ARISS Moderator Script Generator - Helper

By: Ken McCaughey (N3FZX)
On: 2026-02-04
Ver 4.0.0

This file tires to 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 to be able to use tool.

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.

Python Command Line

This tool requires Python version 3.x. For Linux and Mac `python3` is usually used at the command line. Depending on how the system is setup, `python` can be used. Be aware that `python` often refers to Python 2. For Windows, users can use `python` or `py` at the command line. This document references `python3`, but substitute what your system is setup for.

Python Virtual Environment

Some OS's require a users to run Python in a virtual environment. Windows and MacOS do not, Linux does. This is because some OS's use Python for various OS functions (true for Linux). To prevent the user from corrupting OS dependent Python components, a separate "sandbox" area is created. This is easy to setup once Python is installed.

For information on the Python virtual environment see:

https://docs.python.org/3/library/venv.html

https://realpython.com/python-virtual-environments-a-primer/

Setup Virtual Environment

To setup for Linux, at a command line terminal enter:

```
python3 -m venv /path/to/new/virtual/environment
```

To setup for Window, at a command line terminal (i.e. DOS window) enter:

```
python -m venv C:\path\to\new\virtual\environment
```

To setup for MacOS, at a command line terminal enter:

```
python3 -m venv /path/to/new/virtual/environment
```

Activate Virtual Environment

In the command line terminal, change folders into the `venv` folder created above.

For Linux and MacOS, in the command line terminal, enter:

```
$ source venv/bin/activate
(venv) $
```

For Windows, in the command line terminal, enter:

```
PS> venv\Scripts\activate
(venv) PS>
```

Note the new prompt. From here you can run the `pip` tool to install Python libraries. For example:

```
(venv) $ python -m pip install <package-name>
```

Run Python scripts from here are well.

Deactivate Virtual Environment

For Linux and MacOS, in the command line terminal, enter:

```
(venv) $ deactivate
$
```

For Windows, in the command line terminal, enter:

```
(venv) PS> deactivate
PS>
```

Suggested File Structure

It is suggested to create a folder called `Python_Projects`. Within this folder create the local Python virtual environment (`venv`).

Place the `ARISS_mod_script_gen-main` folder under `Python_Projects`. Within the `ARISS_mod_script_gen-main` folder add **copies** of the `Working` folder each ARISS event.

For each ARISS event, run the Python script within the folder setup for that event. The folder for each event will contain all the files needed and created for that event. See example file structure below.

```
Python_Projects
├── ARISS_mod_script_gen-main
│   ├── Examples
│   ├── Templates           <-- Make copies of templates as needed
│   ├── Working             <-- Make copies of this folder
│   ├── My_school_event_1   <-- Copy of "Working" folder
│   ├── My_school_event_2   <-- Another copy of Working folder
│
└── venv                    <-- Python virtual environment
```

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` or `python3 -m pip install docxtpl` or `py -m 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.

Run a Python Test Script

A short simple Python test script is provided for beginners. This script will be successful if the required libraries are present. To run the test script open a terminal (or command) window with a command line interface. Make sure you are the folder with the test script. Use a change directory command (i.e. `cd <folder>`) as needed. Check for the presence of the script using a file listing command (i.e. `ls` or `dir`). Run the script with the following command:

```
python3 ARISS_Python_Test.py
```

Output should reassemble...

```
Python script: ARISS Python Test.py
           V.: 1.1.0
           By: Ken McCaughey, N3FZX
           On: 2025-02-16

This is a simple test script for new Python users.
It is intended to make sure scripts can be executed.

Hello World!

The current date and time is 2025-02-16 11:18:55.113509

Success! Congratulations, you just ran a Python script.
```

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.

For Windows and the Mac it is best to install for current user only, not all users. This should not require admin privileges.

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-main.zip`. Find the the `ARISS_mod_script_get-main` 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 in the **README** file. 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.