Overview of the “*SiriusQuality-BioMa-Irradiance-Component”* folder

1. Content of the “*SiriusQuality-BioMa-Irradiance-Component*” folder

The folder “SiriusQuality-BioMa-Irradiance-Component” contains:

* The source code of the *SQ-Irradiance* *BioMa* Component (*SiriusQuality-Irradiance-Component* folder), see *SQ\_Irradiance\_component* document
* The BioMa dll which are mandatory to use the component (*BioMa-DLL* folder)
* A console application which provides an example for the use of the component (*SiriusQuality-IrradianceConsole*
* Unit tests for the component (*TestIrradianceComponent* folder)
* A visual studio solution which allows to run the console application and the unit tests (*SiriusQuality-BioMa-Irradiance.sln*)
* A detailed documentation about both the calculation scheme and the equations of the component (*Documentation* folder)

1. How to use the component:

The *SQ\_Irradiance\_component* can be added to a *BioMa* solution with the *CLIC* tool or be plugged to your model via a wrapper. Here the composition with *CLIC* will not be presented (see instead [*BioMa* solution documentation](http://bioma.jrc.ec.europa.eu/tutorials/Creating_modeling_solution.pdf)). We will first make a quick overview of the component, then we will present the wrapper and finally we will explain how it is used via a console application.

* 1. Overview of the component

The component (*SiriusQuality-Irradiance-Component)* contains six folders:

* The SiriusQuality-Irradiace*DomainClass* folder. Here can be found:
  + The getter and setter of the states, the rates and the exogenous variables (States.cs, Rates.cs, Exogenous.cs)
  + The metadata on the states (StatesVarInfo.cs, RatesVarInfo.cs, ExogenousVarInfo.cs)
* The *strategies* folder where the simple strategies and the composite one (*Irradiance.cs*) can be found. The composite strategy allows to call sequentially the simple strategies via its *Estimate* function
* The *API* folder containing the classes for the Application Programming Interface
* An *XML* folder where the xml files used to generate strategies and domain classes with the BioMa tools can be found
* The *obj/Debug* folder containing the dll of the component after having built it and the BioMA dll which are mandatory for the project
* The *bin* folder for binaries

In the composite class a loop on input variables makes possible the calculation of outputs for Photosynthetically Active Radiations (PAR) and Infra Red Radiadtion (NIR).

* 1. Irradiance wrapper

The wrapper (*SiriusQuality-Irradiance-Console/ IrradianceBiomaWrapper.cs*) makes possible:

* The initial loading of the parameters
* The day by day valorization of the inputs
* The daily call of the component
* And the daily export of the outputs

For these purposes *states, rates and exogenous* objects are instantiated. These objects are used to valorize inputs and export outputs (via getter). In addition, an object of the composite class is instantiated (*abosorbedIrradiance*). It is used to valorize the parameters and call the *Estimate* function of the composite.

The valorization of the parameters is done in the constructor via the *loadParameters* function. The *Estimate* function can be called everywhere in the code. Its arguments are the values of the input for the current day. Two steps are necessary in the *Estimate* function:

* The current day inputs are valorized
* The *Estimate* function of the composite class is called
  1. Console application

The *SiriusQuality-IrradianceConsole/Program.cs* class can be divided in four parts:

* Inputs grouped in tables. The table counts correspond to the number of days in the simulation
* An object *IrradianceBiomaWrapper* is instantiated to be able to call the *Estimate* function and export outputs
* The option of calculation (Daily/Hourly, Canopy/Layer, Global/Sunshade, Spherical/Ellipsoidal leaf angle distribution) can be chosen in the console
* The *Estimate* function of the wrapper is call each day of the simulation until the number of day exceeds the table counts of step 1. As in the universe of *SiriusQuality* a while loop is used. The number of days to be considered can be chosen in the console
* Outputs of the component are called from the wrapper and printed

1. List of the provided libraries

Six libraries are mandatory to be able to run the component. Five of them are part of the BioMa framework and are loaded both in the component and in the console application projects, while *CRA.AgroManagement2014.dll*, *CRA.AgroManagement2014.Impacts.dll* and *CRA.AgroManagement2014.dll* are used for the management of agronomic events (irrigation, fertilization…), *CRA.Core.Preconditions.dll* is used for the test of parameter, input and output values and *CRA.ModelLayer.dll* is dedicated to the generation of domain class and strategies.

When building the *SiriusQuality-Irradiance-Component* project a library called *SiriusQuality-IrradianceDomainClass.dll and a* library called *SiriusQuality-IrradianceStrategies.dll* is created. It is loaded in the console application project (and the corresponding includes is done on top of the wrapper class).

These six libraries have to be loaded in any project aiming at working with the irradiance component, Unit Tests for example.

1. List of inputs, outputs and parameters of the component:

An exhaustive list of inputs and outputs for the component is given in *Documentation/SQ\_Irradiance\_component*. A list of parameter is provided as well, but, only the following ones are specific to the option of calculation and are not presented in *SQ\_Irradiance\_component*:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 1:** List of parameters for the choice of calculation option in the *Irradiance* component**.** | | | |  |
| **Name in the code** | | **Description** |
| IsSunShadeUsed | | 0: if average intercepted irradiance is considered, 1: if leaves are separated into a lighted and a shaded area |
| IsHourlyUsed | | 0: for a hourly basis calculation, 1: for daily basis |
| IsCanopyUsed | 0: if leaf layers are considered, 1: if an average irradiance is considered at the level of the canopy (Big Leaf model) | |
| IsSphereDistUsed | | In case the leaf area is separated into a lighted and a shaded part, 0: for an ellipsoidal leaf angle distribution, 1: for spherical one |