Dymola

Dynamic Modeling Laboratory

Dymola Release Notes

Dymola 2016 FD01

The information in this document is subject to change without notice.

Document version: 1

Dassault Systèmes AB
Ideon Gateway
Scheelevägen 27 – Floor 9
SE-223 63 Lund
Sweden

Support: http://www.3ds.com/support
URL: http://www.Dymola.com
Phone: +46 46 270 67 00

Contents

1 Important notes on Dymola

Installation on Windows

To translate models on Windows, you must also install a supported compiler. The compiler is not distributed with Dymola. Note that administrator privileges are required for installation. Two types of compilers are supported on Windows in Dymola 2016 FD01:

Microsoft Visual Studio C++

This is the recommended compiler for professional users. Note that **free** Microsoft compiler versions earlier than Microsoft Visual Studio Express 2008 are not supported (concerning **full** versions, some earlier versions are supported). Refer to section “Compilers” on page 34 for more information.

GCC

Dymola 2016 FD01 has limited support for the MinGW GCC compiler, 32-bit and 64-bit. For more information about GCC, see section “Compilers” on page 34; the section about GCC compilers.

Installation on Linux

To translate models, Linux relies on a GCC compiler, which is usually part of the Linux distribution. Refer to section “Supported Linux versions and compilers” on page 35 for more information.

2 About this booklet

This booklet covers Dymola 2016 FD01. The disposition is similar to the one in Dymola User Manual Volume 1 and 2; the same main headings are being used (except for, e.g., Libraries and Documentation).

3 Dymola 2016 FD01

3.1 Introduction

3.1.1 Additions and improvements in Dymola

A number of improvements and additions have been implemented in Dymola 2016 FD01. In particular, Dymola 2016 FD01 provides:

- Improvements in the Documentation layer:
    - Improved support for style sheets (page 8).
- Improved plot handling (page 16).
    - Drag and drop from variable browser to plot windows / table windows.
    - Drag and drop between plot windows and table windows.
- Post-processing after simulation (page 14).
- Support for Scientific Data Format (.sdf) (page 16).
- Setting of FMI options available also when exporting and importing FMUs (page 19).
- Extended FMI support in Simulink (page 21):
    - FMU Export from Simulink:
        - Added functionality, for example:
            - Loading of binary MEX S-functions
            - Structured naming for block hierarchy and I/O buses.
            - Black-box FMU generation
    - FMU Import into Simulink

3.1.2 New and updated libraries

New libraries

The following libraries are new:

- Battery Library, version 1.2.
- Claytex Library, version 2015.3
- Engines Library, version 2015.3.

For more information about the new libraries, please see the section “New libraries” starting on page 23.

Updated libraries

The following libraries have been updated:

- Air Conditioning Library, version 1.11.
- Electric Power Library, version 2.2.2.
- Engine Dynamics Library, version 1.2.4.
- Fuel Cell Library, version 1.3.2.
- Heat Exchanger Library, version 1.4.
- Human Comfort Library, version 2.0.0.
- Hydraulics Library, version 4.3.
- Hydro Power Library, version 2.5.
- Liquid Cooling Library, version 1.4.1.
- Model Management Library, version 1.1.5.
- Modelica_DeviceDrivers, version 1.4.3.
- Modelica_LinearSystems2, version 2.3.3.
- Pneumatics Library, version 1.8.
- Power Train Library, version 2.3.1.
- Thermal Power Library, version 1.11.
- Vapor Cycle Library, version 1.2.2.
- Vehicle Dynamics Library, version 2.2.
- Vehicle Interfaces Library, version 1.2.3.

For more information about the updated libraries, please see the section “Updated libraries” starting on page 25.

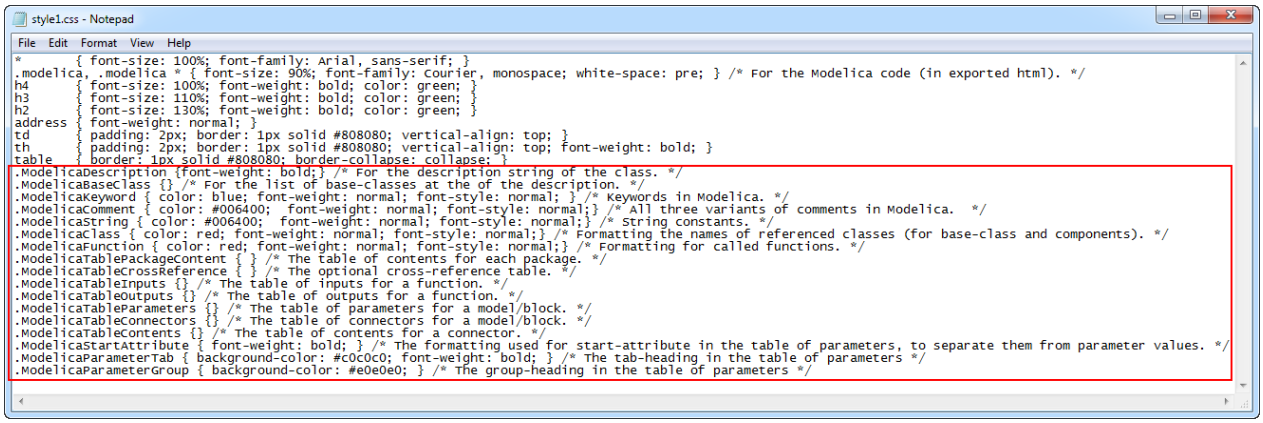
3.2 Developing a model

3.2.1 Improved documentation

Improved support for configuring the style of the documentation of Modelica packages

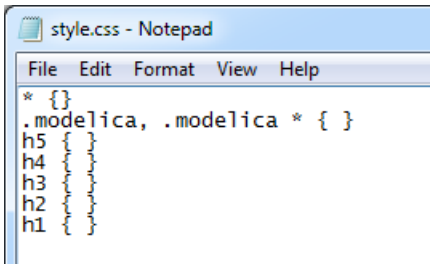
The documentation can be configured using cascading style sheets: there was already limited support for this in previous versions.

The default style sheet is `style1.css`, located in the `insert` folder in the Dymola distribution. This style sheet can be used as a reference, since it lists all the tags. Below the content of the style sheet, with new items added in this version marked.



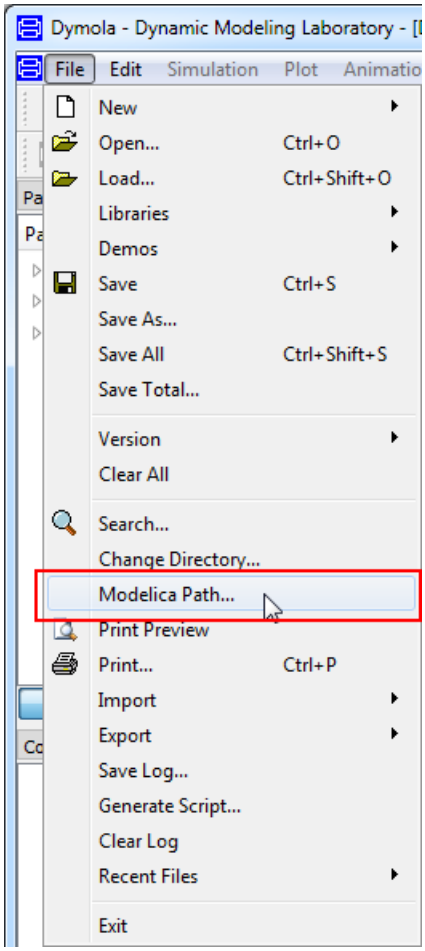
If a library has a style sheet `style.css` located in the `Resources` folder of the library, it will be cascaded. This allows for overriding specific tags.

Note: There is currently a limitation when creating a user-defined style sheet; headers must be entered in a descending order to work, if e.g. adding a header 5, the order must be (in a minimal style sheet just to show the order):

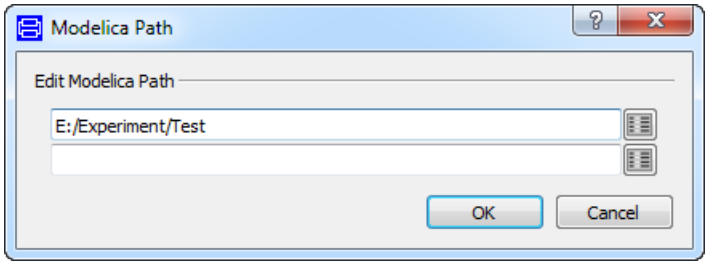


3.2.2 Managing Modelica path setting from Dymola GUI

In Dymola 2016 FD01, it is possible to manage the Modelica path (MODELICAPATH) by the command **File > Modelica Path…**.

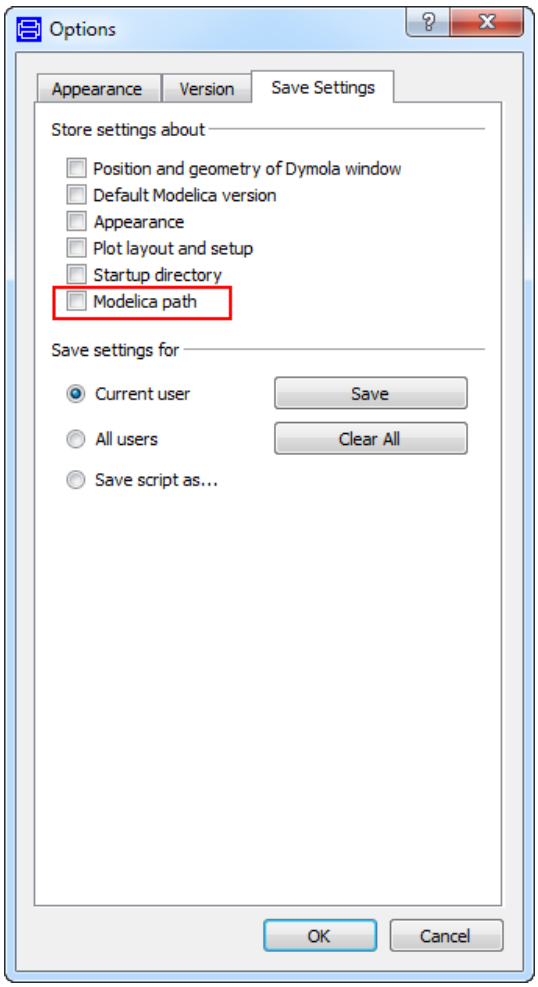


The command **Modelica Path…** opens a dialog the makes it possible to change, add, and delete directories in the MODELICAPATH environment variable from inside Dymola. (MODELICAPATH specifies a semicolon separated list of directories where packages will be searched for when e.g. opening a library or a model.)



Already present directories are listed, except the default directory dymola/Modelica/Library. You can browse for directories in the dialog.

To keep the change of Modelica path between sessions, this function has to be added in setup.mos. This can be done by ticking the setting **Modelica path** in **Edit > Options…**, the **Save Setting** tab, and then click **Save**:



3.2.3 Minor improvements

Flipped images in diagram

The handling of images in diagrams now supports creating flipped images. The changes are:

- Regardless of how you create the image the coordinates will be created in the same non-flipped way.
- You can flip the images using the **Edit** menu.

Existing libraries can be updated to ensure that all images are handled correctly in this respect by using the command `updateModelicaAnnotations("`*`LibraryName`*`")` where *`LibraryName`* is the name of the library to be updated. The library must be loaded before giving this command.

All libraries delivered in Dymola 2016 FD01 are updated by this command.

Support for SVG images

You can now use SVG images in diagram layer and icon layer.

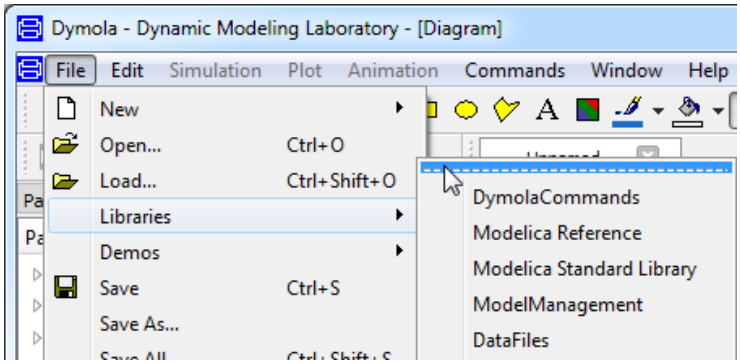
Improved handling of UniCode (UTF-8) versus iso-latin-1coding

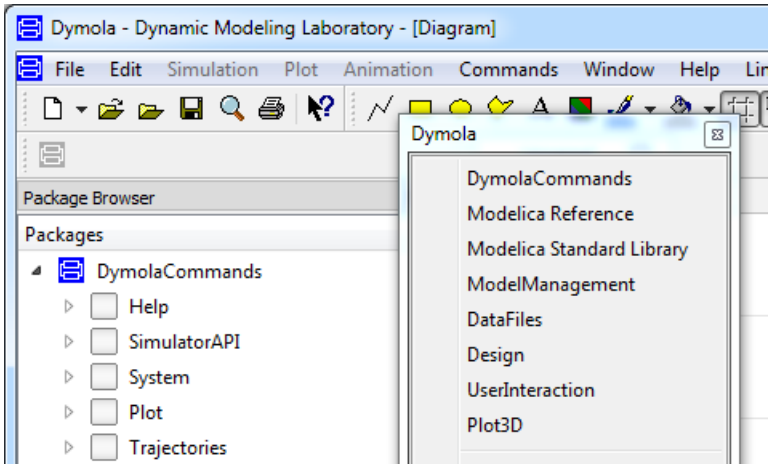
A flag `Advanced.PreferWritingLatin1` is now available to set what coding to use. The default value of the flag is true, meaning that iso-latin-1 will be preferred instead of UTF-8 when writing files.

However, if any non iso-latin-1 character is present, UTF-8 will always be used.

Easier opening of several libraries using a tear-off menu

In Dymola 2016 FD the **File > Libraries** menu is a tear-off menu; if you click the dotted tear-off line in the top of the menu, the menu is detached as a separate window.



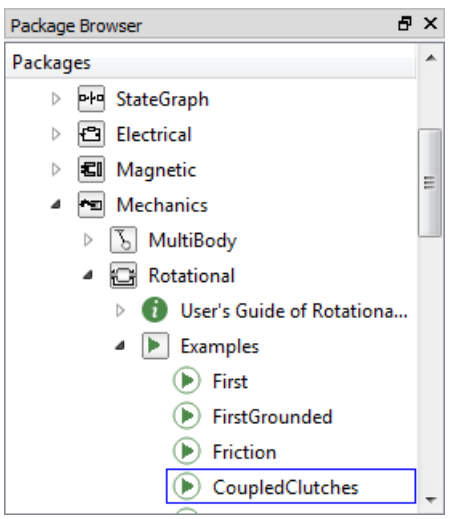


You can use this new window to open several libraries without having to open the menu again. When you close this window, you also restore the menu – the next time you use **File > Libraries** it will look like the first image above.

Improved user interface experience

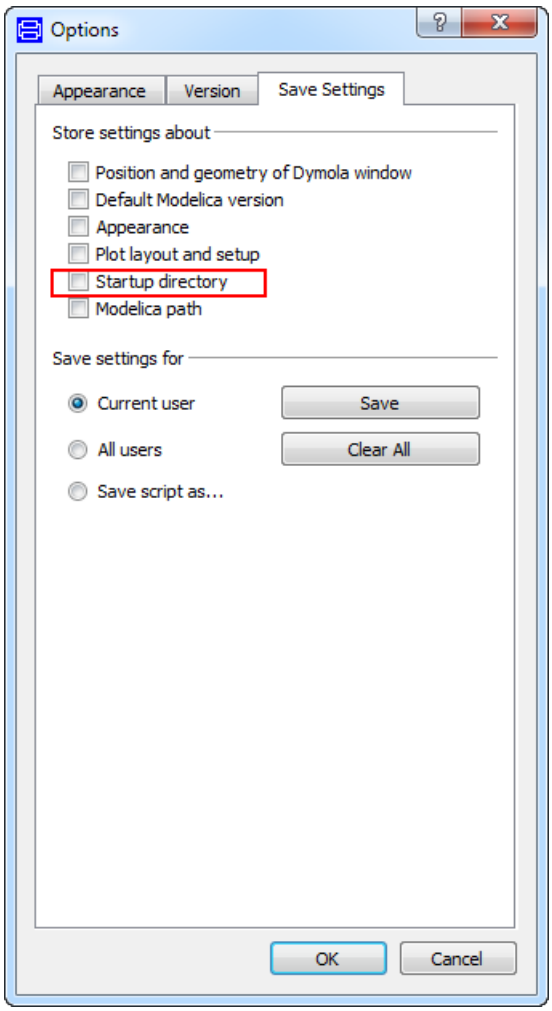
The user interface framework has been upgraded, resulting in a more up-to-date look and feel of the user interface.

As an example, the display of the node structure in the package browser and component browser has been changed (previously "+" and "-" were used to indicate expandable or collapsible nodes, and there were a number of lines in the structure). An example:



Saving the current directory as startup directory by GUI

In Dymola 2016 FD01, the current directory can be saved as startup directory by the command **Edit > Options...**, in the **Save Settings** tab, by ticking the option **Startup directory**, and selecting **Save**.



3.3 Simulating a model

3.3.1 Post-processing after simulation

Dymola has the capability to automatically run an external post-processing command after finished simulation. One or more such commands can be selected from a list in **Simulation > Setup...**, the **Output** tab.

Setting up post-processing commands

Available post-processing commands are defined in Dymola using a script function, for example:

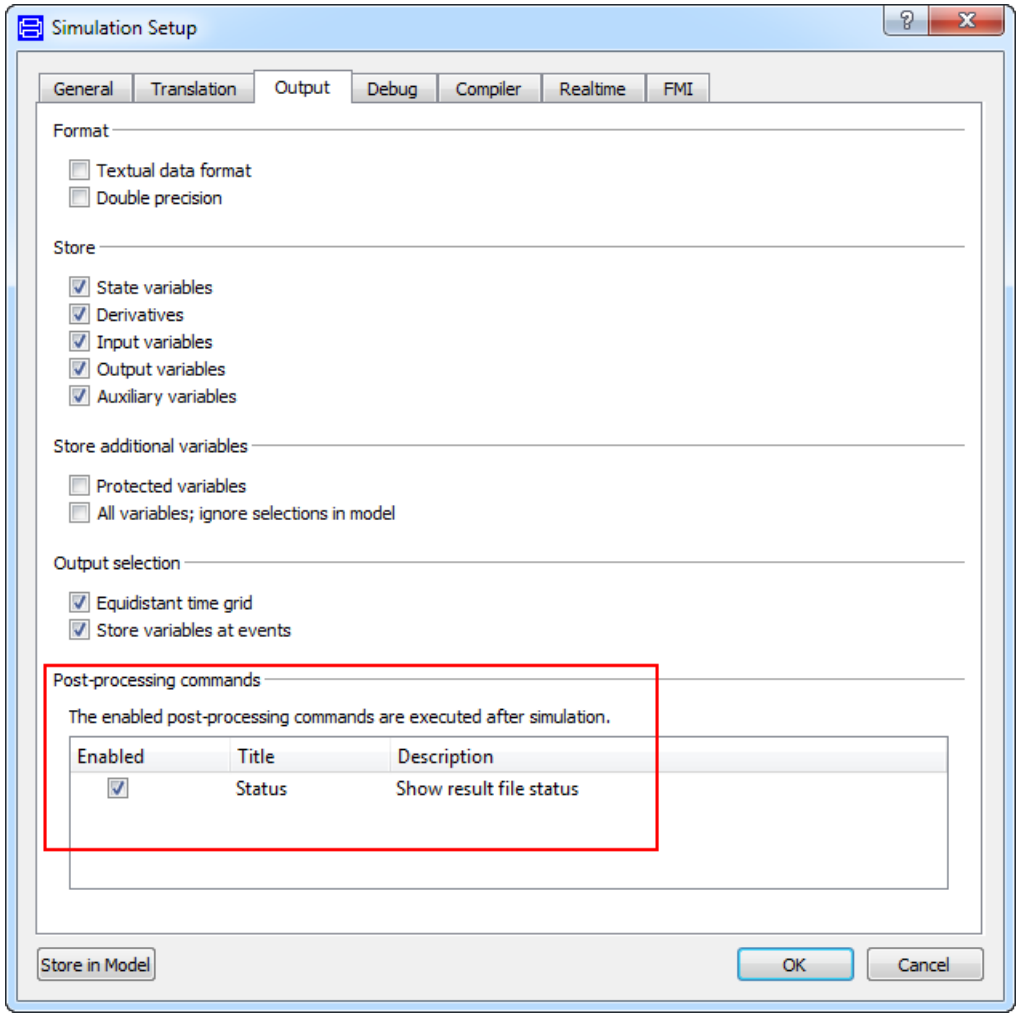
```
definePostProcessing("Status", "Show result file status",
    "Modelica.Utilities.System.command(
     \"dir %RESULTFILE%.* & pause\")", true);
```

The arguments are a short name for identifying the command, a longer description string, a command string, and lastly a flag which says if the command should be enabled by default. The command string supports some very simple macro expansion, where `%RESULTFILE%` is expanded to the name of the simulation result file without extension (see below for an example of the expansion).

The command is executed by Dymola. To execute a command in the standard command shell of the operating system, `Modelica.Utilities.System.command()` can be used.

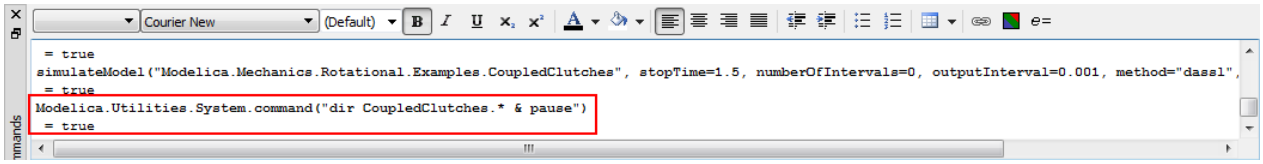
Selecting post-processing commands

Which of the available post-processing commands are executed is selected marking the appropriate checkboxes in **Simulation > Setup...**, the **Output** tab. The commands will be executed in the order listed.

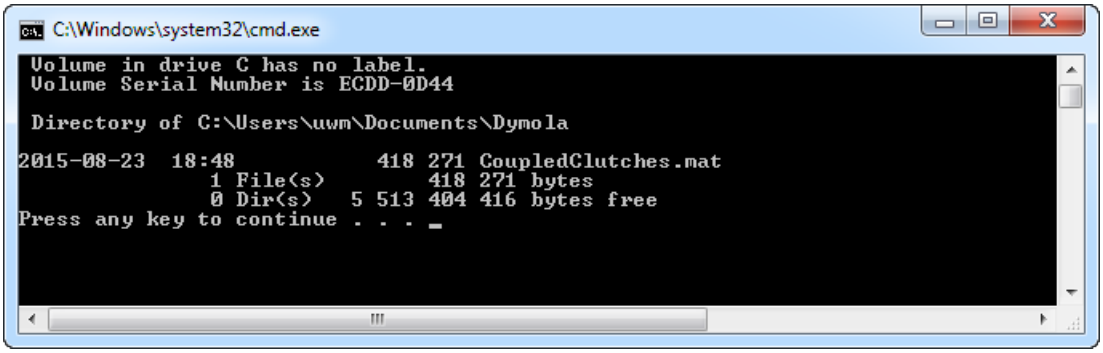


Execution

In the example above, we defined a "dir" command (on Windows) displaying the size of the result file. This command is automatically executed after the user runs the simulation:



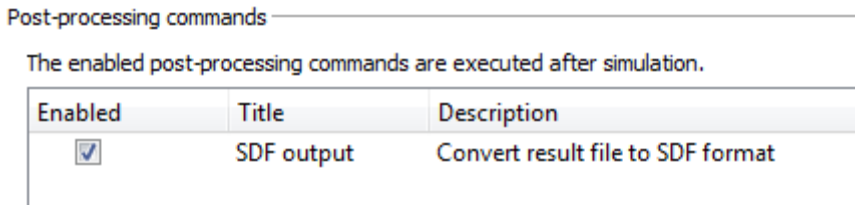
The "pause" command ensures that the command window stays visible:



3.3.2 Support for Scientific Data Format

The Scientific Data Format (SDF) is a file format for storing table data and simulation results, using a data model based on HDF5 (https://www.hdfgroup.org/HDF5). Such files can be processed by several tools supporting HDF5.

Dymola 2016 FD01 supports copying of the simulation result file to SDF using a pre-defined post-processing command (see previous section). The new file has the same name as the result file, but with extension .sdf.



Utility functions for loading and saving SDF files in Matlab are available in the Matlab package SDF provided by the Dymola distribution. See `Program Files (x86)\Dymola 2016 FD01\Mfiles\+SDF\load.m` for documentation.

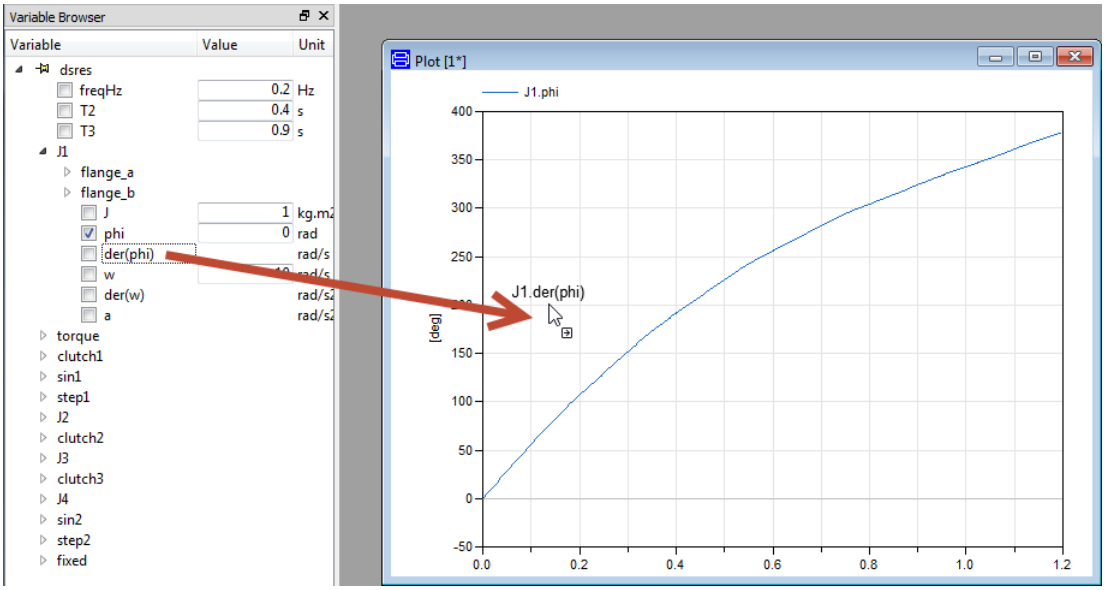
3.3.3 Improved multicore support

A number of previous limitations have been removed. The result is that now also FMUs are supported, both with Visual Studio and MinGW GCC compilers, on Windows.

3.3.4 Plot window

Drag and drop of variables from the Variable Browser to plot windows and table windows

In Dymola 2016 FD01 you can drag and drop variables from the variable browser to plot windows or table window to create curves/tables.



If you drop on the background, a new plot window is opened where the curve is added.

If you drop in a text window, the signal path is inserted.

Dragging of curves between plots and tables

You can drag curves between plot windows and table windows. When dragging from table windows, drag the header in the left column.

You can copy by keeping **Ctrl** pressed while performing the dragging.

Copying curves between plots/diagrams by dragging

Already in previous version curves could be moved between plots or diagrams by dragging the corresponding legends.

Curves can now be copied as well by keeping **Ctrl** pressed when performing the dragging of the corresponding legends.

3.3.5 Minor improvements

Preventing multiple C code files when translating

On Windows, you can set the following flag to prevent splitting of the generated C code (dsmodel.c) across multiple files:

```
Advanced.SeparateFilesCcode=false;
```

The default value of the flag is true.

Note that some compilers may have problems with compiling large files.

3.4 Installation

For the current list of hardware and software requirements, please see chapter "Appendix – Installation: Hardware and Software Requirements" starting on page 34.

3.5 Other Simulation Environments

3.5.1 Dymola – Matlab interface

Compatibility

The Dymola – Simulink interface now supports Matlab releases from R2010a (ver. 7.10) up to R2015a (ver. 8.5). Only Visual Studio C++ compilers are supported to generate the DymolaBlock S-function. The LCC compiler is not supported.

3.5.2 Real-time simulation

Compatibility – dSPACE

Dymola 2016 FD01 generated code has been verified for compatibility with the following combinations of dSPACE and Matlab releases.

dSPACE DS1005 and DS1006 platforms

- dSPACE Release 6.6 with Matlab R2010a
- dSPACE Release 7.0 with Matlab R2010bSP2
- dSPACE Release 7.1 with Matlab R2011a
- dSPACE Release 7.2 with Matlab R2011b
- dSPACE Release 7.3 with Matlab R2012a
- dSPACE Release 7.4 with Matlab R2012b
- dSPACE Release 2013-A with Matlab R2013a
- dSPACE Release 2013-B with Matlab R2013b
- dSPACE Release 2014-A with Matlab R2014a
- dSPACE Release 2014-B with Matlab R2014a, and R2014b
- dSPACE Release 2015-A with Matlab R2014a, R2014b, and R2015a

SCALEXIO

- dSPACE Release 2013-A with Matlab R2013a
- dSPACE Release 2013-B with Matlab R2013b
- dSPACE Release 2014-A with Matlab R2014a
- dSPACE Release 2014-B with Matlab R2014a, and R2014b

- dSPACE Release 2015-A with Matlab R2014a, R2014b, and R2015a

The selection of supported dSPACE releases focuses on releases that introduce support for a new Matlab release and dSPACE releases that introduce a new version of a cross-compiler tool. In addition, Dymola always support the three latest dSPACE releases with the three latest Matlab releases. Although not officially supported, it is likely that other combinations should work as well.

Compatibility – xPC Target

Compatibility with Matlab xPC Target has been verified for all Matlab releases that are supported by the Dymola – Simulink interface, which means R2010a (xPC Target ver. 4.3) to R2015a (Simulink Real-Time ver. 6.2). Only Microsoft Visual C compilers have been tested.

3.5.3 FMI Support in Dymola

Unless otherwise stated, features are available both for FMI version 1.0 and version 2.0.

Improved unit handling for FMU import

FMI version 2.0 supports unit handling where an FMU exporter can define any unit for inputs and outputs as long as conversion to base units according to the FMI standard is available. This allows for proper unit checking for inputs and outputs between FMUs.

Dymola 2016 FD01 supports this; units are automatically converted to base units for inputs and outputs of imported FMUs. Such unit handling for parameters in FMUs is also supported.

The unit conversion can be disabled by setting the flag

```
Advanced.FMI.DoNotDeclareUnits = true;
```

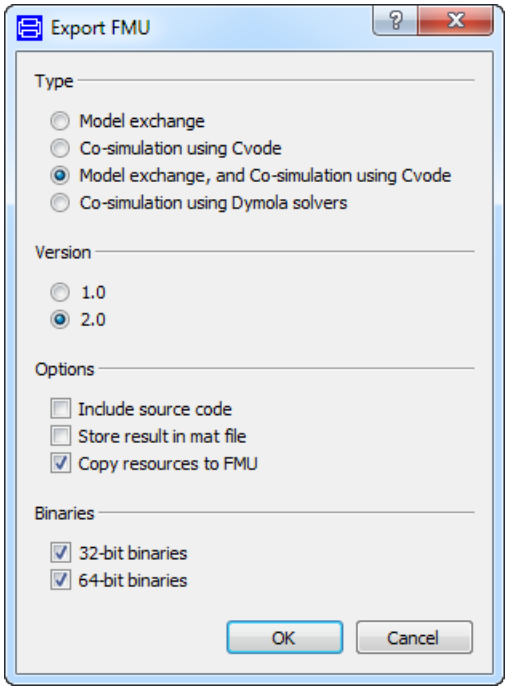
Setting this flag means ignoring the unit declarations completely. The flag is by default false.

Setting of FMI options available also when exporting and importing FMUs

Setting of FMI options (the FMI tab in the simulation setup) is now also available when importing and exporting FMUs.

Exporting FMUs

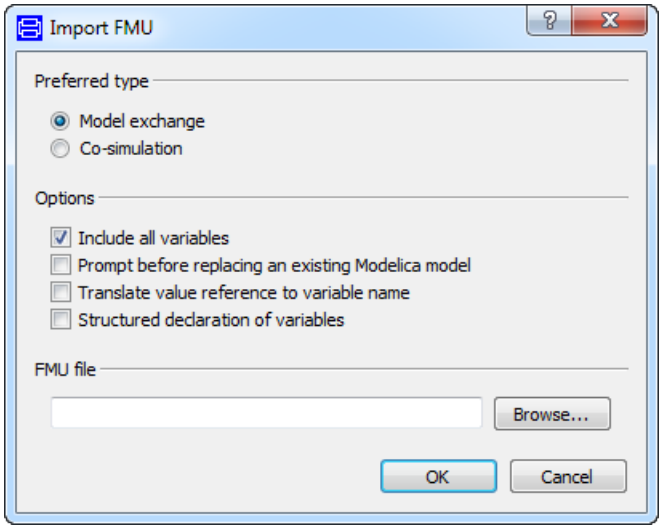
When exporting an FMU by using the command **Simulation > Translate > FMU**, the following menu appears:



The menu corresponds to the Export part of the FMI tab of the simulation settings.

Importing FMUs

When importing an FMU by using the command **File > Import > FMU...**, or by dragging an FMU into the Dymola window, the following menu appears:



When dragging an FMU, the FMU file path is prefilled in the menu.

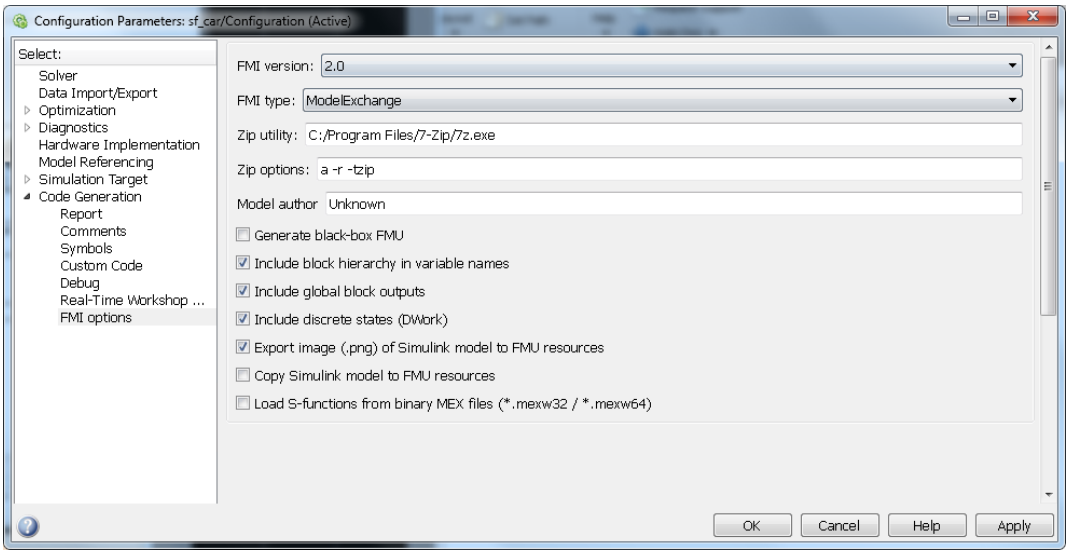
The menu corresponds to the Import part of the FMI tab of the simulation settings.

Enhanced FMI support in Simulink

FMU export from Simulink

A new and enhanced version (2.1.2) of the Dassault Systèmes FMU Export from Simulink package is available. The new supported features are:

- Loading of binary MEX S-functions
- C++ source S-functions
- Simulink I/O buses with structured naming
- Black-box FMU generation
- Block hierarchy in variable names
- Support for Matlab R2015a and R2015b



The FMU Export from Simulink package is located in the Mfiles folder of the Dymola 2016 FD01 distribution as rtwsfcnfmi.zip. See the README file in the package or consult the Dymola manual for full installation and usage instructions.

The package can be used for free without any license key.

Support and maintenance is offered to Dymola customers through the regular support channel at www.3ds.com/support.

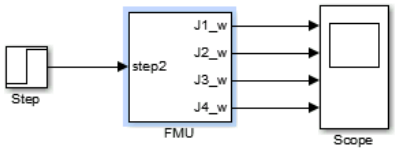
The package is independent of Dymola and updates are sometimes released in between the official Dymola releases. Information about new released versions can be found at www.dymola.com/FMI.

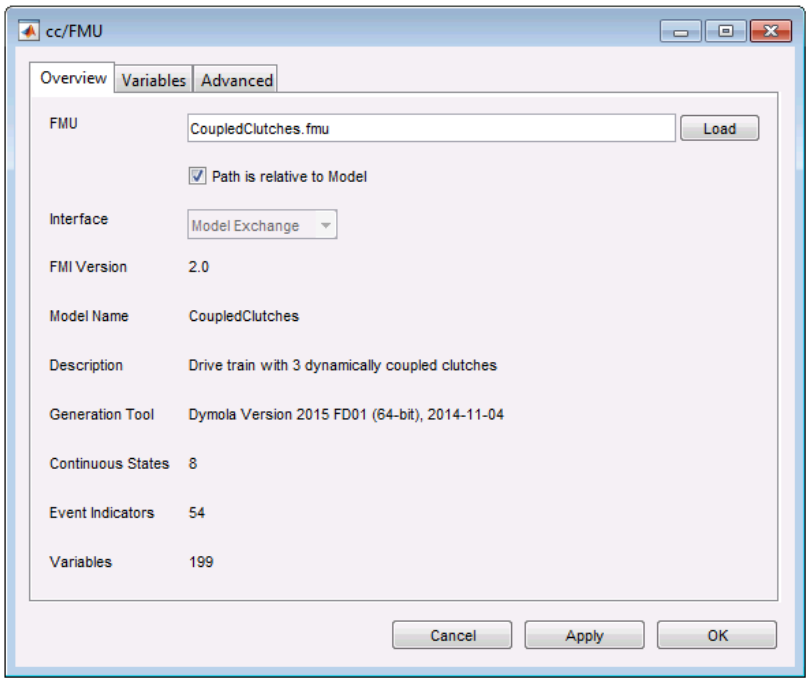
FMU import into Simulink

Dymola 2016 FD01 introduces *FMI Kit for Simulink*, which allows you to import and embed FMUs into your Simulink models. It supports co-simulation and model exchange (FMI 1.0 & 2.0) and works with both 32-bit and 64-bit Matlab. Supported Matlab releases are R2010a to R2015a.

Import of Dymola source code FMUs is supported if exported with any of the following releases; Dymola 2015 FD01 (2014-12-15), Dymola 2016 (2015-04-15), or Dymola 2016 FD01. Import of source code FMUs enables the use of the following simulation targets: Radid Accelerator, RSIM, GRT, and dSPACE ds1005 and ds1006.

FMI Kit for Simulink is located in the `Mfiles\FMIKit_for_Simulink` folder of the Dymola 2016 FD01 distribution. Add this folder to your Matlab path to get started. See the README file for further usage instructions.





The package can be used for free without any license key.

Support and maintenance is offered to Dymola customers through the regular support channel at www.3ds.com/support.

3.6 Modelica Standard Library and Modelica Language Specification

The current version of the Modelica Standard Library is version 3.2.1. The current version of the Modelica Language Specification is 3.3, Revision 1.

Note that the Modelica Standard Library is compliant with the Modelica Language Specification 3.3, Revision 1, which is the most up-to-date definition at the time of this Dymola release. (So this version of the Modelica Language Specification should be used for reference, and not the Modelica Language Specification 3.2, Revision 2.)

3.7 New libraries

Below is a short description of new libraries. For a full description, please refer to the libraries documentation.

3.7.1 Battery Library

The Battery Library contains components for modeling of cells and battery packs in a wide range of applications, including automotive, aerospace, industrial equipment, and process industry.

The Battery Library is tailored to model cells and battery packs for a broad range of applications. The library can be used to model a variety of different cell types and aid in dimensioning, battery system performance studies under varying temperatures, ageing studies, as well as control system development and evaluation.

The basis for the Battery Library is a customizable equivalent circuit approach that can be used for either function based or table-based cell models. Preparation of measurement data for the table-based models can be easily done by using the provided optimization scripts. The thermal model of the cell can be discretized in one dimension. The pack models come in two discretization variants, the scaled model uses a single cell model whereas the discretized pack models use arrays of cells, such that the behavior of each single cell can be simulated. In order to account for varying fabrication tolerance, parameters can be varied using a Gaussian distribution. Cyclic and calendaric aging can be considered with semi-empiric approaches.

The cell models are ready to use, the Battery provides an all in one solution for simulating batteries. Convenient data handling, prepared experiments and easily customizable models make the library easy to use. Colored buttons indicate visually the dependencies of the cell components at a glance. The library accounts for different geometries of the cells and the packs and enables the user to make detailed thermal models of the cell.

Library structure

The library is divided into a number of sub-packages for different component types.

The Cells package contains the packages for the electrical, thermal and aging models and subcomponents.

Various examples for battery pack models can be found in the Packs package as well as the thermal models for housing.

The BMS package contains an example of a battery management system as well as its various subcomponents.

All packages above contain an Examples package.

Used Icons and Functions can be found in the Common package. Also within this package is the Material package, which contains material property models. Users can easily create their own material models by duplicating one of the models and adding their own data.

Prepared Tests according to ISO 12405 are stored in the Examples package.

3.7.2 Claytex Library

The Claytex library is a base library, used in the Engines Library below.

3.7.3 Engines Library

The Engines Library is available in two versions:

- The MVEM (Mean Value Engine Models) version is used to predict the cycle averaged performance of the engine.
- The CAREM (Crank Angle Resolved Engine Models) version uses crank angle resolved combustion models to predict the instantaneous torque and air-flow through the engine.

The CAREM version is an extension to the MVEM and adds the additional components necessary to introduce the more refined combustion models.

The intake, exhaust, and most of the mechanical system is common to both variants of the library.

Model philosophy

The Engines library was built on a concept where subsystem models are parameterized individually in isolation then used as "off the shelf" models to build a larger system. Data records slots will be present in most subsystem models and subsystem component models for parameterization purposes. To date, these data records will require individual parameterization according to the specification of the subsystem or component.

3.8 Updated libraries

Below is a short description of updated libraries. For a full description, please refer to the libraries documentation.

3.8.1 Air Conditioning Library

A new version 1.11 has been released.

Improvements

- Improved library documentation.
- Improved component icons for greater system readability.
- Icons in the package browser.
- User's Guide available in-built in the documentation layer of the library. Please note that the .pdf will be discontinued from the next version onwards.

Conversion of user libraries

No conversion script is needed.

3.8.2 Electric Power Library

A new version 2.2.2 has been released. The documentation of the library has been improved.

No conversion of user libraries is needed.

3.8.3 Engine Dynamics Library

A new version 1.2.4 has been released. The release is a maintenance release.

No conversion of user libraries is needed.

3.8.4 Fuel Cell Library

A new version 1.3.2 has been released. This release is a maintenance release.

No conversion of user libraries is needed.

3.8.5 Heat Exchanger Library

A new version 1.4 has been released. The parameter dialogs of the heat exchangers are improved.

Conversion of user libraries

User models based on version 1.3 do not need any conversion.

3.8.6 Human Comfort Library

A new version 2.0.0 has been released. The new HumanComfort Library 2.0 includes the addition of a CFD-based air model. Using view factors, an exact calculation of the thermal radiation between visible surfaces is supported. The new model uses the finite-volume method to subdivide the air volume, making it possible to reduce the number of required cells. The calculation is made on the basis of Navier Stokes equations, which achieves conservation of momentum. It allows simultaneous dynamic simulation of zones and air conditioning systems. With simultaneous simulation of air-conditioned zones (e.g. in buildings and vehicles) and the air conditioning system, it is possible to draw up an analysis of the interactions between the system and the comfort in the zone.

New models:

- HumanComfortLib.CFD.CoarseGrid.Grid model
- HumanComfortLib.Examples.BESTEST package
- HumanComfortLib.Examples.CFD examples
- New option to disable internal long wave radiation (OnlySolarGain radiation model)
- New atmospheric loss model due to EnergyPlus (new default model)

Modifications:

- New library design concept
- New library structure
- Revision of partition model (dialog, grid connection)
- Revision of radiation models (ignore zero area surfaces)
- Revision of atmospheric loss

3.8.7 Hydraulics Library

A new version 4.3 has been released.

New features

- Completely reworked the documentation structure in the library. New User's Guide structure, tutorial for the Lines package, detailed information section for every sub-package (Cylinders, RotaryActuators, Restrictions, Fluids etc).
- New components. TwoWayValveCurrent (current actuated two way valve), TurbulentMultiple (analytical solution to multiple connected orifice models) and OpenVolume (an open tank volume with constant pressure and variable volume).
- Enabled the opportunity to switch off the Boolean parameter useThermal in ThermoHydraulic oil. This enables much faster simulation speed if using ThermoHydraulic media with useThermal=true than using fluids within Fluids.PreDefined.Tabular.
- Added gas-mixing equations to the ThermoHydraulic oil. This enables constant gas content in the ThermoHydraulic oil as well as it prevents negative pressure from being negative.

Improvements

- Appropriate tabs and groups now exist for all top level components. Parameters that are typically affecting on system level can be found in the General tab. While detailed parameters for component design are found in the Advanced tab. There are separate tabs for Volumes, Initialization and Visualization parameters.
- Pressure dependent color can now be set from the outer oil component.
- Merged FlexibleLine and LongLine component. LongLine now also covers flexible lines.
- Hydraulics library passes through ModelManagement style checking tool when it comes to comments on classes and parameters.
- Moved HydraulicResistances to Restrictions.Fittings.

Conversion of user libraries

Automatic conversion of user libraries from version 4.2 is supported using the included conversion script. However, since Hydraulics.DirectionalControl has been completely reworked, DCV- and Spool-models are working, but conversions do not fully support locally duplicated DCV- and Spool-models.

3.8.8 Hydro Power Library

A new version 2.5 has been released.

Improvements

The release has been focused on improving the documentation; there has been a major revision of the model documentation.

Conversion of user libraries

Automatic conversion of user libraries from version 2.4 is supported using the included conversion script.

3.8.9 Liquid Cooling Library

A new version 1.4.1 has been released. This is a maintenance release.

No conversion is needed.

3.8.10 Model Management Library

A new minor version 1.1.5 has been released, with minor improvements.

3.8.11 Modelica_DeviceDrivers

A new version 1.4.3 has been released. The following improvements have been done:

- Switched to semantic versioning. See http://semver.org.
- Migrated to new release process motivated by impact-on-library-developers.

- Added support for external trigger signals to trigger communication blocks.
- Added support to configure byte ordering in communication blocks.
- Added support for TCP/IP communication for Windows.
- Added serial port support for Windows (was already available for Linux).
- Added compiler support for MinGW and Cygwin.
- Added support for all 32 joystick buttons.
- Added serial port support for Linux (already in version 1.3).

3.8.12 Modelica_LinearSystems2

A new version 2.3.3 has been released.

In this version, the handling of large linear systems (linear systems containing several hundred states) with largely varying eigenvalues has been improved. Zeros, poles and Bode plots for such systems are now computed in a numerically more reliable and robust way.

Examples of systems that benefit from this improvement are large hydraulic systems or flexible bodies with many elastic modes.

This library version is backward compatible to the previous version 2.3.2.

3.8.13 Pneumatics Library

A new version 1.8 has been released.

New features

- The new Pneumatics.Elements package provides necessary components required to build any pneumatic valve from scratch. The valves built using this package will be slightly more detailed in nature than the existing functional valve models in Pneumatics library.
- New component DCV_4_3: Directional control valve with four ports and three stable positions.

Improvements

Appropriate tabs and groups now exist for all top level components. Parameters that are typically affecting on system level can be found in the General tab, while detailed parameters for component design are found in the Advanced tab. There are separate tabs for Volumes, Initialization and Visualization parameters.

Conversion of user libraries

Automatic conversion of user libraries from version 1.7 is supported using the included conversion script.

3.8.14 Power Train Library

A new version 2.3.1 has been released. This release is a maintenance release.

This version is backward compatible to version 2.x.y.

3.8.15 Thermal Power Library

A new version 1.11 has been released.

Improvements

The documentation is improved; there has been a major revision of the model documentation.

Conversion of user libraries

Automatic conversion of user libraries from version 1.10 is supported using the included conversion script.

3.8.16 Vapor Cycle Library

A new version 1.2.2 has been released.

Improvements

- Improved Tutorial
- Improved documentation for the heat exchangers.

Conversion of user libraries

There is no need for conversion of user libraries.

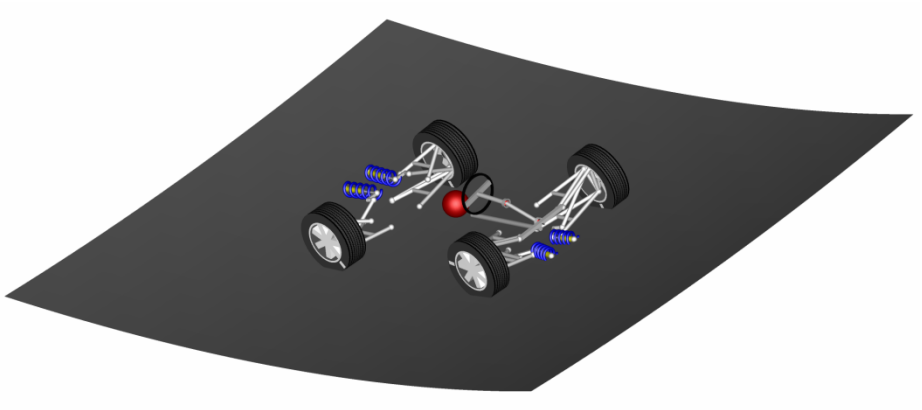
3.8.17 Vehicle Dynamics Library

A new version 2.2 has been released.

New features

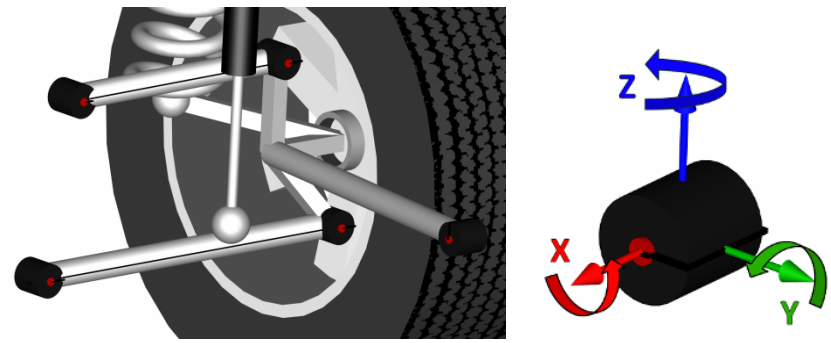
Cornering Ground Model

A new Cornering ground model has been added. This ground model allows modification of the ground surface position, orientation and curvature based on input signals. This ground model can be used in quasi-steady state simulations to evaluate chassis states at a particular straightline or cornering condition. The chassis velocity and curvature result in lateral, longitudinal and vertical accelerations being applied to the vehicle. This ground model is particularly useful in motorsports applications to evaluate chassis states at different points on the race track. Examples using the Cornering ground can be found in QuasiSteadyState and BankedCornering.



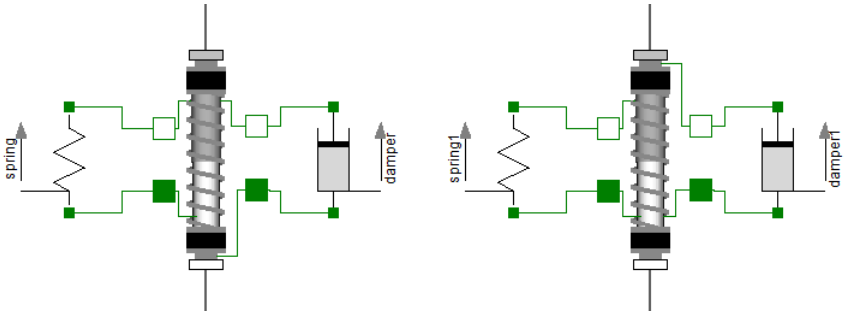
Bushing Visualization

The bushing visualizer has been changed to make it easier to see the relative displacement between frame_b (red cylinder) and frame_a (dark grey cylinder). The updated bushing visualizer can be seen in the image below. The visualizer also makes it easier to understand the axes related to the force {fx, fy, fz} and torque {tx, ty, tz} components. This can be seen in the annotated image below.



New strut components

New bushing mounted strut components have been introduced. These struts are mounted to the chassis and suspension with bushings. They are found in Suspensions.Linkages.Struts.Mounted. The bushing components are replaceable, and Strut2 has two sets of translational flanges where one can be parameterized to attach either inside or outside the mount bushings. This is useful when springs and dampers are co-located in the suspension but attach on different sides of the mount bushings. Two examples are shown here:



New pitman-idler steering model

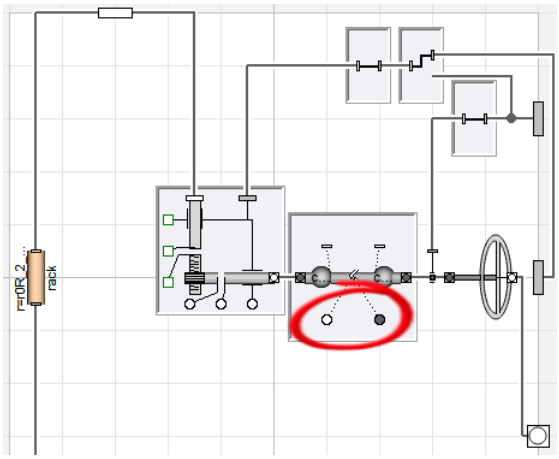
A new pitman-idler steering model has been introduced. This model bas been used previously in NASCAR and is now included as another steering model option.

Improvements/Changes

Changes to Steering Templates

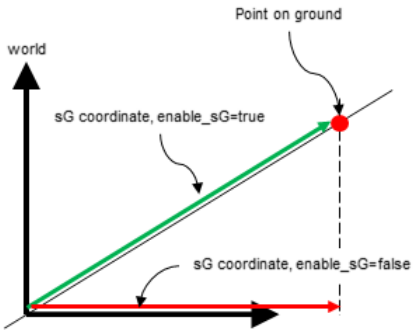
The steering templates have been modified to have additional construction options including a replaceable steering shaft. If you were using either of these templates, you must add a connection between the steeringShaft axle flanges (axle_a and axle_b) as shown in the image below. The following templates were changed:

- GearSteeringWheel
- RackSteeringWheel



Ground function update

There was a discrepancy in how ground coordinates (sG) were defined in Grounds.Flat depending on whether the enable_sG flag was true or false. For enable_sG=false, the sG[1,2] coordinates would be defined in the x,y plane in world coordinates even when the road was inclined while when enable_sG=true they would be defined in the inclined road plane. See illustration below.



The flat_rG_0 function has now been updated so the surface coordinate (sG) are always defined in the inclined ground plane (green arrow) regardless of if sG functions are used.

New Hub Component

A new Hub component has been added. This component was added to properly handle alignment parameters for wheel carriers. Wheel carriers can either be aligned by toe and camber or a spin axis (n0H). The sign of this parameter is different depending on whether applied to a left or right wheel carrier. Previously, the sign change was often done in the suspension linkage. With the new hub component, the same toe/camber can be entered for

both the left and right linkage and the left_linkage parameter in the Hub component will handle the sign changes.

User's guide moved

All content from the external User's guide has been migrated into online documentation in the library. Basic information is found in the User's guide section while more specific information on subcomponents is found in Information classes in each subpackage, see for example Grounds and Chassis.

Conversion of user libraries

Automatic conversion of user libraries from both version 2.0 and 2.1 is supported using the included conversion script.

3.8.18 Vehicle Interfaces Library

A new version 1.2.3 has been released. This version is a maintenance release.

The version is backward compatible with version 1.1.x and 1.2.x.

3.9 Documentation

In the software distribution of Dymola 2016 FD01 Dymola User Manuals of version “September 2015” will be present; these manuals include all relevant features/improvements of Dymola 2016 FD01 presented in the Release notes.

3.10 Appendix – Installation: Hardware and Software Requirements

Below the current hardware and software requirements for Dymola 2016 FD01 are listed.

3.10.1 Hardware requirements/recommendations

Hardware requirements

- At least 1 GB RAM
- At least 400 MB disc space

Hardware recommendations

At present, it is recommended to have a system with an Intel Core 2 Duo processor or better, with at least 2 MB of L2 cache. Memory speed and cache size are key parameters to achieve maximum simulation performance.

A dual processor will be enough if not using multi-core support; the simulation itself, by default, uses only one execution thread so there is no need for a "quad" processor. If using multi-core support, you might want to use more processors/cores.

Memory size may be significant for translating big models and plotting large result files, but the simulation itself does not require so much memory. Recommended memory size is 4 GB of RAM for 32-bit architecture and 6 GB of RAM for 64-bit architecture.

3.10.2 Software requirements

Microsoft Windows

Dymola versions on Windows and Windows operating systems versions

Dymola 2016 FD01 is supported, as 32- and 64-bit application, on Microsoft Windows 7 and Windows 8.1. Since Dymola does not use any features supported only by specific editions of Windows ("Home", "Professional", "Enterprise" etc.), all such editions are supported if the main version is supported.

Compilers

Please note that for the Windows platform, a Microsoft C/C++ compiler, or a GCC compiler, must be installed separately. The following compilers are supported for Dymola 2016 FD01 on Windows:

Microsoft C/C++ compilers, free editions:

- Visual Studio 2008 Express Edition (9.0)
- Visual C++ 2010 Express (10.0)
- Visual Studio 2012 Express Edition (11.0)

- Visual Studio 2013 Express Edition (12.0)

Microsoft C/C++ compilers, professional editions:

- Visual Studio 2005 (8.0)
- Visual Studio 2008 (9.0)
- Visual Studio 2010 (10.0)
- Visual Studio 2012 (11.0)
- Visual Studio 2013 (12.0)

GCC compilers

Dymola 2016 FD01 has limited support for the MinGW GCC compiler. The following versions have been tested:

- For 32-bit GCC: version 4.8.1.
- For 64-bit GCC: version 4.9.2.

To download any of these free compilers, please visit http://www.Dymola.com/compiler where the latest links to downloading the compilers are available. Needed add-ons etc are also specified here. Note that you need administrator rights to install the compiler.

Current limitations with 32-bit and 64-bit GCC:

- Embedded servers (DDE or OPC servers) are not supported.
- Commercial libraries: Only limited testing has been done.
- Support for external library resources is implemented, but requires that the resources support GCC, which is often not the case.
- No support for run-time license.
- For 32-bit simulation, parallelization is only supported with any of the Lsodar, Dassl, Euler, Rkfix2, Rkfix3, or Rkfix4 algorithms.

Dymola license server

For a Dymola license server on Windows, all files needed to set up and run a Dymola license server on Windows, except the license file, are available in the Dymola distribution. (This includes also the license daemon, where Dymola presently supports FLEXnet Publisher version 11.11. This version is part of the Dymola distribution.)

Linux

Supported Linux versions and compilers

Dymola 2016 FD01 runs on SUSE Linux (Release 11), 32-bit and 64-bit, with gcc version 4.9.2, and compatible systems. In addition to gcc, the model C code generated by Dymola can also be compiled by clang. To change compiler, change the variable `CC` in `/opt/dymola-<version>-<architecture>/insert/dsbuild.sh`, where `<architecture>` is `i586` for 32-bit application, and `x86_64` for 64-bit application. As an example, for a 64-bit Dymola 2016 FD01 application:

```
/opt/dymola-2016FD01-x86_64/insert/dsbuild.sh
```

Dymola 2016 FD01 is supported as a 32-bit and 64-bit application on Linux.

Notes

- 32-bit compilation might require explicit installation of 32-bit libc. E.g. on Ubuntu: `sudo apt-get install g++-multilib libc6-dev-386`
- Dymola is built with Qt 5.5 and thereby inherits the system requirements from Qt. However, several xcb helper libraries are bundled with Qt (in detail, QT was built with the flag –qt-xcb) in order to reduce the system dependencies as much as possible.
- For rendering of jpg files, libjpg62 must be installed.

Note on libraries

- Please note that you have to use the Optimization library version 2.x or higher to use multi-criteria design optimization on Linux; the older Design.Optimization package does not support multi-criteria design optimization on Linux.
- The library UserInteraction is not supported on Linux.

Dymola license server

For a Dymola license server on Linux, all files needed to set up and run a Dymola license server on Linux, except the license file, are available in the Dymola distribution. (This also includes the license daemon, where Dymola presently supports FLEXnet Publisher version 11.11.)