European
Commission

BioMA

CLIC

Composition Layer Interactive Code

User Guide

Joint
Research
Centre

Release	Issue	Date
1	5	September 2019

Copyright

Disclaimer

On any of the MARS pages you may find reference to a certain software package, a particular contractor, or group of contractors, the use of one or another sensor product, etc. In all cases, unless specifically stated, this does not indicate any preference of the Commission for that particular product, party or parties. When relevant, we include links to pages that give you more information about the references.

Feel free to contact us, in case you need additional explanations or information.

Contents

Help contents

1

This Guide is targeted to advanced users of the BioMA Software Framework.

In particular, this Help describes how to use CLIC (Composition Layer Interactive Code), the desktop application for model composition.

The target users of this application are modelers with some code development skills in .NET C#..

The topics are organized as follows:

Topic	Contents
Introduction to CLIC	• Composition Layer overview • A quick glimpse of what CLIC can be used for • User Interface overview
Exploring existing models	How to load and explore existing components.
Creating new model runners	• How to create model components and how to add them to the model • How to link components one another • How to define the model’s properties

See also:

- Composition Layer Documentation

Tip:

To access the documentation of the BioMA-related software components, go to https://agri4cast.jrc.ec.europa.eu/DataPortal/Index.aspx?o=s

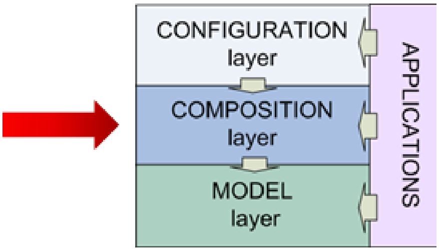
Introduction to CLIC

2

In this section:

- “About the Composition layer” on page 6
- “What CLIC is used for” on page 7

About the Composition layer



Tip:

For a comprehensive documentation on how the Composition Layer is coded, please refer to the Composition Layer Documentation.

The Composition layer allows linking model components to build model solutions for a specific purpose.

The aim of the composition layer is to compose several simulation components into one model (modelling solution) structure. Each simulation component encapsulates the logics of a well-defined part of the model and the logic to use the required data to run the model (e.g., the weather data). Components can be developed independently one another and are not directly inter-related.

Composing two components implies to:

- Execute them in a given sequence
- Set the desired output (from one component) - input (to another component) connection

The connection between components is named '**link**'.

Each component includes specific inputs, outputs, and parameters, as well as a '`run time data`' property, which includes all the domain classes that the component needs to access.

For further information, refer to CREA Web site: http://www.biomamodelling.org).

The '`run time data`' values of a component are read/written by other components through the `link` classes.

What CLIC is used for

CLIC (Composition Layer Interactive Coder) is a desktop application aimed to assist modelers in creating a ModelRunner C# project.

A model runner is an instance of the Composition Layer.

(For further information on the BioMA multi-layer architecture, please refer to the **BioMA Framework User Guide**, which you can find here: http://agri4cast.jrc.ec.europa.eu/DataPortal/Index.aspx, chapter “BioMA Architecture”)

As a modeler, CLIC allows you to:

- Create the simulation components from components Composition layer-independent
- Define the links between components, the model runner class and the C# project structure.
- Explore an already compiled project containing a model runner
- Draw a graph of the structure, that is, drawing a list of components that shows how components are linked one another ((the graphic metaphore is at early stage of development)
- Extract one or more components so as to re-use them for creating another model runner.

Workflow example

1. The modeler explores a modelling solution composed by four components.
2. He/she decides to re-use two components to create a new model, as well as to add to this latter other components from scratch.
3. Given that each component code is independent, the modeler must create only:
   a. The code to create a simulation component only for the ***new*** components
   b. The links between the new component and the ones already present in the modelling solution

   The code to interface the other components to the Composition layer instance remains unchanged.

As a result:

- The output of the application is a folder containing all the C# class files that are required for the project (`.cs` files), as well as the project structure file (`.csproj` file). The code is generated referring to the .NET 3.5 runtime and the project file is relative to Visual Studio 2008 format (which can be easily converted to a newer version of Visual Studio).
- Each class file that is generated includes areas that the user can fill with custom code by writing it manually within the ad hoc areas delimiters. This code will be preserved in a future code generation.
- CLIC displays, in a different window, a graph showing the connections between components.

Tip:

In the **Agri4Cast Software Portal** at https://agri4cast.jrc.ec.europa.eu/DataPortal/Index.aspx?o=s, you can find a step-by-step tutorial, where CLIC is being used to compose a modeling solution. To access it:

- After logging-in, scroll-down to the **Software Tools and Documentation** table and, in the **Tutorial - How to create a Modelling Solution** row, download the setup files and the Tutorial.

Related topics:

- “About the Composition layer” on page 6
- “Installing and launching CLIC” on page 10
- “User Interface overview” on page 11

Using CLIC to compose models

3

Installing and launching CLIC

CLIC is provided as a plug-in of BioMa Spatial and is used to create a modelling solution.

To launch CLIC:

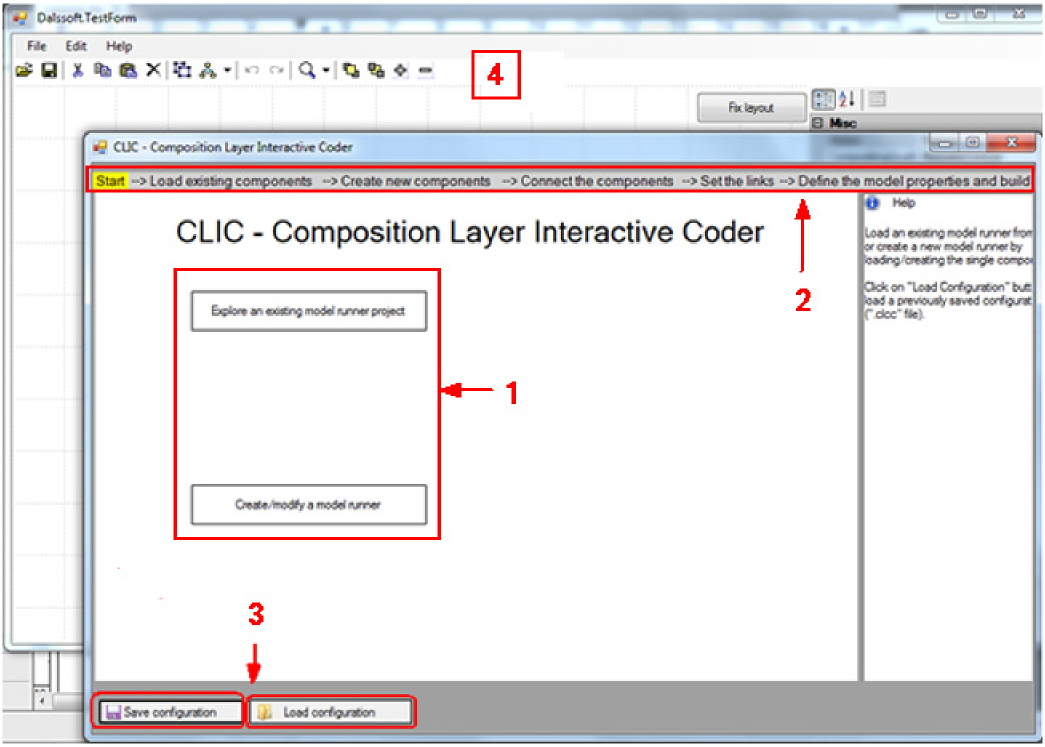
1. Go to the **Agri4Cast Resources Portal** (http://agri4cast.jrc.ec.europa.eu/DataPortal/Index.aspx?o=s).
2. If you are not registered yet, click **Register** at the top-right, then follow the instructions to register.
3. Scroll-down to the **Software Tools and Documentation** table.
4. In the **Tutorial - How to create a Modelling Solution** row, click the 💿 icon.
5. Unzip the **TrainingCreateModelingSolution.zip** package we have provided in a folder at your choice in your PC.
6. Copy the CLIC folder that you find in the **TrainingCreateModelingSolution.zip** package in any location in your computer.
7. Since no installation is required, to launch CLIC, double-click the **CLIC.exe** executable file that you find in the same folder.

See also:

- “User Interface overview” on page 11
- “Exploring existing models” on page 13
- “Creating new model runners” on page 14

User Interface overview

The following shows the **Start** page that is displayed after launching the application:


Dalssoft TestForm
File
Edit
Help
4
Fix layout
Misc
CLIC - Composition Layer Interactive Coder
Start --> Load existing components --> Create new components --> Connect the components --> Set the links --> Define the model properties and build
CLIC - Composition Layer Interactive Coder
Explore an existing model runner project
Create/modify a model runner
1
2
3
Help
Save configuration
Load configuration

Load an existing model runner fron
or create a new model runner by
loading/creating the single compo

Click on "Load Configuration" butt
load a previously saved configurat
("clcc" file).

Note that, in the back of the CLIC start page, another window is displayed, which provides a graphical view of the connected components.

The following table describes the main elements of the UI and provides links to the relevant sections for further information:

Element	Description	More...
1	The buttons that allows you choosing whether to access an existing model or to create a new one, respectively.	page 13 page 14

Element	Description	More...
2	The application pages. After choosing whether you want to use an existing model or create a new one, you will be able to move back and forward using the **Next** and **Back** buttons that will be made available.	
3	The buttons that allows you to save and load a configuration. These buttons are available in all pages.	page 25
4	The window will provide a graphical view of the components you connected allowing you to view and edit the links' details.	page 22

Related topics:

- "Exploring existing models" on page 13
- "Creating new model runners" on page 14

Exploring existing models

This option allows you to explore and reuse an existing, compiled, model runner.

Procedure:

1. In the **Start** page of the application, click the **Explore an existing model runner project** button.
2. In the **Select the model runner library (dll)** area that is displayed, click the browse button and select the DLL where the model runner is located, as well as its dependencies.

   (**Note**: If no dependencies are selected, CLIC will explore the model runner, but it will not be able to reuse components).
3. The **Load existing components** page is displayed. Click the **Back** button if you need to go back to the previous page.

See also:

- "Loading existing components" on page 14
- "Creating new components" on page 16
- "Connecting components" on page 17
- "Linking the connected components" on page 18
- "Defining the model properties and building the project" on page 22
- "Saving and loading configurations" on page 25

Creating new model runners

This option allows you creating a new model runner from scratch or to modify an existing one.

Procedure

1 In the **Start** page of the application, click the **Create/modify a model runner** button.

2 The **Load existing components** page is displayed. (Click the **Back** button if you need to go back to the previous page).

See also:

- “Loading existing components” on page 14
- “Creating new components” on page 16
- “Connecting components” on page 17
- “Linking the connected components” on page 18
- “Defining the model properties and building the project” on page 22
- “Saving and loading configurations” on page 25

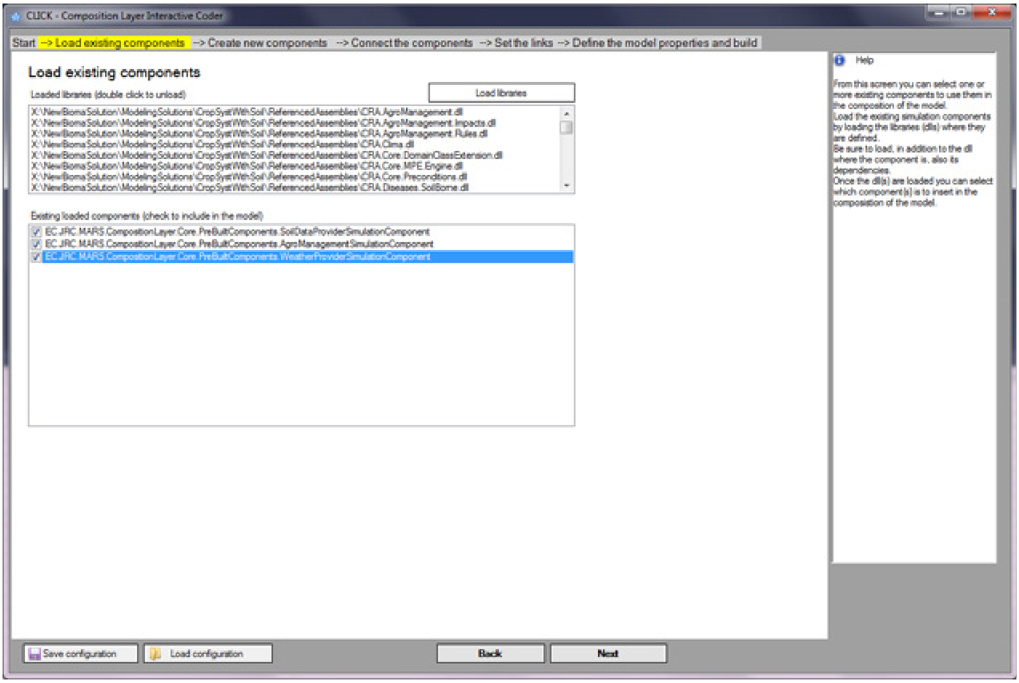
Loading existing components

The **Load existing components** page allows loading existing simulation components (valid implementation of the `ISimulationComponent` interface) that will be used to compose the model.

To do this, you must load the libraries (DLLs) where the simulation components are defined, as well as the related dependencies.

Once the libraries are loaded, you can select the component(s) you want to use for composing the model.

Figure 1 Load existing components page


CLICK - Composition Layer Interactive Coder
Start --> Load existing components --> Create new components --> Connect the components --> Set the links --> Define the model properties and build
Load existing components
Load libraries
Save configuration
Load configuration
Back
Next
Help

Loaded libraries (double click to unload)

```
X:\NewBioma.Solution\ModelingSolutions\CropSystWithSoil\ReferencedAssemblies\CRA.AgroManagement.dll
X:\NewBioma.Solution\ModelingSolutions\CropSystWithSoil\ReferencedAssemblies\CRA.AgroManagement.Impacts.dll
X:\NewBioma.Solution\ModelingSolutions\CropSystWithSoil\ReferencedAssemblies\CRA.AgroManagement.Rules.dll
X:\NewBioma.Solution\ModelingSolutions\CropSystWithSoil\ReferencedAssemblies\CRA.Clima.dll
X:\NewBioma.Solution\ModelingSolutions\CropSystWithSoil\ReferencedAssemblies\CRA.Core.DomainClassExtension.dll
X:\NewBioma.Solution\ModelingSolutions\CropSystWithSoil\ReferencedAssemblies\CRA.Core.MPE.Engine.dll
X:\NewBioma.Solution\ModelingSolutions\CropSystWithSoil\ReferencedAssemblies\CRA.Core.Preconditions.dll
X:\NewBioma.Solution\ModelingSolutions\CropSystWithSoil\ReferencedAssemblies\CRA.Diseases.SoilBorne.dll
```

Existing loaded components (check to include in the model)

- ☑ EC.JRC.MARS.CompositionLayer.Core.PreBuiltComponents.SoilDataProviderSimulationComponent
- ☑ EC.JRC.MARS.CompositionLayer.Core.PreBuiltComponents.AgroManagementSimulationComponent
- ☑ EC.JRC.MARS.CompositionLayer.Core.PreBuiltComponents.WeatherProviderSimulationComponent

From this screen you can select one or more existing components to use them in the composition of the model.
Load the existing simulation components by loading the libraries (dlls) where they are defined.
Be sure to load, in addition to the dll where the component is, also its dependencies.
Once the dll(s) are loaded you can select which component(s) is to insert in the composition of the model.

1 The first step is selecting the libraries. There are two possible scenarios:

   a. If you are exploring an existing model runner, both lists are already filled with the components of the model runner you loaded in the **Start** page (see “Exploring existing models” on page 13).

   b. If you are creating a new model runner, you must click the **Load libraries** button in the **Loaded libraries** area and then select the DLL files you want to use. Be sure that you select the dependencies also.

2 Once the libraries are selected, select the components you want to use in the model runner by clicking the checkbox next to the component’s name in the **Existing loaded components** list.

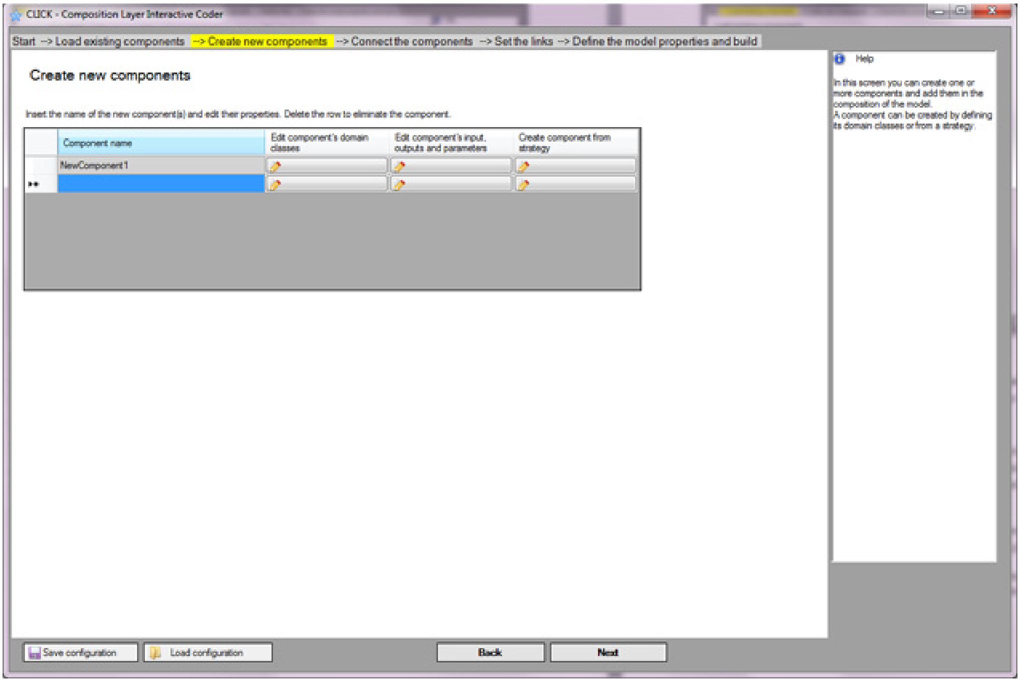
3 When finished, click **Next** to continue.

Creating new components

The **Create new components** page allows creating one or more components and then add them to the model.

You can create a component either by defining its domain classes or from a Model Layer's strategy.

Figure 2 Create new components page


CLICK - Composition Layer Interactive Coder
Start --> Load existing components --> Create new components --> Connect the components --> Set the links --> Define the model properties and build
Create new components
Help
Save configuration
Load configuration
Back
Next

Insert the name of the new component(s) and edit their properties. Delete the row to eliminate the component.

	Component name	Edit component's domain classes	Edit component's input, outputs and parameters	Create component from strategy
	NewComponent1			
▸*				

In this screen you can create one or more components and add them in the composition of the model.
A component can be created by defining its domain classes or from a strategy.

To create a component:

1 Enter the new component name in the **Component name** column. Note that a progressive number is automatically added to the component's name to avoid names duplication.

2 Create the component by setting its domain classes and its inputs/outputs:

   a. Click **Edit component's domain classes**. In the window that is displayed, set the domain classes as required and then click **Save changes**.

   b. Click **Edit component's input/outputs and parameters**. In the the window that is displayed, check the inputs and outputs you want to add to the new component and then click **Save changes**.

3 Alternatively, create the component from a strategy:

   a. Click **Create component from strategy**.

   b. In the window that is displayed, click **Add libraries** if you want to add libraries to the current list.

   c. In the list of available strategies, select the checkbox to add a strategy. To view the strategy details, click the **View details** button in the relevant row.

   d. When finished, click **OK**.

4 Click **Next** to continue to the next step.

Connecting components

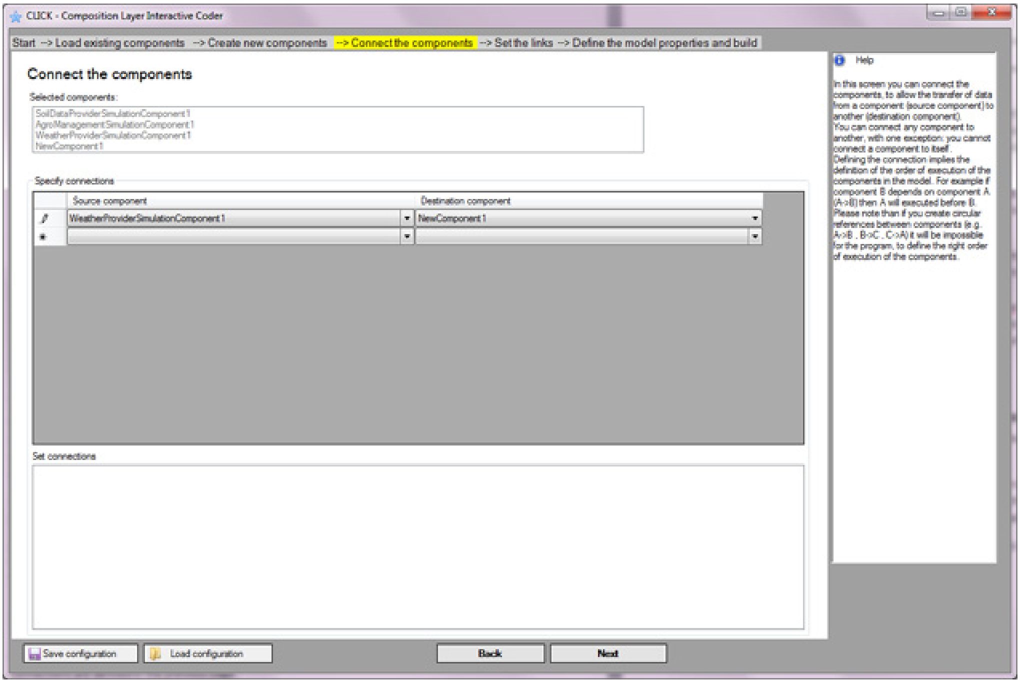
The **Connect the components** page allows you to set the connections between the components, that is, to define the order of execution of the components in the model.

The user can connect any component to another: for example, if component B depends on component A (A->B), then A will executed before B.

Warnings:

- Do not create circular references between components (e.g., A->B , B->C , C->A) otherwise the program will not be able to define the right order of execution of the components so resulting in a random order.
- Do not connect a component to itself. This will generate an error message.

Figure 3 Connect the components page


CLICK - Composition Layer Interactive Coder
Start --> Load existing components --> Create new components --> Connect the components --> Set the links --> Define the model properties and build
Save configuration
Load configuration
Back
Next

Connect the components

Selected components:

SoilDataProviderSimulationComponent1
AgroManagementSimulationComponent1
WeatherProviderSimulationComponent1
NewComponent1

Specify connections

	Source component	Destination component
	WeatherProviderSimulationComponent1	NewComponent1
*		

Set connections

Help

In this screen you can connect the components, to allow the transfer of data from a component (source component) to another (destination component).
You can connect any component to another, with one exception: you cannot connect a component to itself.
Defining the connection implies the definition of the order of execution of the components in the model. For example if component B depends on component A (A->B) then A will executed before B.
Please note than if you create circular references between components (e.g. A->B , B->C , C->A) it will be impossible for the program, to define the right order of execution of the components.

To connect the components:

1. From the **Source component** dropdown list, select the component from which to transfer the data (that is, the outputs).
2. From the **Destination component** dropdown list, select the component to which you want to transfer the outputs (as inputs of the destination component).

   As a result, all connections you have set will be summarized in the **Set connections** pane.
3. Click **Next** to continue to the next step.

Linking the connected components

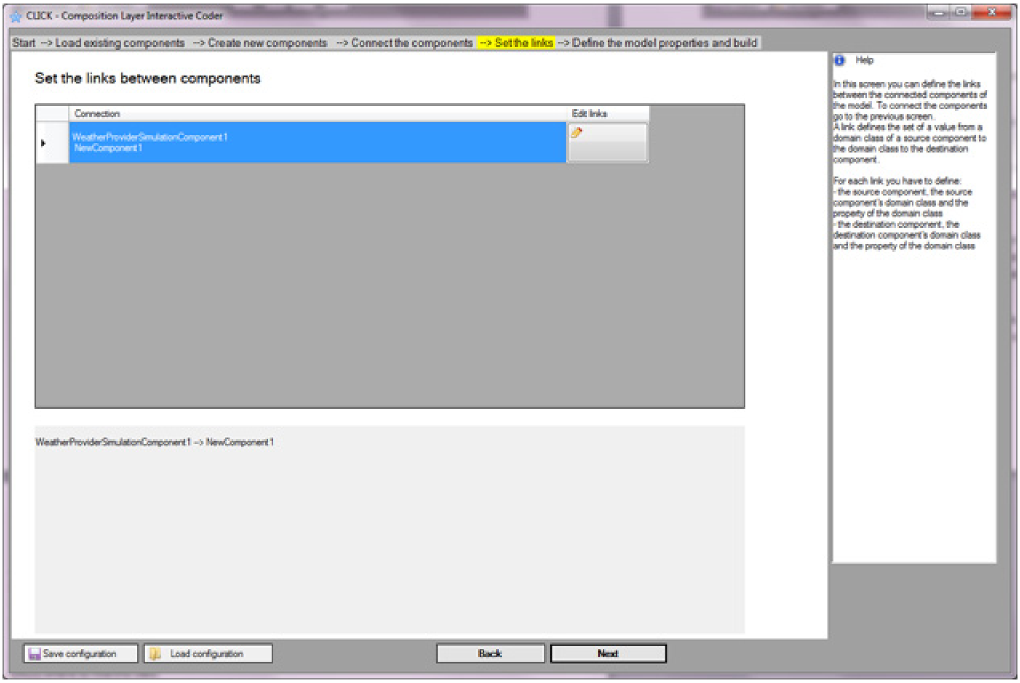
The **Set the links** page allows you to define the links among the connected components.

A link allows setting a value from a domain class of a source component to the domain class of the destination component. For each link the user must define:

- The source component, the source component's domain class and the property of the domain class.
- The destination component, the destination component's domain class and the property of the domain class.

The page shows the components connections that have been defined in the previous page (see “Connecting components” on page 17):

Figure 4 Set the links page


CLICK - Composition Layer Interactive Coder
Start --> Load existing components --> Create new components --> Connect the components --> Set the links --> Define the model properties and build
Save configuration
Load configuration
Back
Next

Set the links between components

	Connection	Edit links
▸	WeatherProviderSimulationComponent1 NewComponent1	

WeatherProviderSimulationComponent1 --> NewComponent1

Help

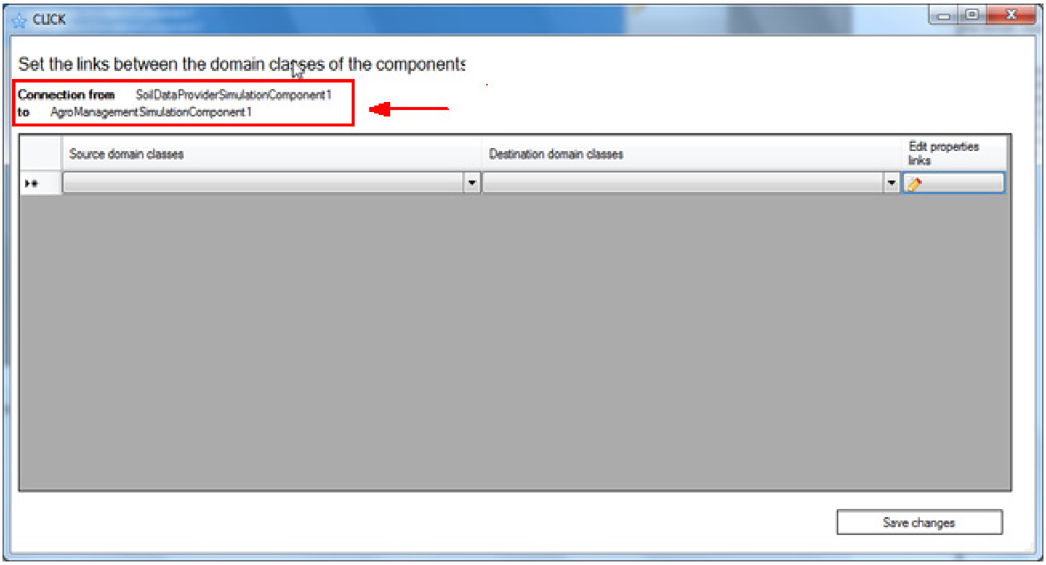
In this screen you can define the links between the connected components of the model. To connect the components go to the previous screen.
A link defines the set of a value from a domain class of a source component to the domain class to the destination component.

For each link you have to define:
- the source component, the source component's domain class and the property of the domain class
- the destination component, the destination component's domain class and the property of the domain class

To define the links among components:

1 Click **Edit links** next to the **Connection** that includes the two connected components.

The **Set links between the domain classes of the components** is displayed:



Set the links between the domain classes of the components

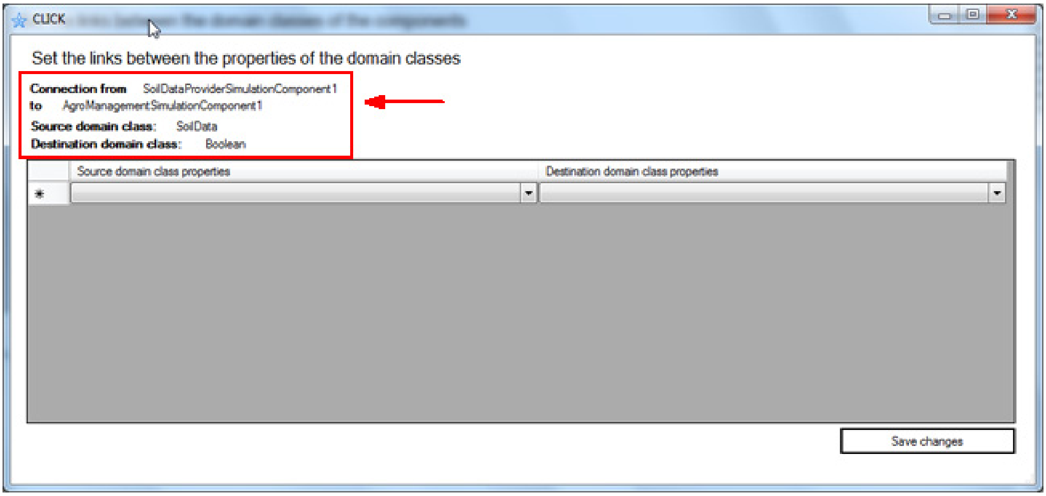
Connection from SoilDataProviderSimulationComponent1
to AgroManagementSimulationComponent1

Source domain classes	Destination domain classes	Edit properties links

Note that, at top of the window, is displayed the currently selected connection. Each component includes specific domain classes, which are listed in the relevant dropdown list.

2 From both the **Source domain classes** and the **Destination domain classes** dropdown lists select the classes you want to link to each other and then click **Edit links**.

The **Set links between the properties of the domain classes** is displayed:



Set the links between the properties of the domain classes

Connection from SoilDataProviderSimulationComponent1
to AgroManagementSimulationComponent1
Source domain class: SoilData
Destination domain class: Boolean

	Source domain class properties	Destination domain class properties
*		

Note that now, at the top the window, are displayed the connections and the links you have already set. Each domain class includes specific properties, which are listed in the relevant dropdown list.

3 From both the **Source domain class properties** and the **Destination domain class properties** dropdown lists select the properties you want to link to each other and then click **Save changes**.

4 The application returns to the **Set the links between components** page. Click **Next** to continue to the next step.

Note:

When setting a link among properties it may happen that the types of the two properties differ, which might result in the code not compiling. CLIC automatically uses a pre-defined converter, if any is available, so as to create, in the generated class, the code that allows converting the source property type into the destination property type.

If no converter is available, the generated code will not compile, but the application will generate a warning message to inform the user. Once the project code has been created, the user must adjust it manually.

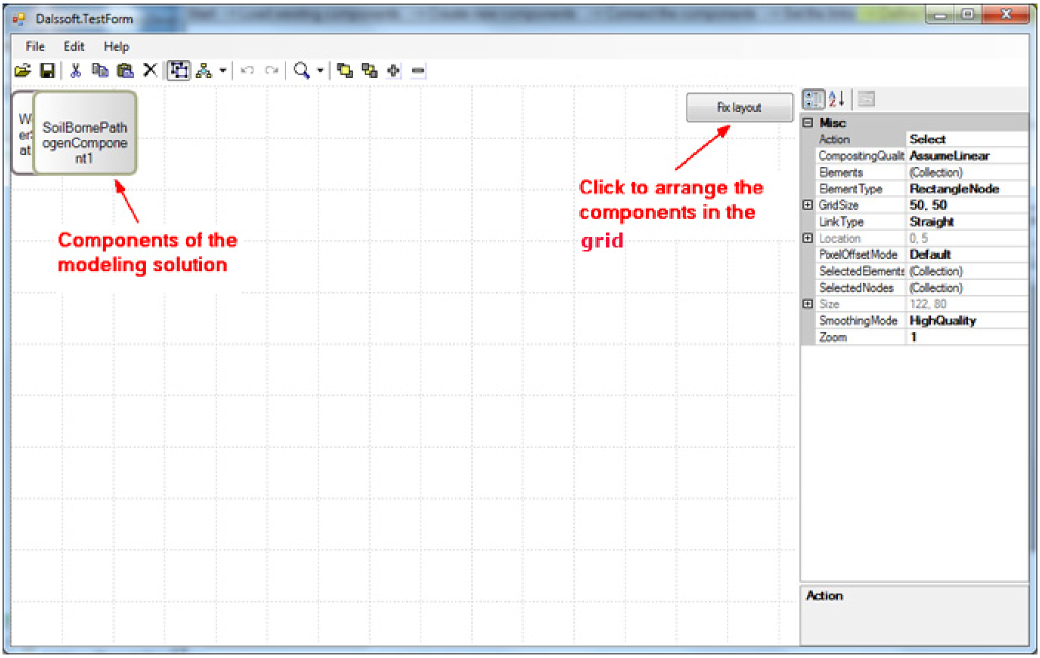
A set of converters is defined by default (e.g., the Double to String and the Decimal to Double converters), but you can define and add a new set of custom converters that CLIC can use. To do it, you must create a converter as implementation of the EC.JRC.MARS.CompositionLayerClassCoder.TypeConverter interface and register it into the TypeConvertersForLinks.xml file that is available in the main directory of CLIC. In this file, you must indicate the two types (source and destination), the full name of the converter class and the libraries (DLLs) where to find this class.

Viewing the links graphically

You can use the window that opens in the back when starting CLIC to view and edit the results of the components connection.

To view the connections:

1 Click the window on the back of CLIC:



Components of the modeling solution

Click to arrange the components in the grid

Misc	
Action	Select
CompositingQualit	AssumeLinear
Elements	(Collection)
Element Type	RectangleNode
Grid Size	50, 50
Link Type	Straight
Location	0, 5
PixelOffsetMode	Default
SelectedElements	(Collection)
SelectedNodes	(Collection)
Size	122, 80
SmoothingMode	HighQuality
Zoom	1

2 If the components icons are piled, click the **Fix layout** button so as to arrange the components in the grid.

3 Once the components are displayed in the grid, click the connection line to view the details.

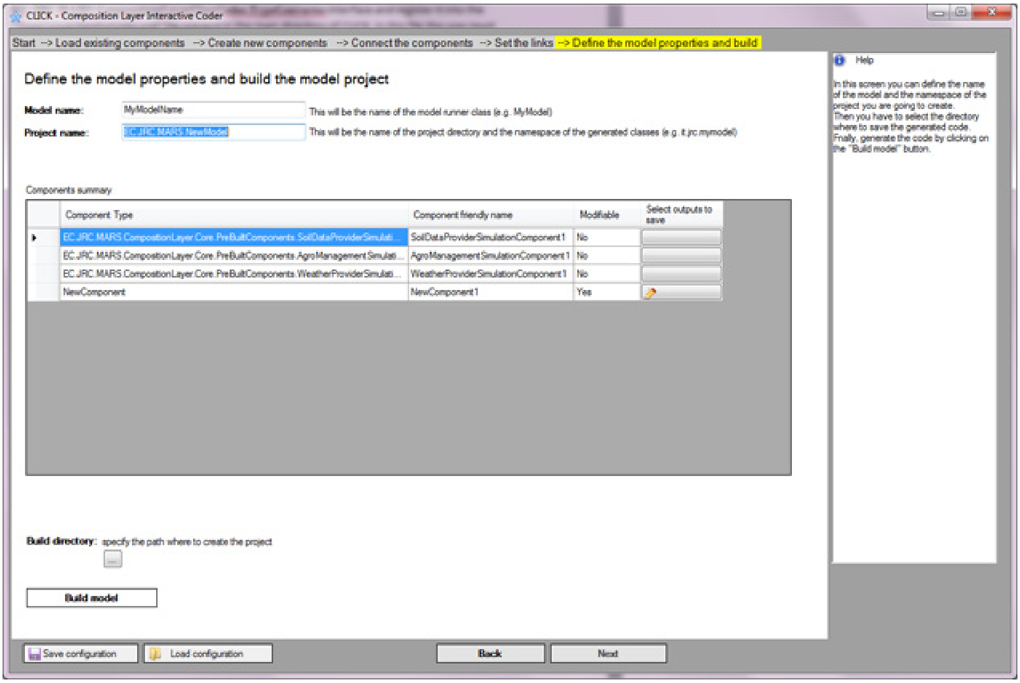
4 The **Set the links between the domain classes of the components** window will be displayed that allows you to view and/or edit the links.

Defining the model properties and building the project

The **Define the model properties and build the model project** page allows you to complete the process.

\

Figure 5 Define the model properties and build page



CLICK - Composition Layer Interactive Coder

Start --> Load existing components --> Create new components --> Connect the components --> Set the links --> Define the model properties and build

Define the model properties and build the model project

Model name: MyModelName — This will be the name of the model runner class (e.g. MyModel)

Project name: EC.JRC.MARS.NewModel — This will be the name of the project directory and the namespace of the generated classes (e.g. it.jrc.mymodel)

Components summary

	Component Type	Component friendly name	Modifiable	Select outputs to save
▸	EC.JRC.MARS.CompositionLayer.Core.PreBuiltComponents.SoilDataProviderSimulati...	SoilDataProviderSimulationComponent1	No	
	EC.JRC.MARS.CompositionLayer.Core.PreBuiltComponents.AgroManagementSimulati...	AgroManagementSimulationComponent1	No	
	EC.JRC.MARS.CompositionLayer.Core.PreBuiltComponents.WeatherProviderSimulati...	WeatherProviderSimulationComponent1	No	
	NewComponent	NewComponent1	Yes	

Build directory: specify the path where to create the project

Help

In this screen you can define the name of the model and the namespace of the project you are going to create.
Then you have to select the directory where to save the generated code.
Finally, generate the code by clicking on the "Build model" button.

To complete the process and build the model:

1 In the **Model name** text box, enter the name for the model runner class.

2 In the **Project name** text box, enter the namespace of the project (e.g., `it.jrc.mymodel`).

3 Optionally, for each component to be created, define which of its output properties you want to save in the DataCollection returned by the model runner by clicking **Select outputs to save** in the relevant row.

Note: This operation is not available for already existing simulation components.

4 Click **Build model** to complete the model creation.

See also:

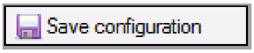
- "Loading existing components" on page 14

- “Creating new components” on page 16
- “Connecting components” on page 17
- “Linking the connected components” on page 18
- “Saving and loading configurations” on page 25

Saving and loading configurations

CLIC allows you to save a configuration or to load an existing configuration file at any step of the model composition.

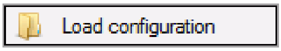
To save a configuration:

1 In any page of the application, click , at the bottom of the window.

2 Specify the location and the **File name** for the configuration, then click **Save**.

The configuration will be saved as a `.clcc` file (in XML format) that you will be able to re-load for further editing.

To load a previously saved configuration:

1 In any page of the application, click , at the bottom of the window.

2 Browse to locate the previously saved `.clcc` file and then cilck **Open**.

See also:

- “Loading existing components” on page 14
- “Creating new components” on page 16
- “Connecting components” on page 17
- “Linking the connected components” on page 18
- “Defining the model properties and building the project” on page 22