## Installation

Our code is distributed by conda envrionment, so all necessary dependencies to run said envrionment are contained within a single environment. The necessary steps to set this up are:

1. Clone the following github repository: https://github.com/Team-ARES-Airflow-Processing-Pipeline/AirFlowPipeline.git

2. Navigate to installation directory

3. Create a new conda environment referencing the included .yml file with the command:

```
        conda create -n <env_name> -f env.yml
```

Now, it is necessary to alter apache airflow to allow dags to be saved and loaded from the included files. To do this, navigate to airflow's installation folder. This will depend on your system and how airflow was installed, airflow has documents which can be found here: https://airflow.apache.org/docs/apache-airflow/stable/.

Alter the airflow.cfg file's database directory to <airflowpipeline>/bin. <airflowpipeline> being the installation directory of the git repository initially downloaded.

## Running Airflow

Now that everything from the above steps is complete, all that is required to run airflow is to type the following commands:

```
airflow db init
airflow scheduler &
airflow webserver –port 8080 &
```

Navigating to 127.0.0.1:8080 or https://localhost:8080 will bring you to airflow's webserver and allow generated DAGs to be viewed and edited.

## Architecture Overview

There are a few important files:

1. ISIS_AirFlowWrappers.py includes the translations from Kalasiris, the Python wrapper for ISIS.
2. ISIS3_Enums.py includes the enum translation of ISIS3 modules.
3. ISIS3_Mods.py includes the class definitions, or reifications of ISIS3 module runs.
4. airflow_builder.py is the actual translation that generates DAGs.
5. example_recipes.json is an example of how to write pipelines for the software.

For further questions regarding architecture and installation please reach out to Isaiah Raspet at isaiahraspet@gmail.com or by phone (714)319-7815.