ExoVista Tutorial Task List

For this tutorial, you should have installed ExoVista and wget.
This tutorial is intended to be done in a command line environment, e.g. Terminal in Mac or Linux, or Command Prompt in Windows.
You should begin this tutorial in the main `ExoVista/` directory.

Task 1: Create a single planetary system scene using ExoVista with the `ExoVistaSystem.py` script.

1. Open the `solar_system.dat` input file in your preferred text editor. Note the three sections for star, planet, and disk properties.
2. This input file is designed to recreate our Solar system. We want to change it to create a distinctly different system, and to avoid filename conflicts.
    a. In the `Star` section, change the value for `ID` to `998`.
    b. Change the value for `TYC2` to `SUNv2`.
    c. In the `Planets` section (note that the planets are listed in order of orbital distance), change the eccentricity for Earth to `0.500`.
    d. Change the inclination of Mars to `181.85061` (making it retrograde).
    e. Change the radius of Venus to `11.209` (making it equal to Jupiter). Do **not** change its mass.
3. Use ExoVista to create a scene for this new solar system, using the script for single systems. (This will take several minutes.) Command:
   `python ExoVistaSystem.py solar_system.dat`
4. Look at the `ExoVista/output/` subdirectory. There should be a new FITS file there beginning with "998"
    a. If this failed due to problems with the input file, a working build of the new input file has been provided, named `solar_system_tutorial.dat`. You may rerun the above command with this file:
       `python ExoVistaSystem.py solar_system_tutorial.dat`
5. Visualize the output file with the provided "readfits" script. Command:
   `python readfits.py`
    a. Enter option `0`. (This will work with the new file you created, or with the provided example file if running ExoVista was unsuccessful.)
    b. In addition to the plots, note the statement about transits and eclipses on the command line.

Task 2: Create a universe of planetary systems using ExoVista with the `ExoVista.py` script.

1. Open the `ExoVista.py` script. We will make two changes to speed up the code execution.
    a. In the line that begins "`settings = Settings.Settings`…" change the `timemax` parameter to `1.0`. (This will shorten the orbital integration baseline.)

    b. In the same function, add the new parameter `diskoff=True`. (This will skip the calculation of the disk flux.)
2. Run ExoVista to generate a batch of planetary systems. (This should take between half a minute and a few minutes, depending on the random draw of planets.) Command: `python ExoVista.py`
3. Look at the `ExoVista/output/` subdirectory. There should be seven new FITS files labeled `0` through `7` (excluding `1`, which is automatically cut because the star is off the model grid).
4. You may visualize these files using readfits, the same as before. (Note that the disk will be absent.)

Task 3: Downloading existing data products from EMAC.
Note: the GUI on the webpage is scheduled for a major overhaul soon.

1. Go to https://tools.emac.gsfc.nasa.gov/exovista/
2. Recommended: read the instructions on the webpage.
3. Download a single FITS file.
    a. Select the first FITS file on the list, beginning with "`1000`"
    b. Click "Download Selected File" in the sidebar.
    c. Move the downloaded file to your `ExoVista/output/` directory.
4. Download a wget list to download multiple FITS files.
    a. Select the first five FITS files on the list, labeled `1000` through `1009`. Do **not** select an entire folder.
    b. Click "Download wget List" in the sidebar. This will download a text file called "`selected-files`"
    c. Move the downloaded file to your `ExoVista/output/` directory.
5. Download a batch of FITS files.
    a. Go back to your Terminal or Command Prompt.
    b. Navigate to `ExoVista/output/`
    c. Run wget to download the files. Command:
    `wget --input selected-files`
6. You may visualize these files using readits, the same as before. (However, the outputs will be slightly different because they were created with ExoVista 1.)

Wrap-up: other-postprocessing scripts

1. Navigate to the main `ExoVista/` directory.
2. Open `load_scene.py`
3. Note the inputs and outputs of the `load_scene()` function. This function can be used and/or modified as the interface between ExoVista data products and other codes.
4. Learn more about ExoVista formats and features in the User Guide.