FMI_FMUexport_Rectifier

1 Test Goal

Within the scope of the new FMI cross-check rules vendors have to prove their tool's ability to export models as FMUs utilizing the FMI for model exchange which can be imported into other tools and/or checked with the ComplianceChecker. This document aims to deliver an easy test set-up based on basic Modelica examples to validate the code export function in SimulationX through export from and import into SimulationX and checking with the ComplianceChecker.

The export will be done for Win32 and Win64 operating systems. This test set will show the usage of FMU parameters and outputs.

2 Test Environment

SimulationX 3.5.707.4 x64

ComplianceChecker.win64 1.0.2

3 Test Steps

3.1 FMU Export

1. Open SimulationX.
2. Locate Modelica.Electrical.Analog.Examples.Rectifier and open in diagram view.
3. Open the parameterization dialogue of *SineVoltage2* and change the parameter *phase* from *"-2/3*Modelica.Constants.pi"* to *"0"*.
4. Repeat step 3 for the element *"SineVoltage3"*.
5. Save as .ism or .isx in a folder in *<path>.*
6. Make a new folder in *<path>* called *"Exports"*.
7. Start the code export wizard by clicking the code export button (see Figure 1).

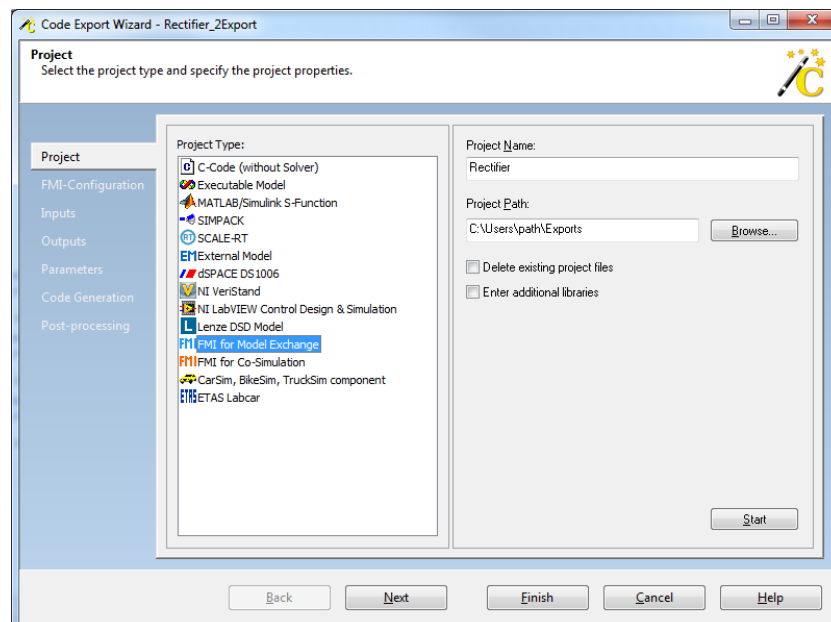**Figure 1: Code Export Button.**

Figure 2: First Page of Code Export Wizard.

8. The first page of the code export wizard will show three columns: An enumeration of different FMU parameters on the left, the type column in the middle (which specifies the parameters), and a column on the right which will show the final parameters.
Choose the project type *"FMI for Model Exchange"* in the middle column, assign the name

"Rectifier", and enter the project path "<path>\Exports" in the right column. Compare to Figure 2. Click "Next".

9. In the next step, check the boxes "Win32 Platform (x86)" and "Win64 Platform (x64)" in section "Platform Support". Click "Next".

10. In the Inputs property, the middle column lists elements with available inputs while the right column lists selected inputs. In this example, no inputs are available. Skip this step by hitting "Next".
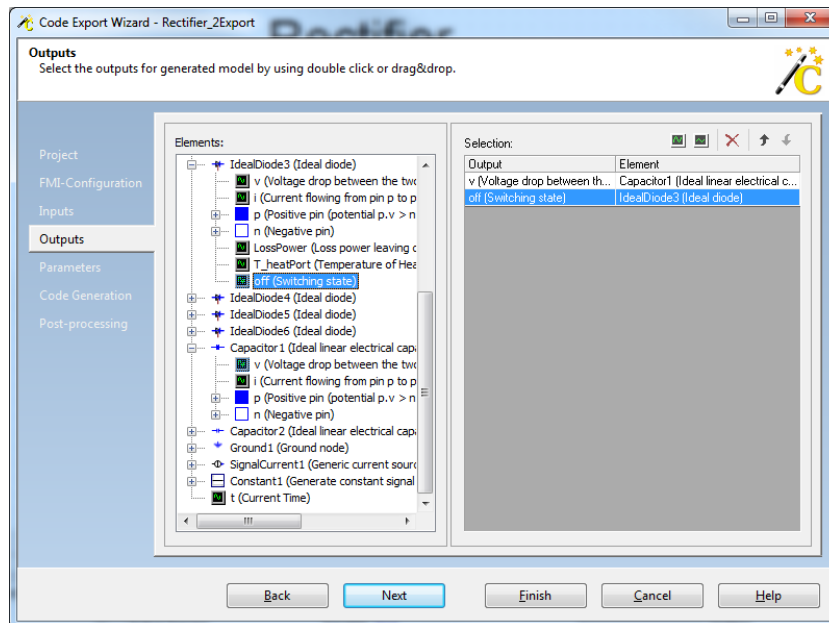

Figure 3: Outputs page of Code Export Wizard.

11. The output property window looks similar to the input property window and allows you to define outputs of the exported FMU. Choose the parameter "v" of the element "Capacitor1" and "off" of "IdealDiode3" (compare to Figure 3). Click "Next".

12. The parameter window allows you to define user-defined parameters of the FMU. As shown in Figure 4, choose the "phase" attribute of "SineVoltage1", "SineVoltage2", and "SineVoltage3". Hit "Next".
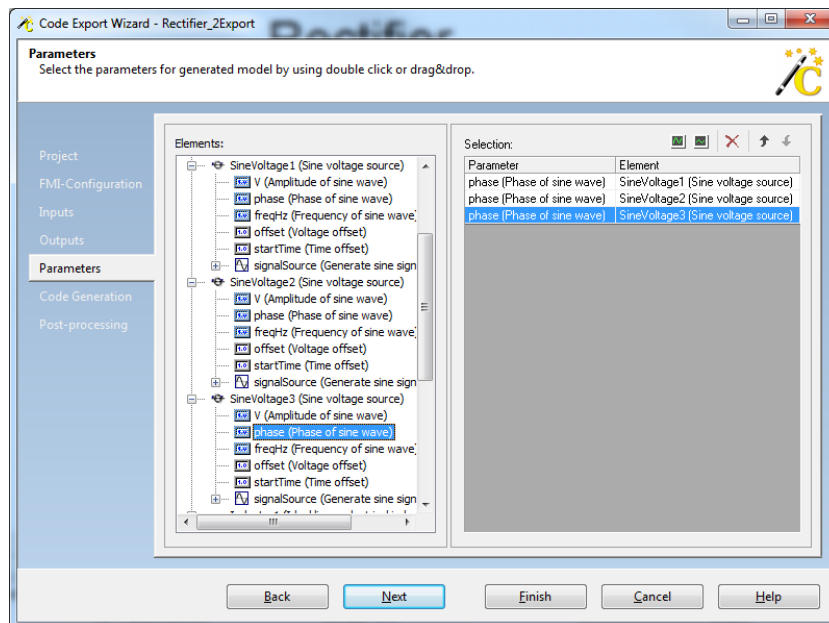

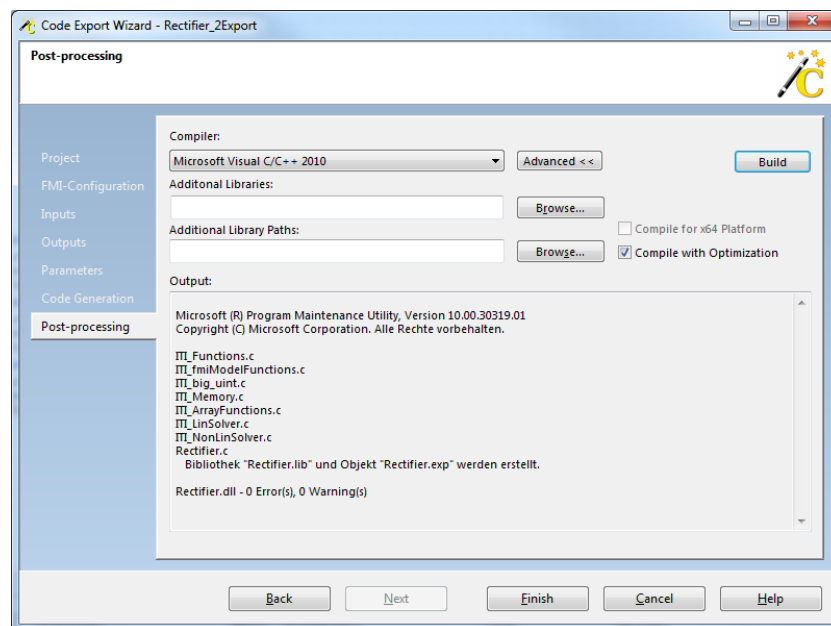**Figure 4: Properties page of Code Export Wizard.**

Figure 5: Post-Processing page of Code Export after build process has finished.

13. The next tab will invoke code generation. Here, a Global Symbolic Analysis will be performed and the correspondent code for the Model will be generated. You will see a status bar on top which shows progress. When the output shows *"Rectifier - 0 Error(s), 2 Warning(s), 0 Information"*, the code generation was successful. The warnings refer to undefined start attributes of *"inductor2.i"* and *"inductor3.i"* and can be neglected. Click *"Next"*. If it was not successful, go back to Step 7.
14. In the Post-Processing Tab, the actual FMU will be generated. Choose compiler *"Microsoft Visual C/C++ 2010"* and hit *"Built"*. The code will be generated and the output window will show the progress. Two build routines will run since the FMU is supposed to be compatible to both win64 and win32 platforms. After the build process has completed, the output according to Figure 5 will show up. Hit finish.

3.2   FMU Check with ComplianceChecker

15. Locate the ComplianceChecker in the system path *<ccpath>*.
16. Open a command shell in windows.
17. To start the ComplianceChecker and run a simulation of the model with a stop time of $0.1\,s$, a step width of $10\,\mu s$, and a log level of 5 (verbose messages), type the following in the shell. The ComplianceChecker log (log.txt) and the simulation results (result.csv) will be located in *"<path>"*.

```
<ccpath>\fmuCheck.win64 -e <path>\log.txt -o <path>\results.csv -s 0.1 -h
1e-6 -l 5 <path>\Exports\Rectifier.fmu
```

4   Expected Results

1. FMU Export should run without problems. The model is pre-tested and should not give errors neither in the Global Symbolic Analysis, nor the Code Generation, nor during the actual build process.
2. The ComplianceChecker should run through without warnings or errors from either the FMU or the checker.

Version	Date	Editor	Description
1.0	2013.04.22	Bastian Binder	Created