DHM Software Python

# Introduction

The “dhmsw” Python software is a software packaging for

# Table of Contents

[Introduction 1](#_Toc526236102)

[Table of Contents 2](#_Toc526236103)

[Prerequisites 3](#_Toc526236104)

[Quick Start 3](#_Toc526236105)

[Run “dhm\_streaming” 3](#_Toc526236106)

[Run “dhmsw” 3](#_Toc526236107)

[Start reconstruction display 3](#_Toc526236108)

[Start image display 3](#_Toc526236109)

[Commanding 4](#_Toc526236110)

[“dhm\_streaming” Software 4](#_Toc526236111)

[Install and Setup “dhm\_streaming” Software 4](#_Toc526236112)

[Running “dhm\_streaming” Software 4](#_Toc526236113)

[Run code: (log frames to disk) 5](#_Toc526236114)

[Run code WITHOUT logging to disk 6](#_Toc526236115)

[Datafiles 7](#_Toc526236116)

[Stop the “dhm\_streaming” Software 10](#_Toc526236117)

[“dhmsw” Software 10](#_Toc526236118)

[Install and Setup the “dhmsw” 10](#_Toc526236119)

[Run the “dhmsw” 10](#_Toc526236120)

[Commanding 10](#_Toc526236121)

[FRAMESOURCE Commands 11](#_Toc526236122)

[RECONSTRUCTION Commands 11](#_Toc526236123)

[HOLOGRAM Commands 11](#_Toc526236124)

# Prerequisites

You need the following packages in order to run the DHM Software:

* “dhm\_streaming” software. Provided by JPL
* “dhmsw” Python software: Provided by JPL
* “shampoo” Python module: Provided by JPL

# Quick Start

The following instruction assume that you have installed all required packages. In these steps you will need to have 5 terminal windows open.

## Run “dhm\_streaming”

Step1: Open terminal (referred to as Terminal A)

Step 2: Change directory to where “dhm\_streaming” software is located

Step 3: Run: (NOTE: See “dhm\_streaming” Software for details.)

./bin/CameraStreaming

## Run “dhmsw”

Step 4: Open terminal (referred to as Terminal B)

Step 5: Change directory where the “dhmsw/dhmsw” software is located

Step 6: Run: (NOTE: See Run the “dhmsw” section for details.)

python3 main.py

## Start reconstruction display

Step 7: Open terminal (referred to as Terminal C)

Step 8: Change directory to the “dhmsw/dhmsw”

Step 9: Run

python3 reconst\_display.py

## Start image display

Step 10: Open terminal (referred to as Terminal D)

Step 11: Change directory to the “dhmsw/dhmsw”

Step 12: Run

python3 rawframe\_display.py

## Commanding

Step 13: Open terminal (referred to as Terminal E)

Step 14: Change directory to the “dhmsw/dhmsw”

Step 15: Process frames from the camera, issue the following:

python3 ./dhm framesource mode=file

# “dhm\_streaming” Software

The “dhm\_streaming” software is the code that reads frames from the camera and logs them to disk and/or sends the frames to all connected clients.

## Install and Setup “dhm\_streaming” Software

Step 1: Download the “dhm\_streaming” code either from disk or from the following GIT repo (access to JPL Network required): <https://github.jpl.nasa.gov/DHM/dhm_streaming>

Step2: Modify the “Makefile” and change the following to match your system configuration: VIMBASDK\_DIR, INCLUDE\_DIRS, LIBS\_DIRS

Step 3: Run “make clean; make”. Verify that the file “bin/CameraStreamer” application exists.

## Running “dhm\_streaming” Software

PRESETS 1: The ‘dhm\_streaming’ software will create a ‘Holograms’ directory and a daily directory within it at the location the program is executed. Ensure that you have write permissions when running the software.

PRESTEPS 2: The empty file /tmp/.vmbcamlock must exist before running the software. To create one simply issue: touch /tmp/.vmbcamlock

USAGE:

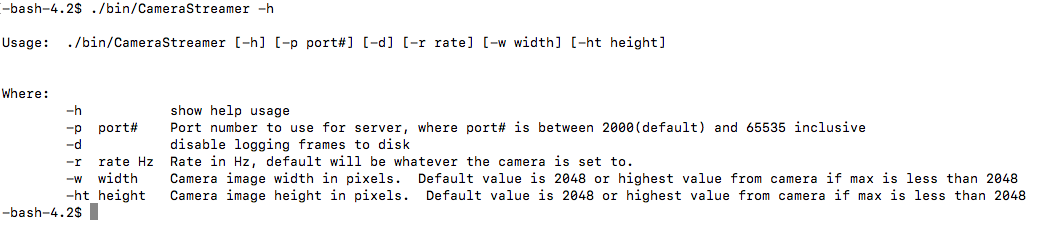


Figure : Screenshot of the ‘CameraStreamer’ application usage print.

### Run code: (log frames to disk)

./bin/CameraStreamer

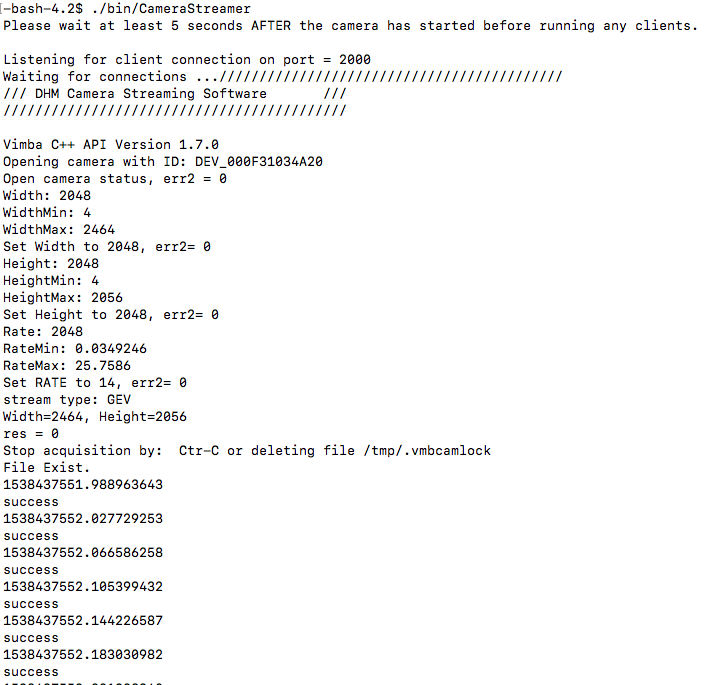


Figure : Screenshot of dhm\_streamer software running and logging to disk

### Run code WITHOUT logging to disk

./bin/CameraStreamer -d

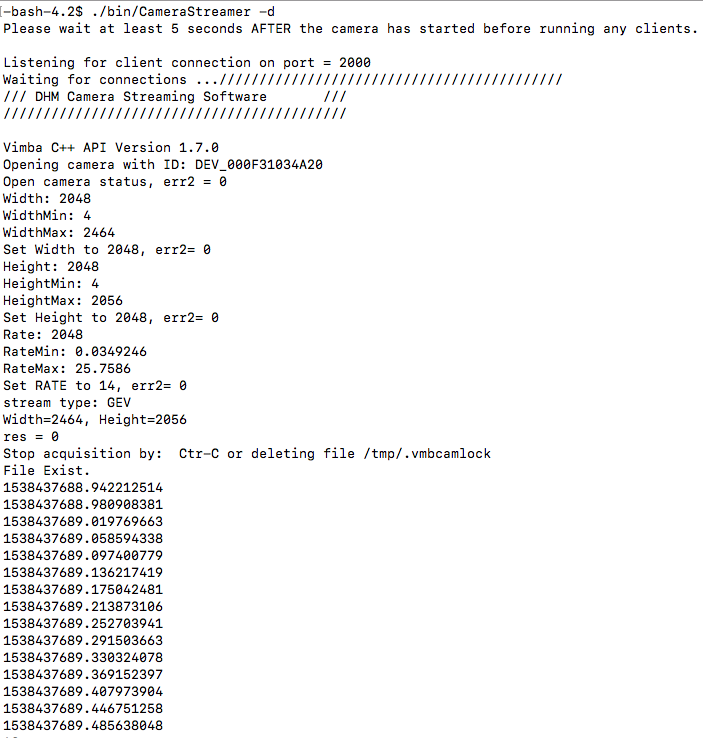


Figure : Screenshot of dhm\_streaming software running witout logging to disk.

## Datafiles

The frames are logged into the “Holograms” directory as TIF files. A ‘timestamp.txt’ file is created automatically (and appended to on additional runs). This file is used to make the data usable with the KOALA software.



Figure : Screenshot showing frame files created during two runs of the dhm\_streaming code.

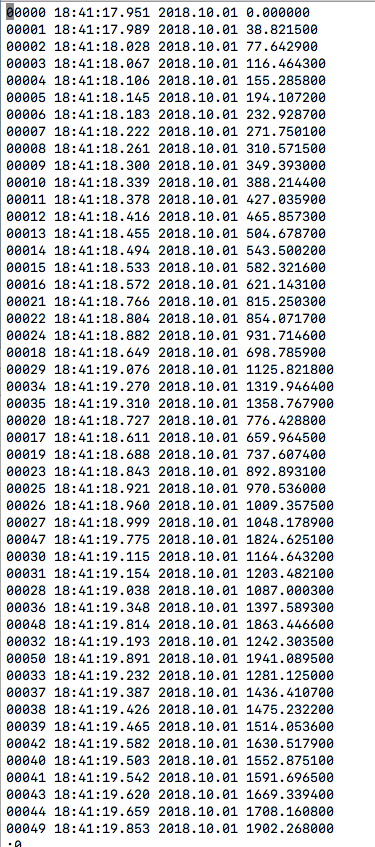


Figure : Screenshot of the timestamp.txt file that is created/appended with every run of the dhm\_streaming software.

## Stop the “dhm\_streaming” Software

To stop the software there are essentially three methods.

Method 1: Ctrl-C. This is a rough stop but it works

Method 2: Delete the /tmp/.vmbcamlock . The code checks the existance of this file at 1Hz. It it doesn’t exist then the code will exit and close all handles properly.

Method 3: Number of seconds defined in the #define variable

# “dhmsw” Software

This is the software that either obtains frames from file or from the “dhm\_streaming” software and performs the reconstructions

## Install and Setup the “dhmsw”

## Run the “dhmsw”

PRESTEP: To run the software we must FIRST ensure that the PYTHONPATH environment variable is pointing to the proper version of the SHAMPOO module. The value of the environment variable is defined in the ‘source\_profile’ file in the ‘dhmsw/dhmsw’ directory.

NOTE: Modify the ‘source\_profile’ directory and ensure the path to the location of the SHAMPOO version we will be using is set to the PYTHONPATH environment variable.

Run the following to source:

source source\_profile

To run the “dhmsw” execute the following:

python3 main.py

## Commanding

All commands must be issued using the following Python script which open a client connection to the “dhmsw” and send the command string.

python3 ./dhm

### FRAMESOURCE Commands

The framesource commands are commands that affect where the frames are obtained from

FRAMESOURCE MODE=[OFF|CAMERA|FILE] FILEPATH=<string>

Example:

python3 ./dhm framesource mode=camera

python3 ./dhm framesource filepath=../tests/simulated\_frames/\*.bmp

python3 ./dhm framesource mode=file

### RECONSTRUCTION Commands

The reconstruction commands affect the reconstruction parameters

RECONST PROPAGATION\_DISTANCE=<float values. Comma separate for more than one> COMP\_SPECTRAL\_PEAK=[ON|OFF] COMP\_DIGITAL\_PHASE\_MASK=[ON|OFF] PROCESSING\_MODE=[OFF|AMP|PHASE|ALL]

Example

python3 ./dhm reconst propagation\_distance=0.1,0.1,0.1

python3 ./dhm reconst processing\_mode=amp

python3 ./dhm reconst comp\_spectral\_peak=on

### HOLOGRAM Commands

The hologram commands affect parameters applied to the image which includes the wavelength.

HOLO WAVELENGTH=<floats. Separate more then one using commas> DX=<float> DY=<float> CROP\_FRACTION=<float> REBIN\_FACTOR=<float> BGD\_SUB=[ON|OFF] BGD\_FILE=<string>

Examples

Python3 ./dhm holo wavelength=439e-9

Python3 ./dhm holo wavelength=439e-9,539e-9,639e-9

Python3 ./dhm bgd\_sub=on bgd\_file=./mybackground.bgd

## Define Fourier Mask

At the moment this feature has not been implemented as a command. To define a fourier mask, you need to modify the DEFAULT file before running the ‘dhmsw’ software. Modify the centerx, centery, radius values in the ‘else’ condition, where each set corresponds to each wavelength specified. See Figure 6. If you are only specifying one wavelength the just modify the first one.

## DEFAULT File

The purpose of the DEFAULT file is to specify parameters that will be loaded on startup.

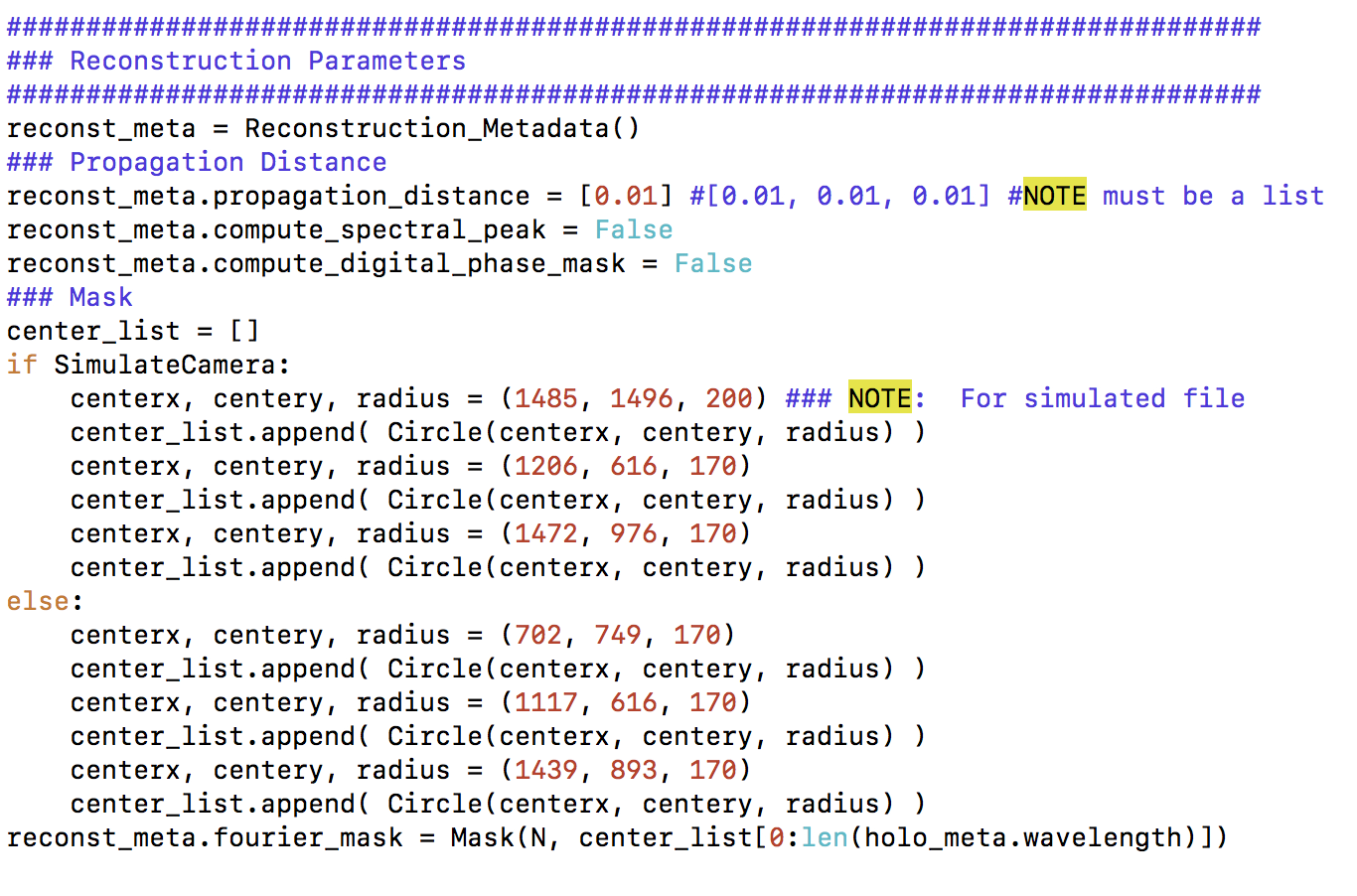


Figure : Screenshot of the DEFAULT file showing the Fourier mask definition.