**W.M. Keck Observatory**

# Exposure Time Calculator Documentation

**January 15th, 2020**

## Overview

The WMKO Exposure Time Calculator (ETC) provides a more cohesive and standardized exposure time calculator hub than the current solution. This is a web-based calculator that is updated as new instrument calculators are added, and any of these instrument calculators can be accessed from the same page without needing to run them from the command line or download IDL (as the previous solution required). The ETC can be downloaded and run on your personal computer, or the released version can be run in most browsers at this link: https://www2.keck.hawaii.edu/inst/PIILogin/etcgui/

The GitHub Repository is here: https://github.com/KeckObservatory/exposureTimeCalculator

## Requirements

This project is written in Python 3 and should be run with Python 3.5 or above. The current Python library requirements for this project are as follows:

- astropy
- bokeh [v1.0.4]
- flask
- getpass
- importlib
- matplotlib
- numpy
- psutil
- pysynphot
- scipy

## How to Download and Run the ETC

Navigate to the desired location on your computer and clone this repository by running the command:

```
git clone https://github.com/KeckObservatory/exposureTimeCalculator.git
```

The file structure will be as follows:

```
./exposureTimeCalculator/*.py
                        /static
                        /templates
                        /datafiles
```

Edit line 27 of etc.py to change the port to an unoccupied value (if the default of 50008 is unoccupied, there's no need to change this unless you want to). Then, edit line 52 of manager.py to point to your local install of Python 3. In the following command examples, I have aliased 'python3' to running the my local install of Python 3 with the full path. Afterwards, run the following command in the exposureTimeCalculator directory:

```
python3 manager.py etc start
```

You can also run the equivalent command:

```
python3 etc.py
```

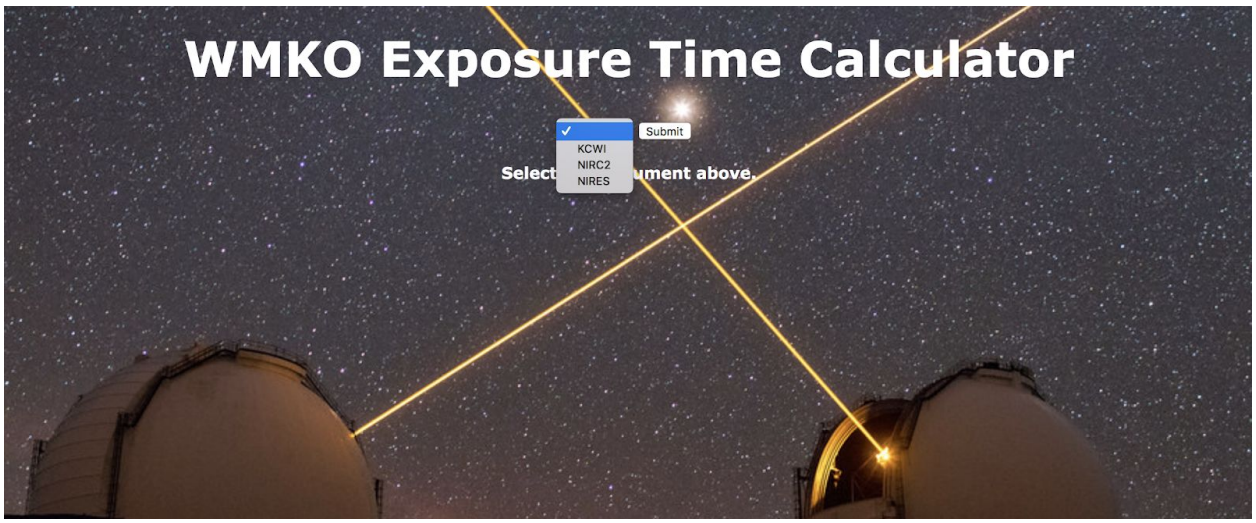
(More details on the manager.py script are provided in the 'Using the ETC' section.)

This will start a webpage on your localhost at the specified port number.

In your web browser, enter the following line in the URL bad (substitute new port number if you changed it earlier):

```
localhost:50008/etcgui/
```

## Using the ETC




Selecting the desired instrument will initialize the instrument-specific input window. See instrument pages for more details.

The homepage itself is generated in etc_gui.py.

For an in-depth of the Signal-to-Noise Ratio calculations, please see http://www.astrosurf.com/buil/us/spectro8/spaude02_us.htm.

### etc.py

This script serves to initialize the Python Flask server on the specified port. It calls etc_gui.py in orer to render the HTML template of a selected instrument, or a default blank template if nothing is selected.

### etc_gui.py

Initializes the instrument selection page, and loads the ETC of the selected instrument.

### manager.py

The purpose of this script is to facilitate the start/stop/restart of the ETC Python Flask server. It keeps track of the process ID assigned to the ETC so that it may be more easily stopped or restarted as desired.

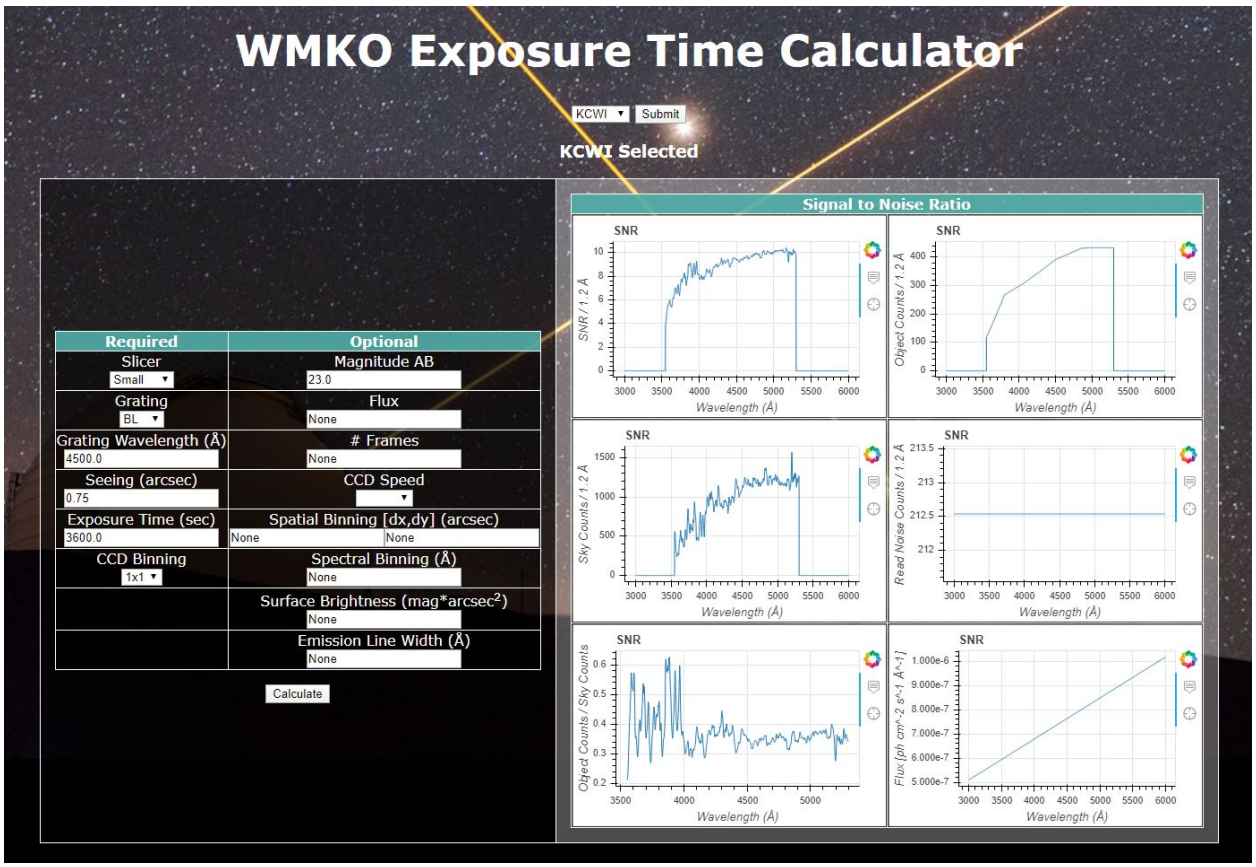
The available calls to this function are:

```
python3 manager.py etc start
```

```
python3 manager.py etc stop
```

```
python3 manager.py etc restart
```

## KCWI

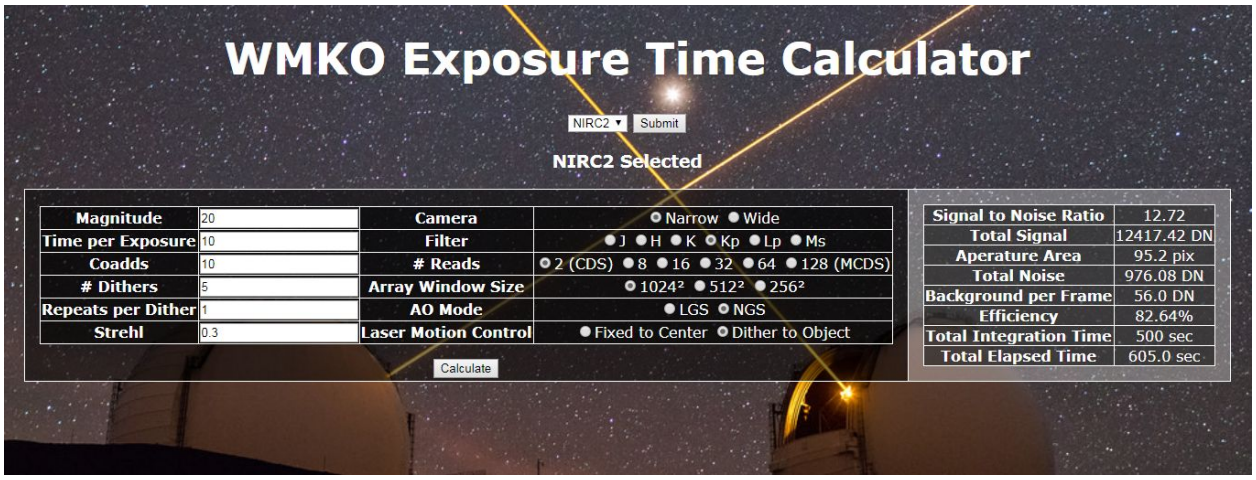



The KCWI ETC yields 6 different plots (x-axis = wavelength):

- SNR
- Object Counts
- Sky Counts
- Read Noise Counts
- Object Counts ÷ Sky Counts
- Flux

kcwi.py is the main script to take the inputs from the input window, plog those values into the KCWI ETC (etc_kcwi.py), and create the output plots. This ETC in particular is dependent on bokeh version 1.0.4, in order to take full advantage of the tooltip properties of the plots.

## NIRC2

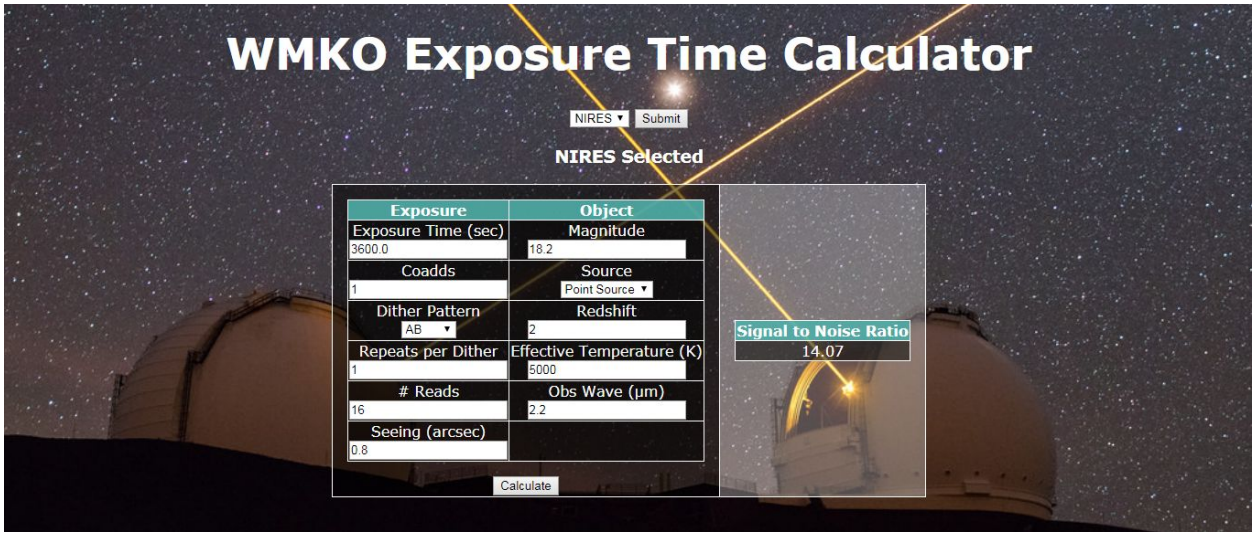



The NIRC2 ETC calculates a number of outputs, which are as follows:

- SNR
- Total Signal
- Aperature Area
- Total Noise
- Background per Frame
- Efficiency
- Total Integration Time
- Total Elapsed Time

nirc2.py is the main script to take the inputs from the input window, plog those values into the NIRC2 ETC (etc_nirc2.py), and initialize the output cells.

## NIRES




The NIRES ETC only outputs a SNR, and due to the use of pysynphot and a heavy interpolation routine during the calculation, takes a few seconds to appear. We are currently aiming to reduce this latency.

nires.py is the main script to take the inputs from the input window, plug those values into the NIRES ETC (etc_nires.py), and initialize the output cell. The NIRES ETC is dependent upon pysynphot, a Python library to simulate photometric data and spectra (see https://pysynphot.readthedocs.io/en/latest/ for more detail).