Data Transfer

I. Introduction

The job of the data transfer is 1) to ensure all the data from the machine gets to CPL's side. 2) The data on CPL's side is organized. And, 3) **Web tracking documents** are produced so analysts can have a quick look at the data. To facilitate this task I have written the **transfer** script. This document will give a brief explanation of the **transfer** script and how to use it to accomplish this task.

II. The Transfer Script and Config File

The **transfer** script is written in bash and has the following path: **/nrims/common/scripts/transfer**. It requires a **config file** to set certain parameters so that it can run correctly. The config file for the **Prototype** has path **/nrims/common/transfer_cfg/transfer-Prototype.cfg** and config file for the **50L** has path **/nrims/common/transfer_cfg/transfer-50L.cfg**. See the attached Appendix for a detailed explanation of the contents of the config file.

III. Running the Transfer Script

To run the **transfer** script, type the following command into the command line:

```
>> transfer <config_file_path>
```

or, for 50l (config file and log location are built-in):

```
>> /nrims/common/transfer_cfg/50l-transfer
```

If everything is set up correctly, the script will copy over all the ***.im*** files from the *SOURCE* directory to the *DESTINATION* directory (as specified in the config file). Then, depending on the exact settings in the config file, the script will then generate ***.nrrd*** files, ***.png*** files of the sum images, HSI images, generate a **web tracking document**, and upload it online.

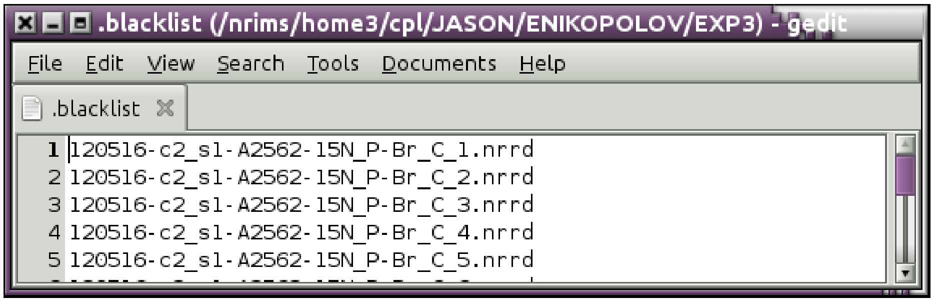
The user may also run the transfer script with the **-now** flag. This avoids the *WAIT_TIME* setting in the config file and runs the transfer script immediately. This is useful for cases when the user want the transfer of files to occur immediately. Usage:

```
>> transfer <config_file_path> -now
```

NOTE: The transfer script is responsible for transferring large amounts of data and can be computationally intensive. Be careful to make sure all parameters in the config file are set correctly and watch the output for any errors or undesirable behavior.

IV. Blacklist files

It is very likely that there are “bad” data files that do not need to be transferred, or should be transferred but should not be included in the tracking document. To exclude files one needs to create a file in the **DESTINATION** directory called “**.blacklist**”. Any file name that is entered in the “**.blacklist**” will not be included in the final tracking document. It does not matter if you use the *.nrrd* or *.im* extension, only the prefix is used to exclude files.



```
120516-c2_s1-A2562-15N_P-Br_C_1.nrrd
120516-c2_s1-A2562-15N_P-Br_C_2.nrrd
120516-c2_s1-A2562-15N_P-Br_C_3.nrrd
120516-c2_s1-A2562-15N_P-Br_C_4.nrrd
120516-c2_s1-A2562-15N_P-Br_C_5.nrrd
```

V. Running the Transfer Script with Cron

To keep CPL's side in near sync with the data on the machine side. I have set up the transfer script to run as a cron job every on darius every hour. There are actually 2 cron jobs related to the transfer of data; one for the Prototype and one for the 50L. The user should probably fimiliarize themseleves with cron (for example, cron needs to be resarted when the machine is rebooted). Type ``**cron -l**″ into the command line on **darius** and you should at least get the following two lines (there may be additional output):

Note that the 50l transfer uses a wrapper script that creates log and error files in /nrims/common/transfer_cfg/log.

```
>> crontab -l
```

```
0 * * * * export DISPLAY=:0 && /nrims/common/scripts/transfer /nrims/common/transfer_cfg/transfer-Prototype.cfg > /tmp/transfer-Prototype.log
0 * * * * export DISPLAY=:0 && /nrims/common/transfer_cfg/50l-transfer /nrims/common/transfer_cfg/transfer-50L.cfg
```

Run every hour

Path of the transfer script.

Path of the config file.

Set the display. This is required in order to run the plugin and imageJ.

Appendix 1 – Example Configuration File with Notes.

SOURCE: The directory that *.im* files get written to by the machine. See sshfs (http://fuse.sourceforge.net/sshfs.html) for details how to do mount this directory. I use the following commands:

```
>> sshfs ims1@newims:/export/home/ims/data /mnt/prototype
>> sshfs mimsop1@shaman:/nrims/mims_data /mnt/50l
```

```
#######################
## Transfer parameters
#######################
SOURCE=~zkaufman/50l/Enikolopov/Exp3
DESTINATION=~cpl/JASON/ENIKOPOLOV/EXP3
WAIT_TIME=240
```

DESTINATION: The destination directory on CPL's side.

WAIT_TIME: Number of seconds before rechecking file sizes. Files that have grown are assumed to be mid acquisition and are not copied over.

```
#######################
## Generate .nrrd files
#######################
NRRDS=true
TRACK=true
TRACK_MASS=26
```

NRRDS: True mean *.nrrd* files will be generated. This almost always set to true.

TRACK: True means the image will be tracked using TRACK_MASS before it is converted to .nrrd file.

```
#######################
## Generate .png files
#######################
PNGS=false
PNG_DIRECTORY=pngs
PNG_OVERWRITE=false
```

PNGS: True means *.png* files of the mass images and HSI images will be generated, and placed in /DESTINATION/PNG_DIRECTORY

PNG_OVERWRITE: If true, the transfer script will regenerate all the *.png* files every time it runs. Otherwise, it will only generate those that are needed.

```
#######################
## Generate HSI
#######################
             HSI= "13/12  27/26"
HSI_THRESH_UPPER= "180    50"
HSI_THRESH_LOWER= "107    37"
     HSI_RGB_MAX= "51     51"
     HSI_RGB_MIN= "0      0"
```

HSI: The set of HSI images to be generated. Leave blank if none desired.

HSI_THRESH_UPPER/LOWER: Parameters for generating HSI images.

HSI_RGB_MAX/MIN: Parameters for generating HSI images.

```
#######################
## OpenMIMS parameters
#######################
USE_SUM=true
MEDIANIZE=true
MEDIANIZATION_RADIUS=1.5
```

USE_SUM: Creates "SUM" images for the HSI image. Almost always set to true.

MEDIANIZE: Medianizes the HSI images.

MEDIANIZATION_RADIUS: The radius used for the medianization filter.

```
#######################
## Generate tracking
#######################
TRACKING=false
TRACKING_OVERWRITE=true
TRACKING_DOCUMENT_NAME=Enik_Exp3.html
FTP=true
```

TRACKING: Generates a web tracking document if set to true.

TRACKING_OVERWRITE: If true, the transfer script will regenerate the web tracking document every time it runs. Otherwise, only when new data is transferred.

TRACKING_DOCUMENT_NAME: Name of tracking document.

FTP: If true, the transfer script will upload a tracking document to Thedi. Otherwise, only a local copy will be generated.