USER GUIDE

GUI-doc

ATLAS Schema INI Editor

Prepare the three ATLAS INI files, inspect the available inputs and run ATLAS in a selected Conda environment. This guide describes the supplied English-language GUI.

Quick start

1. **Start the GUI**
   Extract the ZIP into the project root, then run `python src/atlas_actris/GUI/atlas_schema_gui.py`.
2. **Choose call_atlas.ini**
   At startup, click **OK** and select the file. Its values load in the first tab; its path is copied to **Run**.
3. **Prepare the other files**
   Open **config_file.ini** and **settings_file.ini** in their own tabs. Edit the inputs, use **Check ini-file entries**, then save all changes.
4. **Run ATLAS**
   In **Run**, check the INI path and choose the Conda environment containing ATLAS. Click **Start** and follow the output below.

Before starting
Opening config/settings files in the editor does not rewrite the file references inside call_atlas.ini. Check those references when switching to another configuration.

Find your topic

Topic	Page
Setup and project files	2
Editor at a glance	3
Inputs, filters and validation	4
Paths and saving	5
Run ATLAS with Conda	6
Troubleshooting	7
Reference and verification	8

USER GUIDE / 01

Setup and project files

The ZIP starts at **src/**. Extract it into the folder that already contains your ATLAS **src** directory. The outer project folder can have any name.

Purpose	Project-relative location
GUI and its helpers	`src/atlas_actris/GUI/`
Main GUI file	`src/atlas_actris/GUI/atlas_schema_gui.py`
This guide	`src/atlas_actris/GUI/GUI-doc.pdf`

Launch from the project root

```
python src/atlas_actris/GUI/atlas_schema_gui.py
```

Alternatively, open **atlas_schema_gui.py** in Spyder and run it with F5. Source files are located relative to the GUI file, independently of the current working directory.

Existing ATLAS sources required

INI file	Keys, types and defaults	Help descriptions
call_atlas.ini	src/atlas_actris/utils/ **parse_init_file.py**	src/atlas_actris/templates/ **init_template_flavor.py**
config_file.ini	src/atlas_actris/utils/ **parse_config_file.py**	src/atlas_actris/templates/ **config_template_flavor.py**
settings_file.ini	src/atlas_actris/utils/ **parse_settings_file.py**	src/atlas_actris/templates/ **settings_template_flavor.py**

The six source files are part of your ATLAS installation; replacement copies are not included in the GUI archive. Restart the GUI after changing their declarations.

Dependencies
The editor uses Python and Tkinter only. Running ATLAS additionally requires Conda and a working ATLAS installation in the chosen environment. A system PDF viewer opens this guide via **GUI-doc**.

USER GUIDE / 02

Editor at a glance

Each INI file has a main tab. Its sections are displayed as sub-tabs, generated from the Python declarations. The **Run** tab is used for execution.

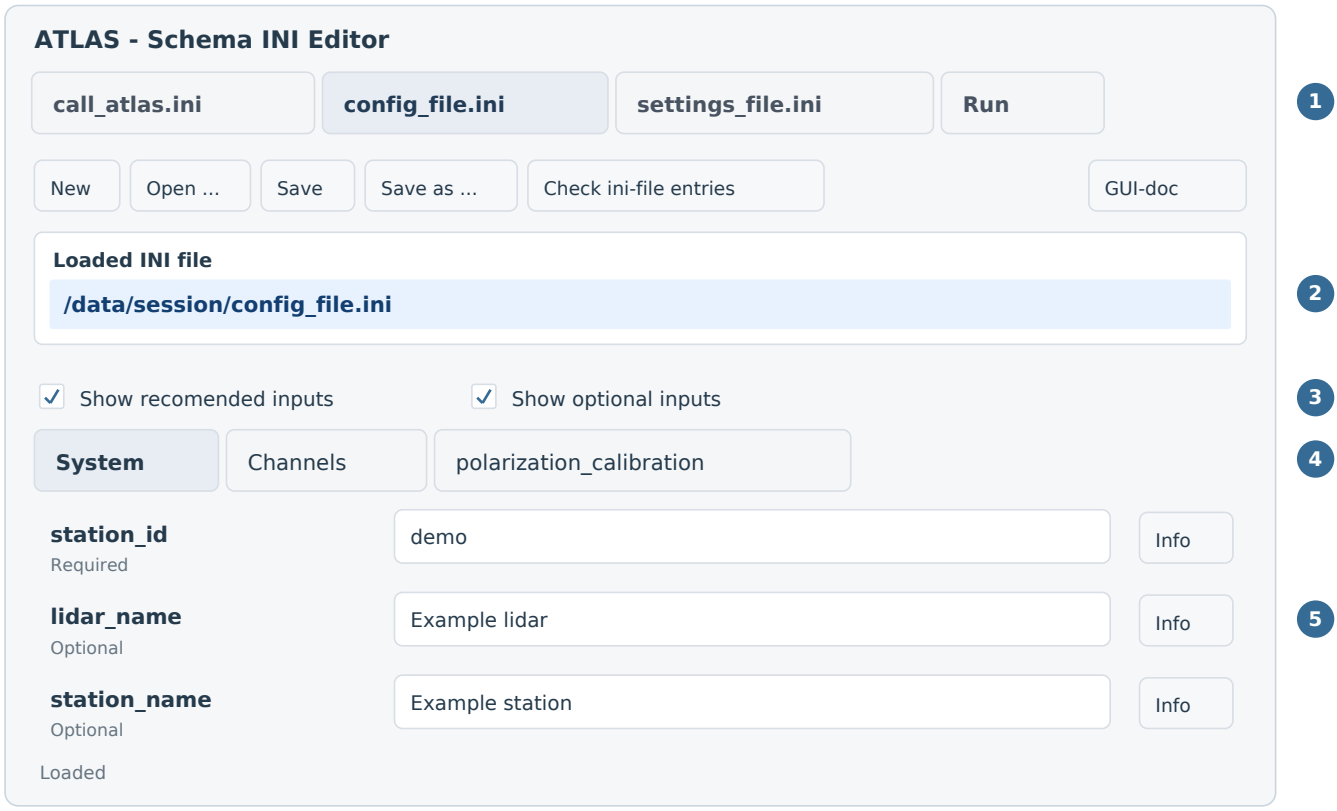


Figure 1. Schematic editor view with example values. Layout details vary by operating system; this is not a screenshot. Numbered markers are explanatory only.

No.	Control and purpose
1	**Main tabs.** Switch between call_atlas.ini, config_file.ini, settings_file.ini and Run. Main labels and the active section label use stronger text.
2	**Loaded INI file.** The highlighted path identifies the file being edited. It is read-only, but can be selected and copied. Open and Save as change this path.
3	**Visibility filters.** Show recommended or optional inputs. Their state stays synchronized across all three INI tabs.
4	**Section sub-tabs.** The active section has a subtle blue-gray background. In config_file.ini, gluing, water_vapour and temperature are hidden.
5	**Info.** Read the description, category, type, defaults and available constraints for that input. Path inputs may also have a browse button.

USER GUIDE / 03

Inputs, filters and validation

Use the descriptions beside individual inputs as the main parameter reference. The GUI shows the declarations supplied by the ATLAS source files; it does not infer additional scientific requirements.

Category	How it is displayed
Required	Always visible. Required values are checked by Check ini-file entries.
Recommended	Controlled by the checkbox labelled **Show recomended inputs**.
Optional	Controlled by **Show optional inputs**. Fields without an explicit category are treated as optional.

Entering values

Blank fields: leave them empty to retain ATLAS automatic/default behavior. A displayed default is guidance; the GUI does not fill it in as an explicit value.

Boolean and choice fields: use the suggested drop-down values. Boolean spelling is `True` or `False`, including capitalization.

Lists: the updated parsers accept commas with or without spaces, or semicolons. In config_file.ini, _, None and empty list positions mean missing values. Required lists must have every position filled; errors identify the missing positions.

Info buttons: open help from the flavor files, including available allowed values, minimum/maximum, list size, examples and version notes.

What the check button does

Check ini-file entries checks declared types, numeric limits, allowed choices, list sizes and required values. It also reports unrecognized entries retained from a loaded file.

It does not run ATLAS, check file existence or perform the complete runtime checks between fields and channels. A passed GUI check is not a guarantee that ATLAS will complete successfully.

Visibility is not deletion
Filtered inputs and the three hidden config sections retain their values and are included when saving. If a tab becomes empty, enable **Show optional inputs**: all call_atlas.ini inputs and uncategorized settings inputs are optional in the supplied schemas.

USER GUIDE / 04

Paths and saving

There are two distinct roles: a loaded INI file in an editor tab, and the execution target shown in **Run**. A manual Run override does not reload or redirect the editor.

Action	Effect on the Run path
Open call_atlas.ini	Copies the newly loaded file path into Run, including when you explicitly open the same file again.
First Save or Save as	Copies the saved call_atlas.ini path into Run. Save as also updates the highlighted editor path.
Edit inputs or normal Save	Keeps a manually entered Run path. Saving still writes the editor's own loaded file.
New in call_atlas.ini	Creates an unsaved model and clears the Run path. A first save provides the new file path.
Type or Browse in Run	Changes the execution target only. The editor remains on its previously loaded file.
Open config/settings	Does not change Run. The file browser starts in the loaded call_atlas.ini folder; if only Run has a selection, it uses that folder.

Keep file references consistent

In call_atlas.ini, verify inputs such as `atlas_configuration_file` and `atlas_settings_file` when changing configuration files. Opening another config/settings file in the GUI does not update these references automatically.

Saving and preservation

Save and **Save as** allow incomplete drafts. Validation issues are reported after saving. Loaded files retain comments, ordering, capitalization, formatting and line endings; unknown entries are preserved. Newly entered fields missing from a file are inserted into their sections.

Use UTF-8 and separate comment lines beginning with # or ;. Inline comments are treated as part of the value, matching the supplied parsers. Duplicate sections/keys and multiline recognized values are rejected.

External edits
If a loaded file has changed outside the GUI, Save refuses to overwrite it. Reload it to use the external version, or use Save as to preserve the GUI edits separately.

USER GUIDE / 05

Run ATLAS with Conda

The **Start** button reads the current Run path at the moment you click it. The selected Conda environment must already contain ATLAS and its dependencies.

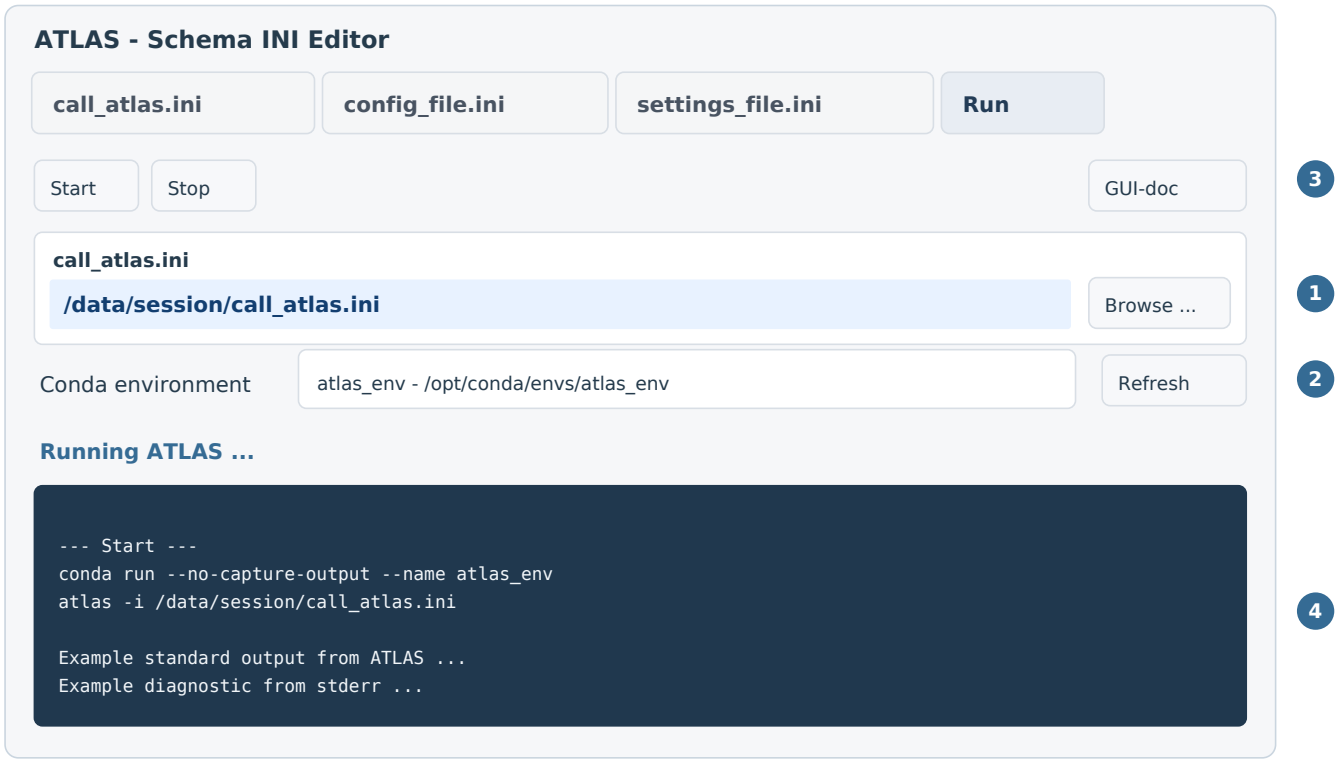


Figure 2. Schematic Run tab. Paths and log lines are illustrative; this is not output from an actual ATLAS run.

1. **Check the execution path**
   Use the editable **call_atlas.ini** field or **Browse ...**. Relative paths entered here are resolved against the GUI folder. Empty or missing files are rejected.
2. **Choose the environment**
   Select an entry from **Conda environment**, or type a name/full environment path. A detected active environment is suggested. Use **Refresh** to reload the list.
3. **Save, then start**
   Save pending edits in all INI tabs. Start invokes `atlas -i` with the current absolute INI path, through `conda run --no-capture-output`.
4. **Follow output or stop**
   Standard output, errors and the final exit code appear below. **Stop** terminates the run. A changed Run path is used on the next start, not by an already running process.

Conda not found?
The **Locate Conda ...** button appears if automatic discovery or environment listing fails. Select **Scripts/conda.exe** on Windows or **bin/conda** on Linux/macOS.

USER GUIDE / 06

Troubleshooting

Use the status line and Run output to identify whether a problem comes from file loading, input checks, Conda or ATLAS itself.

What you see	What to check
Conda not found	Use **Locate Conda ...**, or launch the GUI from Anaconda Prompt. If multiple installations exist, use the intended executable.
Environment list unavailable	Read the logged error. You can enter the environment name manually; use Refresh after correcting the Conda setup.
atlas command unavailable	Choose the environment in which ATLAS is installed. The GUI does not install ATLAS or its dependencies.
ATLAS uses a different INI	Check the current Run field and the command printed at Start. A manual Run override is independent of the editor's loaded file.
Empty or missing sections	Re-enable the visibility filters. gluing, water_vapour and temperature are intentionally hidden in config_file.ini.
Cannot save over a file	Check whether it was modified outside the GUI. Reload it, or Save as to a different location. Also check write permissions.
Documentation not found	Keep GUI-doc.pdf beside atlas_schema_gui.py. A PDF viewer/file association is needed; Linux uses xdg-open.
Missing schema source	Check all six Python files listed on page 2 under src/atlas_actris/utils and src/atlas_actris/templates.

Execution details to keep in mind

The execution working directory is the selected INI file's parent folder. The GUI captures output but does not provide interactive console input. Python output buffering is disabled; native programs can still buffer their own output.

On Windows, the GUI rejects embedded quotes and `&` `|` `<` `>` `^` `%` `!` in launch-related paths or environment names because the Conda wrapper uses command-shell syntax. Spaces in paths are supported.

Stop uses taskkill with its process-tree option on Windows. On Linux/macOS it sends SIGTERM to the process group, then SIGKILL after two seconds. An interrupted run may leave partial outputs. Deliberately detached processes are not managed.

USER GUIDE / 07

Reference and verification

These counts describe the supplied source declarations, not a fixed GUI layout. New declarations can change the available inputs after a restart.

INI file	Sections	Inputs	Required	Recommended	Optional
call_atlas.ini	7	60	0	0	60
config_file.ini	6*	47	5	22	20
settings_file.ini	8	127	0	0	127

* Three config sections are hidden from the interface, while their data remains loaded. Total: 234 inputs. Settings fields without a category are treated as optional.

Schema-reading limits

Config sections use parser groups and SCHEMA order when flavor files have no layout. Older layouts remain supported. The GUI reads declarations and list syntax without executing ATLAS code. Unsupported structural changes produce an error.

The allowed choices for `view_mean_signal_stages`, `view_signal_stages` and `export_stages` depend on `collect_stages()` from utils/cookbook.py. This file was not among the supplied six sources. Those fields accept text and identify the unresolved list in Info.

Compatibility and checks

The code uses Python 3.9-3.13 syntax and the standard library, including Tkinter. Verification used Python 3.12 on Linux. All 40 automated tests passed, covering updated schemas, list formats, required positions, preservation, path synchronization and the runner.

Runner tests simulate Conda with real local subprocesses, covering streaming, UTF-8, errors and child termination. Windows commands are tested with mocks. A full ATLAS/Conda run and native Windows GUI operation were not tested.

About the illustrations
Figures 1 and 2 are vector schematics based on the GUI code. They show example data and explain control placement; native dialogs, fonts and visual details can differ on your computer.

Sources and further reading

GUI behavior: the bundled **atlas_schema_gui.py**, **gui_common.py**, **editor_model.py**, **schema_loader.py** and **atlas_runner.py**. Parameter descriptions: the six ATLAS files listed on page 2.

Conda command reference:
docs.conda.io/projects/conda/en/stable/commands/run.html
docs.conda.io/projects/conda/en/stable/commands/env/list.html

The Run view retains the latest one million output characters. ANSI styling is removed; carriage-return progress messages are shown as separate lines.