## Contents

1. **Important notes on Dymola** ................................................................. 5  
2. **About this booklet** .............................................................................. 5  
3. **Dymola 2020x** .................................................................................... 6  
   3.1 **Introduction** .................................................................................. 6  
      3.1.1 Additions and improvements in Dymola ........................................ 6  
      3.1.2 New and updated libraries ............................................................ 6  
   3.2 **Major Graphical User Interface Update** .......................................... 8  
      3.2.1 A quick summary of the new user interface ............................... 8  
      3.2.2 Some ideas behind the changes and some results ....................... 8  
      3.2.3 About the ribbons ...................................................................... 23  
      3.2.4 Examples of other GUI changes .................................................. 25  
   3.3 **Developing a model** ....................................................................... 28  
      3.3.1 Each Dymola version has its own settings file ............................ 28  
      3.3.2 Minor improvements ................................................................. 28  
   3.4 **Simulating a model** ....................................................................... 31  
      3.4.1 The model layer window ............................................................. 31  
      3.4.2 Plot tab ..................................................................................... 33  
      3.4.3 Animation tab .......................................................................... 33  
      3.4.4 Scripting .................................................................................. 34  
      3.4.5 Minor improvements ................................................................. 35  
   3.5 **Installation** .................................................................................... 38  
      3.5.1 Installation on Windows ............................................................. 38  
      3.5.2 Installation on Linux .................................................................... 40  
      3.5.3 Dymola license server on Windows and Linux ......................... 40  
   3.6 **Model Management** ..................................................................... 41  
      3.6.1 Encryption in Dymola ............................................................... 41  
   3.7 **Other Simulation Environments** ................................................... 42  
      3.7.1 Dymola – Matlab interface ......................................................... 42  
      3.7.2 Real-time simulation .................................................................. 42  
      3.7.3 OPC Communication ............................................................... 43  
      3.7.4 FMI Support in Dymola ............................................................. 43  
   3.8 **Modelica Standard Library and Modelica Language Specification** ................................................................. 44  
   3.9 **Documentation** ............................................................................. 44  
   3.10 **Appendix – Installation: Hardware and Software Requirements**  46
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.10.1</td>
<td>Hardware requirements/recommendations</td>
<td>46</td>
</tr>
<tr>
<td>3.10.2</td>
<td>Software requirements</td>
<td>46</td>
</tr>
</tbody>
</table>
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. Three types of compilers are supported on Windows in Dymola 2020x:

Microsoft Visual Studio C++

This is the recommended compiler for professional users. Both free and full compiler versions are supported. Refer to section “Compilers” on page 46 for more information. Note that from Dymola 2020x, Visual Studio C++ compilers older than version 2012 are no longer supported.

Intel

Dymola 2020x has limited support for the Intel Parallel Studio XE compiler. For more information about this compiler, see section “Compilers” on page 46; the section about Intel compilers.

GCC

Dymola 2020x has limited support for the MinGW GCC compiler, 32-bit and 64-bit. For more information about GCC, see section “Compilers” on page 46; 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 48 for more information.

2 About this booklet

This booklet covers Dymola 2020x. The disposition is similar to the one in Dymola User Manuals; the same main headings are being used (except for, e.g., Libraries and Documentation).
3 Dymola 2020x

Note! This version of Dymola, Dymola 2020x, is the last version that officially supports Microsoft Windows 7.

3.1 Introduction

3.1.1 Additions and improvements in Dymola

A number of improvements and additions have been implemented in Dymola 2020x. In particular, Dymola 2020x provides:

- Major Graphical User Interface Update (page 8)
- Looking at code and documentation when simulating: the model layer window (page 31)
- Each Dymola version has its own settings file (from Dymola 2020x) (page 28)
- Support for Microsoft Visual Studio 2019 compiler (page 38)
- Discontinued support of some old Microsoft Visual Studio compilers (page 39)
- Using the web based license server manager lmadmin for FlexNet Publisher (page 40)
- Improved library encryption (page 41)
- Changes in Dymola documentation (page 44)

3.1.2 New and updated libraries

New libraries

There are no new libraries in this Dymola version.

Updated libraries

The following libraries have been updated:

- Battery Library, version 2.1.3
- Brushless DC Drives Library, version 1.1
- ClaRa DCS Library, version 1.2.1
- ClaRa Grid Library, version 1.2.1
- ClaRa Plus Library, version 1.2.1
- Claytex Library, version 2019.2
- Claytex Fluid Library, version 2019.2
- Cooling Library, version 1.3.2
- Dassault Systemes Library, version 1.3.1
• Dymola Commands Library, version 1.8
• Dymola Models Library, version 1.0.1
• Electric Power Systems Library, version 1.3
• Electrified Powertrains Library (ETPL), version 1.3.1
• Fluid Dynamics Library, version 2.8.0
• Fluid Power Library, version 2019.2
• FTire Interface Library, version 1.0.3
• Human Comfort Library, version 2.8.0
• HVAC (Heating, Ventilation, and Air Conditioning) Library, version 2.8.0
• Hydrogen Library, version 1.3.1
• Pneumatic Systems Library, version 1.3
• Smart Electric Drives, version 1.4.8
• Testing Library, version 1.2.2
• Thermal Systems Library, version 1.5.0
• Thermal Systems Mobile AC Library, version 1.5.0
• VeSyMA (Vehicle Systems Modeling and Analysis) Library, version 2019.2
• VeSyMA - Engines Library, version 2019.2
• VeSyMA - Powertrain Library, version 2019.2
• VeSyMA - Suspensions Library, version 2019.2
• VeSyMA2ETPL Library, version 2019.2
• Visa2Base, version 1.7
• Visa2Paper, version 1.7
• Visa2Steam, version 1.7

For more information about the updated libraries, please see the Release Notes section in the documentation for each library, respectively.
Major Graphical User Interface Update

3.2.1 A quick summary of the new user interface

A quick summary of the new user interface, containing a number of commented slides, is available as a pdf file. To see it, click here.

3.2.2 Some ideas behind the changes and some results

The former mode handling using “Modeling mode” and “Simulation mode” is replaced by tabs, but without any limitations on other commands.

In practice, the previous Modeling mode is replaced in Dymola 2020x by the tabs Graphics Documentation, and Text, while the previous Simulation mode is replaced by the Simulation tab (and related Plot, Animation, and Script Editor subtabs).

In previous Dymola versions, selecting between the two modes “Modeling mode” and “Simulation mode” imposed limitations; in many cases you were forced to go to a certain mode to perform a certain action.

As an example, looking at the File menu in Simulation mode in previous versions of Dymola, a number of commands were not accessible in that mode:
In Dymola 2020x, being on the Simulation tab, which in practice replaces the previous Simulation mode, all commands in the File menu are enabled. An example:
Options to look at Modelica code and documentation when simulating

You can use the model layer window in the Simulation tab to look at Modelica code and documentation while simulating. See “The model layer window” starting on page 31.

The model layers are handled in tabs, but shortcuts are still available

In previous versions of Dymola, selection of what class layer to work in was central in the Modeling mode:

In Dymola 2020x, the handling of the class layers is divided between tabs, but the shortcuts above are still present:
The modeling tabs

**The Graphics tab.**

The icon layer and the diagram layer are handled by the **Graphics** tab, which one to work with is selected in the **Layer** group of that tab:

Note that you usually work with full windows size – such and image would be too small in this document, but the framed part in the tab would then look like:
The documentation layer is handled in the Documentation tab; note that it is now easier to see and select the documentation modes:

It is even easier when working in full windows size:
The Modelica text layer, and the used classes layer are handled in the **Text** tab, which one to work with is selected in the **Layer** group of that tab:

```plaintext
model Pendulum "Simple pendulum"
  parameter Real m=1 "Mass of the pendulum";
  parameter Real l=1 "Length of the pendulum";
  parameter Real g=9.8 "Gravity of acceleration";
  parameter Real J=m*L^2 "Moment of inertia";
  Real phi(start=0.1) "Pendulum angle";
  Real w "Angular velocity";
  equation
    d(phi) = w;
    J*d(w) = -m*g*L*sin(phi);
end Pendulum;
```
Shortcuts to model layers

Shortcuts directly to the model layers are still always available to the lower right in the Dymola window, independent of which tab you are in:

Clicking any button in this model layer toolbar automatically opens the layer and the relevant tab (except when working in the Simulation tab, and the Tools tab, if being activated when in the Simulation tab, see “The model layer window” starting on page 31).

You can also use shortcut keys to model layers (the last shortcut is not really a shortcut to a model layer, but here for completeness):

<table>
<thead>
<tr>
<th>Shortcut keys</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+F1</td>
<td>Displays the icon layer</td>
</tr>
<tr>
<td>Ctrl+F2</td>
<td>Displays the diagram layer</td>
</tr>
<tr>
<td>Ctrl+F3</td>
<td>Displays the documentation layer</td>
</tr>
<tr>
<td>Ctrl+F4</td>
<td>Displays the Modelica text layer</td>
</tr>
<tr>
<td>Ctrl+F5</td>
<td>Displays the Used Classes layer</td>
</tr>
<tr>
<td>Ctrl+F6</td>
<td>Displays the “simulation layer” (the Simulation tab)</td>
</tr>
</tbody>
</table>

Please compare these shortcuts with the shortcut keys for tabs, see section “Shortcut keys to tabs” on page 21.
Handling of simulation: The Simulation tab, and the related Plot, Animation, and Script Editor tabs

The Simulation tab. In Dymola 2020x, the simulation of a model is handled in the **Simulation** tab, opening this tab and simulating a version of the demo coupled clutches gives the result:

The commands in the tab seems somewhat cluttered in this small size of window, but the full ribbon in full window size looks different (divided in two images):
Notes:

- The simulation commands and the simulation setup is handled in the **Simulation** group. Note that the stop time and algorithm are easily directly changeable, while the full simulation setup is reached by the command **Simulation Setup**.
- The command to open or run scripts, or to perform predefined commands, are now located in the Scripting group of the ribbon. There are two commands with connected drop-down menu:

  ![Scripting group](image)

  You can click directly on **Run Script** to open a script to execute, clicking the arrow next to the command, you also see the previous executed scripts to select from. You can also click on the command **New Script** to create a new script, clicking the arrow next to it displays alternative commands. Here you can also find the recent scripts (moved from previous **File > Recent Files** command). The commands are also available in the **Script Editor** tab that opens when you activate the script editor – see the description of the **Script Editor** tab below.

- **Analyze Numerics** is now more accessible that in previous versions.
- For more information about the model layer window (the window with header **Model View** in the figure above), see “The model layer window” starting on page 31.
Above also a plot window is present, although not active; the presence of a plot window results in a Plot tab being visible. It will be activated by activating the plot window:

Note: Many command have been moved from the context commands (for curve, legend, and diagram) to the ribbon of the Plot tab
The Animation tab. If you have an animation window, you will have an **Animation** tab; if you activate the window you will activate that tab:
If you create a new script or open an existing script (using, on the **Simulation** tab, in the **Scripting** group, the **New Script** command or the **New Script > Open Script...** command) you will open and activate a script editor and a corresponding **Script Editor** tab:

![Script Editor screenshot]

**Notes:**
- In the **Simulation** tab, you can use the Scripting group to open or run scripts, or to perform predefined commands. See that tab above.
- The commands for handling the script (**New Script, Open Script** etc.) are moved from the script editor window toolbar to the **Script Editor** tab.
The Tools tab

The last tab is the Tools tab; here a number of “general” commands are collected:

- Export operations (moved from the File menu in previous versions)
- Various tools and options setup (for example, Options that is the previous Edit > Options… command)
- Version handling
- Help and documentation
- Custom menus and toolbars

Notes:
- You can find the Options command as File > Options as well.
• The shortcut buttons to the model layers in the lower right corner works differently depending on what layer you display when in the **Tools** tab; being in the **Simulation** tab and then opening the **Tools** tab, the shortcut buttons behaves as being in the **Simulation** tab, see “Displaying different model layers in the model layer window” on page 32.

**Shortcut keys to tabs**

For the tabs, you can use the following shortcut keys, listed in tab order:

**Note** that you only can go to tabs visible, that is, you can, for example, never be in the **Graphics** tab and display the **Plot** tab by using **Alt+P**; the **Plot** tab is not visible when in the **Graphics** tab.

<table>
<thead>
<tr>
<th>Shortcut keys</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alt+F</strong></td>
<td>Displays the <strong>File</strong> menu</td>
</tr>
<tr>
<td><strong>Alt+G</strong></td>
<td>Displays the <strong>Graphics</strong> tab</td>
</tr>
<tr>
<td><strong>Alt+D</strong></td>
<td>Displays the <strong>Documentation</strong> tab</td>
</tr>
<tr>
<td><strong>Alt+T</strong></td>
<td>Displays the <strong>Text</strong> tab</td>
</tr>
<tr>
<td><strong>Alt+S</strong></td>
<td>Displays the <strong>Simulation</strong> tab</td>
</tr>
<tr>
<td><strong>Alt+O</strong></td>
<td>Displays the <strong>Tools</strong> tab</td>
</tr>
<tr>
<td><strong>Alt+P</strong></td>
<td>Displays the <strong>Plot</strong> tab</td>
</tr>
<tr>
<td><strong>Alt+A</strong></td>
<td>Displays the <strong>Animation</strong> tab</td>
</tr>
<tr>
<td><strong>Alt+C</strong></td>
<td>Displays the <strong>Script Editor</strong> tab</td>
</tr>
</tbody>
</table>

Please compare with the shortcuts to model layers, see section “Shortcuts to model layers” on page 14.

**The relation between the context menus and the new tab commands**

In previous versions, the context menus have consisted of virtually all important commands for an object. In Dymola 2020x it is instead the tabs that hold the majority of the commands – those ones not specific for the item or very general – this means that the context commands are now more lean, not obscuring as much as in earlier versions. As a couple of examples, consider the context menu for a graphical object in the diagram and a curve in a plot, in previous Dymola versions:
In Dymola 2020x:

To see the “missing commands” compared with previous versions, please see the images of the Graphics tab (for the object) and the Plot tab (for the curve) above.

### 3.2.3 About the ribbons

The tabs presented above (Graphics, Documentation, Text, Simulation, Plot, Animation, Script Editor, and Tools tab) are all ribbon tabs, that is, each tab contains a ribbon containing the commands in the tab.

#### Grouping

In the ribbons, the commands are grouped logically.

#### Primary and secondary commands

When the size of the Dymola window permits it, primary commands are presented by larger icons, secondary commands by smaller icons; an example from the Draw group in the Graphics tab:
Commands expanding to menus

Some commands expand to a menu, for example the **Arrange** command from the group above:

Selecting ribbon display mode

The **Minimize Ribbon** command can be used to select how much of the ribbons to display:

The command works in toggle mode, with three levels. The default level, showing both commands and tab ribbons, are seen above. Clicking once on the command, you do not display the names of the groups:

Clicking once more, you do not display the groups and the commands (unless when clicking the tab name):
3.2.4 Examples of other GUI changes

Quick Access Toolbar

A quick access toolbar is present to, by default, create a new model, open or save a model, undo/redo, simulate, or stop simulation:

Note the possibility to configure what commands to display, as above. The configuration is saved between sessions.

More information in the command tooltip

An example:
**New Windows menu**

The new Windows menu are now located in the upper right corner of the Dymola window:

![New Windows menu screenshot](image)

The new Windows menu contains the previous **Recent Windows** command in Simulation mode and the general **Window** command.

**Notes:**

- By clicking on this command, you toggle between the present windows (in the example above Model View, Plot (1), and Animation. (“Model View” is the diagram layer when simulating, see “The Model View is always displayed by default in the model layer window” on page 32.)
- The entry **Docked Windows** corresponds to the previous **Window > Tools**; now listing the displayed browsers and windows in the present tab – you can display/hide them by clicking them in the menu.

**Rearranged File menu**

The commands in the **File** menu have been rearranged in groups, and some command have been moved to e.g. the **Tools** tab.
**Changed location of the displayed model instance hierarchy path**

The model instance hierarchy path is now displayed in the upper left of the diagram layer itself (before it was located in a specific toolbar):
3.3 Developing a model

3.3.1 Each Dymola version has its own settings file

From Dymola 2020x, each new Dymola version will have its own settings file. The name of the settings file will still be setup.dymx, but the file will reside in a new version-specific folder for each new Dymola version. The folder will be located in the folder ...
\Users\<user>\AppData\Roaming\DassaultSystems\Dymola\.

For Dymola 2020x on Windows, a typical path for the setup file is, according to this:
C:\Users\<user>\AppData\Roaming\DassaultSystems\Dymola\2020x\setup.dymx

For Dymola 2020x on Linux, the path is, under the user’s home directory:
    .dassaultsystemes/dymola/2020x/setup.dymx

When starting a newly installed Dymola 2020x, any previous setup.dymx or setup.mos is copied to the new setup.dymx.

The new way of handling settings file means, for example, that different Dymola versions can be set up to use different compilers.

Note that there is no change in previous versions of Dymola, this new setup handling only applies to Dymola 2020x and later.

3.3.2 Minor improvements

Applying experiment annotation from base class when extending

When extending from a class, for example, from a model, any experiment annotation defined in the corresponding base class will be included in the extending. This was the case already in recent versions of Dymola.

(The experiment annotation correspond to the settings in the General tab of the simulation setup; for example, start time, stop time, and algorithm. This means that in the new class, you can see settings in this tab that comes from the base class.)

If the you add an experiment annotation in the new class, that one overrides the base class annotation.

Warning about potentially conflicting Modelica names

Modelica names are case-sensitive, but files in some operating systems, e.g. MS Windows, are not. This may lead to problems when mapping Modelica files to e.g. MS Windows filenames.

To prevent such problems, a warning is now given when creating or renaming a class, if the new class differs only in upper/lower case from an existing class (in the same scope, for example, in the same package).
There are two cases, one is when creating a model that is similar to an existing model in the same package, for example, creating `MYMODEL` when `MyModel` is already created in the same package. You will be given a warning and a suggestion to use another name.

The other case is when you create a model that has similar name as a model name of an instance used in the package, but coming from another package. As an example, dragging, from the package `AnotherPackage`, an instance `MyTest` to a model in the package `ThisPackage`, and then, in `ThisPackage`, create a model `ANOTHERPACKAGE`. In this case you also have the alternative to update the references according to the new model you create.

**New GUI features**

Except from the reorganization of the GUI presented above, there are some new features as well. (Some are presented in their context above, but can also be listed as new features.)

**Pedantic check as GUI selection**

You can select to check in pedantic mode from the Graphics and Text tab, by clicking the arrow next to the Check command and select Pedantic:

![Pedantic check](image)

**Option to have a smaller File > Libraries menu**

In case you have many libraries, the File > Libraries menu may be quite large by default. To have a smaller menu in such a case, you can set the flag

```plaintext
Advanced.File.CompactMenu = true
```

(The flag is by default false.) Note that this flag only works on the File > Libraries menu, not on the general File menu.

**Support for external icons for the menus File > Libraries and File > Demos**

To support external icons in the File > Libraries and File > Demos menus, there is a new input parameter `iconName` of type String added to the built-in function `LibraryInfoMenuCommand`. (This built-in function is used in the file `libraryinfo.mos` to create such menus.)
The default value is "libraryicon.png" which means that the easiest solution to add an external icon is to create an icon with that name in the same folder as libraryinfo.mos. Notes:

- Other file formats are also supported, for example, svg.
- A relative path to the icon can also be added if it should be located in another folder than libraryinfo.mos.
- Oversized icons are scaled to fit in the menus.
3.4 Simulating a model

3.4.1 The model layer window

Looking at the result when simulating a version of the coupled clutches demo:

The model layer window (with the header Model View above) corresponds to the Diagram window in Simulation mode in previous Dymola versions. In Dymola 2020x, the functionality is extended, and since it can show several model layers, not only the diagram layer, a suitable name for the window is now “model layer window”.

**The Model View is always displayed by default in the model layer window**

You don’t have to click any command to display the model layer window, it is always displayed by default, and by default it displays the diagram layer in a read-only mode, as in previous Dymola versions. The header **Model View** of the window indicates that. If you happen to close it in the figure above, just click on the shortcut button to the diagram layer in the lower left corner of the Dymola window (the shortcuts are framed in the figure above).

**Navigating by the instance hierarchy path**

If you right-click any component and select **Show Component**, you open this component as in previous versions of Dymola, but now also the instance hierarchy path appears in the Model View; you can click on the component you want to navigate to.

![Model View](image)

**Displaying different model layers in the model layer window**

If you are in the **Simulation** tab, the model layer shortcuts in the lower right corner of the Dymola window (framed in the first image above) will work on the model layer window. This enables you to, for example, look at the Modelica code and the documentation while working with simulation. Note that you cannot do any editing in the model layer window.

As an example, if you have opened the **flange_a** component as in the figure above, and you want to look at the Modelica text or the documentation of this component, you can click the corresponding model layer shortcut buttons in the lower right corner of the Dymola window as above to get:
To enable convenient editing, if you have, for example, displayed the Modelica text of a certain component in the model layer window, you will the same Modelica text displayed in the Text tab when you click that tab to edit the text.

### 3.4.2 Plot tab

**GUI changes of the plot tab**

Many context commands have been moved from the context menu of curves and legends to the Plot tab. See the general GUI changes above.

**Easier access to changing the independent variable**

You can now change the independent variable by context menus in the diagram and for the curve/legend, as well from the Plot tab, apart from the already present way of changing it from the variable browser.

### 3.4.3 Animation tab

**Animation control for models containing UserInteraction objects**

When you simulate models containing UserInteraction objects, for example, the demo Modelica.Fluid.Examples.Tanks.ThreeTanks, the framed animation control items in the figure below are available in the Animation tab:
3.4.4 Scripting

GUI changes of the script editor

Some specific GUI changes:

- The commands for creating, opening, running etc. a script have been moved from the toolbar in the script editor to the Simulation tab and the Script Editor tab. See the general GUI changes above.

- The command to save a script with a new name has changed from using File > Save As… (when having the script editor active) in previous version, to the command Save Script > Save As… in the Script Editor tab:
Note that in Dymola 2020x, the commands **File > Save > Save** and **File > Save > Save As**... are not working on script files anymore.

**The script created by the command Generate Script opened after generation**

In Dymola 2020x when you use the command **Generate Script**, the generated script is always opened in the **Script Editor**, to make it easy to add a description, and also to acknowledge that the command did work. If you do not want to add something, you can just close the **Script Editor**.

**Improved built-in function translateModelFMU**

The built-in function **translateModelFMU** now uses any experiment setting (that is, settings in the General tab of the simulation setup; for example, start time, stop time, and algorithm) to make it easier to script these.

If this is not desired, you can set the flag **Advanced.FMI.UseExperimentSettings=false** to deactivate the feature (the flag is by default true).

### 3.4.5 Minor improvements

**New flags for advanced setup for simulations**

A number of flags that can be used by advanced users to fine-tune the simulation have been introduced:

- **Advanced.Simulation.StepSizeMin** The minimum step-size of simulators if non-zero, \( h_{\text{min}} \). Use with care, since too large value will lead to simulation failure. The default value is **Advanced.Simulation.StepSizeMin=0.0**

- **Advanced.Simulation.StepSizeMax** The minimum step-size of simulators if non-zero, \( h_{\text{max}} \). Use with care, since too small value will lead to simulation failure. The default value is **Advanced.Simulation.StepSizeMax=0.0**

- **Advanced.Simulation.StepSizeStart** The starting step-size of simulators if non-zero, \( h_0 \). The default value is **Advanced.Simulation.StepSizeStart=0.0**
• **Advanced.Simulation.EventEpsilon**  The time resolution for events, eveps. The default value is `Advanced.Simulation.EventEpsilon=1e-10`

• **Advanced.Simulation.MaxEventIterations**  The maximum number of event iterations, eviter. Use with care, since too small value will lead to simulation failure. The default value is `Advanced.Simulation.MaxEventIterations=20`

### Improved diagnostics for failing to evaluate a model due to `Evaluate=false`

Setting `Evaluate=false` may prevent the translation of a model since some parameters cannot be evaluated. The diagnostics of such cases have been improved in Dymola 2020x. A very simple example of such a model:

```plaintext
model Unnamed
  parameter Boolean b=false annotation(Evaluate=false);
  parameter Real x[:]=zeros(if b then 1 else 2);
end Unnamed;
```

The diagnostics when translating this model is now:

![Diagnostics](image)

### Changed default value of Show Toolbar in the command log pane

The **Show Toolbar** option in the command log pane in the command window is by default not active in Dymola 2020x, meaning that the command log pane by default only shows the command log; you cannot add content and edit that content, by default. This saves space since the editing toolbar is not visible, and increases the performance somewhat since the handling of the log file is simplified.
If you create a stand-alone command window with the command **Windows > New Command Window** that window will have the same state of the toolbar display as the current state of the internal command window. Once created, the windows are independent of each other when it comes to this setting – this enables you to have, for example, the internal window showing only the command log, while you can have the toolbar enabled in the stand-alone window.

The flag `Advanced.CommandLogToolbar` only applies to the internal command log window.
3.5 Installation

For the current list of hardware and software requirements, please see chapter “Appendix – Installation: Hardware and Software Requirements” starting on page 46.

3.5.1 Installation on Windows

For the full list of supported compilers, see “Software requirements” starting on page 46.

Discontinued support for Windows 7 in Dymola 2021

This version of Dymola, Dymola 2020x, is the last version that officially supports Microsoft Windows 7 operating system.

Microsoft Visual Studio compilers

Support for Visual Studio 2019 compiler

Dymola 2020x supports the Visual Studio 2019 compiler, the following editions:
Visual Studio Community 2019 (16)
Visual Studio Enterprise 2019 (16)
Visual Studio Professional 2019 (16)

Microsoft Build Tools 2019 Notes:

- The minimum required selection to run Dymola is:
  - VC++ 2019 version highest listed latest version-number tools
  - C++/CLI support
- Installing this selection, no IDE (Integrated Development Environment) is installed, only command line features
- This installation is not visible as a specific selection when later selecting the compiler in Dymola, the alternative to select is the same as for any Visual Studio 2019 alternative: Visual Studio 2019/Visual C++ 2019 (16).

The compiler is selected in the simulation setup, reached by the command Simulation > Setup, in the Compiler tab:
Discontinued support of some old Microsoft Visual Studio compilers

Due to the fact that Microsoft no longer supports Visual Studio compiler versions older than Visual Studio 2012, the compilers Visual Studio 2005, 2008, and 2010 are not supported in this Dymola version. See the above image for selectable compilers.

**Intel compilers**

Note that the Visual Studio 2019 compiler is not supported as underlying compiler when selecting an Intel compiler.
3.5.2 Installation on Linux

Updated gcc compiler support
Dymola 2020x runs on openSUSE 42.2, 64-bit, with gcc version 5.3.1, and compatible systems. For more information, see “Supported Linux versions and compilers” on page 48.

Updated Qt support
Dymola 2020x is built using Qt 5.12.4 and thereby inherits the system requirements from Qt. Some libraries might require explicit installation:

- libevent-2.0.so.5
- libpng16-0
- libre2.so.0
- libwebp.so.5
- libwebpdemux.so.1
- libwebpmux.so.1

3.5.3 Dymola license server on Windows and Linux

Using the web based license server manager lmadmin for FlexNet Publisher
In Dymola 2020x you can use the web based license server manager lmadmin for FlexNet Publisher instead of the command line based lmgrd both on Windows and Linux.

Some short notes about the handling:
- start: run bin/lmadmin/lmadmin
- access: http://localhost:8090
- stop: Administration - Server Configuration - Stop Server
- log on first time, user: "admin", passwd: "admin"

For more information, please see the documentation for Flexera Software LLC FlexNet Publisher Licensing Toolkit 11.14.
3.6 Model Management

3.6.1 Encryption in Dymola

Library encryption enhancements

In Dymola 2020x it is possible to manage the protection level for a library hierarchy using a new graphical user interface. An example:

The starting point in the example above is a package MyPackage with 10 sub-packages. Right-clicking MyPackage in the package browser and selecting Attributes from the context menu gives the dialog to the right of the package browser above.

Clicking, in this dialog, the button Settings controlling visibility after possible encryption gives the browser to the right in the image above. Each subpackage in this example has been given a different encryption level by right-clicking it and selecting from the context menu, as shown for the last package.

Note that Full access and Text have two levels, one for non-package, and one without. The non-package level means that for such a package, the encryption level is applicable for the components in the package, but not the package itself. For example, for the package My_Subpackage6 you can look at the Modelica text for each included component, but not for the total package, while for the package My_Subpackage 8 you can look at the Modelica text also for the total package.
The selected encryption levels are applied when encrypting the package.

3.7 Other Simulation Environments

3.7.1 Dymola – Matlab interface

Compatibility

The Dymola – Simulink interface now supports Matlab releases from R2014a (ver. 8.3) up to R2019a (ver. 9.6). Only Visual Studio C++ compilers are supported to generate the DymolaBlock S-function. The LCC compiler is not supported.

3.7.2 Real-time simulation

Compatibility – dSPACE

Dymola 2020x officially supports the DS1005, DS1006, MicroLabBox, and SCALEXIO systems for HIL applications. For these systems, Dymola 2020x generated code has been verified for compatibility with the following combinations of dSPACE and Matlab releases:

- dSPACE Release 2014-A with Matlab R2014a
- dSPACE Release 2014-B with Matlab R2014b
- dSPACE Release 2015-A with Matlab R2015a
- dSPACE Release 2015-B with Matlab R2015b
- dSPACE Release 2016-A with Matlab R2016a
- dSPACE Release 2016-B with Matlab R2016b
- dSPACE Release 2017-A with Matlab R2017a
- dSPACE Release 2017-B with Matlab R2017b
- dSPACE Release 2018-A with Matlab R2018a
- dSPACE Release 2018-B with Matlab R2018a and R2018b
- dSPACE Release 2019-A with Matlab R2018a, R2018b, and R2019a

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 supports 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.

Note on dym_rti_build and dSPACE Release 2017-A and later

The function rti_usrtrcmerge is no longer available in dSPACE Release 2017-A and later. As a consequence, it is required to run the standard rti_build function (with the ‘CM’ command) after dym_rti_build to get your _usr.trc content added to the main .trc file. For example:
Compatibility – Simulink Real-Time
Compatibility with Simulink Real-Time has been verified for all Matlab releases that are supported by the Dymola – Simulink interface, which means R2014a (Simulink Real-Time ver. 6.0) to R2019a (Simulink Real-Time ver. 6.10). Only Microsoft Visual C compilers have been tested.

3.7.3 OPC Communication
The OPC Communication support in Dymola will be removed in a future release.

Note that the discontinued support of Microsoft Visual Studio compilers older than Visual Studio 2012 means that in Dymola 2020x only Visual Studio 2012 is supported for OPC communication in Dymola.

3.7.4 FMI Support in Dymola
Unless otherwise stated, features are available both for FMI version 1.0 and version 2.0.

FMU Export

Improved built-in function translateModelFMU
The built-in function translateModelFMU now uses any experiment setting (that is, settings in the General tab of the simulation setup; for example, start time, stop time, and algorithm) to make it easier to script these.

If this is not desired, you can set the flag Advanced.FMI.UseExperimentSettings=false to deactivate the feature (the flag is by default true).

Improved command for FMU Export
Due to the above item, the command Simulation > Translate > FMU has been improved by using the experiment setting, since the command uses the above built-in function.

Improved support for optional FMI 2.0 features
The following options are now supported also for Co-simulation using Dymola solvers, for all Dymola solvers except Cvode:
- canGetAndSetFMUState
- canSerializeFMUState

Improved export of source code FMUs using external c-functions
When exporting source code FMUs using external c-functions the source code of those functions will now be included.
Specifically:

- The contents of includeDirectory will be included under sources/include
- The contents of __Dymola_SourceDirectory (default “…/Resources/Source”) will be included under sources/source

**FMI Kit for Simulink**

The FMI Kit for Simulink is from Dymola 2020x not included in the Dymola distribution, you can instead access it freely from GitHub (including all source code):

[https://github.com/CATIA-Systems/FMIKit-Simulink](https://github.com/CATIA-Systems/FMIKit-Simulink)

The current latest released version is 2.6.0; a direct link to this version is:

[https://github.com/CATIA-Systems/FMIKit-Simulink/releases/tag/v2.6](https://github.com/CATIA-Systems/FMIKit-Simulink/releases/tag/v2.6)

Starting from version 2.6.0, support for FMI Kit Simulink is handled through the issue system at GitHub.

---

### 3.8 Modelica Standard Library and Modelica Language Specification

The current version of the Modelica Standard Library is version 3.2.3. The current version of the Modelica Language Specification is 3.4.

Note that the Modelica Standard Library version 3.2.3 is compliant with the Modelica Language Specification 3.4. (So this version of the Modelica Language Specification should be used for reference, and not the Modelica Language Specification 3.2, Revision 2.)

---

### 3.9 Documentation

**General**

The new GUI has a major impact on particularly the former Dymola User Manual Volume 1. This manual has been updated, but it is also now split in two manuals:

- “*Dymola User Manual 1A: Introduction, Getting Started, and Installation*”, containing the following chapters:
  - What is Dymola?
  - Getting started with Dymola
  - Introduction to Modelica
  - Appendix - Installation
- “*Dymola User Manual 1B: Developing and Simulating a Model*”, containing the following chapters:
These manuals are updated to the new GUI, and have the version “September 2019 (Dymola 2020x)”. These manuals also include all relevant features/improvements of Dymola 2020x presented in the Release Notes. Due to the large changes, change marks in the margin are not used in the manuals for this version.

The manual “Dymola User Manual Volume 2” is not updated in this version; it will be available in the next version, but can also be downloaded earlier, when ready, from http://www.Dymola.com. It is planned to be ready at the end of 2019.

**Documentation for FMI Kit for Simulink**

The latest documentation for FMI Kit for Simulink is from Dymola 2020x not included in the “Dymola User Manual Volume 2” since it is not updated (see above); you can access the documentation from GitHub instead; see section “FMI Kit for Simulink” on page 44.

In the updated version of “Dymola User Manual Volume 2” the documentation for the FMI Kit for Simulink will be removed since it is handled from GitHub.
Appendix – Installation: Hardware and Software Requirements

Below the current hardware and software requirements for Dymola 2020x are listed.

3.10.1 Hardware requirements/recommendations

Hardware requirements
- At least 2 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 6 GB of RAM.

3.10.2 Software requirements

Microsoft Windows

Dymola versions on Windows and Windows operating systems versions
Dymola 2020x is supported, as 64-bit application, on Microsoft Windows 7\(^1\), Windows 8.1, and Windows 10. 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, an Intel compiler, or a GCC compiler, must be installed separately. The following compilers are supported for Dymola 2020x on Windows:

Microsoft C/C++ compilers, free editions:
- Visual Studio 2012 Express Edition (11.0)

---

\(^1\) This version, Dymola 2020x, is the last Dymola version that officially supports Windows 7.
• Visual Studio 2013 Express Edition for Windows Desktop (12.0)
• Visual Studio 2015 Express Edition for Windows Desktop (14.0)
• Visual Studio 2017 Desktop Express (15) Note! This compiler only supports compiling to Windows 32-bit executables.
• Visual Studio Community 2017 (15)
• Microsoft Build Tools 2017 Notes:
  o The minimum required selection to run Dymola is:
    ▪ VC++ 2017 version highest listed latest version-number tools
    ▪ C++/CLI support
  o Installing this selection, no IDE (Integrated Development Environment) is installed, only command line features
  o This installation is not visible as a specific selection when later selecting the compiler in Dymola, the alternative to select is the same as for any Visual Studio 2017 alternative: Visual Studio 2017/Visual C++ 2017 Express Edition (15).
• Microsoft Studio Community 2019 (16)
• Microsoft Build Tools 2019 Notes:
  o The minimum required selection to run Dymola is:
    ▪ VC++ 2019 version highest listed latest version-number tools
    ▪ C++/CLI support
  o Installing this selection, no IDE (Integrated Development Environment) is installed, only command line features
  o This installation is not visible as a specific selection when later selecting the compiler in Dymola, the alternative to select is the same as for any Visual Studio 2019 alternative: Visual Studio 2019/Visual C++ 2019 (16).

**Microsoft C/C++ compilers, professional editions:**
• Visual Studio 2012 (11.0)
• Visual Studio 2013 (12.0)
• Visual Studio 2015 (14.0)
• Visual Studio Professional 2017 (15)
• Visual Studio Enterprise 2017 (15)
• Visual Studio Professional 2019 (16)
• Visual Studio Enterprise 2019 (16)

**Intel compilers**
The Intel compilers Intel Parallel Studio XE 2016, XE 2017, and XE 2018 are supported.
Note that you must also select a Visual Studio compiler when selecting any Intel compiler.
Supported combinations of Intel compilers and Visual Studio compilers:
Intel Parallel Studio XE 2018: Visual Studio 2013, 2015, or 2017

Current limitations:
- Embedded server (DDE or OPC) is not supported.
- Export DLL is not supported.

**GCC compilers**

Dymola 2020x has limited support for the MinGW GCC compiler. The following versions have been tested:
- For 32-bit GCC: version 4.8.1, 5.3, 6.3, and 8.2
- For 64-bit GCC: version 4.9.2, 5.3, 7.3, and 8.1

Hence, at least the versions in that range should work fine.

To download any of these free compilers, please visit [http://www.Dymola.com/compiler](http://www.Dymola.com/compiler) where the latest links to downloading the compilers are available. Needed add-ons during installation 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.
- Support for external library resources is implemented, but requires that the resources support GCC, which is not always the case.
- FMUs must be exported with the code export option\(^2\) enabled.
- For 32-bit simulation, parallelization (multi-core) is currently not supported for any of the following algorithms: RadauIIa, Esdirk23a, Esdirk34a, Esdirk45a, and Sdirk34hw.

**Dymola license server**

For a Dymola license server on Windows, all files needed to set up and run a Dymola license server on Windows using FLEXnet, except the license file, are available in the Dymola distribution. (This includes also the license daemon, where Dymola presently supports FLEXnet Publisher version 11.14. This version is part of the Dymola distribution.)

**Linux**

**Supported Linux versions and compilers**

Dymola 2020x runs on openSUSE 42.2, 64-bit, with gcc version 5.3.1, and compatible systems (see [http://doc.qt.io/qt-5/supported-platforms.html](http://doc.qt.io/qt-5/supported-platforms.html)). Any later version of gcc is typically compatible. In addition to gcc, the model C code generated by Dymola can also be compiled by clang.

\(^2\) Having the code export options means having any of the license features **Dymola Binary Model Export** or the **Dymola Source Code Generation**.
To change compiler, change the variable CC in /opt/dymola-<version>-x86-64/insert/dsbuild.sh. As an example, for a 64-bit Dymola 2020x application:

```
/opt/dymola-2020x-x86_64/insert/dsbuild.sh
```

Dymola 2020x is supported as a 64-bit application on Linux.

Notes

- 32-bit compilation for simulation might require explicit installation of 32-bit libc. E.g. on Ubuntu:
  ```
sudo apt-get install g++-multilib libc6-dev-i386
  ```
- Dymola is built with Qt 5.12.4 and thereby inherits the system requirements from Qt. Some libraries might require explicit installation:
  - `libevent-2.0.so.5`
  - `libpng16-0`
  - `libre2.so.0`
  - `libwebp.so.5`
  - `libwebpdemux.so.1`
  - `libwebpmux.so.1`
- For rendering of jpg files, `libjpeg62` must be installed.
- For FMU export/import to work, `zip/unzip` must be installed.

Note on libraries

- 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 11.14.)