ARISS Moderator Script Generator - README

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

Contents

- Goals

- How It Works

- Setup

- Instructions and Work Flow **<-- Lots of useful info here!**

- Suggestions for Best Success **<-- Lots of useful info here!**

- Known Issues and Troubleshooting Tips

- Notes About the Moderator Script Format

- Dictionary of Script Document Variables

Goals

This tool is intended to help ARISS moderators create a script for ISS telebridge contacts.

The goals of this tool are to:

1. Have good consistent formatting that is easy to read and follow.

2. Have all the checklists and important details included.

3. Automate filling out the template to ensure accuracy.

4. Produce an event timeline outline to support event planning.

5. Automate the calculation of the event timeline. No manual time math!

6. Allow for easy customization for contact needs.

7. Speed up the script generation time.

8. Provide predictability for mentors and moderators.

How It Works

Two files are needed to get started; a moderator script template in MS-Word `docx` format, and a completed text file form containing the values of needed variables. The Python script basically works like a mail merge to populate the template with the variable values (such as names, callsigns, etc.) set in the form file. It also takes care of the script timeline and does time math.

Two files are generated; the moderator script in MS-Word `docx` format, and an event outline text file. The outline is useful in finalizing the event plan. It maps out the start time for each event block, planned durations, and time to ISS rise of the given event block. The file is easy to e-mail.

The outline time calculations use the start of the conference and ISS rise times as anchors. The calculations assume typical time durations for ARISS related activities, such as audio checks, and ARISS program. The school/group program start time and duration are calculated as a remainder of time left after accounting for the ARISS portions of the event. The baseline durations are specified in the form file and can be changed if needed.

Setup

Before you begin, make sure things are set up correctly.

1. **DOWNLOAD** from **GitHub** the files as a Zip file

   - **GET** the latest version at: https://github.com/twk6809/ARISS_mod_script_gen

   - **CLICK** on the green `<> Code` button and choose `Download ZIP`.

2. **UNZIP** and and folder called `ARISS_mod_script_gen-main` will appear.

   The following files in a folder called `ARISS_mod_script_gen-main`:

   - `README.md` and/or `README.pdf` - this file
   - `HELPER.md` and/or `HELPER.pdf` - more help with Python
   - `ARISS_Python_Test.py` - test script
   - `Working` - folder with the main files
   - `Examples` - folder with sample output files
   - `Templates` - folder with `docx` script template files

3. **MOVE** the folder structure to where ever you want it.

4. **CHECK** that you have Python 3.x is installed.

   - **ENTER** `python3 --version` in a terminal window.
     (Note: Depending on your OS and how Python is configured, the command to use might be `python3` or `python` or `py`.)

   - **INSTALL** if needed. See https://www.python.org/downloads/
     (Also see the **HELPER** file.)

5. **CHECK** that you have the `docxtlp` library installed.

   - **ENTER** `pip show docxtpl` or `python3 -m pip show docxtpl` in a terminal window.
     (Need to be in your virtual environment if you have one.)

   - **INSTALL** using `pip install docxtpl` or `python3 -m pip install docxtpl`.
     (Also see https://pypi.org/project/docxtpl/ or the **HELPER** file.)

6. **RUN** the test script to confirm needed libraries are present.

   - **OPEN** a terminal window with a command line interface.

   - **RUN** test script: `python3 ARISS_Python_Test.py`
     (Note: Depending on your OS and how Python is configured, the command to use might be `python3` or `python` or `py`.)

   - **CHECK** 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.
     ```

Instructions and Work Flow

1. **MAKE** a working folder.

   - **COPY** the `Working` folder to a new folder named for the ARISS event or school name.

     ```
     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
     ```

   - **CHECK** for these two required files in the working folder:

     - `ARISS_mod_script_gen.py`

     - `ARISS_mod_script_form_blank.txt`

2. **MAKE** a working input form file.

   - **COPY** the file `ARISS_mod_script_form_blank.txt` and name it `ARISS_mod_script_form.txt`. This file must be in the working folder.

   - **NOTE** running the Python script generates a new clean form file named `ARISS_mod_script_form_blank.txt` to ensure there is a good blank form available.

3. **MAKE** a working script template `docx` file.

   - **COPY** the desired master `docx` template file to be used in the working folder created in step 1. See the `Templates` folder for some options. The `Template_Catalog` file will have descriptions.

   - **RENAME** the master template file to `ARISS_mod_script_temp.docx`. The working template file will be edited.

     - For example, copy and rename `ARISS_mod_script_temp_master.docx` to `ARISS_mod_script_temp.docx`.

   - **NOTE**, there is a useful checklist on the last page of the template file.

4. **COMPLETE** the text file form, `ARISS_mod_script_form.txt` .

   This should be done by the mentor and/or moderator. There are instructions within the file. Fill out as much as you can using a text file editor.

   - Do **NOT** leave blank variables.
     - Leave the default values to be filled in later.
     - Enter `None` or `N/A` is not used.

5. **EDIT** the working template file `ARISS_mod_script_temp.docx` , not the `master` .

   In general the provided templates should be sufficient to cover the needs for a contact without much editing. Edits should be based on the plans for the contact in terms of event flow and planned components, such as videos and school/group program. Some edits may be needed based on type of conference being used (i.e. Verizon vs. Zoom).

   - **PASTE** in the questions from ARISS Ops Uplink file.

   - **EDIT** the script text and add/remove cues and notes as needed.
     (See dictionary of variable below.)

   - **REMOVE** unused/optional blocks, such as the videos.

     - **DELETE** table row to remove an optional event block.
     - Do **NOT** add more event blocks or change the event block names or numbers.

   - **REMOVE** the red colored text script notes after reading/following them.

6. **RUN** the Python script using an IDE or at the command line.

   - **CHECK** input files. The two input files must be in the same folder as the Python script. The files must have the following names:

     - Working template: `ARISS_mod_script_temp.docx`
     - Working form: `ARISS_mod_script_form.txt`

   - **ENTER** `python3 ARISS_mod_script_gen.py` on the command line. See the `HELPER` file for information on running Python.

   - **CHECK** for messages to scroll across the screen. There is limited error checking. See **Known Issues and Troubleshooting Tips** below if problems are encountered. It will report success if there were no errors. See sample output below.

```
Python script: ARISS_mod_script_gen.py
         V.: 4.0.0
         By: Ken McCaughey, N3FZX
         On: 2026-02-04

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...
Time check...
 Conf start time: 02:00
 ISS rise time: 04:00
 Time check: Good
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_Example_V1_outline.txt
 Complete script --> ARISS_mod_script_Example_V1.docx

Success!

Python Script Done
```

- **NOTE** two new files will be generated. If they already existed, they will be overwritten. These are the final product files.

  - `ARISS_mod_script_<short_name>_V<ver#>.docx`
  - `ARISS_mod_script_<short_name>_V<ver#>_outline.txt`

7. **OPEN** the new moderator script file and review it for completeness and correctness.

   - **INSPECT** the script. Does the script look right?
     - Is the version number correct?
     - Does the outline look correct?
     - Are the moderator notes correct for the type of conference call?
     - Are all the variables (in bold text) filled in correctly?
     - Are there any unexpected default variables, i.e. `{{variable}}` ?

8. **UPDATE**. If the moderator script needs to be updated, there are two ways to handle this. The first method is the preferred path.

   - **EDIT** the working template file, and/or form file. Be sure to update the version number variable in the form file. **RERUN** the Python script. New `docx` and outline files will be created overwriting the old ones.

   - **EDIT** the moderator script directly and save. This is best path for minor edits IF no more edits are anticipated and the Python script will not be re-run.

9. **DISTRIBUTE** the script.

   - **MAKE** a PDF version of the script.

   - **SEND** out. OK to send the `docx` version, PDF recommend as well. PDF cannot be changed and its format/fonts are fixed.

Suggestions for Best Success

These suggestions for mentors and moderators are based on observations and lessons learned from past experience.

1. The script generation tool is intended for the use by ARISS mentors and moderators. Is is not intended for use by the school/groups themselves.

2. Getting Python setup can be a challenge for some. Please ask for help. Users do NOT need to know Python to make this tool work! Plan on taking some time to become familiar with how the tool works. It is recommended that users play with the tool well in advance of needing to use it for a contact. Once Python is setup, it is pretty easy to use and upgrade.

3. The resulting moderator script is best viewed as an **ARISS script** for managing the **ARISS portions** of the event. If the school/group has a program planned for Event #5, the details are best NOT included in the script. The script only needs to capture the start and end times of what the school/group wants to do. This greatly simplifies the script production and decouples it from any plans not directly relevant to the ARISS portion of the event.

4. If extra activities are being planned by the school/group for the event, they are best to be scheduled before the start of the conference, within the time constraints of Event block #5, the **School/Event Program**, and/or after the concluding ARISS remarks. This minimizes potential timing errors and reduces the risk of upsetting the look and feel of the final script.

5. To get the quickest turn around time, it is best to minimize changes to the templates. Adding significant amounts of new content is discouraged, especially if time is short. Picking the right template speeds up publication.

6. It is recommended to NOT include the conference credentials or the any live stream links in the script. It is best NOT to attempt to duplicate critical information from e-mails, as this is error prone and more work when late changes occur.

7. The script tool and templates are built around a fixed number of event blocks. These should not be renumbered nor more added. Unneeded blocks can be removed (i.e. for videos), but the remaining blocks should not be renumbered. This preserves the timing and behind the scenes time math. Event blocks should never exceed one page in length.

8. The time zone abbreviation and UTC time offset are for informational purposes only. These values are NOT error checked or used in any of the time math. The time math is based on the `ISS rise in event local time` and `Start of conference in event local time`. The UTC times are likewise informational. These should be check carefully.

9. Pay attention to the status of the ARISS videos in the form file. Selecting the ARISS videos, or not, in the form file only affects the timeline calculations (time math). It does not affect the content of the template. Either the correct template needs to be selected (i.e. long vs. short), or the universal template can be adjusted accordingly for the ARISS videos planned.

10. The templates were intentionally designed to be readable if printed on a black and white printer. Be cautious if adding colors or highlights as the readability can be compromised if a color printer is not used. What looks good on a computer screen may not turn out the same when printed.

Known Issues and Troubleshooting Tips

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

Always run the tool in a command line window so that you can see any error messages that are generated.

1. The correct **Python** command to be used at the command line depends on your system's configuration and the desired Python version. It could be `python` or `python3` or `py`. On many systems, `python` is aliased to `python3`, meaning both commands will run Python 3. However, it's not universally true, and some systems might still have `python` pointing to Python 2 or not aliased at all. To ensure you are using Python 3, it is best to use `python3`, except for Windows. On Windows, you can use `py` as a generic launcher, and specify the version using switches like `py -3` for Python 3. You can also use `python` or `python3` if they are correctly configured in your system's PATH environment variable. This document refers to `python3` for command line entries. Substitute the command your system is setup for.

2. The tool can fail if the input **file names** are not as expected. The input file names are fixed in the Python code. Check spelling for file names.

3. Most **errors** are likely to be made in the form file. If this occurs the tool will give you some clues, starting with the first line that has an issue. To start, check for blank variables. Check for proper date and time formatting. Check for erroneous text or comments missing leading "#". Check field name spelling, for missing colon, or missing space after colon. Use `ARISS_mod_script_form_blank.txt` as a reference if needed.

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

5. There is limited error checking for times. A **time error** message will occur if the conference start time is after ISS rise time. Times must be in the 24 hour format. A **time error** message will also occur if there is not enough time for all the events. This would be due to the conference time starting too late with respect to ISS rise time. The fix is to correct the times in the form file.

6. If a **variable** never seems to get updated in the script, a field name in the form file may have been corrupted. The tool has default variables for each field name in the form file. If a field name becomes misspelled in the form file, it gets ignored and the default value (usually in the form of `{{varialbe}}`) is used and placed in the script.

7. 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!

8. 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, or using a form file from an older version of the tool. 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.

9. 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).

10. Feel free to post `Issues` on **GitHub** or leave comments and feature suggestions in the `Discussion` area.

Notes About the Moderator Script Template Format

- The script template file must be in MS-Word `docx` format (not `doc`). These files can be edited with MS-Word or Libre Writer. If using Libre Writer, be sure to save in the `docx` format.

- The MS-Word `docx` script template file needs to have variables inside double braces, i.e. `{{variable}}` . The form text file maps all the variables to their values. Only use variables that are listed below.

- The moderator script is divided into event blocks. Do NOT reorder.

- Typical durations have been established for ARISS blocks. The durations are specified in the form and can be changed if necessary.

- The event blocks in the script are in a one column table. Each event block is in a row. The table is set up so that page breaks are not permitted within a row. This keeps each event block together on the same page, helping with readability and flow during the event. It is very helpful to turn on the "view table gridlines" feature to see the hidden boarders.

Dictionary of Script Document Variables

The form file is the vehicle to specify the variable values for the script. The form consists of a list of field names and variables pairs.

```
Field name: {{variable}}  =  field/variable pair
```

The variables take the form of double braces around a variable name (i.e. field name `ISS callsign to be used` variable is `{{ISS_callsign}}`). These names are fixed in the Python script and should not be changed.

Below is the list of field names and their corresponding variables that are supported by the Python script. The variables can be placed anywhere, and repeated, in the `docx` moderator script working template document. It is not necessary to use all of the available variables in a script. Note that most variables are specified in the form file, while others are only generated by the Python script. If additions are needed, the Python script would need to be updated.

The default form file only has the minimum necessary field/variable pairs. To use additional pairs, add then to the form file and template from the list below (not including the calculated timing variables).

The field/variable pairs marked with a "*" at the end are included by default in the form file.

To use a field/variable pair not included by default, follow this two step process:

1. Edit the form file and copy the desired pair from the list below.
2. Then use the variable in the script template file.

For example copy/paste **School teacher name:** `{{teacher_name}}` as a new line in the form file. Then assign a value for the `{{teacher_name}}` variable (i.e Ms. Brooks). In the template file place the variable `{{teacher_name}}` where you want it to appear. When the tool is run the teacher's name will be populated in the script.

Script version

Script version: `{{version}}` *

School/Group Information

School/group name: `{{group_name}}` *
School/group location: `{{group_city_state}}` *
Coordinator at event: `{{group_coordinator_name}}` *
School principal name: `{{principal_name}}`
School teacher name: `{{teacher_name}}`
School/group presenter name: `{{presenter_name}}` *
Event phone number: `{{event_phone}}`
Emergency back-up phone number: `{{event_backup_phone}}`

ISS Information

Astronaut name: `{{astronaut_name}}` *
Astronaut callsign: `{{astronaut_callsign}}` *
ISS callsign to be used: `{{ISS_callsign}}` *

ARISS Moderator Information

Moderator name: `{{moderator_name}}` *
Moderator callsign: `{{moderator_callsign}}` *
Moderator phone number: `{{moderator_phone}}`
Moderator will be On-site/Remote for the contact: `{{moderator_location}}` *

ARISS Mentor Information

Mentor name: `{{mentor_name}}` *
Mentor callsign: `{{mentor_callsign}}` *
Mentor phone number: `{{mentor_phone}}`
Mentor will be On-site/Remote for the contact: `{{mentor_location}}` *

ARISS Ground Station/Telebridge Information

Ground station callsign: `{{gnd_sta_callsign}}` *
Ground station location: `{{gnd_sta_location}}` *
Ground station telephone number: `{{gnd_sta_phone}}`
Ground station operator name: `{{gnd_sta_op_name}}` *
Ground station operator callsign: `{{gnd_sta_op_callsign}}` *
Ground station operator phone number: `{{gnd_sta_phone}}`
Ground station support name(s) and callsign(s): `{{support_names_callsigns}}`
Ground station audio interface): `{{gnd_sta_audio}}` *
Ground station video interface: `{{gnd_sta_video}}` *

ARISS Optional Videos

Contact from student perspective (Yes/No): `{{student_video}}` *
Contact from the ISS perspective (Yes/No): `{{ISS_video}}` *

Live Streaming

Live steam planned (Yes/No): `{{livestream_plan}}` *
Live stream operator name: `{{livestream_name}}` *
Live stream operator callsign: `{{livestream_callsign}}`
Live stream operator phone number: `{{livestream_phone}}`

Fixed Timing

Contact date: `{{contact_date}}` *
Conference start, event local time (HH:mm): `{{conf_etz}}` *
ISS rise predict, event local time (HH:mm): `{{AOS_etz}}` *
Conference start, UTC time (HH:mm): `{{conf_utc}}` *
ISS rise predict, UTC time (HH:mm): `{{AOS_UTC}}` *
Event time zone abbreviation (CAPS): `{{etz}}` *
Event time zone shift to UTC (+/-HH): `{{TZ_shift}}` *

Calculated Timing

The variables listed below are calculated by the Python script. These are predominately used to time tag the event blocks and generate the outline report file. These cannot be used in the form file.

Event	Start Time	Duration	ISS Rise
Conference start time	{{T01}}	{{D01}}	{{R01}}
Ground station checklist	{{T02}}	{{D02}}	{{R02}}
Contact preparation checklist	{{T03}}	{{D03}}	{{R03}}
Practice run through w/students	{{T04}}	{{D04}}	{{R04}}
School/group program/slack time	{{T05}}	{{D05}}	{{R05}}
Start ARISS program	{{T06}}	{{D06}}	{{R06}}
ARISS introduction	{{T07}}	{{D07}}	{{R07}}
Video from student perspective	{{T08}}	{{D08}}	{{R08}}
Video from ISS perspective	{{T09}}	{{D09}}	{{R09}}
Introduce the ground station	{{T10}}	{{D10}}	{{R10}}
Handover to ground station	{{T11}}	{{D11}}	{{R11}}
ISS rise and AOS	{{T12}}	{{D12}}	{{R12}}
ISS Contact!	{{AOS_etz}}	-	-
ISS set and LOS	{{T13}}	-	-
Closing & end of ARISS program	{{T13}}	{{D13}}	-
Contact preparation duration	-	{{D14}}	-
Shool/group program duration	-	{{D15}}	-
ARISS program duration	-	{{D16}}	-
Total event duration	-	{{D17}}	-