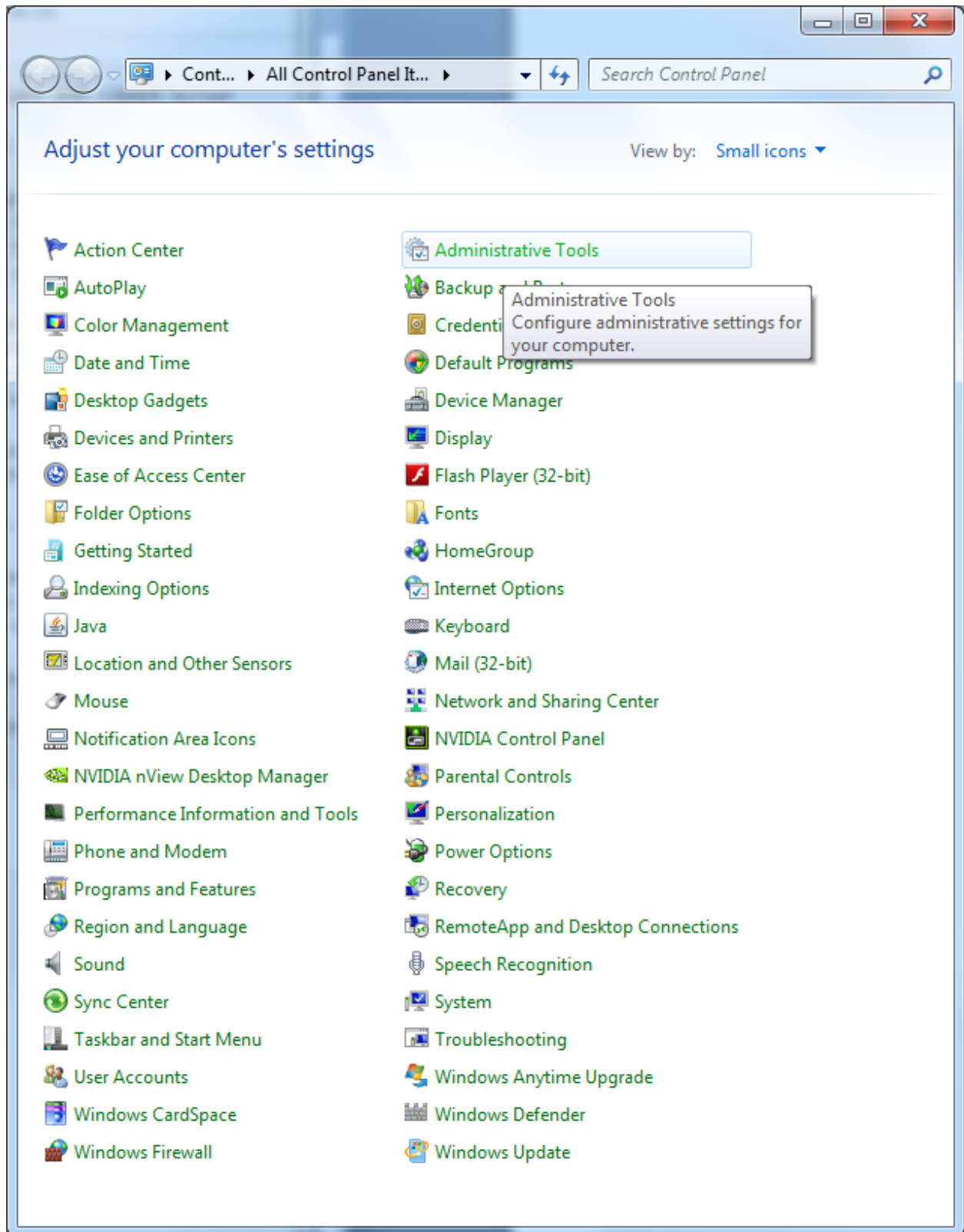


Windows Task Scheduler instructions

Copyright 2015 Gregory C. Hancock d/b/a Grid Insight

To get a logging script to run in the background on Windows, you can run it as a Windows task that starts at boot and keeps running. This is easier and simpler than setting up a Windows service, but it requires a little initial work on the part of the local system administrator. This document shows all the relevant settings to work with the `amrlogger.py` script from Grid Insight, but it does not go so far as to walk the administrator through each step. This is a preliminary document for use by those already familiar with the subject matter. Contact Grid Insight for assistance.



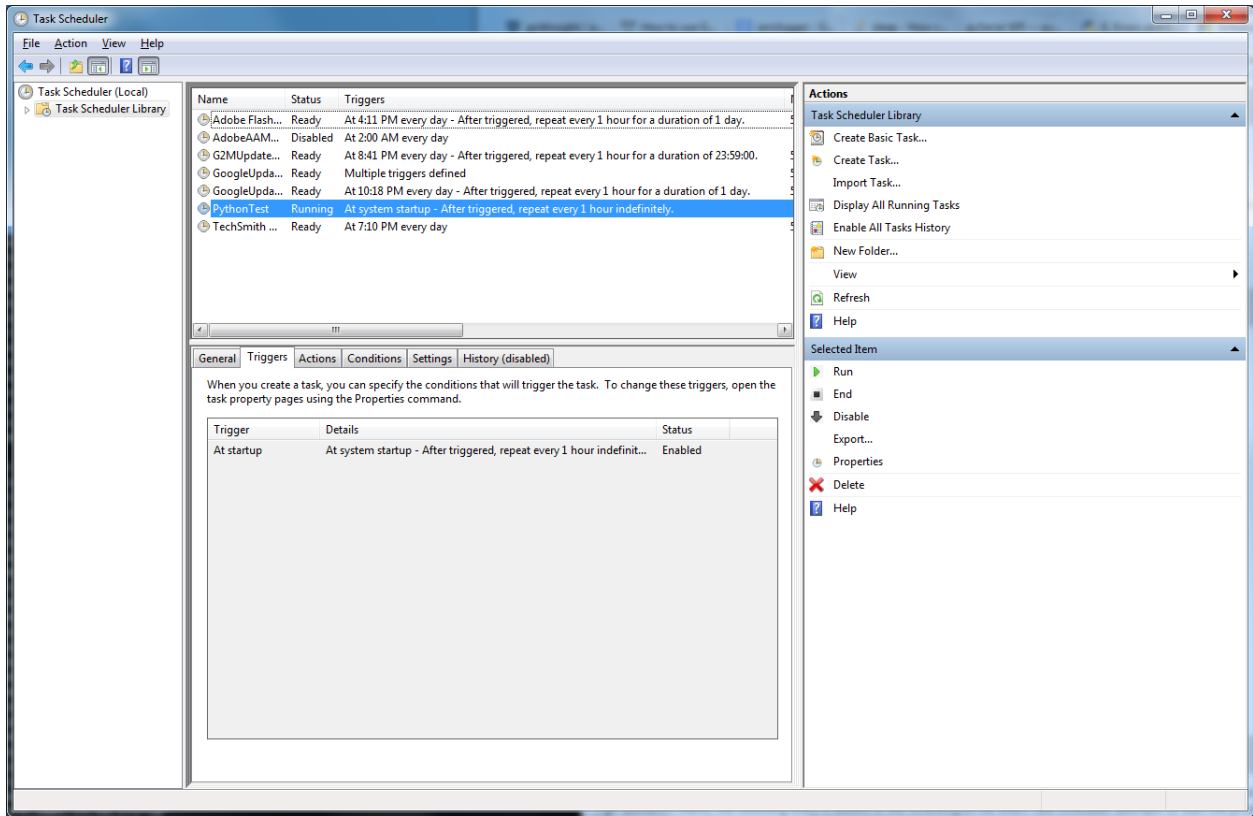
Administrative Tools

Organize Open Burn

Search Administrative Tools

Name	Date modified	Type
Component Services	7/13/2009 9:57 PM	Shortcut
Computer Management	7/13/2009 9:54 PM	Shortcut
Data Sources (ODBC)	7/13/2009 9:53 PM	Shortcut
desktop.ini	4/25/2014 1:35 AM	Configuratio
Event Viewer	7/13/2009 9:54 PM	Shortcut
iSCSI Initiator	7/13/2009 9:54 PM	Shortcut
Local Security Policy	4/25/2014 1:35 AM	Shortcut
Microsoft .NET Framework 2.0 Configura...	6/13/2014 2:31 PM	Shortcut
Performance Monitor	7/13/2009 9:53 PM	Shortcut
Print Management	4/25/2014 1:35 AM	Shortcut
Services	7/13/2009 9:54 PM	Shortcut
System Configuration	7/13/2009 9:53 PM	Shortcut
Task Scheduler	7/13/2009 9:54 PM	Shortcut
Windows Firewall with Advanced Security	7/13/2009 9:54 PM	Shortcut
Windows Memory Diagnostic	7/13/2009 9:53 PM	Shortcut
Windows PowerShell Modules	7/13/2009 10:32 PM	Shortcut

Task Scheduler Date modified: 7/13/2009 9:54 PM Date created: 7/13/2009 9:54 PM
Shortcut Size: 1.23 KB

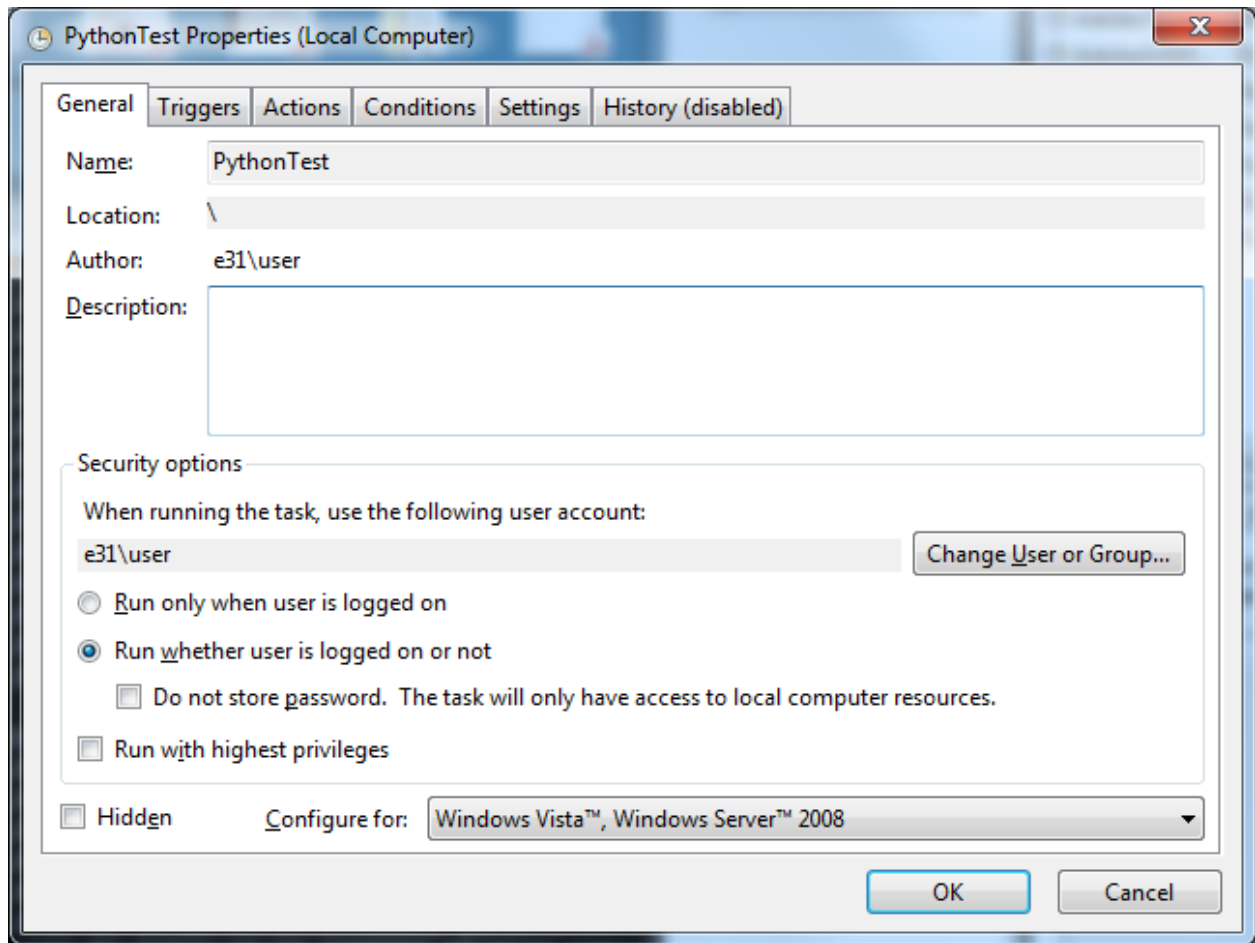


This is what it looks like when you are done. To get there, you have to create a new task. Click “Create Task...” in the right-hand Actions menu to get started. The following screens detail some example settings for a new task that will run `amrlogger.py` at start up, leave it running, and check every hour to make sure it is still running.

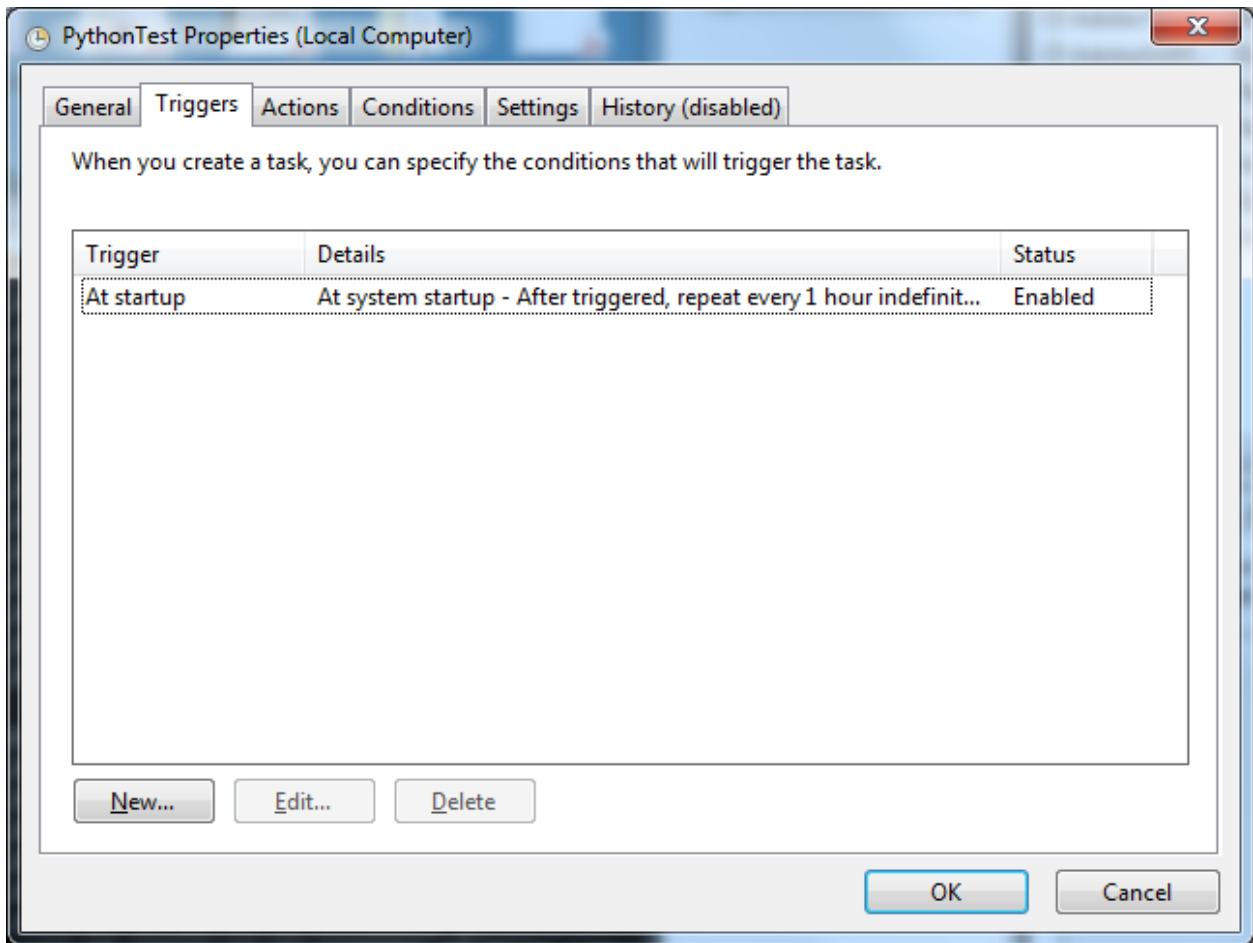
This assumes that you already have the necessary serial port driver installed, that Python 2.7 is installed, and that the Oysterville receiver module has already been manually tested using, for example, TeraTerm.

Python download link:

<https://www.python.org/download/releases/2.7/>



Name the task something descriptive, but do not include any spaces in the name. On each of these configuration screens, **make sure all options match what is shown in the screenshots.**



Edit Trigger



Begin the task: At startup

Settings

No additional settings required.

Advanced settings

Delay task for: 1 minute

Repeat task every: 1 hour for a duration of: Indefinitely

Stop all running tasks at end of repetition duration

Stop task if it runs longer than: 3 days

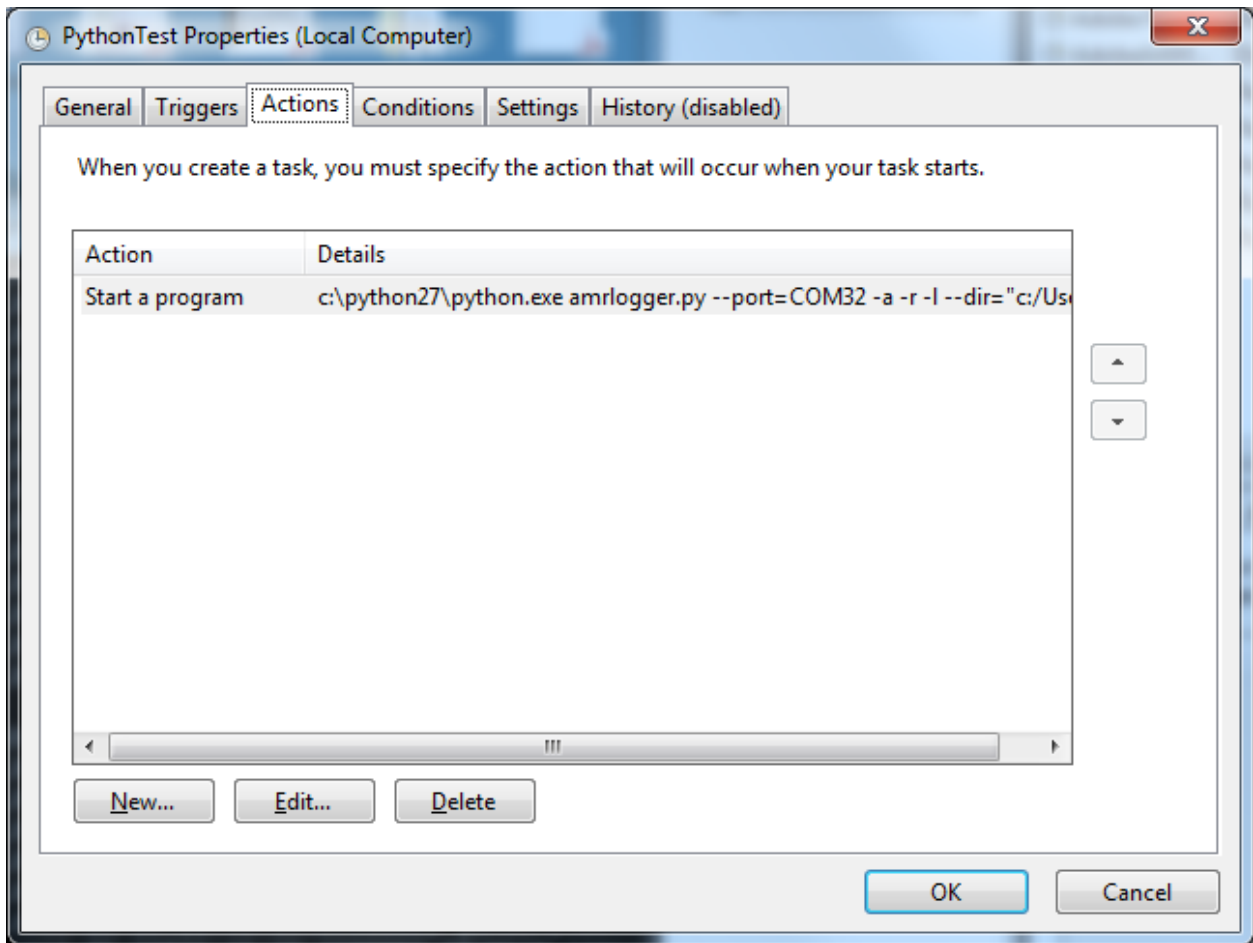
Activate: 5/19/2015 2:54:20 PM Synchronize across time zones

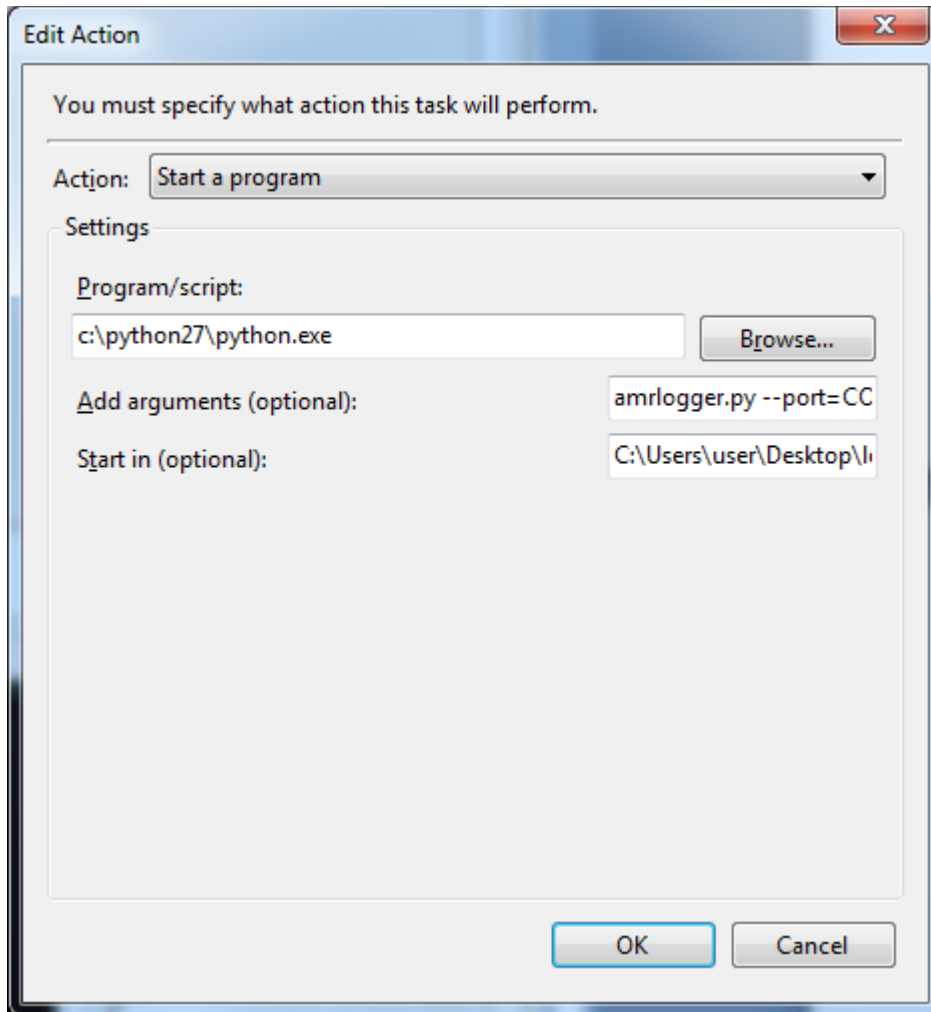
Expire: 5/19/2016 2:54:20 PM Synchronize across time zones

Enabled

OK

Cancel





Program/script:

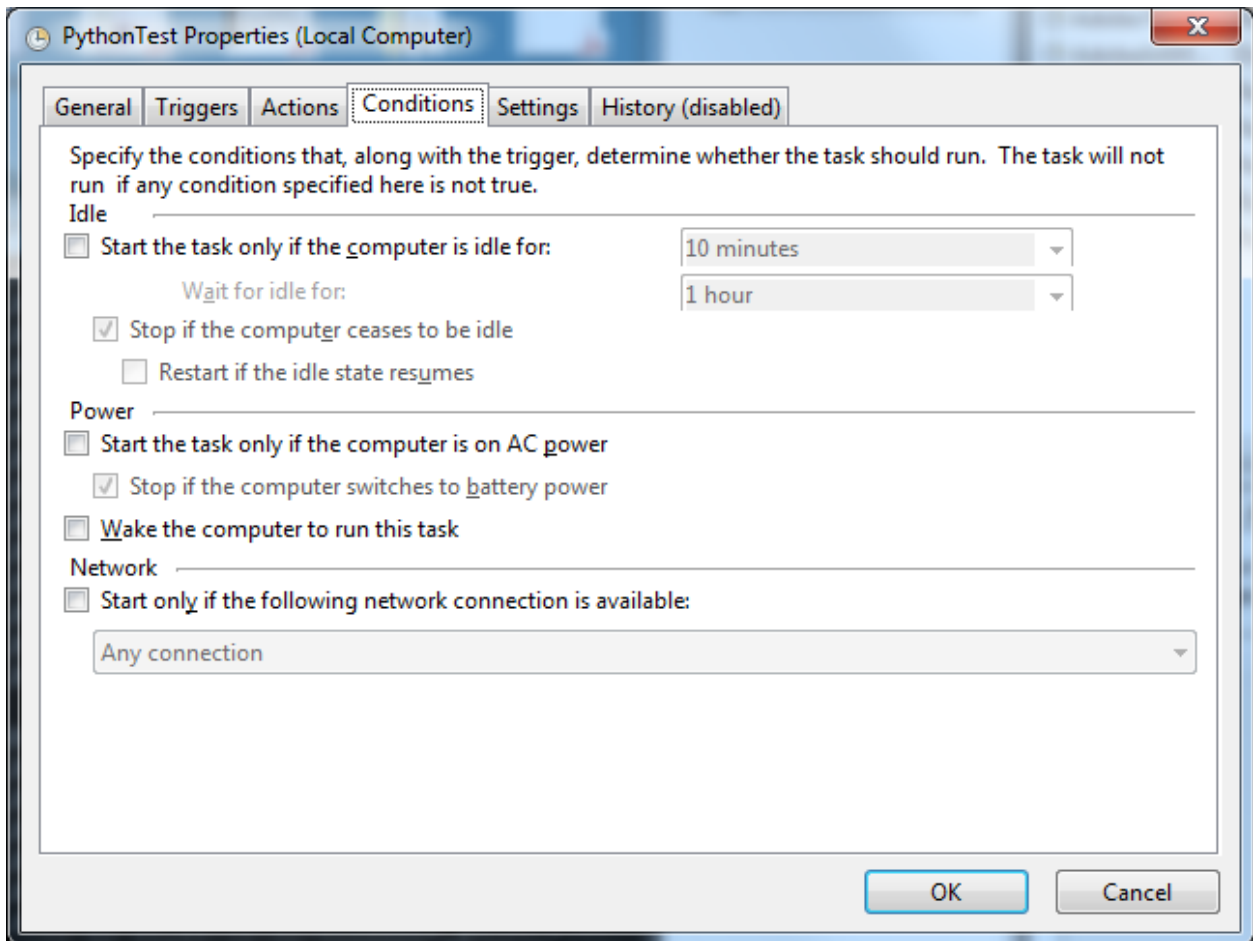
c:\python27\python.exe

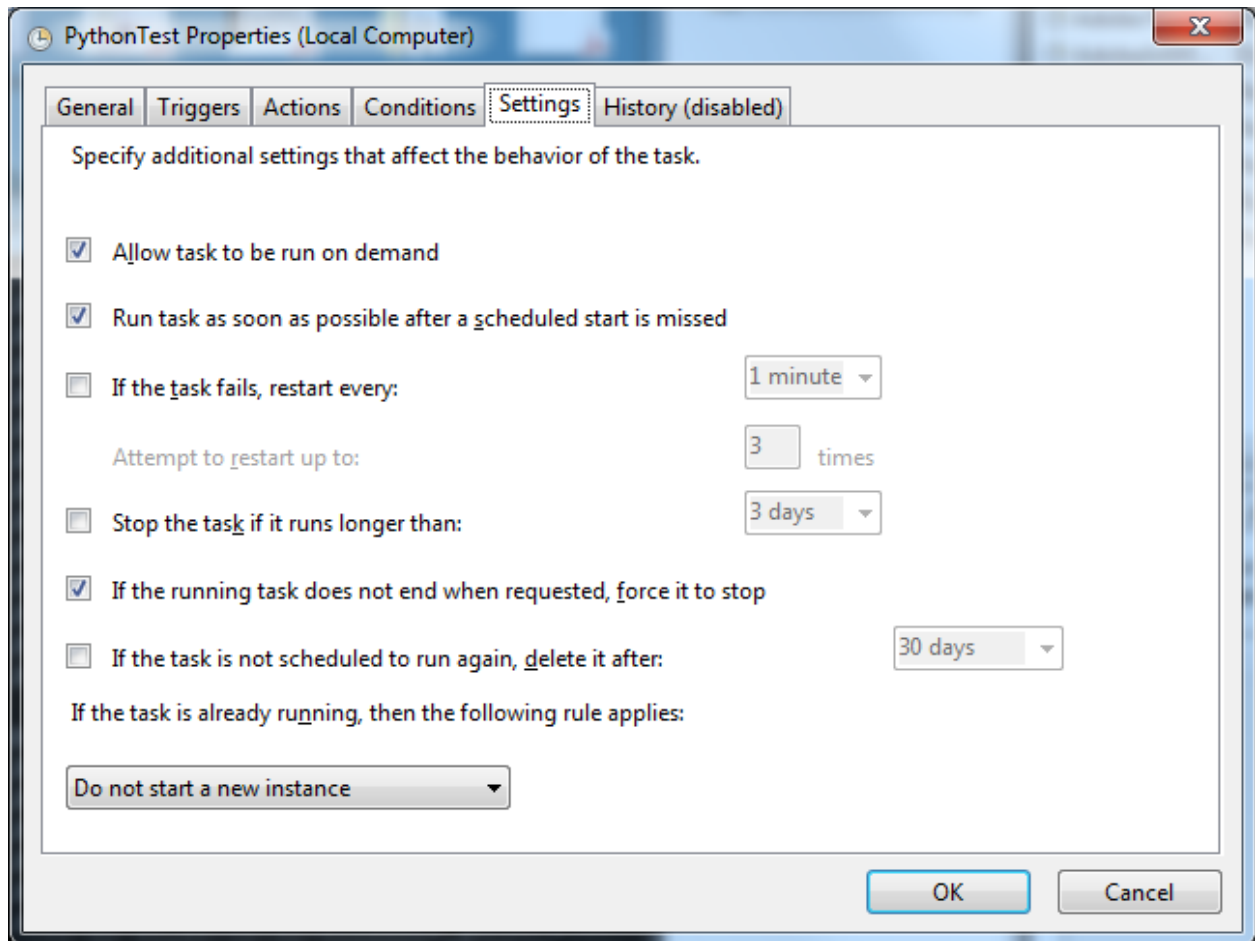
Add arguments (optional):

amrlogger.py --port=COM32 -a -r -l --dir="c:/Users/User/Desktop/log" --file="_amr_log_file.txt"

Start in (optional):

C:\Users\user\Desktop\log





This is the exported XML from Windows Task Manager for the above configuration:

```
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Date>2015-05-19T14:28:46.7138835</Date>
    <Author>e31\user</Author>
  </RegistrationInfo>
  <Triggers>
    <BootTrigger>
      <Repetition>
        <Interval>PT1H</Interval>
        <StopAtDurationEnd>>false</StopAtDurationEnd>
      </Repetition>
      <Enabled>>true</Enabled>
      <Delay>PT1M</Delay>
    </BootTrigger>
  </Triggers>
  <Principals>
    <Principal id="Author">
      <UserId>e31\user</UserId>
      <LogonType>Password</LogonType>
      <RunLevel>LeastPrivilege</RunLevel>
    </Principal>
  </Principals>
  <Settings>
```

```
<MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
<DisallowStartIfOnBatteries>>false</DisallowStartIfOnBatteries>
<StopIfGoingOnBatteries>>true</StopIfGoingOnBatteries>
<AllowHardTerminate>>true</AllowHardTerminate>
<StartWhenAvailable>>true</StartWhenAvailable>
<RunOnlyIfNetworkAvailable>>false</RunOnlyIfNetworkAvailable>
<IdleSettings>
  <StopOnIdleEnd>>true</StopOnIdleEnd>
  <RestartOnIdle>>false</RestartOnIdle>
</IdleSettings>
<AllowStartOnDemand>>true</AllowStartOnDemand>
<Enabled>>true</Enabled>
<Hidden>>false</Hidden>
<RunOnlyIfIdle>>false</RunOnlyIfIdle>
<WakeToRun>>false</WakeToRun>
<ExecutionTimeLimit>PT0S</ExecutionTimeLimit>
<Priority>7</Priority>
</Settings>
<Actions Context="Author">
  <Exec>
    <Command>c:\python27\python.exe</Command>
    <Arguments>amrlogger.py --port=COM32 -a -r -l --dir="c:/Users/User/Desktop/log" --
file="_amr_log_file3.txt"</Arguments>
    <WorkingDirectory>C:\Users\user\Desktop\log</WorkingDirectory>
  </Exec>
</Actions>
</Task>
```