

# ABOVE Server Documentation

## Version 1

Casey Daniel

March 30, 2015

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Directory Tree</b>	<b>3</b>
<b>3</b>	<b>TCP Sever</b>	<b>3</b>
<b>4</b>	<b>File Manager</b>	<b>3</b>
<b>5</b>	<b>Summary Plot Generation</b>	<b>5</b>
<b>6</b>	<b>SD reader</b>	<b>5</b>
<b>7</b>	<b>FTP Server</b>	<b>6</b>
<b>8</b>	<b>Shell Scripts</b>	<b>6</b>
<b>9</b>	<b>HTML Page</b>	<b>6</b>
<b>10</b>	<b>Installation</b>	<b>7</b>
10.1	Crontab . . . . .	7

## 1 Introduction

This documentation is meant to be a user guide for the ABOVE server files. These files include the TCP server, Initial Ingestion, plot generation, and shell scripts. All source files can be found on the server on `/data/vlf/src/`, however this may change on the new server once set up.

## 2 Directory Tree

Table 1 provides a summary of the relevant directories and information on where to find certain files.

root	<code>/data/vlf/</code>
source directory	<code>/data/vlf/src</code>
raw data	<code>/data/vlf/RawData</code>
malformed files	<code>/data/vlf/malformedFiles</code>
data tree	<code>Year/Month/Day/Site/Hour</code>
chunk files	<code>/data/vlf/chunks/datatree</code>
full data files	<code>/data/vlf/full_files</code>
logs	<code>/data/vlf/logs</code>
summary plots	<code>/data/vlf/summaryPlots/datatree</code>

Table 1: Directory List

## 3 TCP Sever

The Python TCP server it's self is written in Python 2.7, and is needed to run. In the source directory the server is managed by the executable of the form source `directory/tcp_server/tcp_server_PORTNO.py` where PORTNO is the desired port number. This executable looks for one of 3 commands, start, stop, or restart. (i.e. `./tcp_server_26000.py start`)

If another port is desired a few changes need to be made to the python file. The file needs to be renamed with the new port number, and near the top of the file, the variable `PORT_NUMBER` changed to match.

The server flow is described in figure 1, once I finish making it. This will detail the communications flow between the client on the FPGA and the server.

The client will send over chunk files over the internet connection, and will deposit them in the raw data directory.

## 4 File Manager

The File Manager, as found in ingestion software directory, and is responsible for file management. It runs the same way the tcp server does, and the same commands and format are used. This file looks through the rawData directory

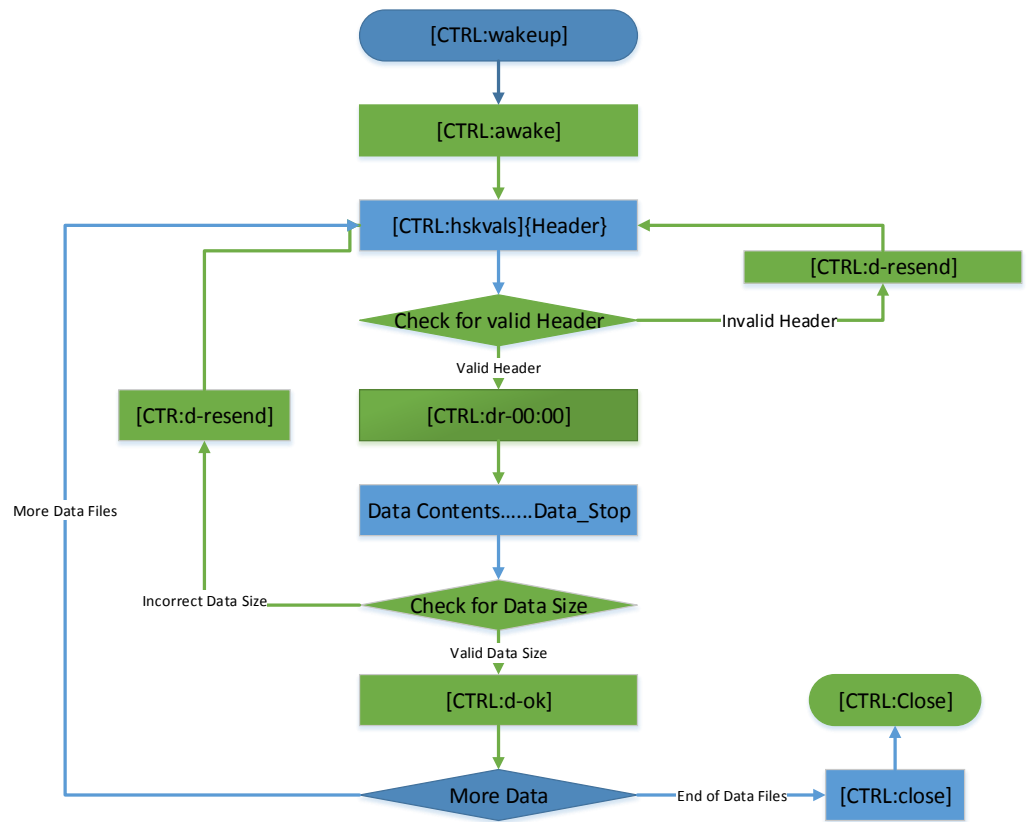


Figure 1: Visual of the connection process. Blue represents information sent by the client, and green represents information or decisions made by the server.

and collects the chunk files produced by the server and combines them into a full file, and deposit in the full file tree.

The File Manager is run on a damon script. When the `clean_up()` method is called, a search of the raw data directory is done. A check for a set of data files is done. If a set contains the expected number of files, as set in the header of the files, they are combined and a full file is saved in the full file directory, and the chunks are then moved to the chunks directory.

If a full set of files is not found, and the time stamp is over one hour old, then the chunk files that are found are then moved to the malformed files directory. File Manager is also responsible for sending RTEMP Information. These packets are formed and sent from this script, for information on format please contact Darren Chaddock.

After the files are formed, Eric Davis' script for the phase data is then started in a separate thread. Once the clean up is complete, a timer is set for 5 minutes for clean up to be called again.

## 5 Summary Plot Generation

Summary plot generation is handled by Matlab scripts. There are four scripts that handle the summary plots and the availability plots, two for each. One of these is called by a cronjob periodically (see the appendix) and will only search through the previous and current days directories, while the other meant to run on its own and search through the entire directory tree. The summary plot files are `plot_maker.m` to be called by the cronjob and `full_plot_maker.m` regenerates all missing summary plots. Similarly `availabilityPlotter.m` is called by the cronjob and `fullAvailabilityPlotter.m` is meant to regenerate all of the availability plots.

The crontab calls a shell script stored in `src/shellScripts/`. The shell scripts then call the Matlab scripts.

All of these can be run from the command line using the command 'Matlab -nodesktop -nodsplay -r run "/path/to/script/scriptname.m"'. It's recommended that when running the regeneration scripts that you move the other Matlab scripts out of the directory to ensure they are not run. Matlab is a resource hog, and leaving these scripts in place can cause a bottle neck on the CPU. There is also a shell script `regenerations.sh` in `src/shellScripts/` directory that will manage these scripts.

## 6 SD reader

The SD reader application is a c++ script that will read the contents of the SD card, and place them in the directory tree with a given root directory. To run this program in the command line `"/sdCard if=/path/to/InputFile.dat od=/root/of/directory/tree"`. There are also 2 optional commands. `-hf` will fix the header values. The early versions of the SD card had a few fields with

incorrect information. `-sd SD_ SIZE` will allow you to change the SD card size. The default is 128GB, however if the card is a different size, then a different one can be use. It's recommended to direct the output to a log file by attaching `"&log.txt"` to the end of the command.

Before running, the SD card should be copied into a separate file. The mounting point for the SD on a unix system can be found by looking through `/dev/` using `diskutil`. Once the mount point is found, the command `dd if=/path/of=out/file.dat` can be used to transfer the documents. The `/out/file.dat` can then be fed into the SD reader.

## 7 FTP Server

The FTP server is handled by VSFTPD, and can be started by the su by running the command `"vsftpd /etc/vsftpd/above_vsftpd.conf"`. This opens an anonymous FTP server with the the summary plot directory as root. At a later time this will also include CDF files. These allow the summary plot page to view the the plots on the web.

## 8 Shell Scripts

There are a few shell scripts. These primarily handle the generation of summary plots. There is also a reboot script that is called by the crontab on boot to restart the server and the file manger. There is also a `root_reboot` shell script that is called by the root user on boot to start the FTP server.

Note: on the current server the reboot scripts don't work and need to be run by hand when the server is rebooted, however with the new server hardware it should be functional.

## 9 HTML Page

In the summary plots page directory sit a simple HTML page to be able to view the summary plots from the browser. It consists of a simple form with the year, month, day, year, hour, site, and a few buttons to be able to flip through both the availability charts and the summary plots.

The year is not automatically generated, and every year the year needs to be added to the form.

Below the form is the java script that will react to the buttons and pull up the requested image from the FTP server, and display an error message if they are not available.

## 10 Installation

To install the server software on a new server, all the files should be placed in a source directory. All the directories in the directory table should be worked out on the new server, and updated in any python and Matlab script. These directories are commonly named and labeled near the top of the files.

Ensure both python and Matlab are installed. The python packages numpy, scipy, and matplotlib also need to be installed.

The TCP server should be started on the desired ports, and the sites can be remotely configured with the updated ports and IP addresses. Following which the file manager should be started.

Cronjob's calling the shell scripts should be set up. It's recommended that the availability plots and the summary plots generation script be timed so they are called at different times to prevent Matlab becoming a huge resource hog. The current set up can be seen in the crontab section, however this can be tailored as desired.

The HTML page also needs to be updated with the new IP address. It's listed as a constant in the java script. The site is currently hosted by Darren Chaddock, and the new HTML page should be sent to him to be updated.

### 10.1 Crontab

Here is the current crontab setup to be used as an example. For more options on crontab, please consult their documentation.

```
Main User Crontab
1,30,50 * * * * /data/vlf/src/shellScripts/matlabSummaryPlot.sh
&> /data/vlf/logs/plottingscript.log
@reboot /data/vlf/src/shellScripts/reboot.sh &>/dev/null
15,45 * * * * /data/vlf/src/shellScripts/matlabAvailabilityPlotter.sh
&> /data/vlf/logs/avail_log.log
Root Crontab
reboot /data/vlf/src/shellScripts/root_reboot.sh &> /dev/null
If you wish to see the output of these scripts, the file can be adjusted.
```