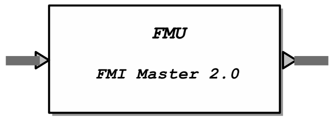
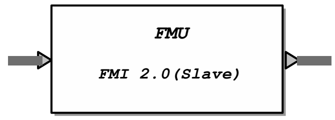
EMTP FMI Toolbox





Emmanuel RUTOVIC, Jerome CORNAU, 8/19/2016 11:49 AM

1 Description

The Functional Mock-Up Interface (FMI) is a tool-independent standard for model exchange and co-simulation. The first version, FMI 1.0, was published in 2010 [1], followed by FMI 2.0 in July 2014 [2].

EMTP is compatible with the FMI Standard for co-simulation as master (version 1.0 and version 2.0) and as slave (version 2.0 only). This document explains how to use the FMI Toolbox for EMTP.

2 Functional Mock-Up Interface (FMI)

The “Functional Mock-Up Interface” (FMI) is a standard initially developed by the European Consortium Modelisar. The standard specification is available for free on the official website https://www.fmi-standard.org/.

The aim of the FMI is to standardize the data exchange between different simulation tools. The motivations for such a standard are numerous:

- No need to manually convert models from one tool to another,
- No need to manually generate and compile code,
- Relatively easy approach to parallelize and distribute computations.

The FMI standard comes in two flavors:

- **FMI for Model Exchange**: FMU (Functional Mock-Up Unit, see Figure 1) contains only the model data and equations. Compatible tools can import FMUs and use their own solver.
- **FMI for Co-simulation**: FMU contains the model data and the simulation solver (or a link towards it), see Figure 2

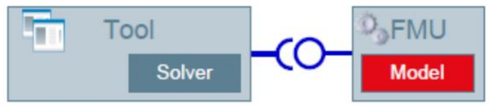


Figure 1: FMI for Model Exchange.

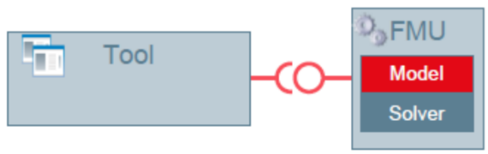


Figure 2: FMI for Co-Simulation.

Only control signals (integer, floating and Boolean) are exchanged.

Specifically, a FMU is a ZIP archive (with the “fmu” extension) containing:

- XML file (modelDescription.xml): for describing the model (including the number, the names and the types of inputs and outputs). The XML file format is standardized and documented in the FMI specifications.
- The model (C code or binaries): the model equations and data for the “FMI for Model Exchange” or the model data and the simulation solver (or a link towards it) for the ”FMI for Co-Simulation”.
- Other files required by the Slave tool.

A "Slave" tool that exports FMUs must generate a FMU that contains the files listed above. A “Master” tool that imports FMUs will have to read the FMU content and manage the exchange of data between the FMU and its calculation engine.

Tools can be compatible with one and/or the other variants of the standard and can import and/or export FMUs. The list of programs compatible with the FMI standard is available on the project website: https://www.fmi-standard.org/tools.

3 FMI Master

As explained in [2], the master program controls the data exchange between subsystems and the synchronization of all simulation solvers (slaves). With the “Master” device from the FMI library, one can load FMUs and use EMTP as FMI master program.

3.1 Load FMU

At first run, the user can select a new FMU file (see Figure 3). The FMU must be compatible with the FMI for co-simulation standard and must contain 32-bit Windows binaries.

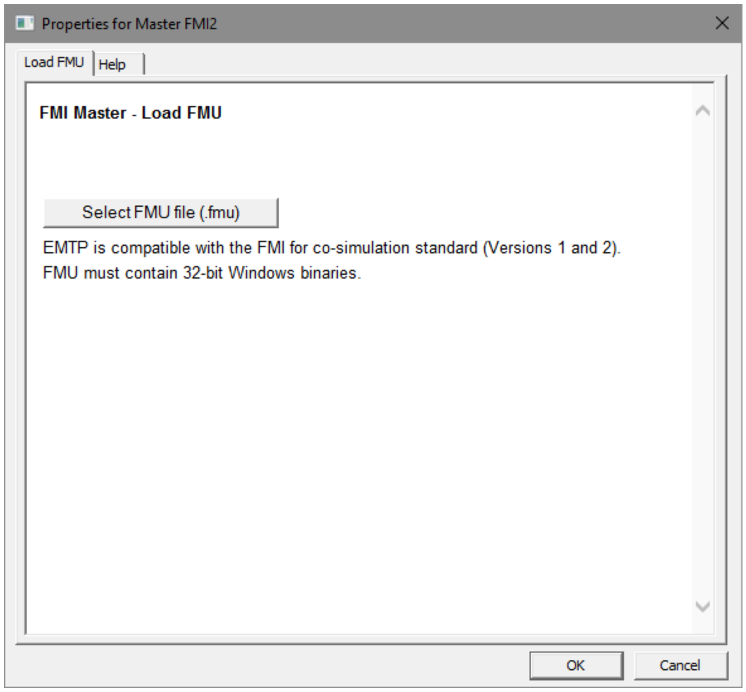


Figure 3: FMU Selection window.

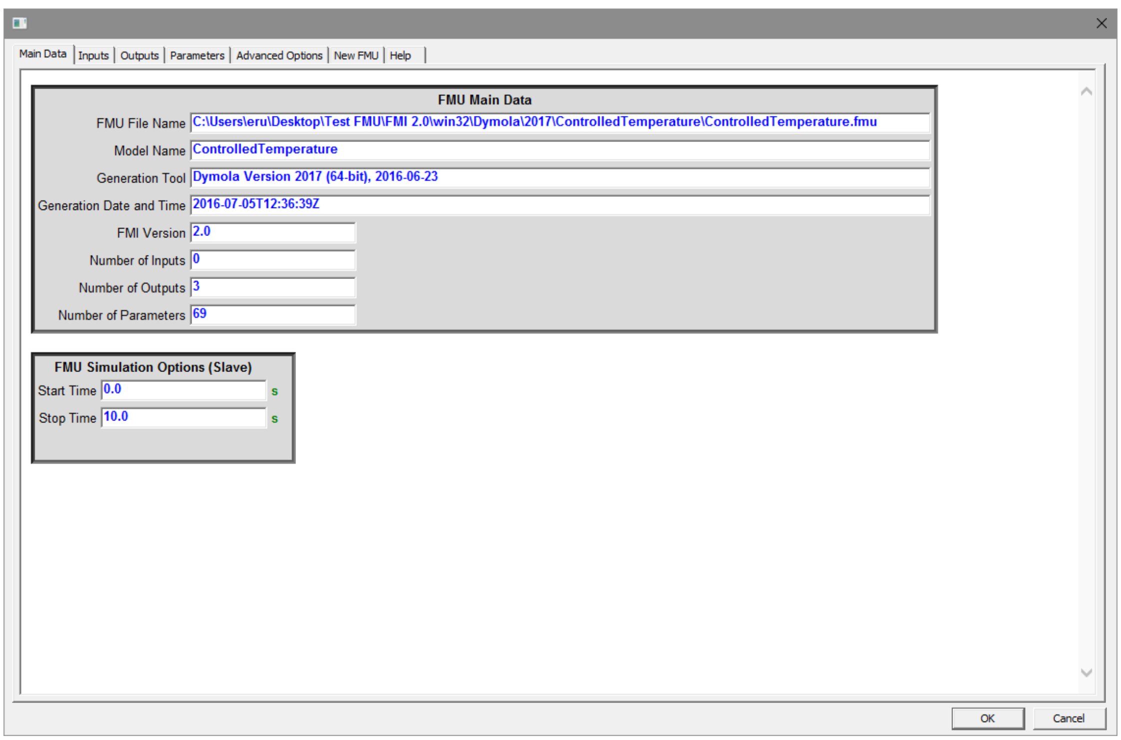


Figure 4: FMU Main Data tab.

Once the FMU is loaded, important FMU information is displayed in the "Main Data" tab (see Figure 4). One can load another FMU using the available option in the "New FMU" tab.

3.2 FMU Inputs and Outputs

The list of inputs and outputs is extracted from the FMU and detailed in the “Inputs” and “Outputs” tabs (see Figure 5). The Input and output bundles are automatically added to the device as shown in Figure 6.

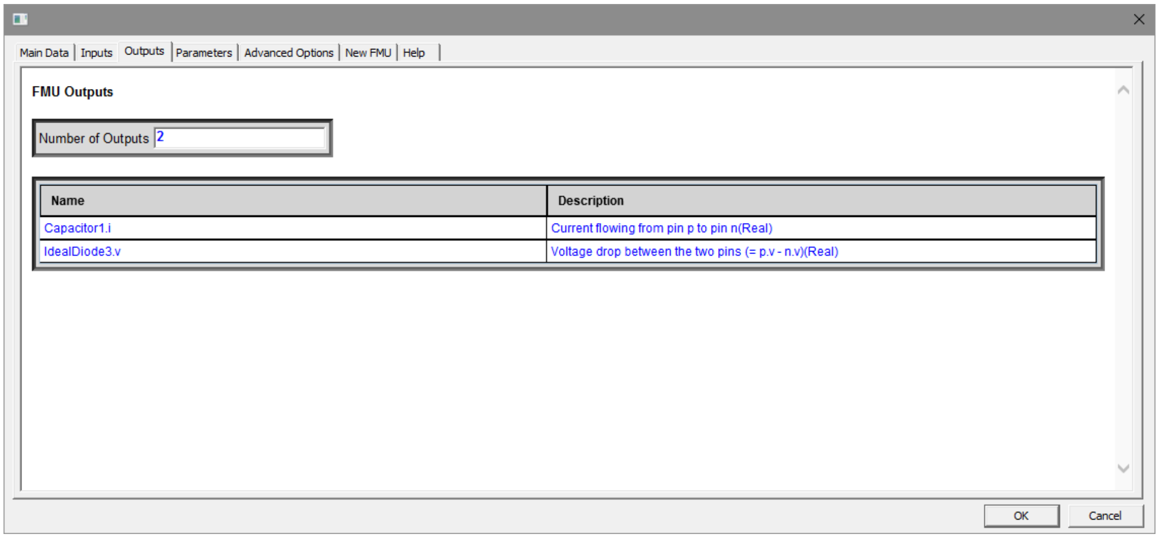


Figure 5: Example of Outputs tab.

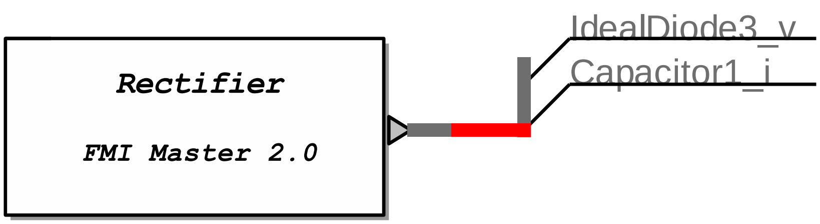


Figure 6: Example of FMU with zero input and two outputs

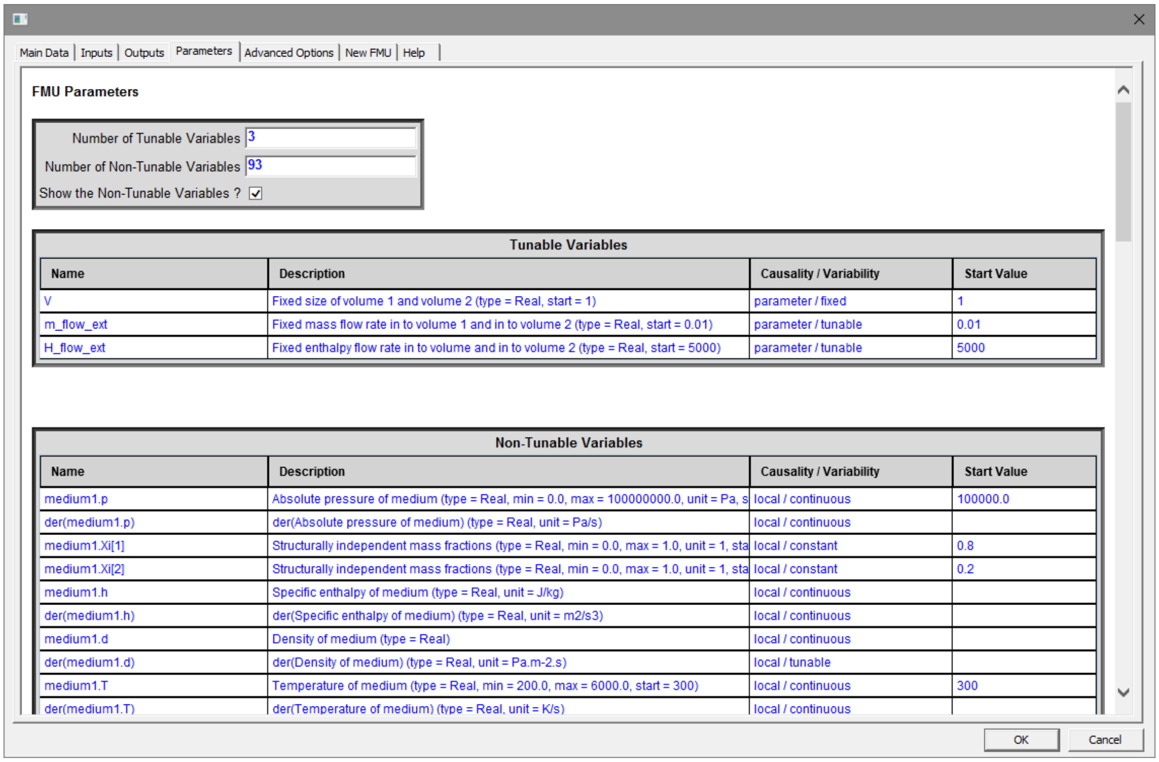
Four types of variable are handled by the FMI standard: String, Boolean, Integer, and Float. String variables are not processed by EMTP and will be set to “0.0” signals. For Booleans, the corresponding EMTP signal values become 1.0 for true and 0.0 for false.

3.3 FMU Parameters

The “Parameters” tab lists all the variables which are not inputs or outputs. The start value is the value of a variable before the first time step (initial condition). Parameters are classified in two categories (see Figure 7):

- Tunable Variables: with tunable start values,
- Non-Tunable Variables: with fixed start values.

Hence, the “start value” column of the “Parameters” tab can be modified by the user to set customized start values.



Main Data | Inputs | Outputs | Parameters | Advanced Options | New FMU | Help

FMU Parameters

Number of Tunable Variables: 3
Number of Non-Tunable Variables: 93
Show the Non-Tunable Variables ? ☑

Tunable Variables

Name	Description	Causality / Variability	Start Value
V	Fixed size of volume 1 and volume 2 (type = Real, start = 1)	parameter / fixed	1
m_flow_ext	Fixed mass flow rate in to volume 1 and in to volume 2 (type = Real, start = 0.01)	parameter / tunable	0.01
H_flow_ext	Fixed enthalpy flow rate in to volume and in to volume 2 (type = Real, start = 5000)	parameter / tunable	5000

Non-Tunable Variables

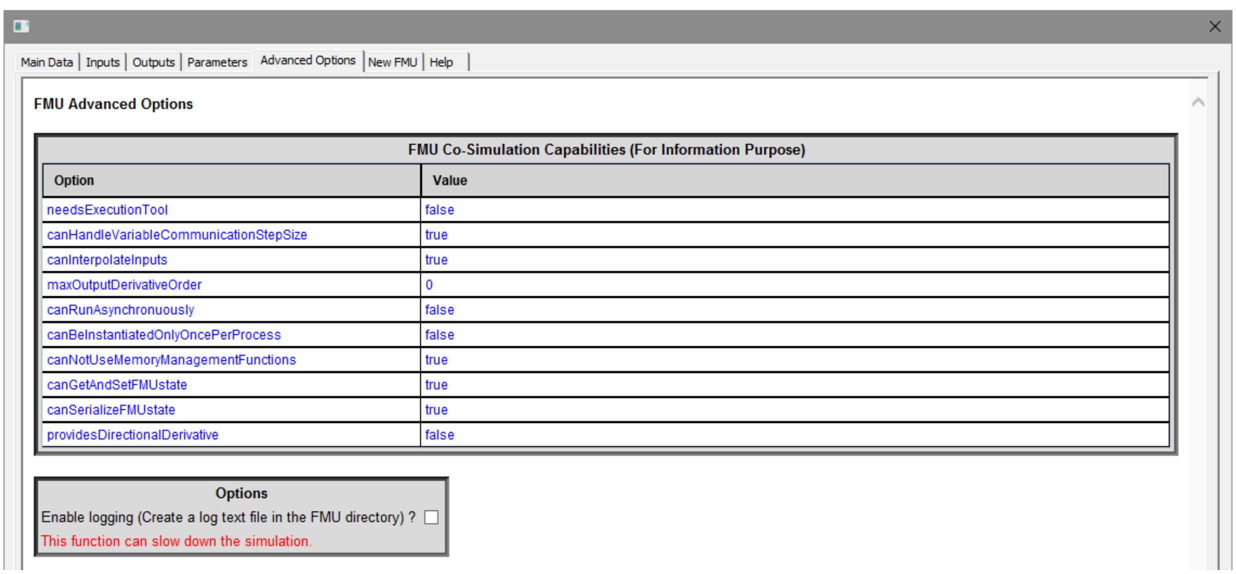
Name	Description	Causality / Variability	Start Value
medium1.p	Absolute pressure of medium (type = Real, min = 0.0, max = 100000000.0, unit = Pa, s	local / continuous	100000.0
der(medium1.p)	der(Absolute pressure of medium) (type = Real, unit = Pa/s)	local / continuous	
medium1.Xi[1]	Structurally independent mass fractions (type = Real, min = 0.0, max = 1.0, unit = 1, sta	local / constant	0.8
medium1.Xi[2]	Structurally independent mass fractions (type = Real, min = 0.0, max = 1.0, unit = 1, sta	local / constant	0.2
medium1.h	Specific enthalpy of medium (type = Real, unit = J/kg)	local / continuous	
der(medium1.h)	der(Specific enthalpy of medium) (type = Real, unit = m2/s3)	local / continuous	
medium1.d	Density of medium (type = Real)	local / continuous	
der(medium1.d)	der(Density of medium) (type = Real, unit = Pa.m-2.s)	local / tunable	
medium1.T	Temperature of medium (type = Real, min = 200.0, max = 6000.0, start = 300)	local / continuous	300
der(medium1.T)	der(Temperature of medium) (type = Real, unit = K/s)	local / continuous	

OK | Cancel

Figure 7: Tunable and Non-tunable variables.

3.4 FMU Advanced Options

This data tab is shown in Figure 8.



Main Data | Inputs | Outputs | Parameters | Advanced Options | New FMU | Help

FMU Advanced Options

FMU Co-Simulation Capabilities (For Information Purpose)

Option	Value
needsExecutionTool	false
canHandleVariableCommunicationStepSize	true
canInterpolateInputs	true
maxOutputDerivativeOrder	0
canRunAsynchronously	false
canBeInstantiatedOnlyOncePerProcess	false
canNotUseMemoryManagementFunctions	true
canGetAndSetFMUstate	true
canSerializeFMUstate	true
providesDirectionalDerivative	false

Options

Enable logging (Create a log text file in the FMU directory) ? ☐
This function can slow down the simulation.

Figure 8: Advanced options.

The co-simulation capabilities of the FMU are listed in the “Advanced Options” tab for information purposes. For compatible FMUs (V2 Only), it is possible to create a log file in the FMU directory.

3.5 Practical considerations regarding time steps

Only synchronous simulations are available at the moment. It means that the master and the slaves perform their calculation sequentially and not simultaneously.
It is preferred to use the same numerical integration time-step in the master (EMTP) and in the slaves. If different time steps are used, it is strongly recommended to use a fixed ratio between the master time-step and the slave time-step ($\Delta t_{MASTER} = k\ \Delta t_{FMU}$ or $\Delta t_{FMU} = k\ \Delta t_{MASTER}$, where k is a positive integer).

Since iterations in the numerical solution process are not currently part of the FMI, an artificial time-step delay is added between the master and the slaves.

3.6 Initialisation

As explained in [2] , the FMUs enter an inititialization mode before the simulation process. Values of FMU inputs at t=0 are sent to the FMU during the inititialization mode. Values of FMU Outputs at t=0 are read from the FMU when it exits the initialization mode (V2 only).

This feature can be used to correctly initialize a control circuit inside a FMU from the EMTP steady-state.

4 FMI Slave

The Slave device from the EMTP library can be used to generate FMUs from EMTP designs. EMTP FMUs do not include the EMTP engine, hence EMTP must be installed and activated on the machine of the master tool.

4.1 Inputs and Outputs

It is possible to set the number of inputs and outputs in the “Main Data” tab (see Figure 9). FMU inputs are the EMTP device outputs and vice versa.
An example of resulting device is shown in Figure 10.

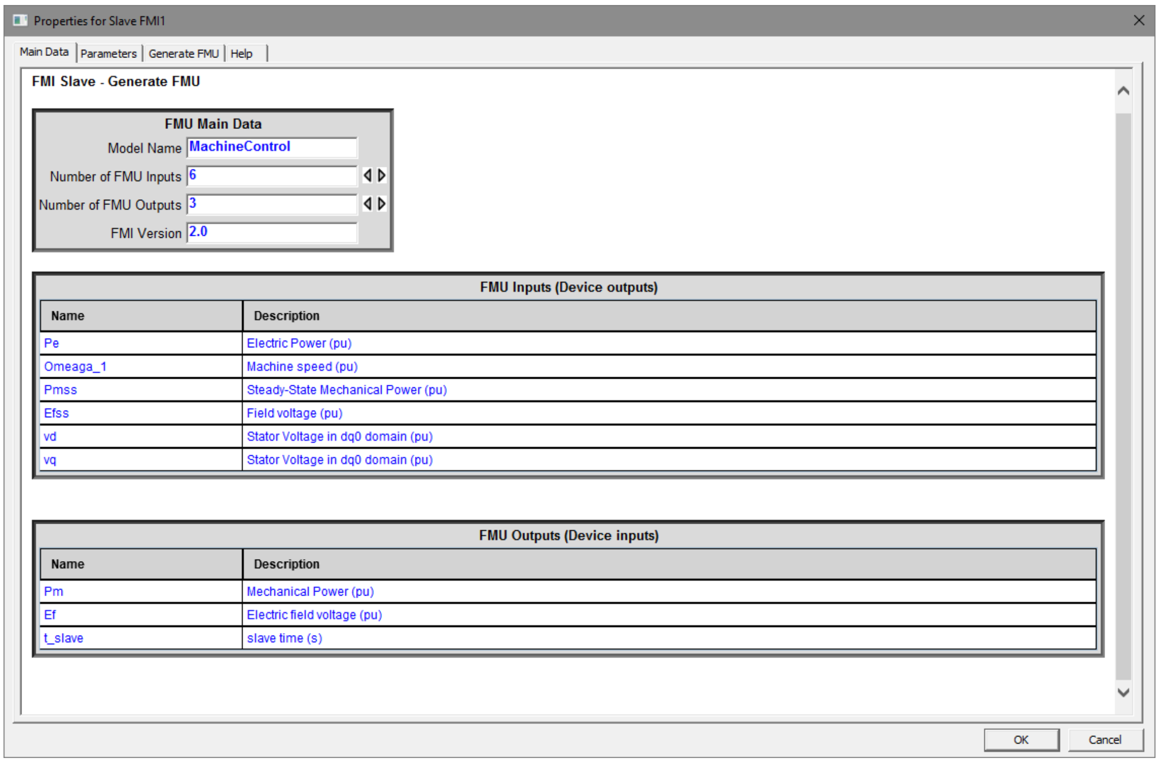


Figure 9: Main Data tab, definition of Inputs and Outputs.

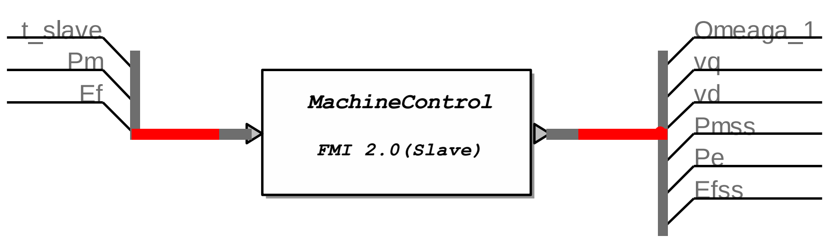


Figure 10: Example of FMI Slave device.

Custom description can be added to comment each variable.

4.2 Simulation options

EMTP FMU works only for time-domain solution. Hence, “Find time-domain solution” should be selected in the EMTP “Simulation options” menu.
The main time-step and the simulation time will be saved to the FMU. It is recommended that the master tool uses the same time-step and simulation time as the FMU.

4.3 Parameters

EMTP FMUs have four parameters (see Figure 11):

- **emtpopt.exe location**: This is the location of the emtpopt.exe file on the machine where the FMU is simulated.
- **emtpstate.ini location**: This is the location of the emtpstate.ini file on the machine where the FMU is simulated.
- **Co-simulation mode:** Two co-simulation modes are available:
    - **Mode 1 (Synchronous):** The master and the FMU perform their calculations sequentially.
    - **Mode 2 (Asynchronous, same time step):** The Master tool must use the same time-step as the FMU. An artificial time-step delay is added. The master and the FMU perform their calculations simultaneously.
- **Time-out (ms):** EMTP starts during the initialization mode of the master tool. It generally takes a few seconds and a time-out is allowed to elapse before an error is thrown. 5000 ms is generally sufficient, but a longer timeout might be required on some computers.

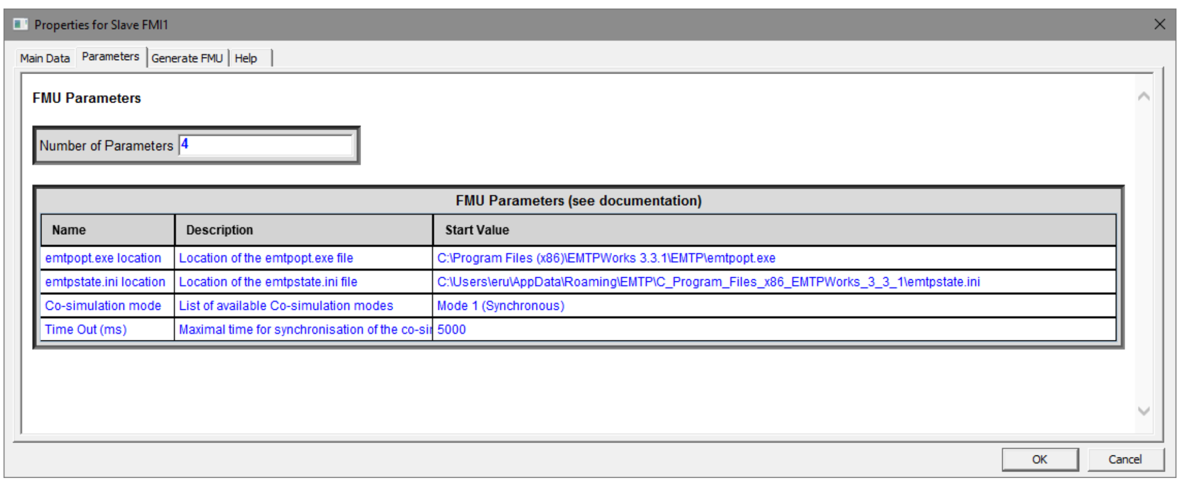


Figure 11: Parameters tab.

Parameters can be modified by the master tool. Default values are defined in the “Start Value” column of the “Parameters” tab in Figure 11.

The location of the emtpopt.exe and emtpstate.ini files are required by the master tool to start EMTP. Since the FMU might not be used on the computer used to create the FMU, it might be necessary to specify the location of those parameters manually.

The location of **emtpopt.exe (**EMTP computational engine**)** is located in “*ApplicationDir\EMTP\emtpopt.exe*”. “*ApplicationDir”* designates the application folder (path):

a. On a 64 bit Windows system the default value is C:\Program Files (x86)\EMTPWorks.
b. On a 32 bit Windows system the default value is C:\Program Files\EMTPWorks.

The file **emtpstate.ini** is located in "*ApplicationDataDir\emtpstate.ini*". "*ApplicationDataDir*" designates the application data folder (path):

a) On a Windows 7, 64-bit system, the typical default value is: C:\Users\ThisUser\AppData\Roaming\EMTP\C_Program_Files_x86_EMTPWorks
b) The following script command line can be run from a JavaScript window or code in EMTPWorks to determine application path: "*writeln(getAppDataDir);*" It will output the application data path in the Script Console window

4.4 FMU Generation

When the EMTP design is ready, one can generate the required FMU by using the option in the "Generate FMU" tab shown in Figure 12.

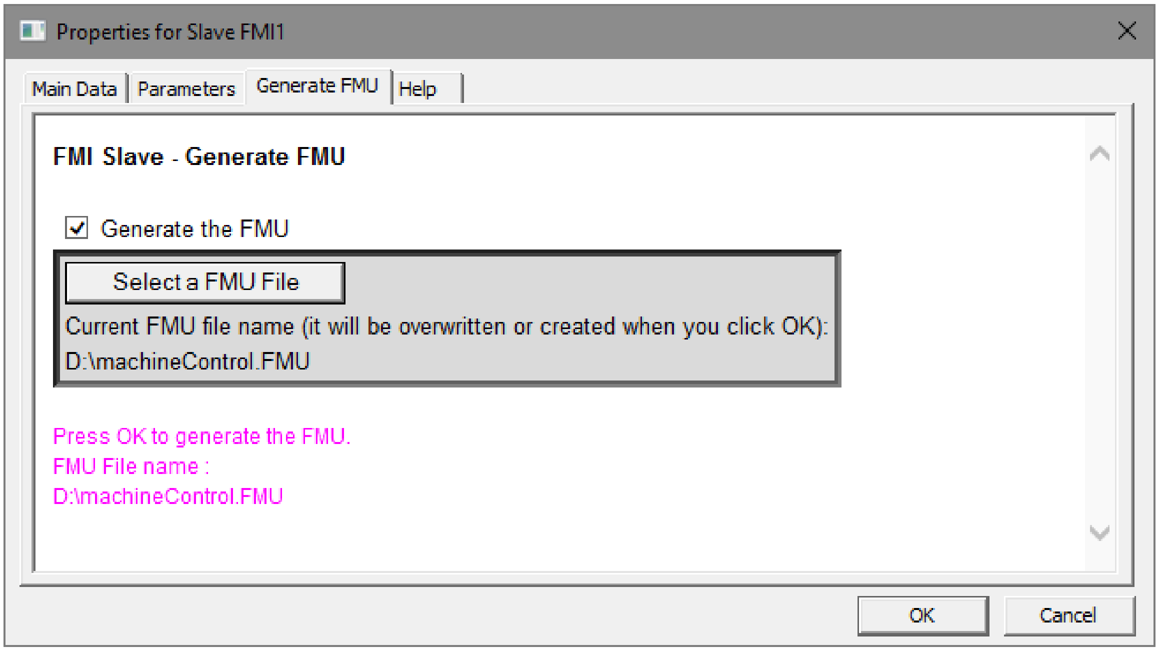


Figure 12: "Generate FMU" tab example.

The details of the generation process are listed in the EMTP Console. When the FMU is generated the following message is displayed.

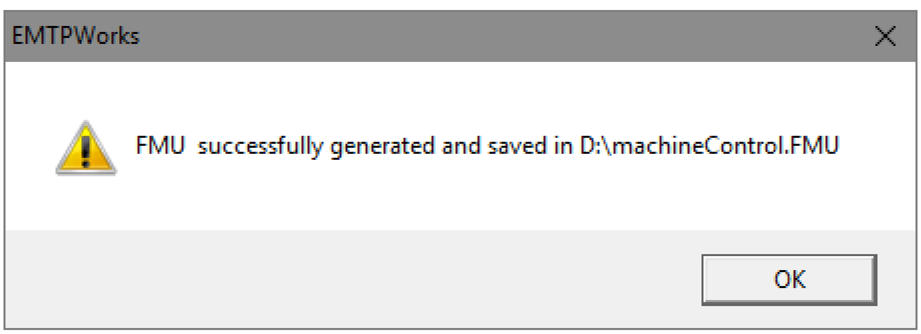


Figure 13: FMU Generated successfully.

The generated FMU can be used on any machine where EMTP is installed and activated.

4.5 Compatibility and limitations

EMTP FMUs are compatible with all the tools that implement the “FMI for Co-Simulation” standard V2.0.
EMTP FMUs passe the “FMU Compliance Checker” successfully. The FMU Compliance Checker is a free software provided by the Modelica Association, implemented by Modelon AB to check a given FMU's compliance with the FMI standard. The FMU Compliance Checker can be downloaded from this link: Compliance Checker.

5 References

[1] MODELISAR, "Functional Mock-up Interface for Co-Simulation - Version 1.0," 2010.

[2] Modelica Association Project “FMI”, "FMI for Model Exchange and Co-Simulation - Version 2.0," 2014.

6 Credits

The FMI Toolbox is based on the initial work done by Electricité de France (EDF) and uses third-party open-source components. You can find the source code of their open source projects along with license information below.

Functional Mock-up Interface

The project website is: www.fmi-standard.org

XMLWritter

XMLWriter - XML generator for Javascript, based on .NET's XMLTextWriter.

Date: 3/12/2008
version 1.0.0
@author Ariel Flesler
http://flesler.blogspot.com/2008/03/xmlwriter-for-javascript.html

MiniZip

miniz.c v1.15 - public domain deflate/inflate, zlib-subset, ZIP reading/writing/appending, PNG writing