Introduction To Python With Pandas

Chapter 13: Integrating With Windows

Chapter Concepts

Command Line Parameters

Windows Task Scheduler

Chapter Summary

Reading Command Line Arguments

- Can use `sys.argv`:

```
import sys
for x in sys.argv:
    print(x)
```

- `argv[0]` is the program name
  - Whether it is a full path, or not, is system dependent
  - If python is invoked with `-c` option (list of commands), it contains '-c'

- `sys.orig_argv` are the arguments passed to python

More Complex Command Lines

- `argv` is suitable for simple options
- If you need more complex parsing, use `argparse`:

```
import argparse

parser = argparse.ArgumentParser(
    prog='program_name',
    description='brief description',
    epilog='at the bottom of help')

parser.add_argument('filename')        # positional
parser.add_argument('-c', '--count') # takes a value

args = parser.parse_args()
print(args.filename, args.count)
```

Files on the Command Line

- If all the command line parameters are files to be processed, consider `fileinput`

```
import fileinput

with fileinput.input() as f:
    for line in f:
        print(line[:-1])
```

- `fileinput` can also be used with a list or tuple of filenames to read any set of files

Chapter Concepts

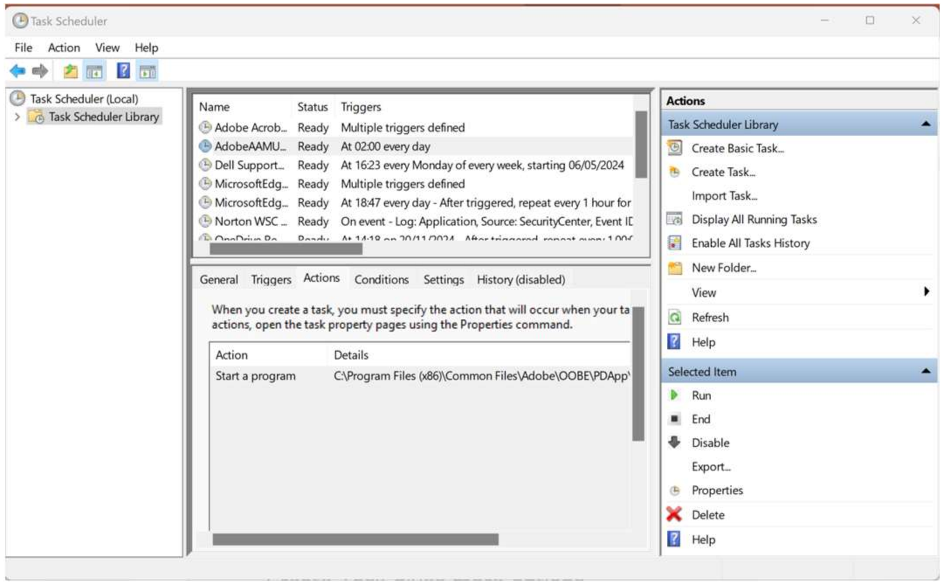
Command Line Parameters

Windows Task Scheduler

Chapter Summary

Windows Task Scheduler

- The Windows Scheduler allows you to schedule tasks
  - At a preset frequency
  - When some event happens
  - It can wake the computer from sleep



Name	Status	Triggers
Adobe Acrob...	Ready	Multiple triggers defined
AdobeAAMU...	Ready	At 02:00 every day
Dell Support...	Ready	At 16:23 every Monday of every week, starting 06/05/2024
MicrosoftEdg...	Ready	Multiple triggers defined
MicrosoftEdg...	Ready	At 18:47 every day - After triggered, repeat every 1 hour for
Norton WSC ...	Ready	On event - Log: Application, Source: SecurityCenter, Event ID

When you create a task, you must specify the action that will occur when your ta
actions, open the task property pages using the Properties command.

Action	Details
Start a program	C:\Program Files (x86)\Common Files\Adobe\OOBE\PDApp\

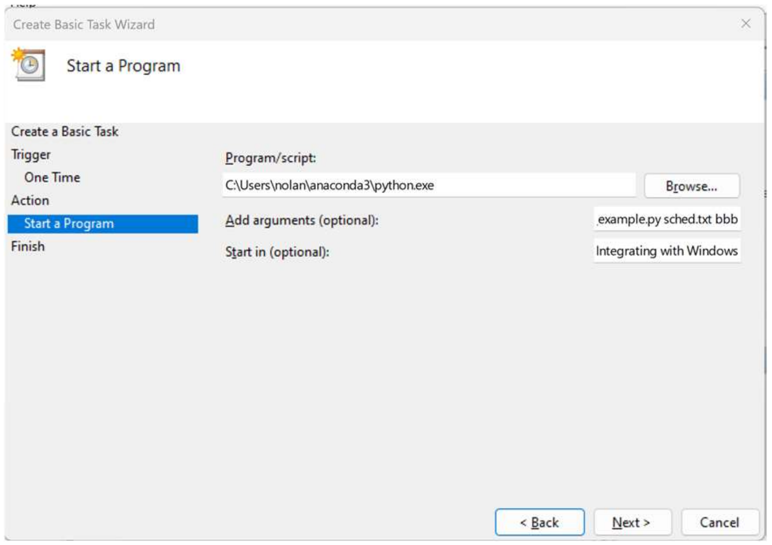
- From Windows Start menu:
  - Task Scheduler
  - Choose Create Basic Task or Create Task
    - Create Basic Task is simpler
      - If you want to execute the task when you are not logged in, you will need to edit the Security parameters afterwards
    - Create Task gives more options, but is more complex

Scheduling a Task

- To schedule any task, you need to consider:
  - The program to run
    - This will be the full path to python.exe
    - With the script as a parameter
    - And the directory where the script is located
  - When to run it
    - Should it run if you are not logged in?
    - Should it run if the computer is asleep?
    - Does it need a network?
  - What to do if it fails

Scheduling a Python Task

- Most of the parameters are self-explanatory
  - Find python.exe using the "Browse" button
  - Enter the script name and any command line arguments
    - If there are spaces in arguments, surround them with double quotes
  - In the "Start in" box, enter the directory that holds the script
    - If there are spaces in the path, do *not* surround the path with quotes
- On the last step, if you are creating a Basic Task and need to run the script when you are offline, check the box to open the properties
- Always try a manual run before relying on the schedule

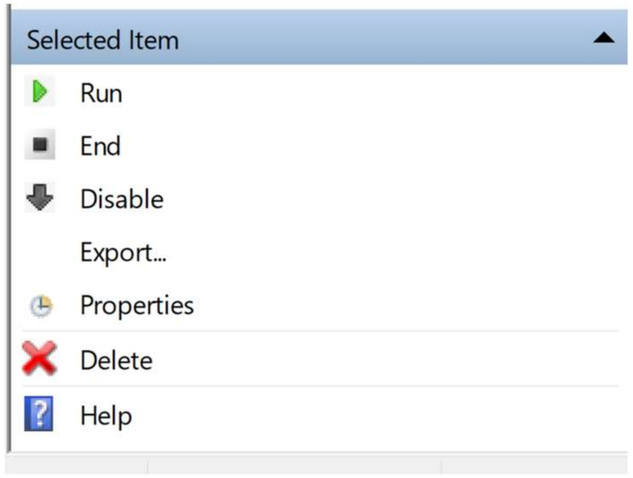

Create Basic Task Wizard

Start a Program

Program/script: C:\Users\nolan\anaconda3\python.exe

Add arguments (optional): example.py sched.txt bbb

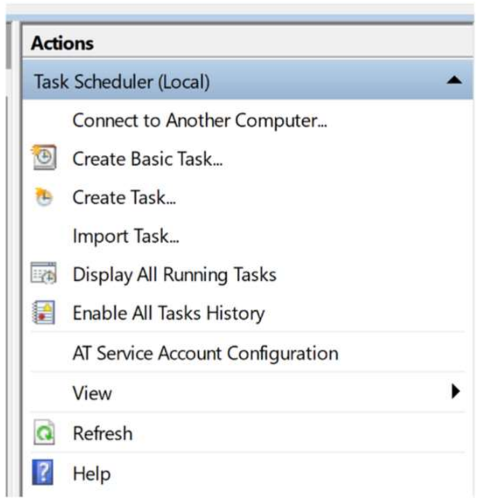
Start in (optional): Integrating with Windows



What To Do If It Fails

- In the top box of the Task Scheduler, you can see the last time the task ran and the result
  - If the result is anything other than (0x0), the job failed (or you set an exit code from your script)

- In the job properties, you can see the full history
  - This is most likely disabled by default



- You may prefer the Event Viewer
  - From the Start Menu, type “Event Viewer”
  - Navigate to:
    - Applications and Services Logs
    - Microsoft
    - Windows
    - TaskScheduler
    - Operational
  - You can also Enable and Disable Logs from here

- Note that a lot of scheduled tasks run
  - Enabling logs may not be something you want to do all the time

Exercise 13.1: Run a Scheduled Task

- Create a program that accepts two command line parameters
  - The first is a filename
  - The second is text to write to that file
- When you run the program, it should write the text and the current time to the file
  - Test it first at the command line
- Set the program to run in the scheduler and verify that it writes the correct time

- Hint:
  - There are a number of ways to get the current time, but you could consider using `datetime.now()` from the `datetime` module
    - You will need to access it as `datetime.datetime.now()`!

Chapter Concepts

Chapter Summary

- Command Line Arguments:
  - `sys.argv`
  - `argparse`
  - `fileinput`
- Windows Task Scheduler
  - Python program location
  - Working directory
  - Script and arguments