FAQ ANL 2025

This FAQ will updated continuously for ANL 2025!

Where can I find further help with coding?

You can look at tutorials on the ANL website: https://yasserfarouk.github.io/anl . Much more extensive documentation of the platform Negmas in general can be found here: https://negmas.readthedocs.io/. Apart from that, you can ask one of the organizers (tamara.florijn@cwi.nl).

Can we submit an agent and see its performance and later on, improve it and submit a new one?

Yes! You can find the online tournament https://scml.cs.brown.edu/anl. You can register, and even submit just the template agent, then you can see how the online tournament works. You can resubmit as many times as you wish.

How can I access a data file in my package?

When your agent is submitted, it is run in an environment different from that in which the tournament will be run. This means that you **cannot** use hardcoded paths in your agent. Moreover, you (and we) do not know in advance what will be the current directory when the tournament is run. For this reason, it is **required** that if you access any files in your agent, you should use a path relative to the file in which the code accessing these files is located. Please note that accessing ANY FILES outside the directory of your agent is **prohibited** and **will lead to immediate disqualification** for obvious security reasons. There are no second chances in this one.

Let's assume that your file structure is something like that:

```
base
├── sub
│   ├── myagent.py
│   └── otherfiles.py
├── data
│   └── myfile.csv
└── tests
```

Now you want to access the file *myfile.csv* when you are inside *myagent.py*. To do so you can use the following code::

```
import pathlib
```

```
path_2_myfile = pathlib.Path(__file__).parent.parent / "data" / "myfile.csv"
```

What file structure should I use when running the tutorials and the template agent?

Make sure your file structure looks something like this:

```
ANL 2025submission/
├── Code_for_tutorials2025/
│   ├── Tutorial_running_a_negotiation.py
│   └── ...
├── my_agent/
│   ├── helpers
│   ├── report
│   └── myagent.py
├── Official_test_scenarios/
│   ├── dinners
│   └── ...
└── venv/
    └── lib/
        ├── ...
        └── ANL2025
```

Can my agent pass data to my other agents between negotiations?

NO Passing data to your agents between negotiations will lead to disqualification.

Can my agent read data from the HDD outside my agent's folder?

NO Your agent can only read files that you submitted to us in your zip file.
It cannot modify these files in anyway during the competition.
It cannot read from anywhere else in secondary storage. Trying to do
so will lead to disqualification.

Can my agent write data to the HDD during the negotiation?

NO The agent is not allowed to write anything to the hard disk during the
competition.

Can I print to the screen to debug my agent?

PLEASE DO NOT

Printing to the screen in your agent will prevent us from monitoring the progress of tournament runs and will slow down the process. Moreover, it is not useful anyway because the tournaments are run in

parallel.

If you really need to print something (e.g. for debugging purposes), please remove all print statements before submission. We will never touch your code after submission so we cannot remove them.

Can I write arbitrary code in my module besides the negotiator class definition?

When python imports your module, it runs everything in it so the top level code should be only one of these:

- Class definitions
- Function definitions
- Variable definitions
- Arbitrary code that runs in few milliseconds and prints nothing

Any other code *must* be protected inside::

```
if __name__ == "__main__"
```

For example, if you want to run a simulation to test your agent. *DO NOT USE SOMETHING LIKE THIS*::

```
anl2024_tournament(....)
```

But something like this:

```
def main():
    anl2024_tournament(....)

if __name__ == "__main__":
    main()
```

This way, importing your module will not run the world simulation.

I ran a simulation using "anl tournament2025" command. Where are my log files?

If you did not pass "--no-log", you will find the log files at *~/negmas/anl2024/[date-time-uuid]*

I implement my agent using multiple files. How should I import them?

Assume that you have the following file structure

```
base
├── subfolder
│   └── component2.py
├── component1.py
└── agent.py
```

In your `agent.py` file, you want to import your other files::

```
import component1
import subfolder.component2
```

This will **not** work because in the actual competition `component1.py` and `component2.py` will not be in python path.

There are two ways to solve it:

The clean way is to use relative imports. You will need to turn your agent int a package by adding empty `__init__.py` files to every folder you want to import from::

```
base
├── sub
│   ├── __init__.py
│   └── component2.py
├── __init__.py
├── component1.py
└── agent.py
```

You can now change your import to::

```
import .component1
import .subfolder.component2
```

Notice the extra dot (`.`)

Another way that does not require any modification of your file structure is to add the following lines
before your imports::

```
import os, sys
sys.path.append(os.path.dirname(__file__))
```

Note that the later method has the disadvantage of putting your components at the **end** of the path which
means that if you have any classes, functions, etc with a name that is defined in *any* module that appears
earlier in the path, yours will be hidden.

I am using sklearn. How can I save my model during training and load it during the competition?

Check this link for details of how to do that.

It is important to note that when an estimator is unpickled with a scikit-learn version that is inconsistent with the version the estimator was pickled with, aInconsistentVersionWarning is raised. This warning can be caught to obtain the original version the estimator was pickled with:

```
from sklearn.exceptions import InconsistentVersionWarning
warnings.simplefilter("error", InconsistentVersionWarning)

try:
    est = pickle.loads("model_from_prevision_version.pickle")
except InconsistentVersionWarning as w:
    print(w.original_sklearn_version)
```

Is it possible to connect to the Internet?

—---

Because of security reasons, it is not possible to access the internet.

Is there a time limit, and can we access it?

—---

There will be a hidden-time limit in the order of minutes per negotiation.