

How to Backup to a USB Drive on Insertion to a USB port in Windows 8

To do this we will setup -

- A Task which will run everytime we log on to our PC

WHICH WILL

- Monitor our USB ports and whenever a USB drive is connected

WILL

- check if the drive is our backup drive, and if it is

WILL

- run a backup script to mirror any changes in our monitored directory (my documents) since the last backup.

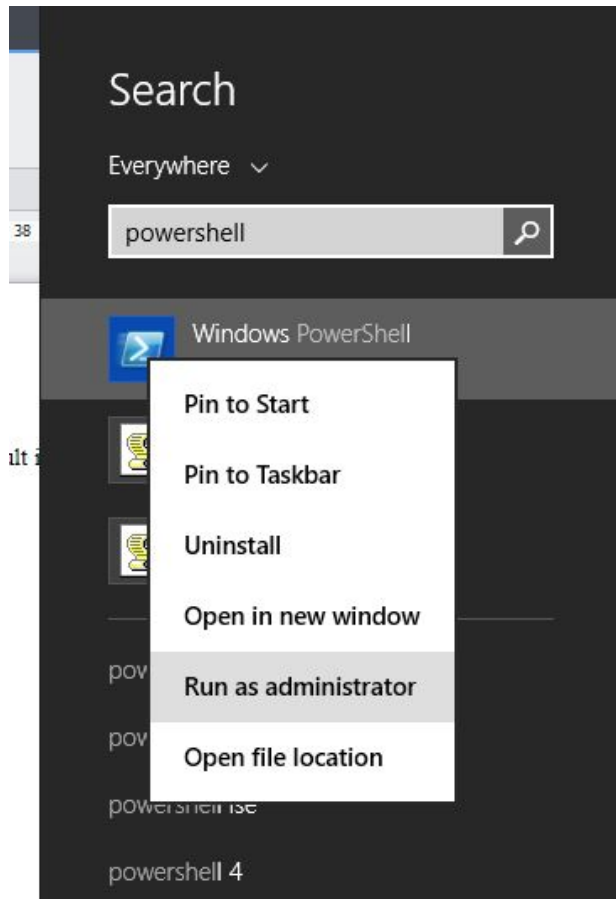
This process involves using the -

- Windows Powershell (which must be set to allow the Execution of scripts)
- Powershell ISE (for writing/editing the script required to monitor the USB port)
- Windows Task Scheduler
- A file sync program which has the ability to be launched, run and closed using a script.

It should also work for other Windows versions, but the windows may look different to those in the screenshots used.

1. Launching Powershell.

To launch Powershell in Windows 8 place the mouse cursor in the top right corner of the screen and from the pop out panel click the 'search' icon, type the word 'Powershell' in the search box, right click on the Windows Powershell option shown and select 'Run as Administrator'.



2. Enabling Scripts in Powershell.

In the window that opens type the words '**Set-ExecutionPolicy Unrestricted**' followed by enter.



Then hit the 'Enter' key to confirm the instruction (no need to type 'Y' as that is the default instruction).

3. Writing the Powershell Script.

Now we need to write a script for powershell which will monitor the USB ports for the connection of a USB device.

To do this open copy the following script then paste it into notepad or some other simple text editor, modifying it as outlined below.

```
=====Script
content=====

#Requires -version 2.0
Register-WmiEvent -Class win32_VolumeChangeEvent -SourceIdentifier volumeChange
write-host (get-date -format s) " Beginning script..."
do{
    $newEvent = Wait-Event -SourceIdentifier volumeChange
    $eventType = $newEvent.SourceEventArgs.NewEvent.EventType
    $eventName = switch($eventType)
    {
        1 {"Configuration changed"}
        2 {"Device arrival"}
        3 {"Device removal"}
        4 {"docking"}
    }
    write-host (get-date -format s) " Event detected = " $eventName
    if ($eventType -eq 2)
    {
        $driveLetter = $newEvent.SourceEventArgs.NewEvent.DriveName
        $driveLabel = ([wmi]"Win32_LogicalDisk='$driveLetter']").VolumeName
        write-host (get-date -format s) " Drive name = " $driveLetter
        write-host (get-date -format s) " Drive label = " $driveLabel
        # Execute process if drive matches specified condition(s)
        if ($driveLetter -eq 'Z:' -and $driveLabel -eq 'Mirror')
        {
            write-host (get-date -format s) " Starting task in 3 seconds..."
            start-sleep -seconds 3
            start-process "C:\BackupScript\mydocuments.cmd"
        }
    }
    Remove-Event -SourceIdentifier volumeChange
} while (1-eq1) #Loop until next event
Unregister-Event -SourceIdentifier volumeChange

=====end of script=====
```

The highlighted parts of the above script should be changed as follows to suit your situation before saving the script.

Z: - should be changed to the drive letter allocated to your backup drive

Mirror - should be changed to the label on your backup drive

C:\BackupScript\mydocuments.cmd - should be changed to the name of your backup script created in the backup program (see later details for the backup program).

Once the changes are made save the script with a suitable name and the extension .ps1.

I used the name 'USBbackup.ps1', - the name is not important, BUT make sure that the save file has the extension '.ps1' and not .txt or anything else.

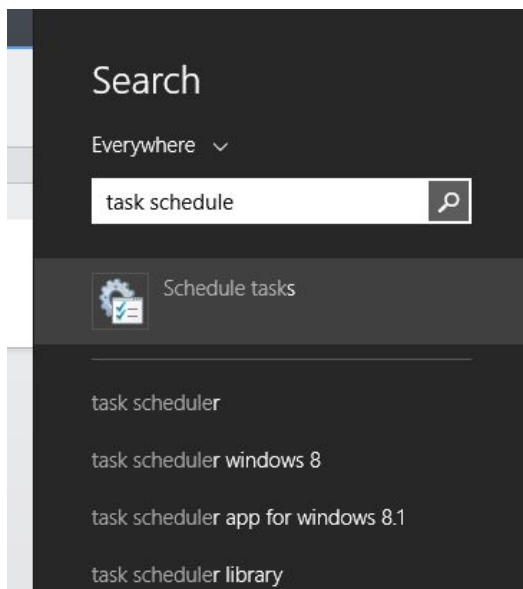
Place the script in a directory of your choice - I put it in a directory called 'BackupScript' on the root of the C: drive.

4. Setting the script to run at windows Log-on.

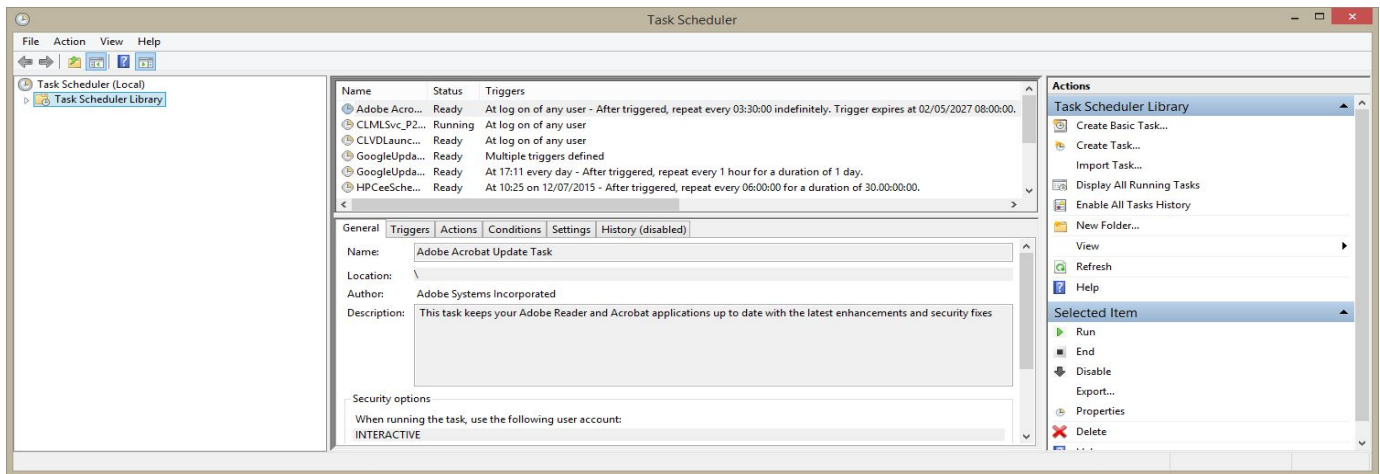
For the backup script to be launched and run automatically when we connect our backup device the script we have just written needs to be running before the device is connected to a USB port. We could run the script manually before connecting the device, but the easiest way to have this happen without having to remember starting it is to have the script launched at Windows Log-on so it is running in the background at all times. (It's a small script and only consumes about 2MB of memory).

To have the script launch at Windows log-on we can use Windows Task Scheduler as follows:

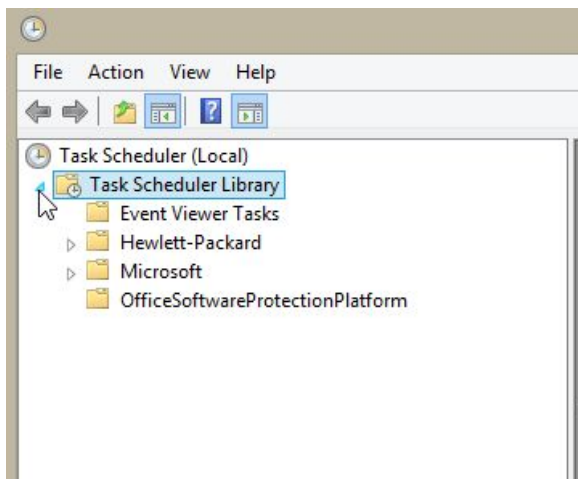
Open the search box like before, but this time type the words 'task schedule' in the search box and click the option 'Schedule Tasks'.



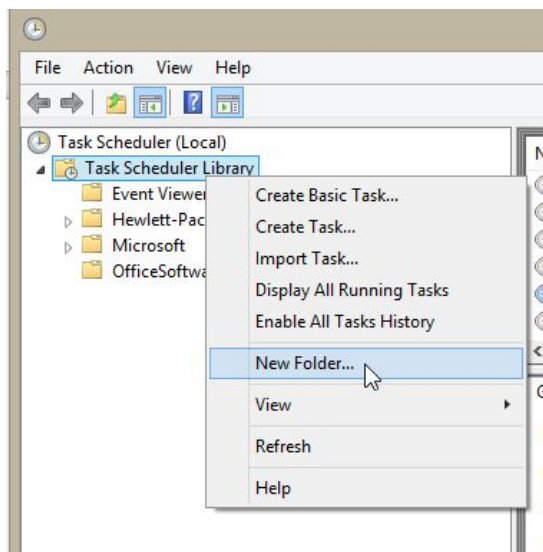
This window opens

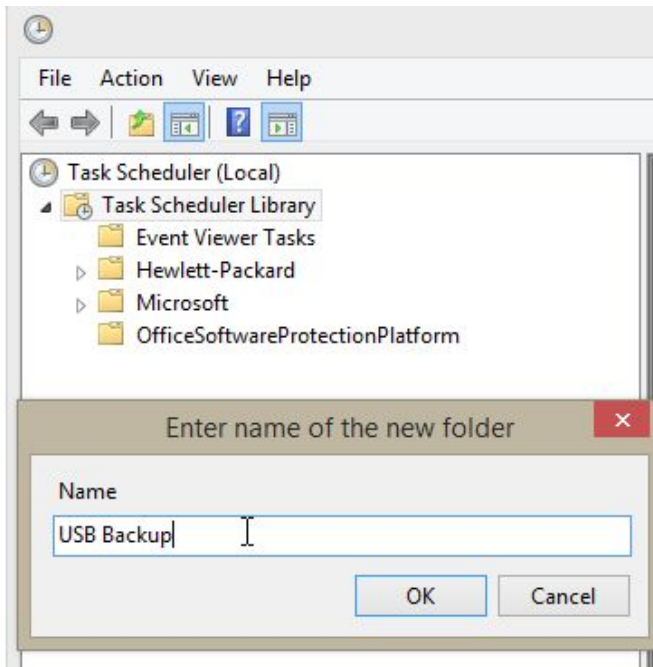


Expand the Library

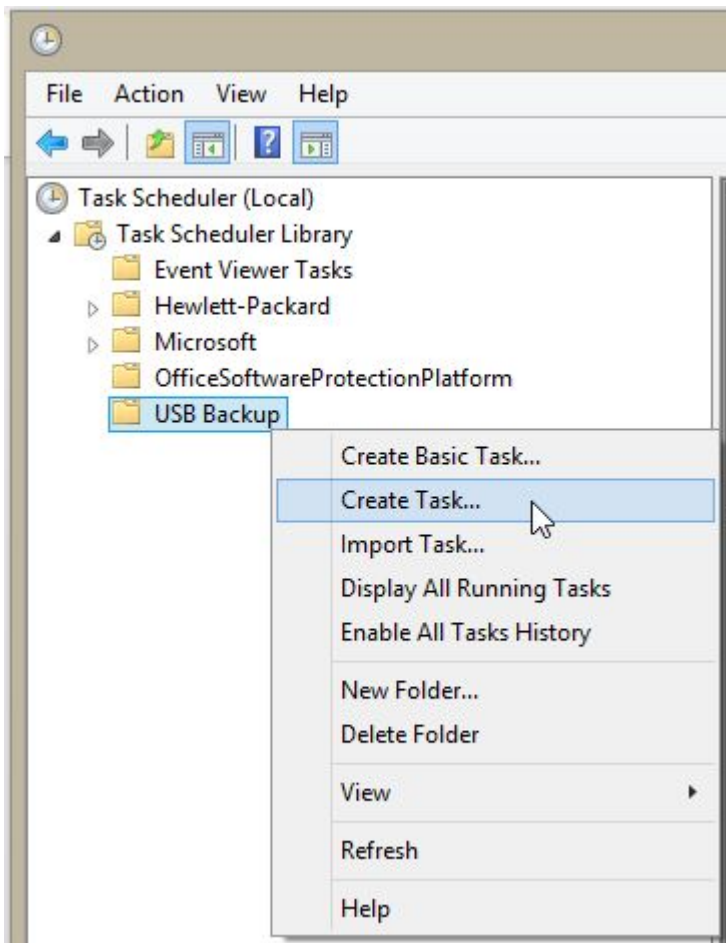


Right click on heading and add a new folder named to suit (in my example it's USB Backup)

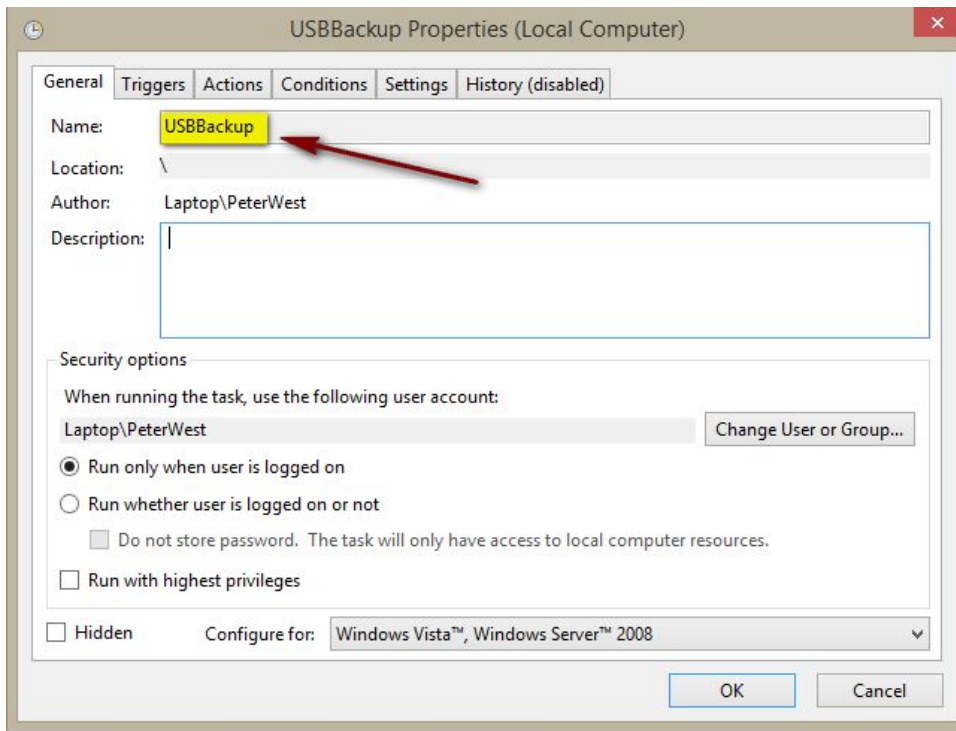




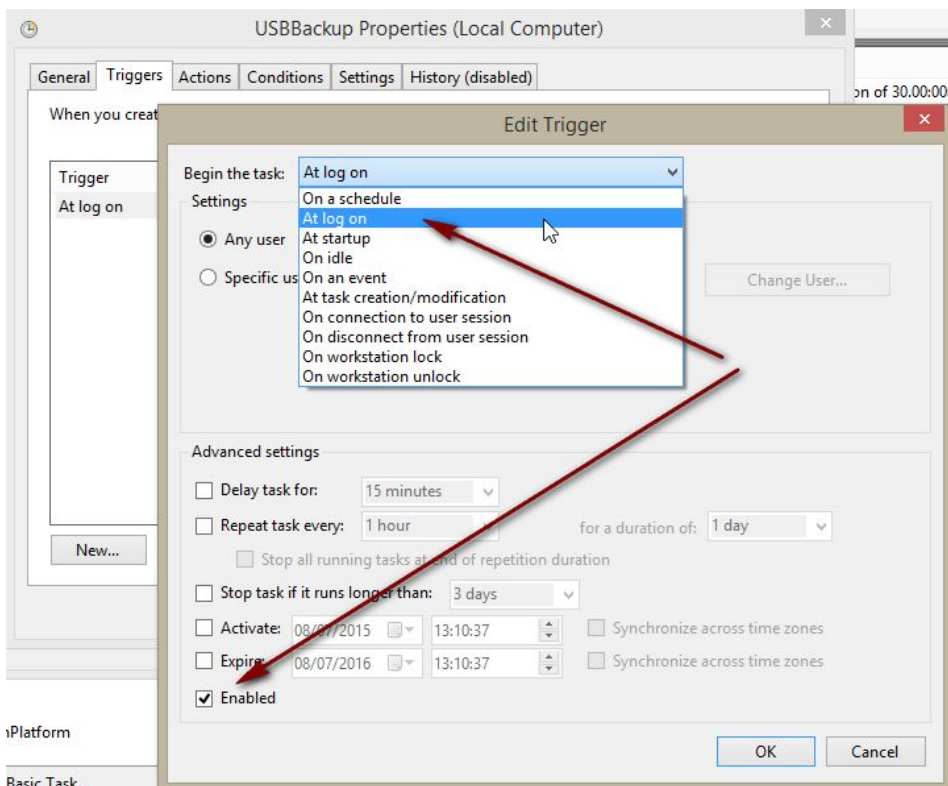
Right click on the new folder name and select 'Create Task'



In the new window give the task a name in the 'General' Tab



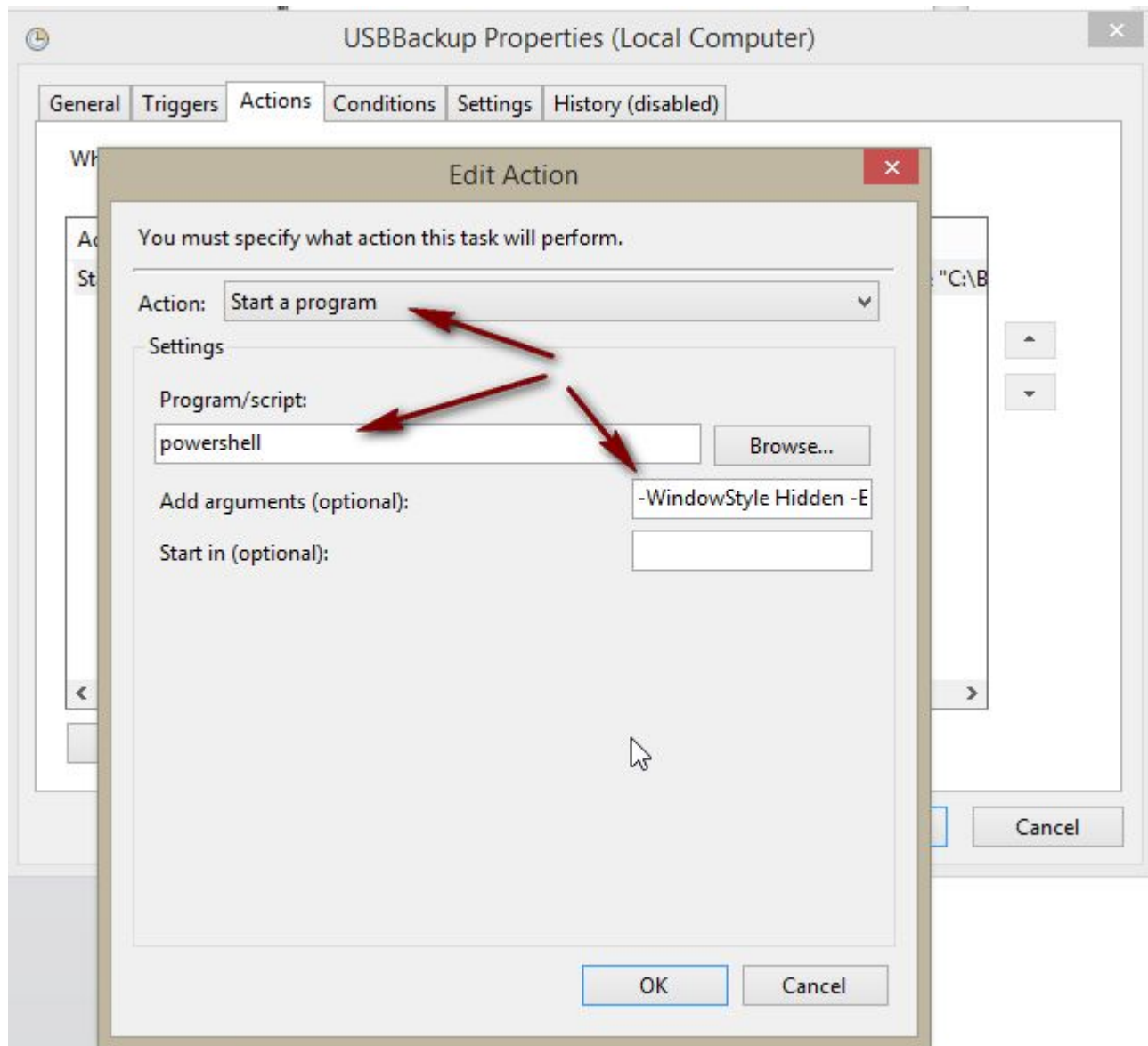
Now select the 'Trigger' tab and from the drop down list choose 'At Log On' and ensure the task is enabled.



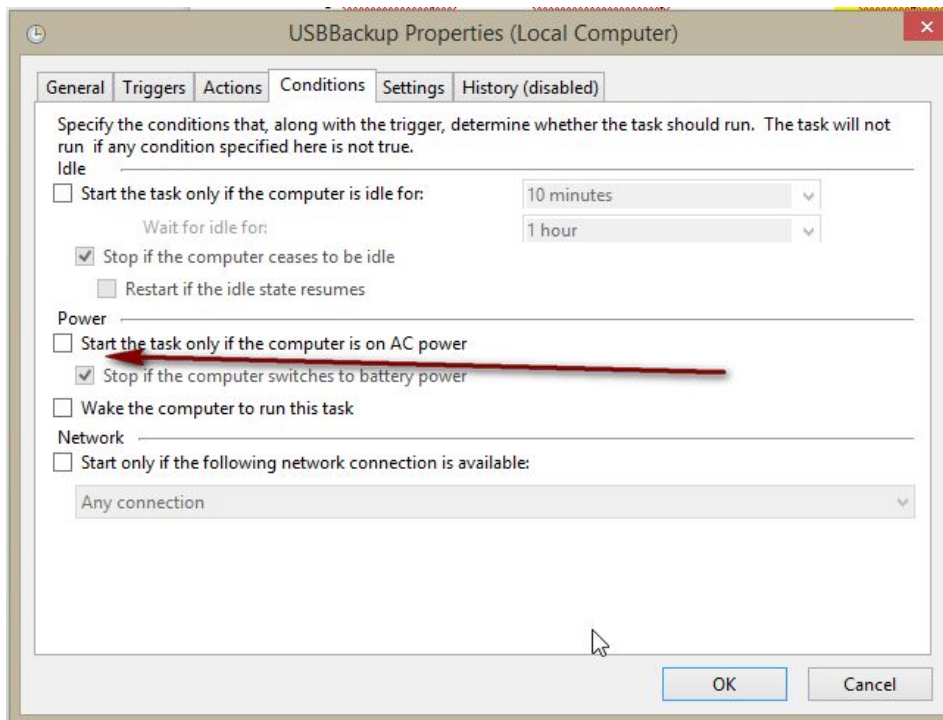
In the actions tab we need to set 'Start a program', tell it which program (powershell), and add the following arguments

[-WindowStyle Hidden -ExecutionPolicy Unrestricted -File "C:\BackupScript\USBbackup.ps1"]

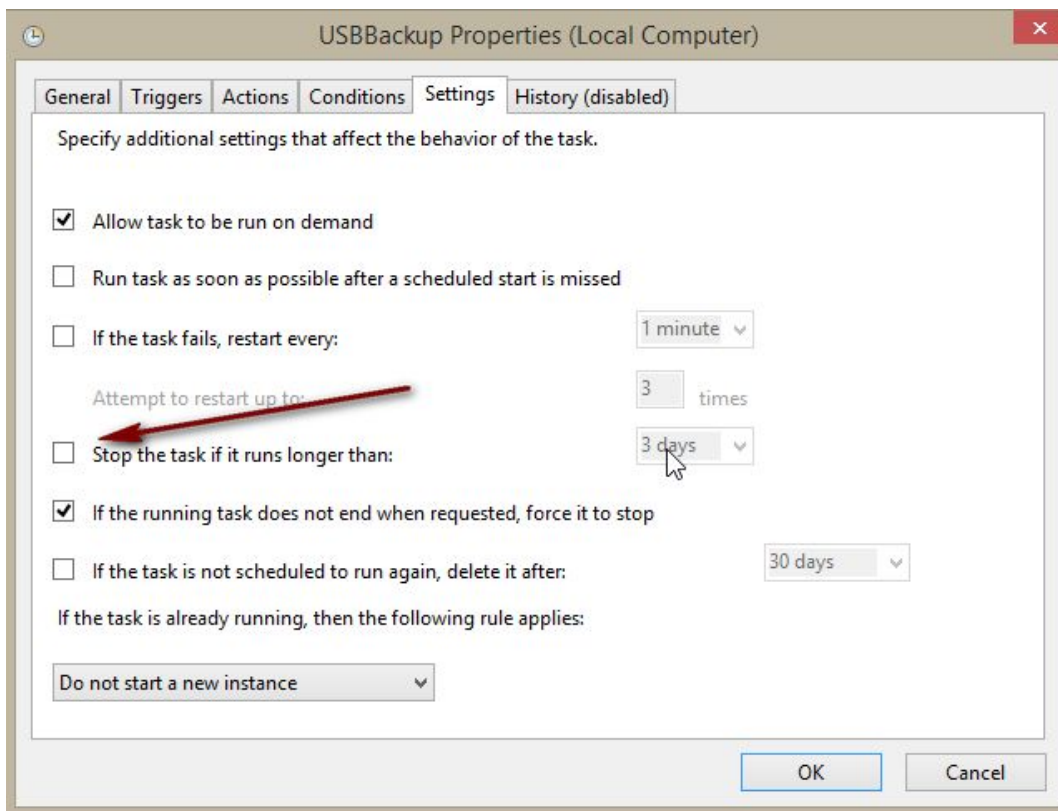
Everything highlighted above between the square brackets [] but not including the brackets and ensuring we set the right script and location (highlighted here) to match the name and location of the script you created earlier.



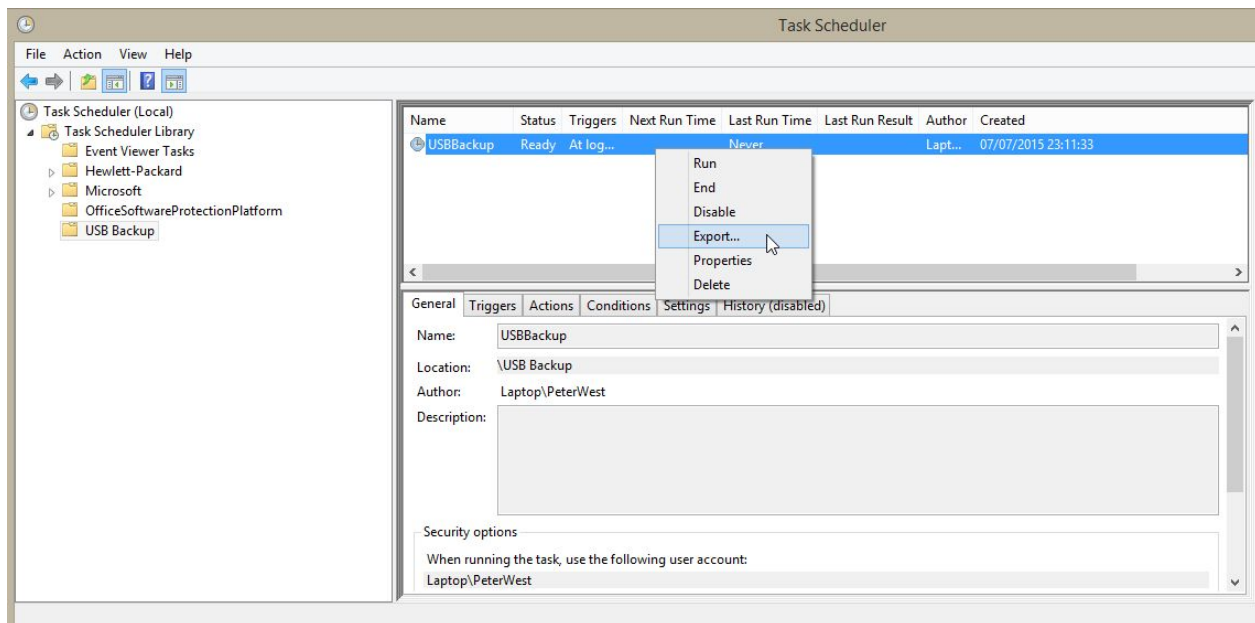
In the 'Conditions' tab, if like me you run a laptop and you want the script to run when you start up the laptop on battery, uncheck the relevant box.



In the 'Settings' tab, if like me you are likely to have your PC on for more than 2 days, uncheck the box so the script does not close.



Having created the Task and clicked OK to save it, we should back it up.
To do this, right click the task in the window and choose 'Export' saving it to a suitable location.



With our Powershell Script created and set to monitor the USB ports once we are logged in to windows, the final thing we need to do is to create a backup script to be run to do the actual backup.

For this I suggest a Free program called DirSync Pro which can be downloaded from here - <http://dirsyncpro.org/>

I won't go in to all of the programs features as you can read all about them on their web site. I will say, however, that the reasons I chose it are as follows:

- It is portable. It does not need to be installed and runs from it's own directory which can be on your PC or on the USB backup drive(s).
- It can be set to run as a service to do realtime backup of files/directories on PC's to second hard drives or USB devices (such as permanently connected Memory Sticks etc.
- It can be used to configure command line scripts which will do what we need here - ie to launch the program, run a backup routine and then close the program again.

It does however require Java be installed on your computer - so if you do not have java you need to download and install it from here - <http://java.com/en/>

Configuring the backup in DirSync Pro.

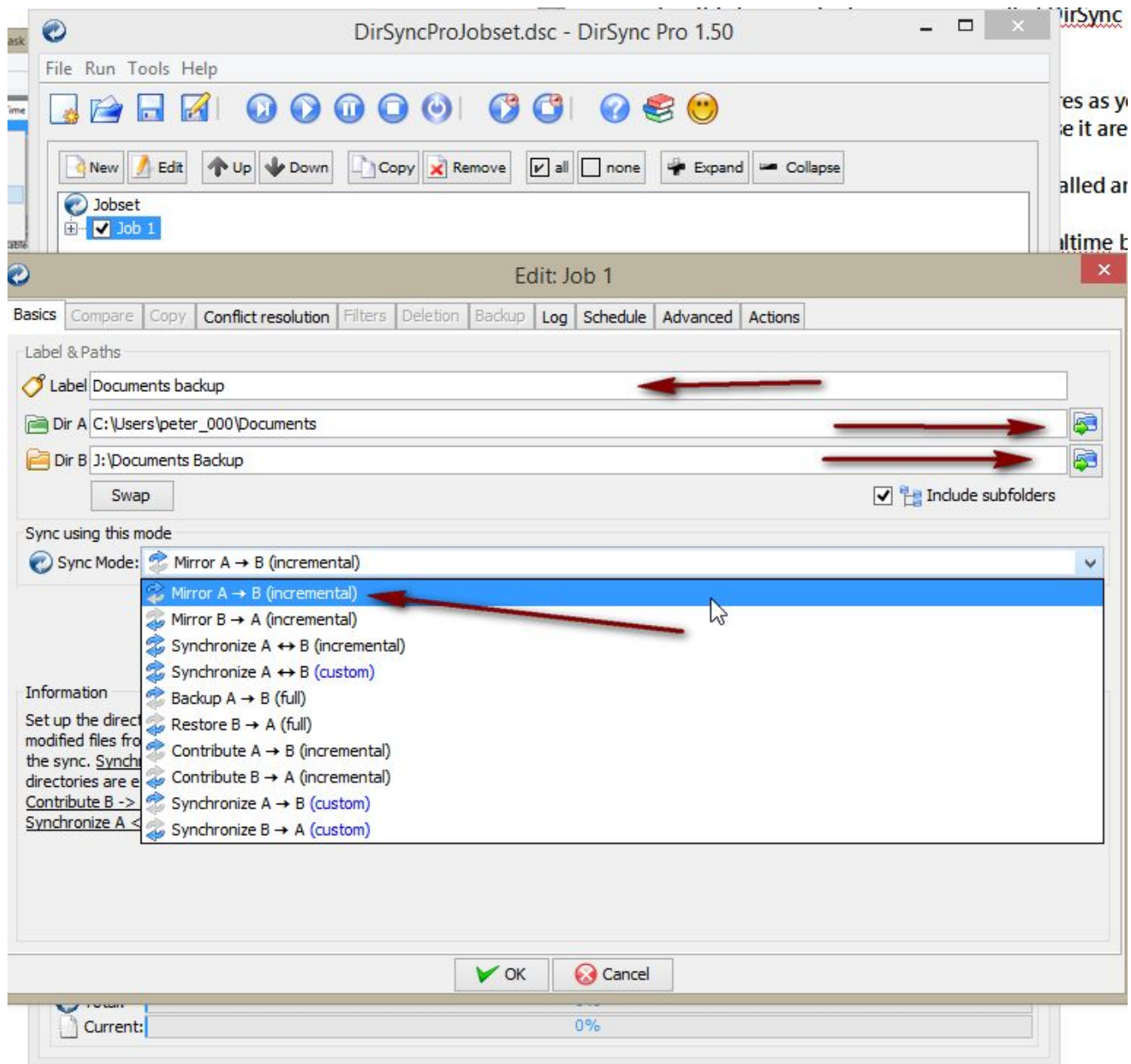
Download the file from the DirSync website saving it to a suitable directory.

NOTE: the suitable directory can be on your PC hard drive, or on the backup drive. For example I saved and extracted the file to a directory called 'Sync' on the external drive I am backing up my files to.

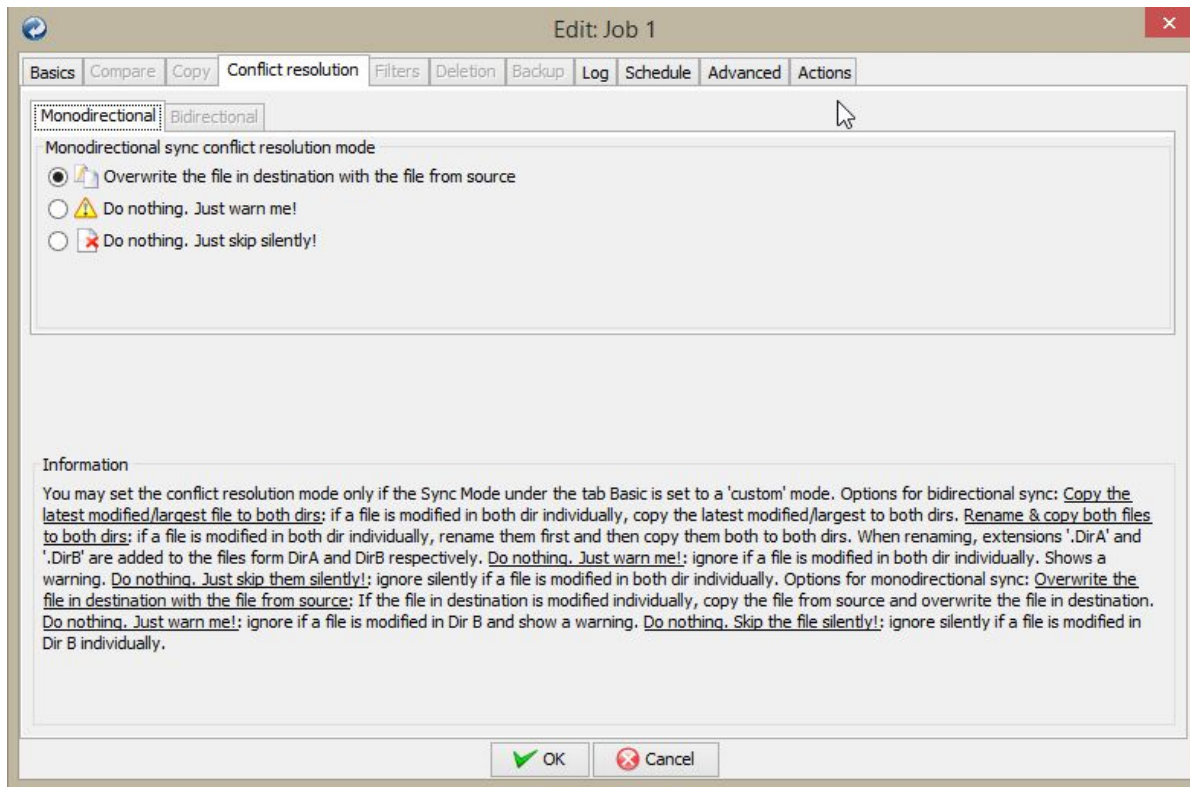
NOTE2: If you have not already done so to undertake NOTE 1, connect the backup drive to the PC to make it available for the next steps.

Once extracted, in the chosen directory double click the program file (DirSyncPro.exe) to run it and on first run accept the license agreement

In the main program window, double click the default Jobset - 'Job 1' - to edit it.
Change the name to something meaningful, select the directory you wish to keep backed up, and then the directory to which you want it backed up, and select the type of backup you want to do. EG Mirror A->B



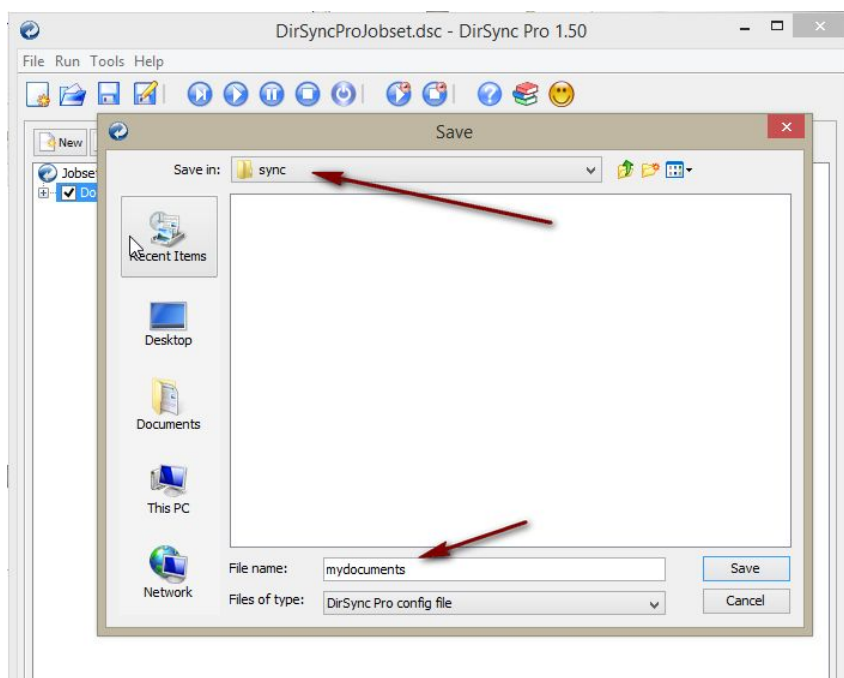
In the 'Conflict Resolution' tab choose how you want conflicts dealt with



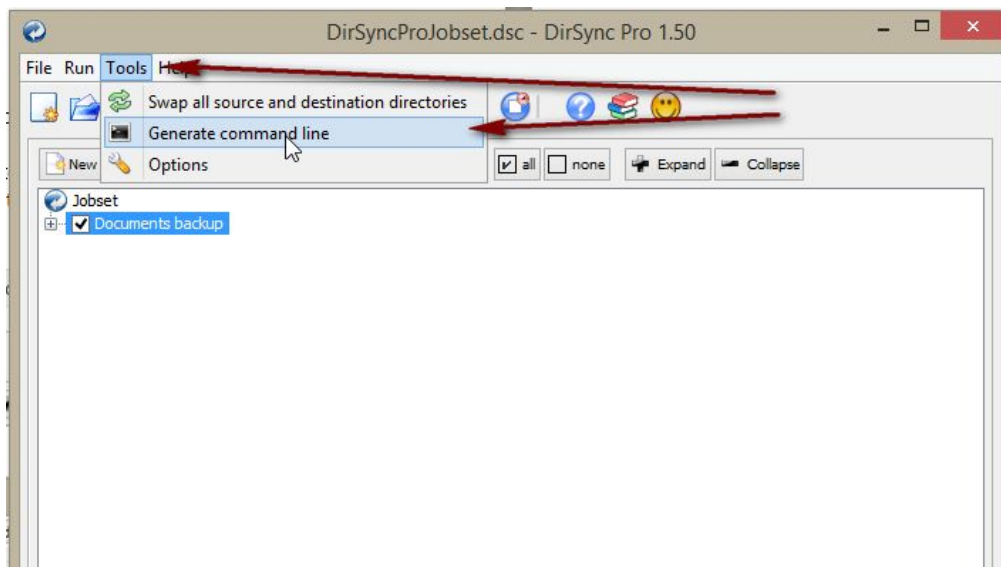
And click OK to save the job.

You can have a look at the other tabs to see if there is anything in them you would like to change - I didn't bother.

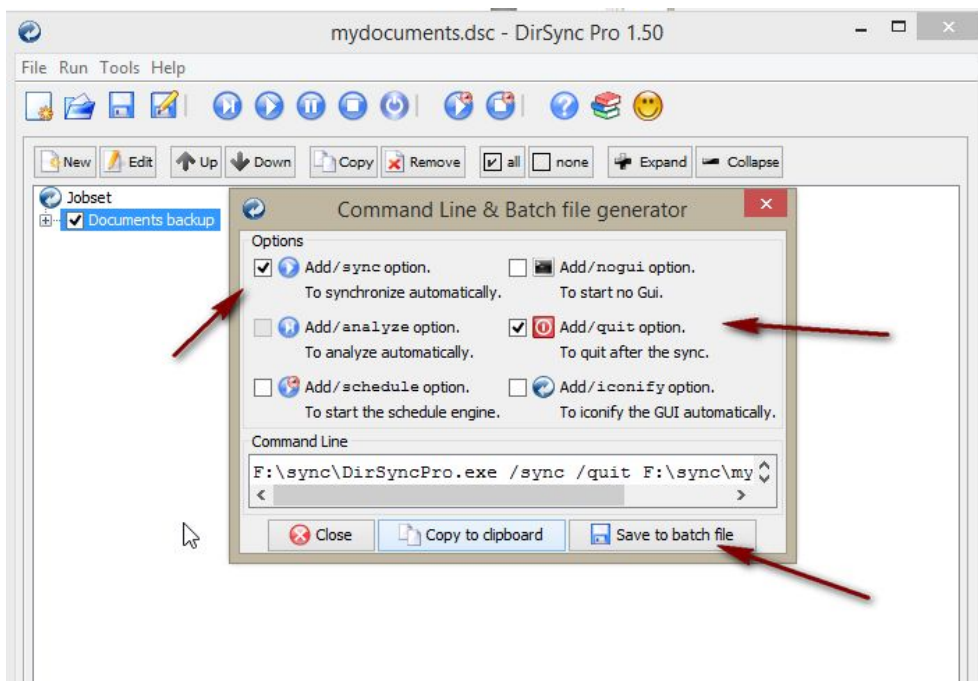
Back in the main window select 'File/Save' and save the job with a suitable name - I saved it into the same directory as the program on my backup drive.



Now select 'Tools' and 'Generate command line'



And check the two items to run a sync and also to quit the program after the sync is completed.



Finally click the option to 'Save to batch file' and save the file to the directory set up earlier - in my example **C:\BackupScript**

That's it.

So we now have setup -

- A Task which will run everytime we log on to our PC

WHICH WILL

- Monitor our USB ports and whenever a USB drive is connected

WILL

- check if the drive is our backup drive, and if it is

WILL

- run a backup script to mirror any changes in our monitored directory (my documents) since the last backup.