Deploying Windows Extractors

Instructions for Creating VM in Nebula

1. Create a Windows VM in Nebula.
    a. Choose the instance boot source to be "Boot from image (create new volume option)".
    b. Choose appropriate VM flavor to get sufficient disk size and number of CPUs
    c. Choose appropriate security and networking
    d. Launch instance
    e. Associate a Floating IP address
    f. Login to the VM and enable remote desktop connection

Install the Extractor

1. Install ArcGIS
    a. Available on the WebStore for free
    b. Instructions with ArcGIS work for the install
    c. ArcGIS also includes the Python that it expects to use
    d. Get the path to ArcGIS as it will be used in the next step (hereafter referred to as <ArcGIS PATH>)
2. Add Python, pip, and 7zip to the Path
    a. These work for Windows 10, future versions of Windows may change how the path is altered
    b. On the control panel search for Edit the System Environment Variables, click the Environment Variables button
    c. The lower half of the new window has the system variables, choose Path and then the Edit button
    d. Click the New button, to add a new section to the path
    e. Add <ArcGIS PATH> to the top of the list of path, and <ArcGIS PATH>/Scripts right beneath it (this will use the Python installed with ArcGIS before anything else)
    f. Add "C:\Program Files\7-Zip\" to the end of the path.
        i. Quote the value due to the space in Program Files
3. Update pip
    a. `python -m pip install -U pip`
4. Install pika and requests
    a. The version is important as newer versions introduced errors
    b. `pip install pika==0.11.2`
    c. `pip install requests==2.18.4`
5. Install Git Bash to run git from the command line
    a. Download and install from gitfromwindows.org
6. Install pyclowder version 1
    a. The extractors have not been updated to the latest pyclowder yet
    b. Set up a Repositories directory in home directory, and change into it
    c. From Git Bash run `git clone https://opensource.ncsa.illinois.edu/bitbucket/scm/cats/zzpyclowder1.git`
    d. Change into the zzpyclowder1 directory
    e. `python setup.py install`
7. Install the extractor
    a. Use the version from Clowder, rather than Browndog
    b. From Git Bash run `git clone https://opensource.ncsa.illinois.edu/bitbucket/scm/cats/extractors-bd-cz.git`
    c. This puts the scripts into place to be run
    d. The config.py for each extractor will need to be updated to point to the proper RabbitMQ URL and Clowder registration endpoints
        i. For browndog.ncsa.illinois.edu use:
            1. RabbitMQ URL: amqp://<USER>:<PASSWORD>@rabbitmq.ncsa.illinois.edu:5672/clowder[-dev]
            2. Clowder Registration: https://browndog.ncsa.illinois.edu/clowder[-dev]/api/extractors?key=<KEY>
        ii. If running everything locally then start RabbitMQ and Clowder locally, and the default values in config.py will likely work
8. Test that the extractor works
    a. Run it from the cmd prompt and watch the outputs to be sure it is working properly.
    b. Also advised to upload some files to Clowder and observe that the extractor is receiving, processing and uploading the results back correctly.

Build Windows Service [Uses NSSM]

1. Create a simple bat file to run extractor and save it in some directory that is easily accessible:

```
cd /d <full_path_to_extractor_directory>
python Landsat7mosaic.py
```

2. Download and install NSSM. At the time of writing this document, the latest stable version is 2.24.
Make sure that NSSM is added to your PATH for easy access from the command prompt.

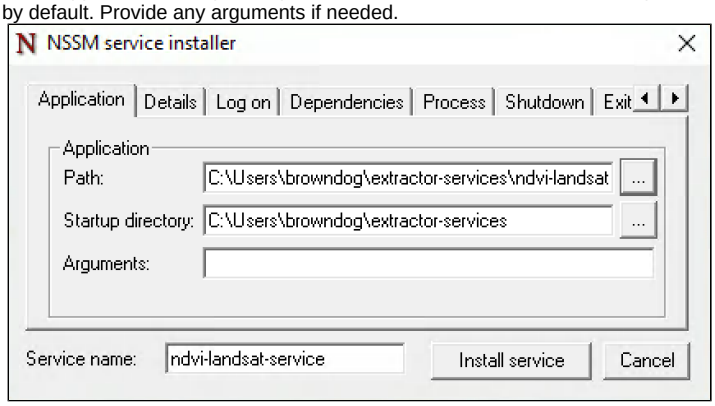
3. From Windows Command Prompt, run the command to install a new service. You will need to provde administrative privileges to execute this command:

```
nssm install <service_name>
```

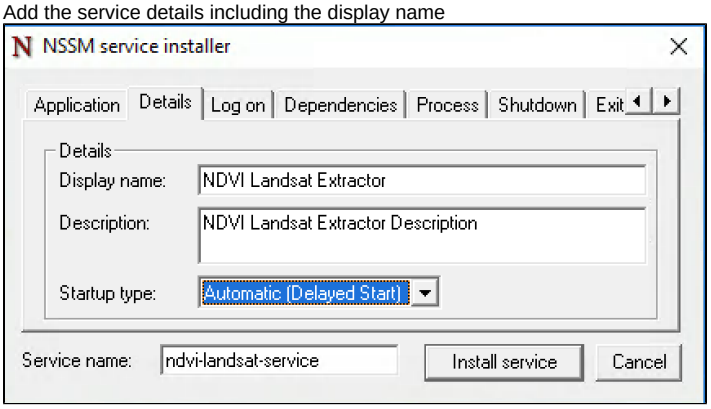
Here the <service_name> is the unique name of the service. This is different from the display name that shows up when you open the Windows Service Manager.

4. Fill in the details of the extractor and the service settings in various tabs

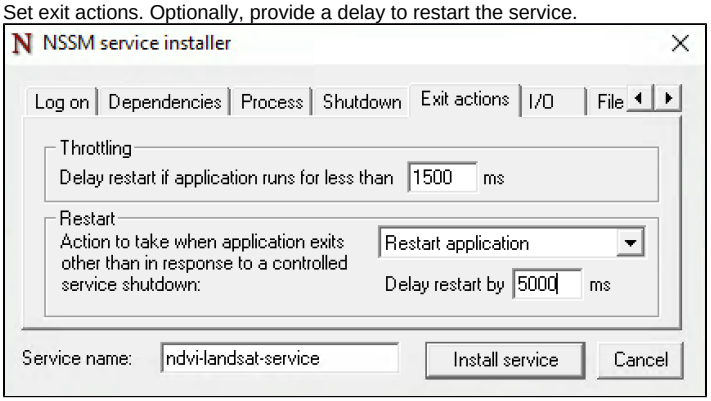
    a. In the Application Path, put the path of the bat file that was created in step 1. Startup directory will be the directory containing the bat file by default. Provide any arguments if needed.



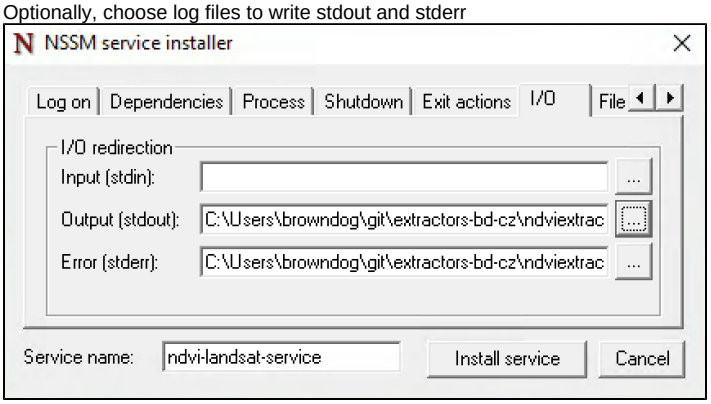
    b. Add the service details including the display name



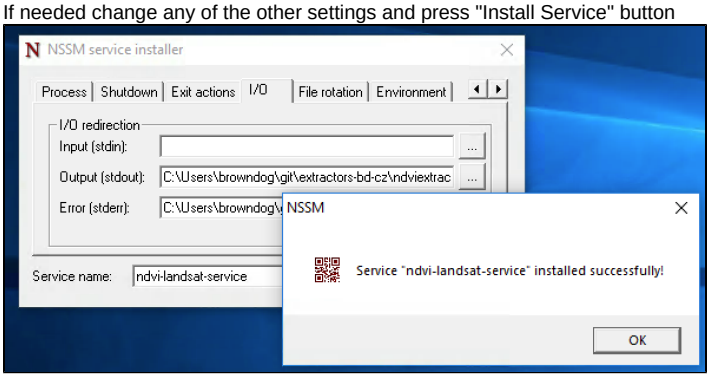
    c. Set exit actions. Optionally, provide a delay to restart the service.



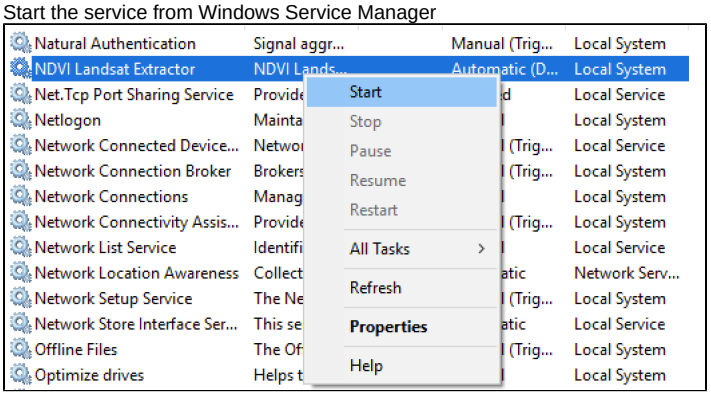
d. Optionally, choose log files to write stdout and stderr



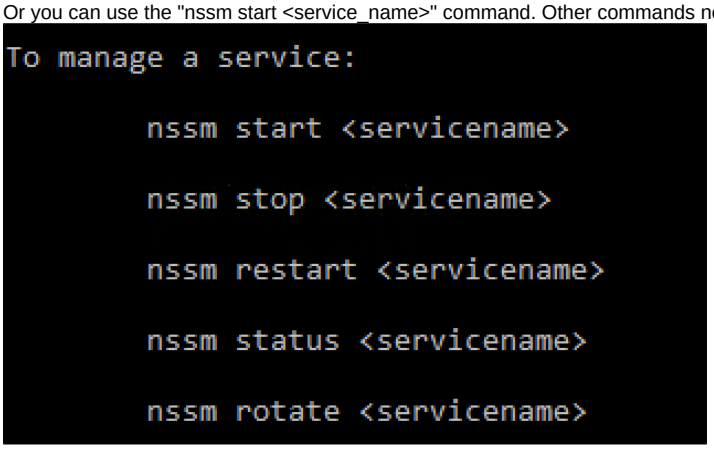
e. If needed change any of the other settings and press "Install Service" button



f. Start the service from Windows Service Manager



Or you can use the "nssm start <service_name>" command. Other commands needed to manage the service are listed below:



For example,

```
nssm restart terex-floodplain
```

g. Make sure that the corresponding extractor queue was created in RabbitMQ. Done!

~~Build Windows Service:~~

*[This section exists only because of historical reasons. The method described here uses srvany and is **not recommended** because of issues with restarting and stopping the service cleanly. DO NOT USE THIS METHOD!]*

~~1. Prepare a simple bat file to run extractor, like:~~

~~cd /d <full_path_to_extractor_directory>~~

~~python cellprofiler_extractor.py >> log_<some_extractor_specific_name> 2>&1~~

~~2. Convert bat file to exe file~~

~~For that can bat_to_exe_converter can be used: http://www.computerhope.com/issues/ch000435.htm~~

~~3. Get srvany.exe~~

~~It can be found on Windows server 2003 resource kit tools: http://www.microsoft.com/en-us/download/details.aspx?id=17657~~

~~4. Use install.bat to build a Windows Service:~~

~~install.bat can be found here: https://opensource.ncsa.illinois.edu/stash/projects/CBI/repos/cyberintegrator-2/browse/edu.uiuc.ncsa.cyberintegrator.server.feature/root/service~~

~~It assumes srvany.exe is on a subdirectory called service (create this subdirectory and copy srvany there)~~

~~Edit install.bat file changing NAME, DESC and EXE to reflect the name, description and executable file of your service.~~

~~Set HOME to the directory containing install.bat script~~

~~Run install.bat as administrator and install the service~~

~~5. Check that the correspondent queue was created on the rabbitmq server that the extractor connects to:~~

~~Delete queue first if already exists and no one else is using~~

~~If not working, try checking log errors in the event viewer (find it by typing eventviewer on the start search box) under windows logs, in system or in application and filtering for errors (filter current log on right panel, check box for "Error") which have "event sources"=Service control manager. (http://stackoverflow.com/questions/1067531/are-there-any-log-file-about-windows-services-status)~~

~~6. Change windows service configuration to "delayed start"~~

~~Open "services" from win start menu (type services on the start search box and click enter)~~

~~Right click on the extractor >properties~~

~~Change startup type to "automatic (Delayed Start)"~~

~~7. Restart VM~~

~~Do not login and see if the queue was created in the rabbitmq; it will take a couple of minutes because of the "delayed start"~~

~~Delete queue first if it already exists to be sure the extractor service can create the queue if necessary~~