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

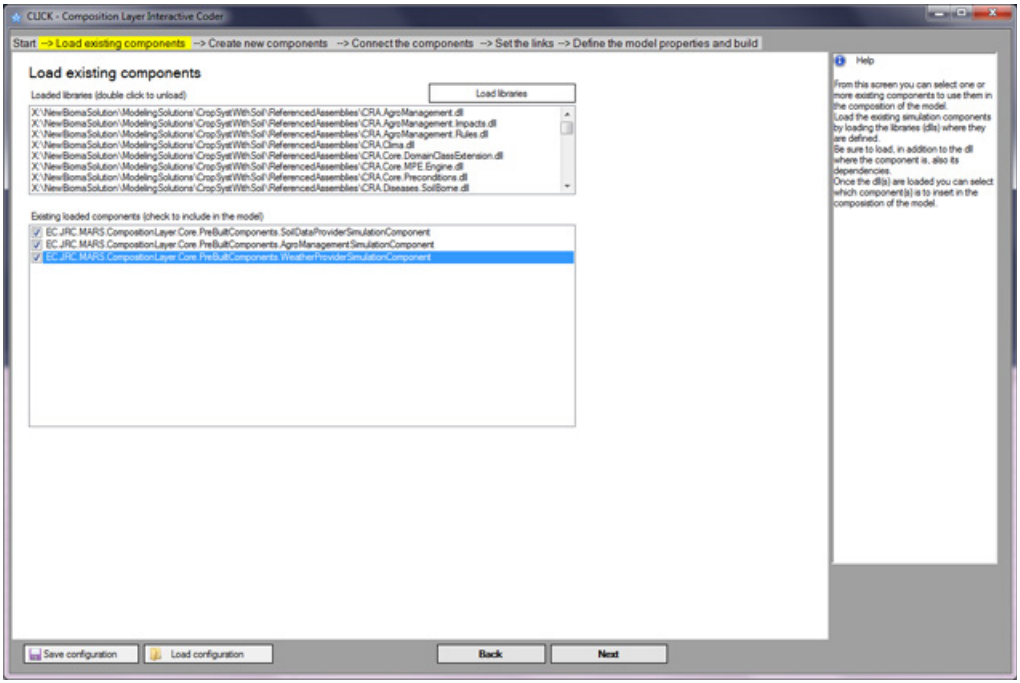
Loading existing components

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 2 Load existing components page



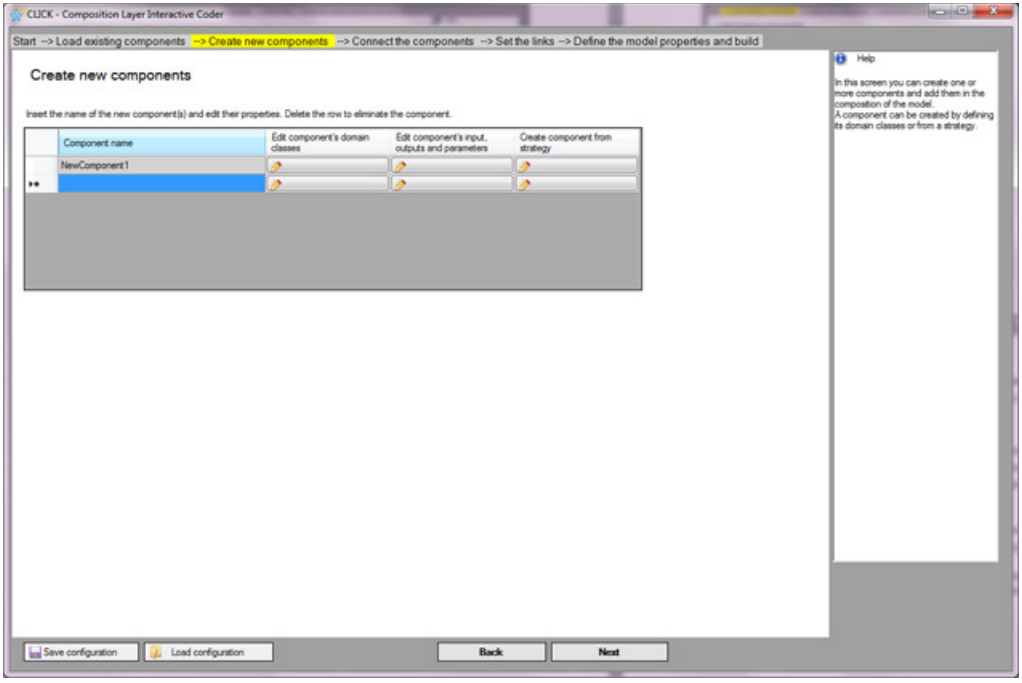
1. The first step is selecting the libraries. There are two possible scenarios:
   a. If you are exploring an existing model runner, both lists are already filled with the components of the model runner you loaded in the **Start** page (see “Exploring existing models” on page 13).
   b. If you are creating 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 3 Create new components page



To create a component:

Tip:

For reference documentation related to domain classes and strategies, please refer to http://bioma.jrc.ec.europa.eu/components/default.aspx?productname=prepostconditions

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 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

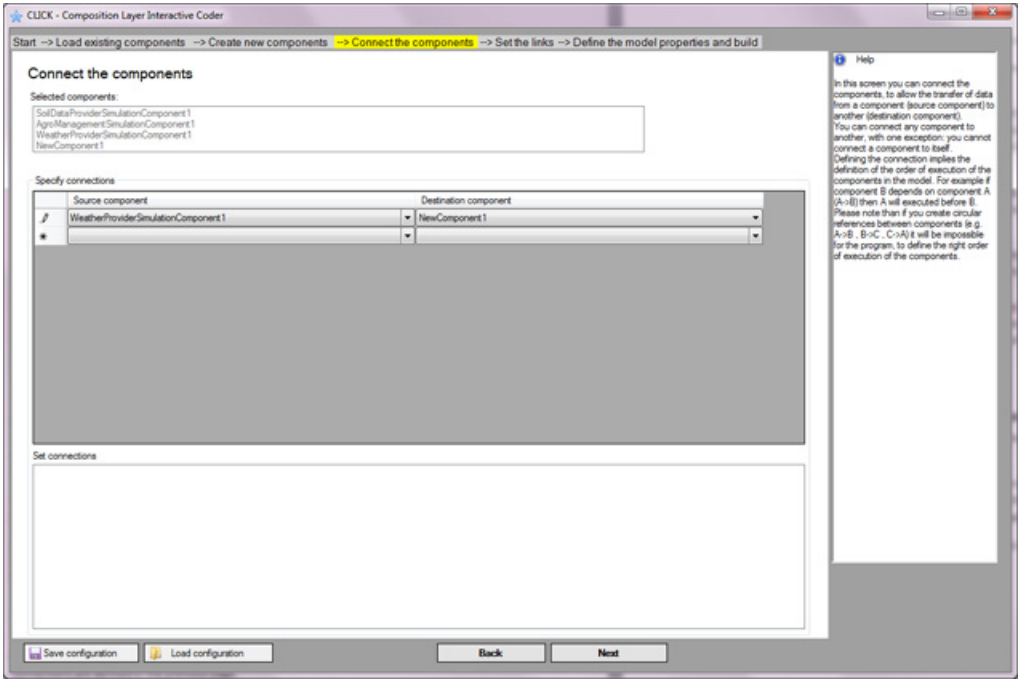
The **Connect the components** page allows you to set the connections between the components, that is, to define the order of execution of the components in the model.

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 4 Connect the components page



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

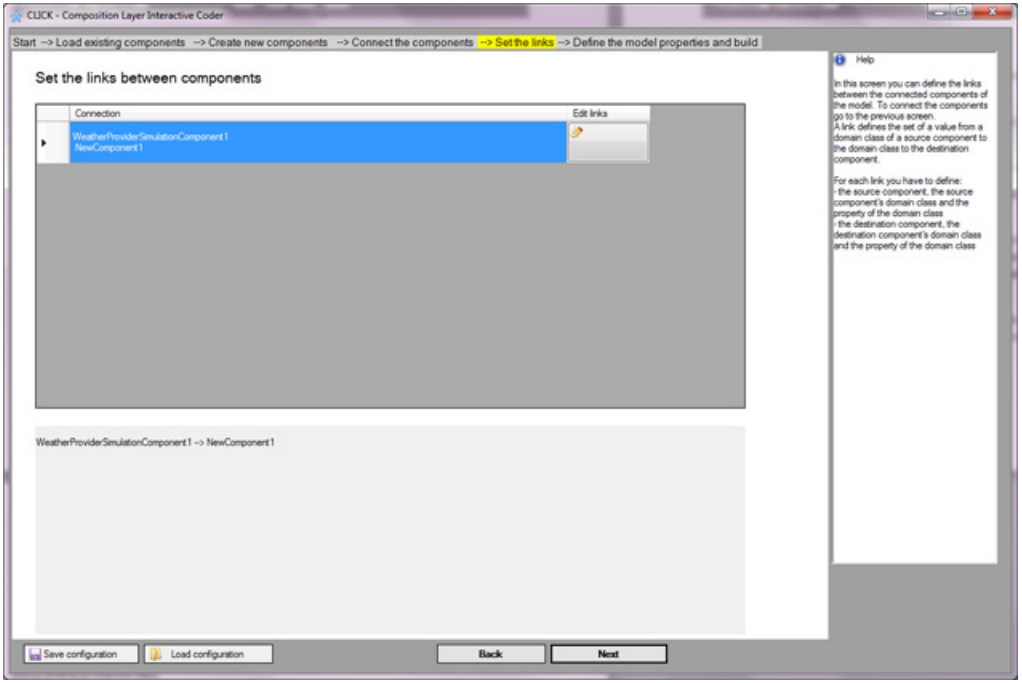
The **Set the links** page allows you to define the links among the connected components.

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):

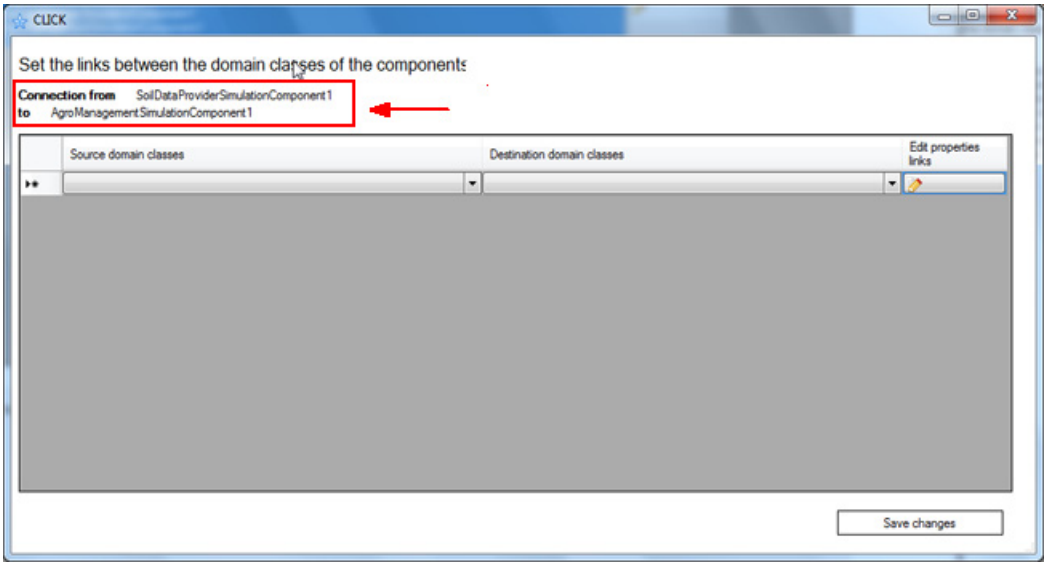
Figure 5 Set the links page



To define the links among components:

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

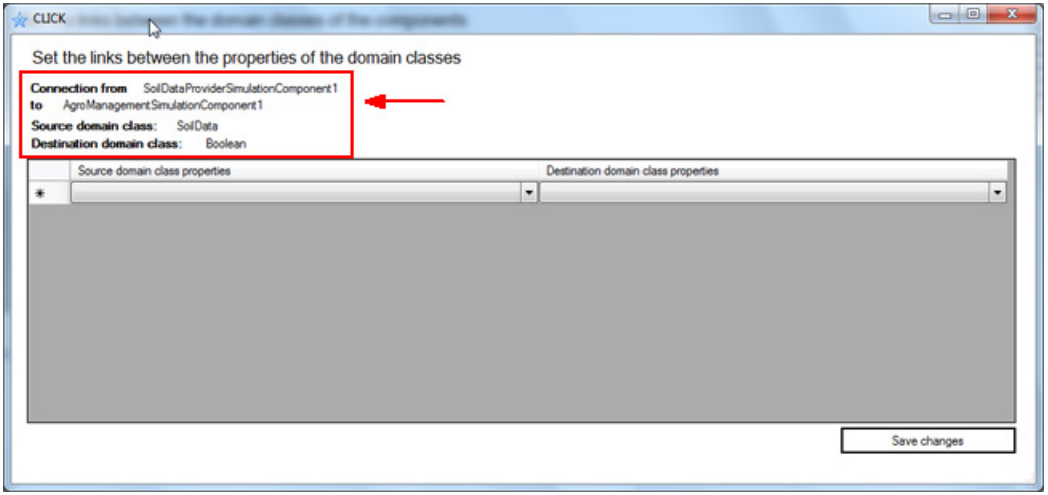
The **Set links between the domain classes of the components** is displayed:



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:



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.

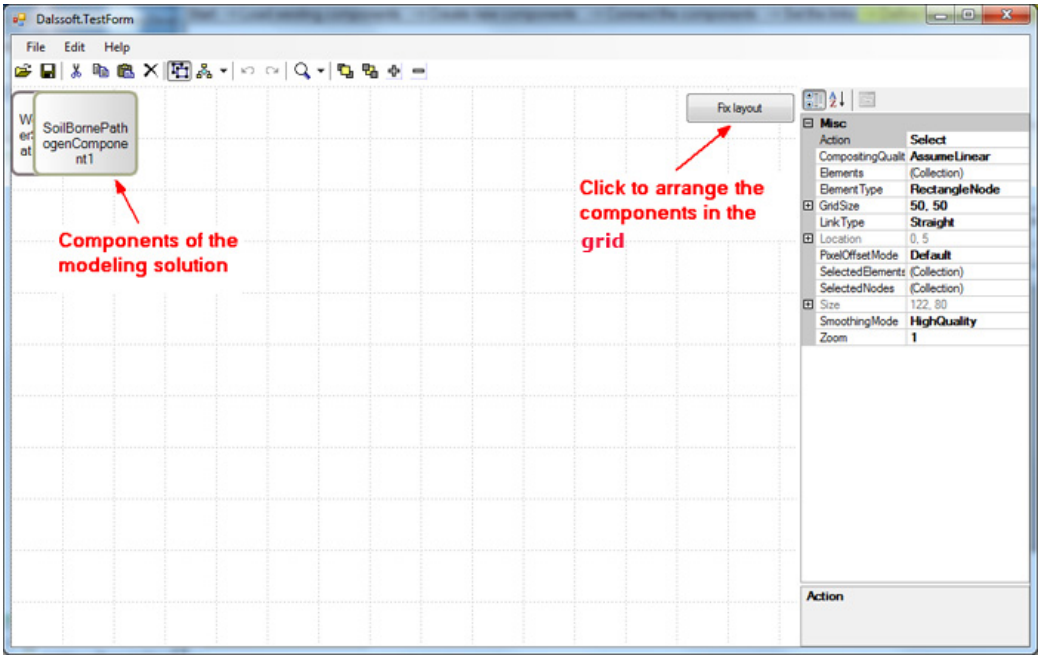
A set of converters is defined by default (e.g., the Double to String and the Decimal to Double converters), but you can define and add a new set of custom converters that CLIC can use. To do it, you must create a converter as implementation of the EC.JRC.MARS.CompositionLayerClassCoder.TypeConverter 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:



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.

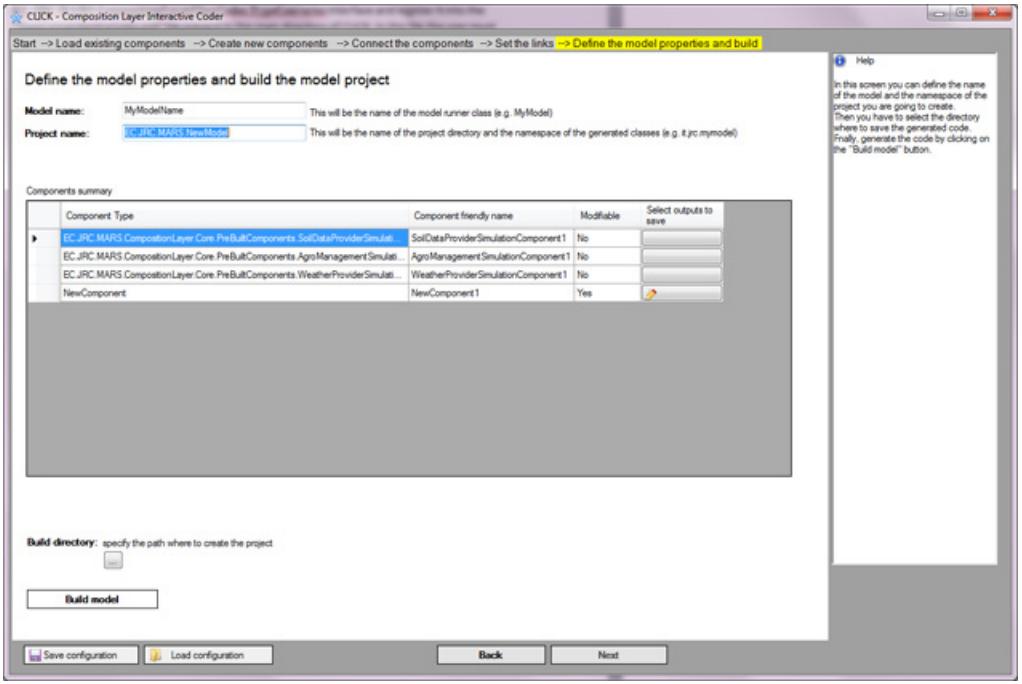
4 The **Set the links between the domain classes of the components** window will be displayed that allows you to view and/or edit the links.

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 6 Define the model properties and build page



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.

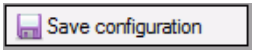
See also:

- “Loading existing components” on page 14

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:

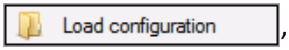
1 In any page of the application, click 

, 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