SyncPy2 Method Wizard UI documentation

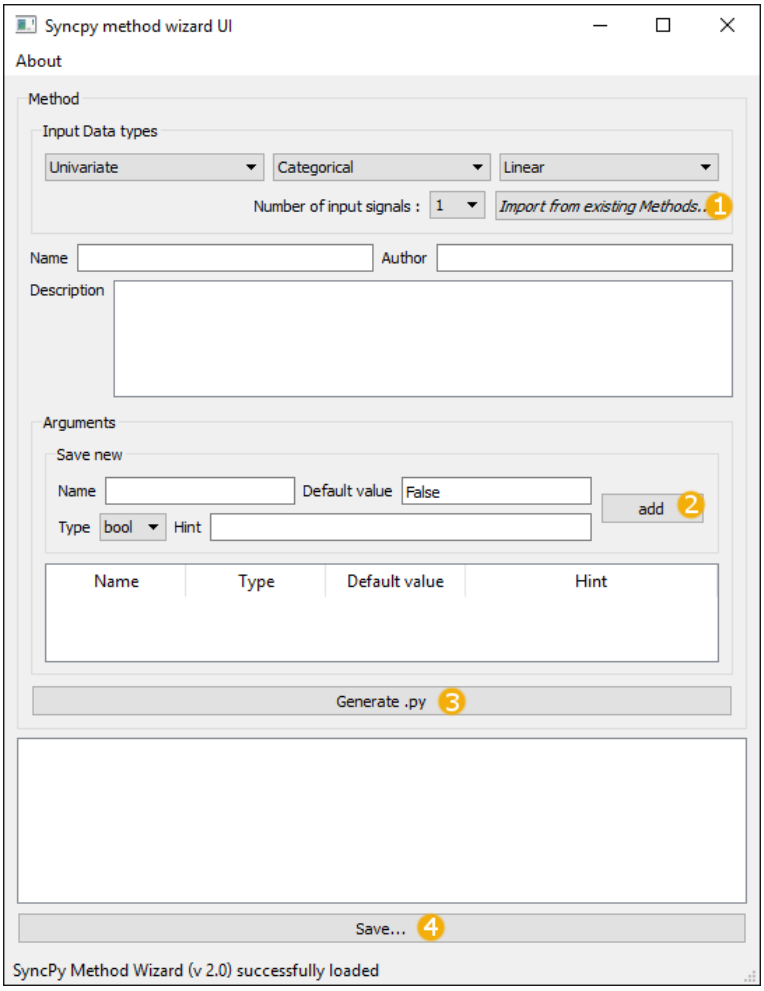
Part 1 – Installation

You need to install anaconda to be able to launch the entire SyncPy suite. If you download the SyncPy installer from the website, it will also install anaconda for you.

You can access the UIs either from the src folder inside the SyncPy installation folder or shortcuts on your desktop if you are on Windows/Mac, or in your home folder if you are on Linux, if you installed SyncPy with the installer.

Part 2 - SyncPy2 Method Wizard UI

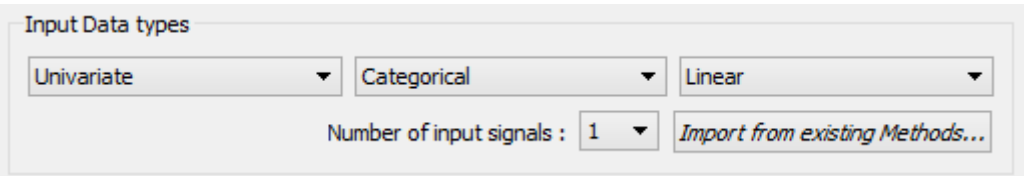
The purpose of this tool is to help to create, from scratch or an example, python coder to implement new Methods that the interface can use to process data.



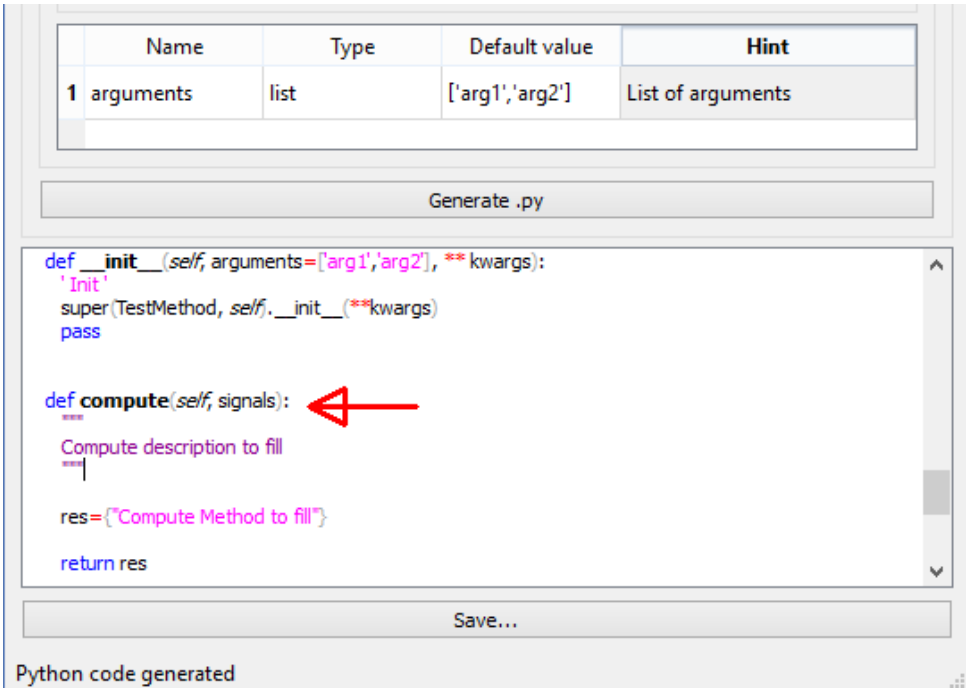
1. Create a method from scratch

a. Choose the input data

First you have to choose on which kind of data the method will work. In the *Input Data types* panel choose the types of the signals, and the number of signals to process.



Or you can use the button to import an existing Method and it will deduce the types for you (see tion do to what you want to do.



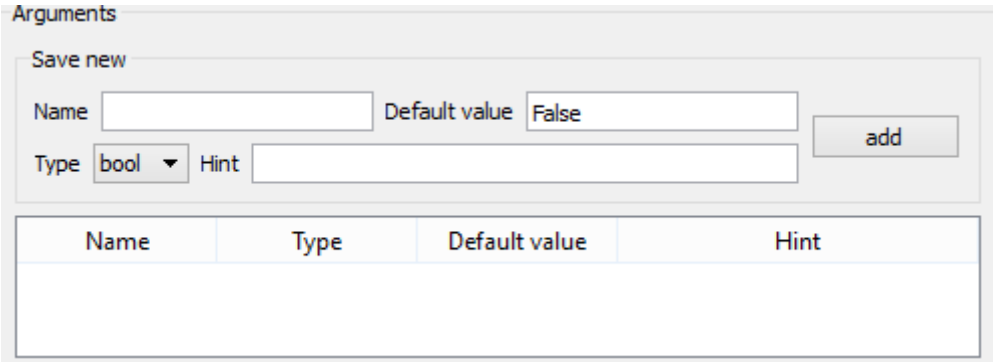
Warning: generating the python file will not save it! To do it click on the *Save* button (4), it will automatically suggest you to save the python file in the right place to be able to use it in the SyncPy UI.

<u>Import an existing method</u> section).

b. Set the main information

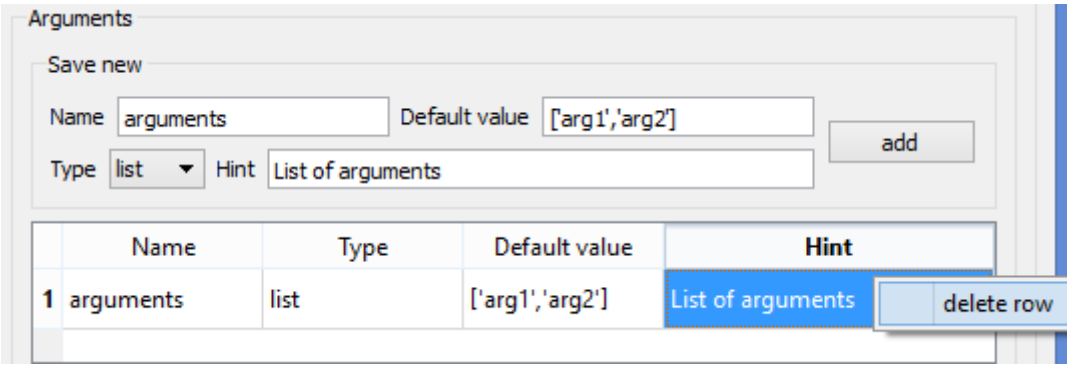
Then you have to choose a name for your method. This will be the name of the generated file and the name displayed in the SyncPy UI. Set the name of the author and a description of what your method is doing.

c. Set the arguments



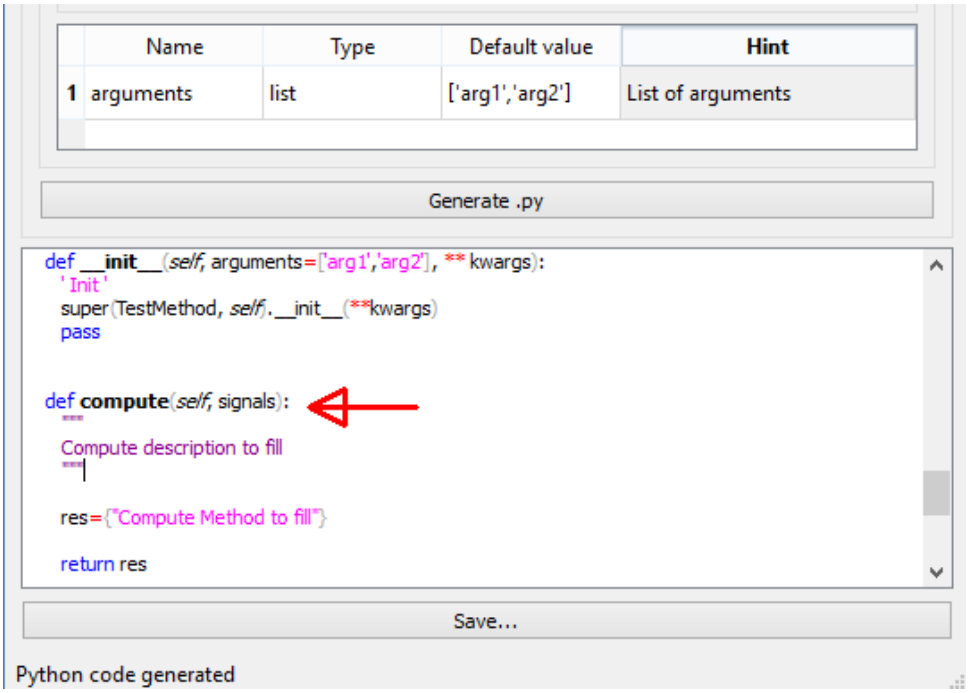
Now you can start to add the arguments of your method. For each argument you **must** specify a name and a default value. For each type of argument, a default value is set to show you the format you must enter. **Warning: if you change the type of the argument you will lose the default value you set!**

If an argument with the same name is already added you can't add it. The hint is optional but useful for the community. Click on the *add* button (2) to add your argument in the arguments list. You can erase at any moment an argument in the list by right clicking on it and choose *delete row*.



d. Generate and save the method

Clicking on the *Generate .py* button (3), the wizard will create your method with all the information you gave. The final step is to edit the *compute* function do to what you want to do.



Warning: generating the python file will not save it! To do it click on the *Save* button (4), it will automatically suggest you to save the python file in the right place to be able to use it in the SyncPy UI.

2. Import an existing method

You can also import an existing method by clicking on the *Import from existing Methods* button (1). Before that you can set the type of the signals and the number of signals, the wizard will only show methods working on the same signals you want to process.

The wizard will fill all the information, add the arguments to the argument list and generate the python file. **Warning: you will have to rewrite the compute method, the wizard will not copy it**.

Be careful to **change the name of your method**, generate the python file and modify the compute function before saving the method.