ARISS Moderator Script Generator - Helper File

By: Ken McCaughey (N3FZX)
On: 2024-11-30
Ver 3.0.0

This file provides helpful information for users new to Python to help run the ARISS Moderator Script Generator. You do not need to know Python to run this tool!

The README file has information on how the tool works. This file has information on how to setup Python and run the tool. This is were known issues are documented and additional tips reside.

Contents

Python

Python comes with Linux and Raspberry Pi's. Macs often have Python, but it is not configured for user projects. Windows doesn't come with Python.

Linux and Raspberry Pi systems also come with Libre Office needed to open MS-Word `.docx` files. Thonny is in the Raspberry Pi software repository and most Linux repositories. These systems are generally easier to setup to use this tool. Libre Office is available for Windows and Macs in addition to Linux.

The two sections below assume Python is already installed on your system. If you are a beginner or Python is not installed, skip the the next section and consider installing Thonny.

Installing the `docxtpl` Python Library

This tool requires the `docxtlp` Python library that is not normally included with a Python installation. It must be added. How it gets added depended on your setup. The latest version of the library can be found at `https://pypi.org/project/docxtpl/`. In general it can be installed using the command line command: `pip install docxtpl`. If you are using a virtual environment (i.e. `venv`), execute the command in that folder with the virtual environment active.

Alternatively, use Thonny and its tool to install library packages. This is detailed below.

Running `ARISS_mod_script_gen.py`

To run the tool, open a terminal and at the command line enter:

```
python3 ARISS_mod_script_gen.py
```

It runs fast and is not interactive. A number of lines of messages will stream down the window. It will resemble the example below in the Thonny section. You can scroll up to see what you may has missed. If it ends with `Success!` then you are all set.

There should now be more or updated files in the `ARISS_mod_script_gen` folder. The files of interest are the script and outline files.

Thonny Python IDE

Thonny is a decent basic Python Integrated Development Environment (IDE). It is free and runs on Linux (and Raspberry Pi's), Macs, and Windows. It also bring along its own Python installation. Installing Thonny gets you a good tool and Python in one step. This is recommended for Python beginners. The software may be in your machine's software repository (or store). Or the files for your OS can be found at `https://thonny.org/`. The wiki for Thonny can be found at `https://github.com/thonny/thonny/wiki`.

Once installed, turn on the file viewer. In Thonny, from the main toolbar click on `View` then click to add a check mark for `Files`. It will add a sub-window to the left side. When you are running the script this should be set to your working directory with all the ARISS Moderator Script files.

Installing Thonny on a Mac

A good guide for installing Thonny on Mac is at the link below. It also has some instructions for adding libraries.

`https://www2.seas.gwu.edu/~cs4all/1012/editor-install/thonny-mac.html`

Thonny Virtual Environment Setup

Starting with Python version 3.11, a virtual environment is required. This is in essence a local container of the Python files for users to use. It isolates any Python files the operating system may be using to protect your OS. Thonny supports the virtual environment. This needs to be setup only once for Thonny. It can use used for all your Python projects.

Note that the virtual environment setup is not required under Windows.

Start by making a Python project folder, i.e. `Python_Projects`. Within the folder create a new empty folder called `venv`. This will be the location for the user virtual environment, which will be setup below.

In Thonny, from the main toolbar click on `Tools` then `Options`. It will open a window. Select the `Interpreter` tab.

Which kind of interpreter... should be `Local Python 3`. If not click on the drop down menu and select `Local Python 3`.

At the bottom right of the window find `New virtual environment`. Click on that, and select the `venv` folder created above.

Now set Thonny to use that virtual environment. `Python executable` has a drop down menu. Click on it and find the path that corresponds to the `venv` folder. Note that the path will end with something like `.../Python_projects/venv/bin/python3`. Click on `OK` to close the `Thonny Options` window.

How To Add Python Libraries in Thonny

In Thonny, from the main toolbar click on `Tools` then `Manage packages...`. A window will open up. All the installed packages are listed in a column on the left. These are all packages only installed in the virtual environment.

In the box at the top you can enter the name of the package (or library) needed. It will provide a list of matches under `Search results`. Click on the one you need. It will then give you a more detailed description with an option to Install. Click on `Install`. A small window will appear as it is installed. Once complete it will appear on the list column on the left. There is also an option to `Uninstall`. Note that if it was already installed, there may be an options to `Upgrade` if a newer version has been released. Click `Close` when done.

Installing `docxtlp` Library in Thonny

To install the `docxtpl` library needed by the ARISS script generator, in Thonny, from the main toolbar click on `Tools` then `Manage packages...`. In the search box, enter `docxtpl`. Click on `docxtpl` in the search results. Click on `Install`. Click `Close` when done.

Setup `ARISS_mod_script_gen.py` in Thonny

Download the Zipped package from GitHub at:

```
https://github.com/twk6809/ARISS_mod_script_gen
```

Unzip the GitHub file `ARISS_mod_script_gen.zip`. Find the the `ARISS_mod_script_get` folder and copy to the `Python_projects` folder.

Running `ARISS_mod_script_gen.py` in Thonny

It is possible to associate Python files (with the `.py` ext) with Thonny. The method varies with OS, so it is not included here. If you do this, a double click on any `.py` file can open it up in Thonny automatically.

Open Thonny and on the main toolbar click on `File`, then `Open` and work your way to the folder with the file `ARISS_mod_script_gen.py` and open the file. It will open the Python script in its own tab.

In the Thonny Files sub-window (left side) you should see all the ARISS Moderator Script files.

Any of the sub-windows in Thonny can be resized. Just grab the edges with the mouse and drag to suit.

Thonny can open and edit text files, such as the `ARISS_mod_script_form.txt` file. It can open the outline report file as well. Clicking on any of the `.docx` files should open them up in MS-Word or Libre-Writer.

To run the script, just click on the `Run` button (green circle with right arrow) on the toolbar. Or on the toolbar click on the `Run` menu, then click on `Run current script...`.

It runs fast and is not interactive. A number of lines of messages will stream down the `Shell` sub-window. It will resemble the example below. You can scroll up to see what you may has missed. If it ends with `Success!` then you are all set.

There should now be more or updated files in the `ARISS_mod_script_gen` folder. The files of interest are the script and outline files.

It is advised to open the script file and check it. Make sure all the variables have been filled in correctly. Please proof read prior to any distribution. Recommend generating a PDF to be included in any distribution.

Example output messages from a successful run of the script.

```
Python script: ARISS_mod_script_gen.py
           V.: 3.0.0
          By: Ken McCaughey, N3FZX
          On: 2024-11-30

This tool creates an ARISS Moderator Script.
It reads data from a form text file and a MS-Word docx template file.
Then generates a complete script and timeline report.

Required Input Files:
  Moderator form  --> ARISS_mod_script_form.txt
  Script template --> ARISS_mod_script_temp.docx

Blank moderator form text file generated...
Form text file found...
Form text file read...
Output filenames generated...
Timeline calculations complete...
Outline report file generated...
Variable dictionary updated...
Template file found...
New moderator script generated...

Output Files Created:
  Blank form file --> ARISS_mod_script_form_blank.txt
  Outline report  --> ARISS_mod_script_outline_Example_V1.txt
  Complete script --> ARISS_mod_script_Example_V1.docx

Success!

Python Script Done
```

Creating a Native Executable

A native binary may be included on the GitHub page for this project. If it exists, it will be in a separate folder that denotes the OS. If it exists, download it and place the binary file in the `ARISS_mod_script_gen` folder with all the other files. If it is not already available for you OS, you can create one with the steps below.

It is possible to make an executable for your operating system. You need to have Python installed. It can be done with Thonny. The advantage is that you can just run it without having to fuss this Python or Thonny. There is some limited portability to copy and run the executable on other machines. The disadvantage is that you need to create and test a new executable with every update that comes out.

Creating a native executable requires an extra file, `ARISS_logo_simple.ico`.

Install `pyinstaller`

To install the `pyinstaller` library needed by the ARISS script generator, in Thonny, from the main toolbar click on `Tools` then `Manage packages...`. In the search box, enter `pyinstaller`. Click on `pyinstaller` in the search results. Click on `Install`. Click `Close` when done.

Running `pyinstaller`

Below are the command line instructions for making a native binary executable for most operating systems. Below are the commands to be run at the command line. A bunch of messages will scroll by and, if successful, end with something similar to the following:

```
    ##### INFO: Appending PKG archive to custom ELF section in EXE
    ##### INFO: Building EXE from EXE-00.toc completed successfully.
```

In all cases additional two folders will be created, `build` and `dist`. The files in the `build` do not need to be saved. The file in the `dist` folder is the native binary executable. This file should be copied to the with all the other ARISS Moderator Script files. A shortcut can be made if desired, but setup to run in a terminal window.

To run the native executable, in a terminal window, enter `ARISS_mod_script_gen`.

Make a Linux (or Raspberry Pi) Executable

- In Thonny on the main toolbar click on `Tools`, then `Open System Shell...` This should open the terminal window with a command line in the folder
  with the ARISS Moderator Script files.

- On Linux/Rasp Pi command line (all on one line):

  ```
  pyinstaller --clean -w -F -i "ARISS_logo_simple.ico" ARISS_mod_script_gen.py
  ```

- If a .spec files exists,on the command line enter:

  ```
  pyinstaller ARISS_mod_script_gen.spec
  ```

Make a Windows Executable

- In Thonny on the main toolbar click on `Tools`, then `Open System Shell...` This should open the terminal window with a command line in the folder
  with the ARISS Moderator Script files.

- On Windows command line (all on one line):

  ```
  pyinstaller --clean -w -F -i "ARISS_logo_simple.ico" ARISS_mod_script_gen.py
  ```

- If a .spec files exists,on the command line enter:

  ```
  pyinstaller ARISS_mod_script_gen.spec
  ```

Make a Mac Executable

- In Thonny on the main toolbar click on `Tools`, then `Open System Shell...` This should open the terminal window with a command line in the folder
  with the ARISS Moderator Script files.

- On Mac command line (all on one line):

  ```
  pyinstaller --clean -w -F -i "ARISS_logo_simple.ico" ARISS_mod_script_gen.py
  ```

- If a .spec files exists,on the command line enter:

  ```
  pyinstaller ARISS_mod_script_gen.spec
  ```

Known Issues and Tips

If you get errors or have issues take a look below first.

Before starting a new script, clear out any old forms, working templates, and scripts.

The tool can generate an outline report with just the **ISS rise time** and the **conference start time**. The script will be very incomplete, but the outline file is usable to aid in early event planning.

If running in Thonny, make sure the `ARISS_mod_script_gen` tab is active before clicking on `Run`. If another tab, such as the form is open and the active tab, it will try to run what is in the form, which is not Python, and generate lots of errors. This is easy to do! Just make the tab with the script the active one an run it.

The tool can fail if the expected file names are not as expected. The input file names are fixed in the Python code. See the README file.

Misspelled variable name(s) in the form, or the template, will not get properly populated. Do not change the variable names. If new variables are needed, contact the author.

If the value of a variable in the form is not known, or not going to be used, do not change the value in the form. For example if the live stream operator will not be used, leave form default value as `{{livestream_name}}`.

If the form file becomes corrupted the tool will fail. Corruptions could be an inadvertently changed variable name, invalid date/time format, a carriage return (line break) in the middle of a variable. One sure way to recover is to delete the form file and an make a new one from the `ARISS_mod_script_form_blanl.txt file`. Whenever the tool is run a new blank form is created.

In the form file, do not use the `#` character in any variables. This is interpreted at a comment and ignores everything afterwards until a carriage return (line break).

Over time more templates may be added, or they maybe updated. You can create your own versions to suit your needs. Just be aware the tool is built around event blocks that take discrete amounts of time to complete. If you deviate from the formula of this design it can create more work and good results cannot be guaranteed.

Pay attention to the status of the ARISS videos in the form file. The use, or not, of these changes the timeline calculation. For the `ARISS_mod_script_temp_master_short.docx` template, make sure both ARISS videos are set to `No`. For `ARISS_mod_script_temp_master_long.docx`, make sure both ARISS videos are set to `Yes`.