**How to Set Up a Cron Job on Windows for Weekly Automated AIS Download**

**16 December 2020**

**Introduction**

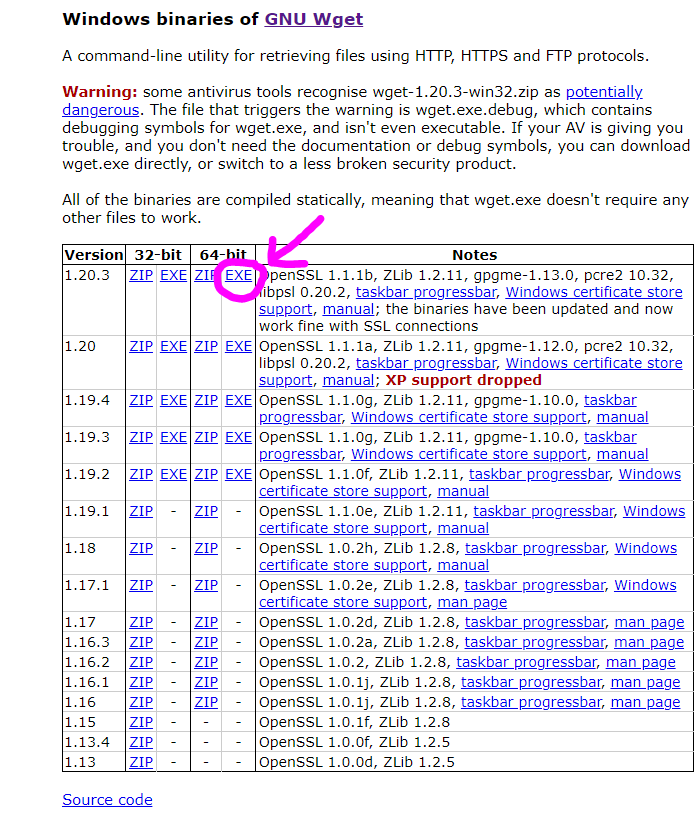
If you work with AIS data, you may know that some websites store their data in thousands of different files. Additionally, some organizations update their AIS data every day, which can make returning to the website every day to download the data very tedious. This documentation will walk you through how to set up a scheduled task to automatically download your AIS data from an HTTP website. You can set up the schedule however you want, once per day/week/month/etc.

This example was done with Windows 10 and wget version 1.20.3.

**Step 1:** Download the wget executable for windows from the GNU website.

Link for download: <https://eternallybored.org/misc/wget/>

wget is a command that you can run in your command prompt to manage your downloads from websites. I placed the executable in the folder that I wanted the files to download to, but you can also save it under Windows the windows folder.

****

**Step 2:** Check if wget is installed correctly.

Before setting up the job, let’s make sure that wget is installed correctly. Open the command prompt and set your directory to where the wget executable is stored. In this case, the executable is stored in a folder labeled “AIS” so I will set the director to that folder with the following line:

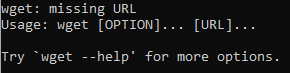
https://lh4.googleusercontent.com/b5SKsEvJ_UGysM4utI7atWhmuhNLZSjZQAiliV_hKzUN86arwplFSfd61g6jcFQD85r0qVyjlOCEzJoIaUyZpFnmDl9IjOZ7nRD66uFLQJRT3bvpDyWOwu2bMDtPA9Amgv3FPK_e

Now, type wget in the command prompt.

If the message below is returned, the command prompt cannot find wget, either because it was not installed properly, or because it is in a folder that cannot be found.

https://lh6.googleusercontent.com/RbACF2CkF9UVomKfEVCjrgixjfhTR1rQ_pYMTZzFfPk3NCFcJ2WAb8OnoOrb_mRwGWO04-YOCcO4YvruH6NNfFod7rcAqTTDvKNjFOIiM-_7p03IG3MVROsVsCasHQcT2empvf1u

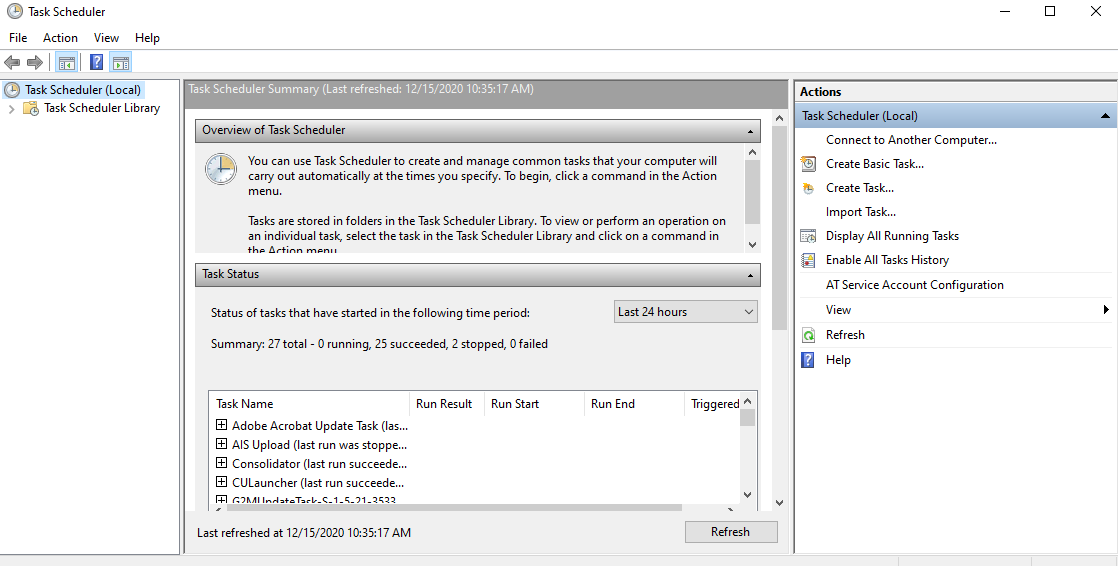
If the below message is returned, then wget is working and you can move to step 3.



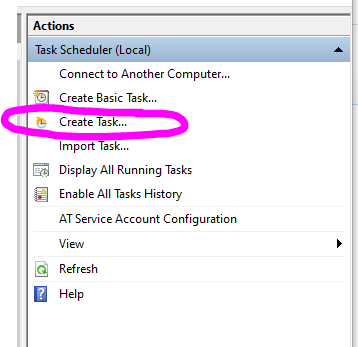
**Step 3:** Setting up the scheduled download.

A “cron job” is a time-based job schedule in Unix-like computer operating systems. With Windows machines, we can set up a scheduled job using the “Task Scheduler.”

**Step 3.1:** Type Task Scheduler into your search box. This is what the app looks like.



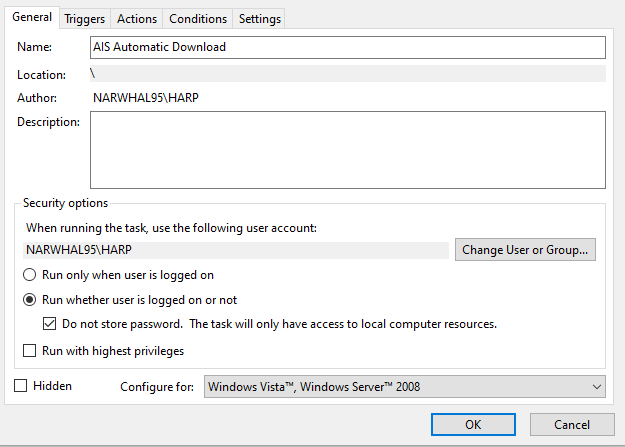
**Step 3.2:** In the right hand toolbar under “Actions,” click on “Create Task…”



**Step 3.3:**

Name the task whatever you would like, I named it “AIS Automatic Download.”

There are many different security options that you can choose from. In this example, I would like the job to run whether I am logged on or not, and I do not want my password stored.



**Step 3.4:** Move to the tab labeled “Triggers” and click “New…” in the bottom left hand corner.This is where you can pick the schedule that you would like the job to run on.As you can see there are manyoptions, and you can be as specific as you want.

Starting in the top right corner, I set “Begin the task:” to “On a schedule.” I then chose the schedule for running the task to occur weekly, starting at 11 AM on Sunday. That is as detailed as I want to make it, so I won’t even mess with any of the “Advanced settings.” Mark the box “Enabled” in the bottom right hand corner to enable the schedule, and then click OK.

**Step 3.5:** Move to the tab labeled “Actions” and click “New…” in the bottom left hand corner.

Starting at the top, I set “Action:” to “Start a program”. Under Settings, click “Browse…” to point the “Program/script:” to where your wget executable lives. Mine lives here “D:\AIS\wget.exe”.

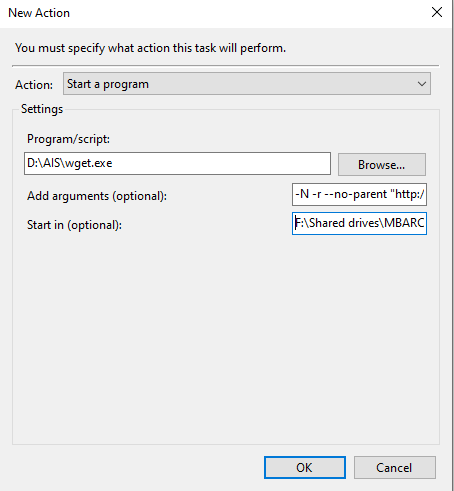
You need to add an argument to tell wget what to do and where to pull files from. The argument I use is

**-N -r --no-parent “**[**http://aisdata.sbwireless.org/**](http://aisdata.sbwireless.org/)**”**

which tells wget to go to the following website to download the files. Note, you need to put quotes around the website!

-r --no-parent tells wget to download all of the website's contents with all the files and subfolders and their files. -N tells wget to check if the remote files have changed, and only download it if it has (this allows wget to only download new files, and not download everything all over again).

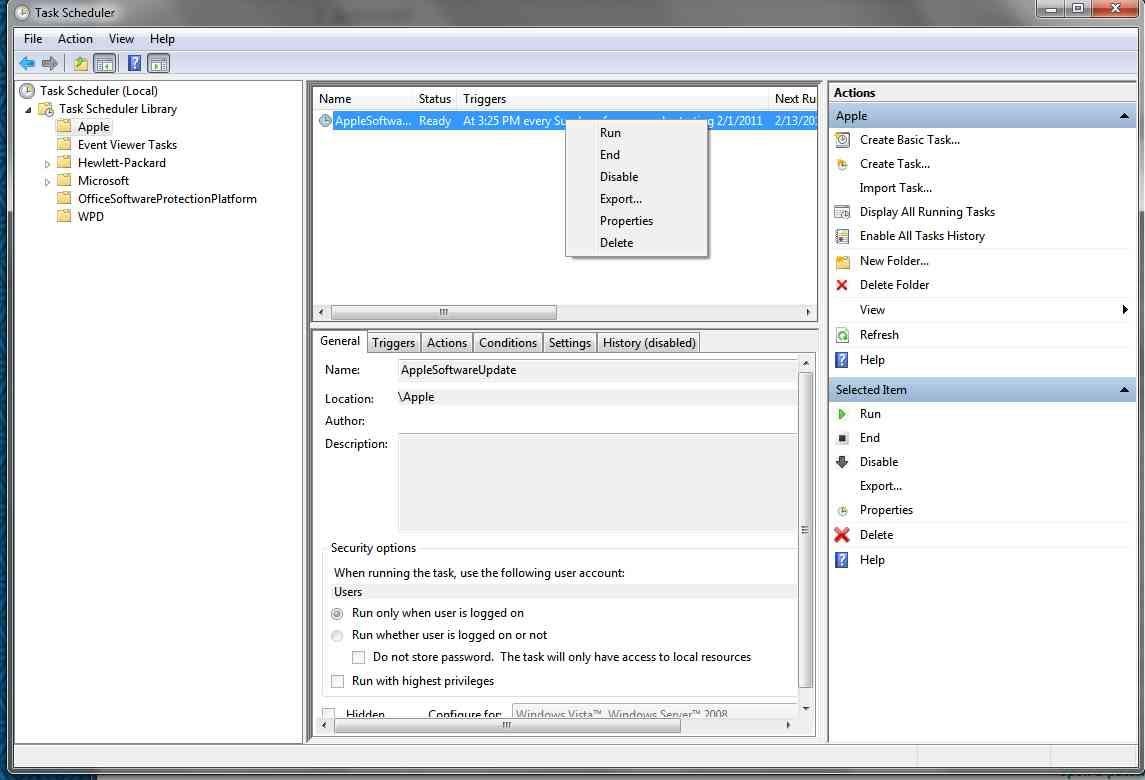
Finally, point to the directory where you would like the files to be saved in the “Start in (optional):” box. I would like the downloads to go into my Google Drive, so I pointed to my Google Drive file stream like so “F:\Shared drives\MBARC\_AIS\SBARC”.

****

**Step 4:** Test to see if the job works.

You probably don’t want to wait until Sunday at 11 AM to see if the job works, right? You could set the job to run at a time in the near future where you could see if it worked, OR you can do the following:

**Step 4.1:** Find the task in the Task Scheduler and right click on it. From the drop down menu, click “Run”. This will force your job to run, even though it is not part of the scheduled run. You can see that the task is running if under “Status” it says “Running”.



**Step 4.2:** Go to where you told the task to save your files and check if your files are being plopped in there. It will take time for the job to run. It will check on every file to see if there has been an update, so it will take ~1.5 hours (if you have thousands of files like me).

If you see your files in the folder, it worked! If they’re not there, then it didn’t work ☹. Look at the additional comments below to see if your issue falls under one the following discussed.

**Additional comments:**

* The job will make the folders before downloading the files in the folders. If you see that the first folders are there, but there are no files in them, WAIT, the job may just be working on making more folders before downloading the files.
* The website in the “Action” needs quotation marks around it, make sure yours has them!
* Do not put the wget command in the “Action”, only put the arguments.

Email me if you need any help with this!

Vanessa ZoBell: vmzobell@ucsd.edu