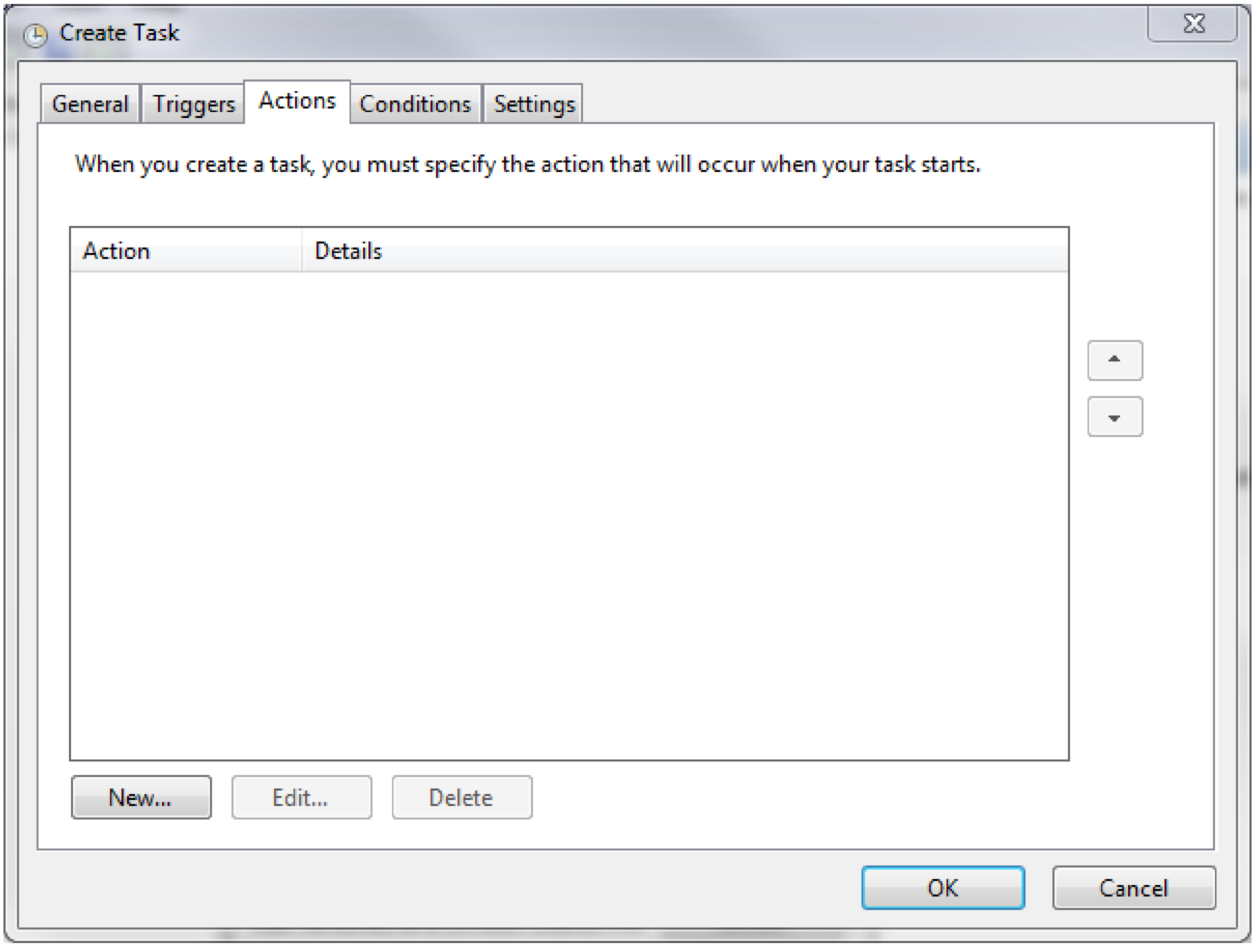
Last refreshed at 8/1/2013 6:21:53 PM

Creates a customized task for advanced users.

RunningTSTool_TaskScheduler

Running TSTool – Creating a Task in the Task Scheduler

A tabbed dialog will appear. Go to the ***Actions*** tab, and click ***New…*** to create a new task.



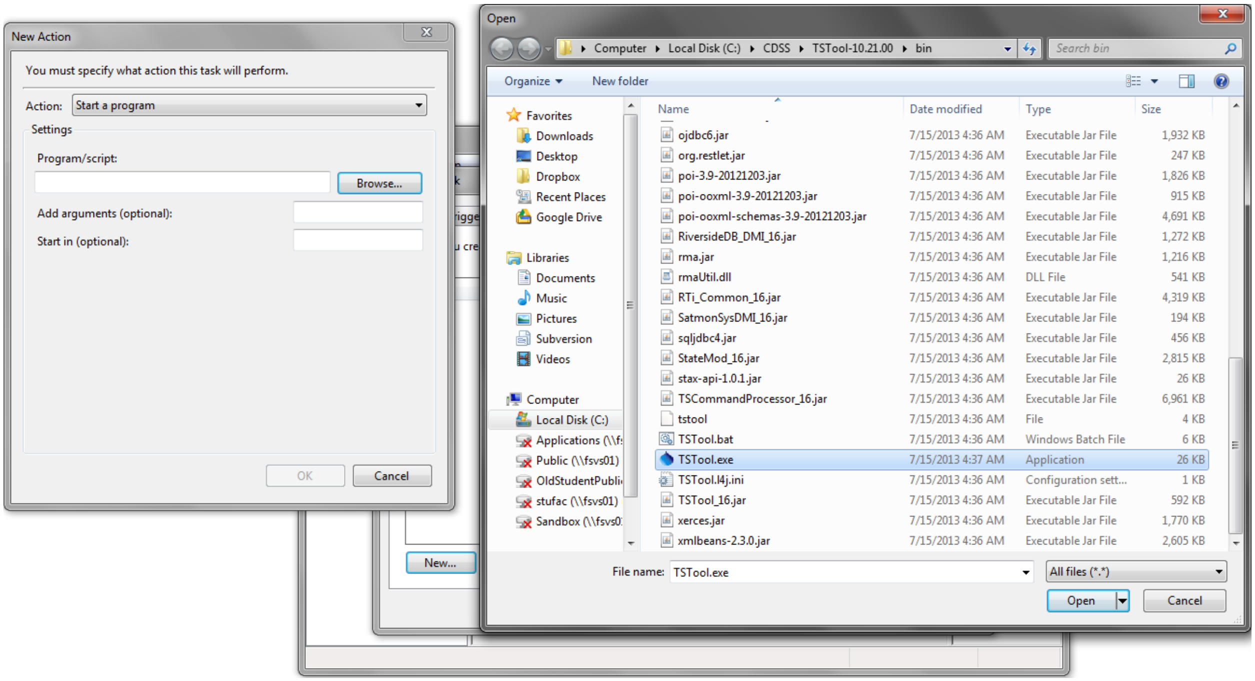
When you create a task, you must specify the action that will occur when your task starts.

Action	Details

RunningTSTool_NewTask0

Running TSTool – Create a New Task

Specify the program to run by using the ***Browse…*** button to pick the TSTool executable program, as shown below. The example shown on the right is a typical location for the install (specify an appropriate version number).



New Action

You must specify what action this task will perform.

Action: Start a program

Settings

Program/script:

Add arguments (optional):

Start in (optional):

Open

Computer ▸ Local Disk (C:) ▸ CDSS ▸ TSTool-10.21.00 ▸ bin

Name	Date modified	Type	Size
ojdbc6.jar	7/15/2013 4:36 AM	Executable Jar File	1,932 KB
org.restlet.jar	7/15/2013 4:36 AM	Executable Jar File	247 KB
poi-3.9-20121203.jar	7/15/2013 4:36 AM	Executable Jar File	1,826 KB
poi-ooxml-3.9-20121203.jar	7/15/2013 4:36 AM	Executable Jar File	915 KB
poi-ooxml-schemas-3.9-20121203.jar	7/15/2013 4:36 AM	Executable Jar File	4,691 KB
RiversideDB_DMI_16.jar	7/15/2013 4:36 AM	Executable Jar File	1,272 KB
rma.jar	7/15/2013 4:36 AM	Executable Jar File	1,216 KB
rmaUtil.dll	7/15/2013 4:36 AM	DLL File	541 KB
RTi_Common_16.jar	7/15/2013 4:36 AM	Executable Jar File	4,319 KB
SatmonSysDMI_16.jar	7/15/2013 4:36 AM	Executable Jar File	194 KB
sqljdbc4.jar	7/15/2013 4:36 AM	Executable Jar File	456 KB
StateMod_16.jar	7/15/2013 4:36 AM	Executable Jar File	2,815 KB
stax-api-1.0.1.jar	7/15/2013 4:36 AM	Executable Jar File	26 KB
TSCommandProcessor_16.jar	7/15/2013 4:36 AM	Executable Jar File	6,961 KB
tstool	7/15/2013 4:36 AM	File	4 KB
TSTool.bat	7/15/2013 4:36 AM	Windows Batch File	6 KB
TSTool.exe	7/15/2013 4:37 AM	Application	26 KB
TSTool.l4j.ini	7/15/2013 4:36 AM	Configuration sett...	1 KB
TSTool_16.jar	7/15/2013 4:36 AM	Executable Jar File	592 KB
xerces.jar	7/15/2013 4:36 AM	Executable Jar File	1,770 KB
xmlbeans-2.3.0.jar	7/15/2013 4:36 AM	Executable Jar File	2,605 KB

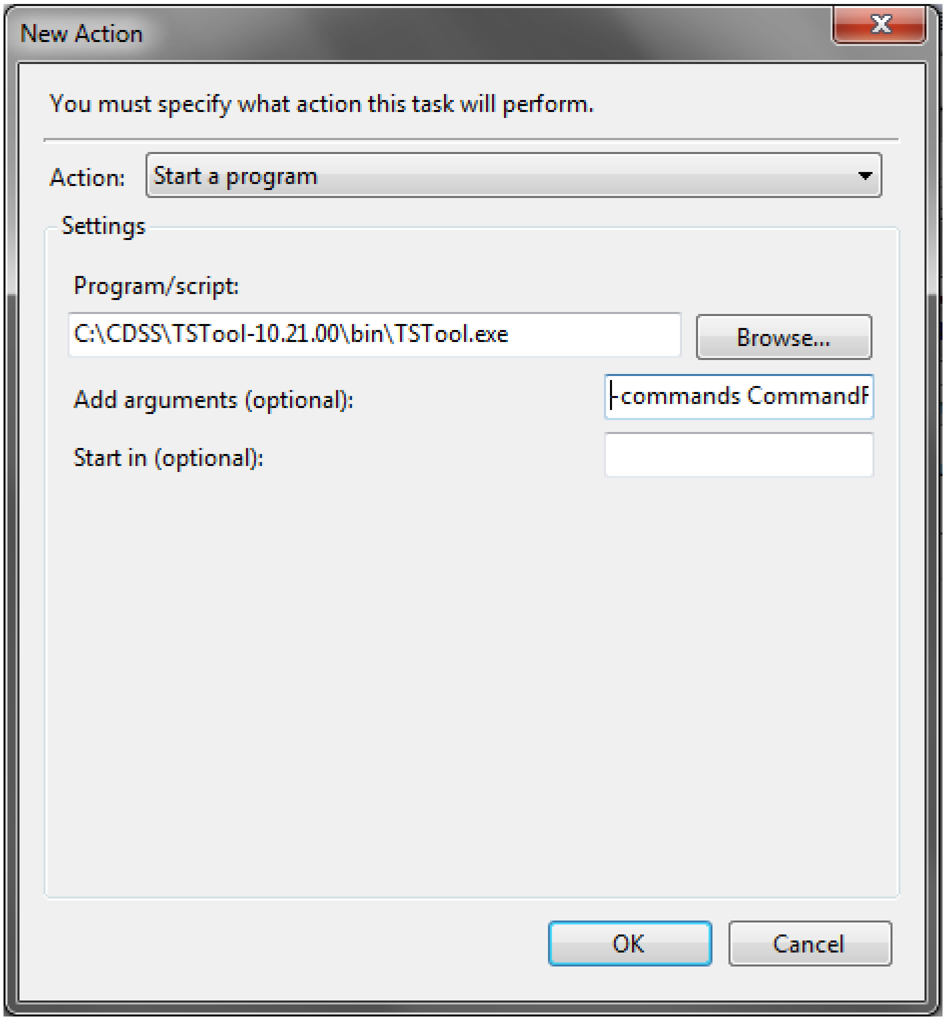
File name: TSTool.exe

All files (*.*)

RunningTSTool_NewTask1

Running TSTool – Specify Program to Run for Task

Specifying the absolute path to the command (starting at the drive) is recommended. Also specify that TSTool should run without the user interface and specify command file to use (`-commands CommandFileName`), as shown in the following figure:



New Action

You must specify what action this task will perform.

Action: Start a program

Settings

Program/script: C:\CDSS\TSTool-10.21.00\bin\TSTool.exe

Add arguments (optional): -commands CommandF

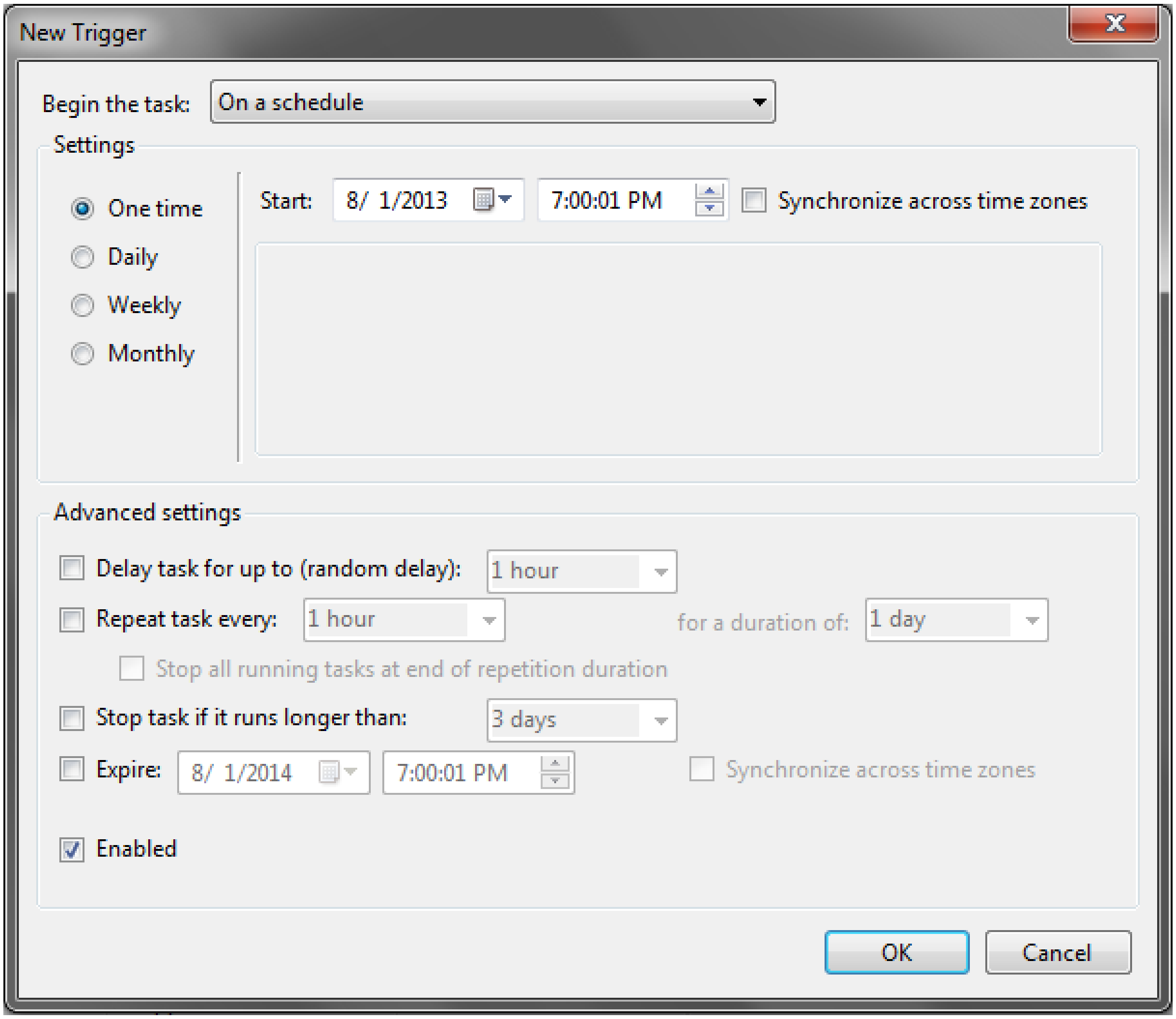
Start in (optional):

RunningTSTool_NewTask2

Running TSTool – Specify Program and Command Line Parameters to Run for Task

TSTool log files will by default be in the logs folder under the TSTool installation. Use the `StartLog()` command to control the log file location.

To actually run the task, define a trigger in the ***Triggers*** tab of the ***Create Task*** dialog (press ***New…*** to create a trigger):



New Trigger

Begin the task: On a schedule

Settings

● One time
○ Daily
○ Weekly
○ Monthly

Start: 8/ 1/2013 7:00:01 PM □ Synchronize across time zones

Advanced settings

□ Delay task for up to (random delay): 1 hour

□ Repeat task every: 1 hour for a duration of: 1 day

□ Stop all running tasks at end of repetition duration

□ Stop task if it runs longer than: 3 days

□ Expire: 8/ 1/2014 7:00:01 PM □ Synchronize across time zones

☑ Enabled

RunningTSTool_Triggers

Running TSTool – Specify Trigger to Indicate Run Task Schedule

Any number of triggers can be chosen, and it can be run at almost any interval, from every five minutes upward. Conditions that will prevent the program from running can also be chosen (***Conditions*** tab). Output files will be overwritten on each run unless the filename is dynamic, such as using the date/time.

Running TSTool as a Scheduled Task – Linux

These instructions apply to variants of Linux and UNIX.

Cron is a background system process that checks every minute to see if any process should be run. To schedule a run, the crontab command is used. It creates or edits an entry in the user’s crontab configuration file. Each user can only have one crontab in most variants of Unix (Debian is slightly different; check the man page for more information).

The command `crontab -e` will edit an existing crontab or create a new one if none exists. If this is the first time a crontab has been created on the system, the command may present a list of editors to choose from. The choice is not important to the functioning of the command.

The syntax of a crontab entry is as follows:

```
minute hour day_of_month month day_of_week command
```

A `*` in any of the first five fields will run the process at every instance, and a range (e.g. `1-5` or `5,10,15`) can be entered into any of the first five fields. For example, to run once an hour, the entry would be as follows:

```
0 * * * * (command)
```

To run TSTool as a cron process, `command` should be specified as follows:

```
/PathToInstallation/bin/tstool -commands PathToCommandFile
```

The paths should be specified as absolute path because cron does not necessarily have a home directory.

The log files will by default be in the logs folder under the TSTool installation. The `StartLog()` command can be used to configure the log file location.

Output files will be overwritten at each execution of the command unless the names are determined dynamically.

For more information, see:

`man cron` (information about cron options and system variations)
`man crontab` (information about options and system variations)
`man 5 crontab` (information about syntax and limitations)

Running TSTool as a Web Server

If TSTool is started with the `–httpServer` option it will run as a web server on port 8000. This functionality is experimental and requires additional development. This may replace or be combined with the experimental REST servlet describe below.

Running TSTool as Batch Server

If TSTool is started with the `–batchServer` option it will run continuously in headless mode. Use the `-batchServerHotFolder Folder` command line parameter to specify the hot folder for command files. This is an experimental feature.

- Each command file that is copied into the folder will be executed and then removed. If necessary, use the `StartLog()` command to create a log file in a different folder to troubleshoot.
- Create a file named *stop* in the hot folder to cause TSTool to stop processing and exit.
- In the future additional features will be added.

Running TSTool as a REST Web Server

It is possible to use the Restlet package (http://restlet.org) to implement a REST API wrapper around TSTool's batch features. Consequently, TSTool can be started as a REST server, listen for requests, process data, and return resources. Preliminary work has been done to implement this capability but much more effort is needed to provide production features.

Running TSTool as a Web Browser Applet

It is possible to integrate TSTool with a web application to run TSTool as an applet. In this case, the web browser runs the Java TSTool application within a Java Runtime Environment within the web browser. This approach was evaluated early in TSTool development as is still enabled in some form, although it has not been used for years. The main challenges with applets are the security issues of accessing local files from the applet. Because TSTool is used for heavy processing using local data, the applet approach was not suitable. Enabling TSTool as an applet, using remote data (such as web services) is possible but will require additional development and testing.