

# Customising Dymola Menus & Startup -- Modelica Library Opener

Raees B. K. Parambu – Senior Project Engineer

**Software/Language used:** Dymola 2019 FD01, MSL 3.2.2

**Keywords:** Custom, Dymola, Menu, Startup, Modelica, Library.

## Introduction

This blog post presents a methodology for opening (loading) a customised set of Modelica libraries in Dymola with a single click. For demonstrating this method, VeSyMA® (Vehicle Systems Modelling and Analysis), a suite of commercial Modelica libraries is used. For making this feature appear by default in the Dymola menus, an additional method is also presented. It is shown that with the aid of the presented method and by carefully calling libraries in a proper order, it ensures that with a single click a customised set of libraries are only loaded into the Dymola Package Browser.

## How to Open a Modelica Library in Dymola

In Dymola, there are three different methods for adding a Modelica library into the **Dymola Package Browser** using the **File** menu, as grouped within a green box in Figure 1. These three methods are as follows:

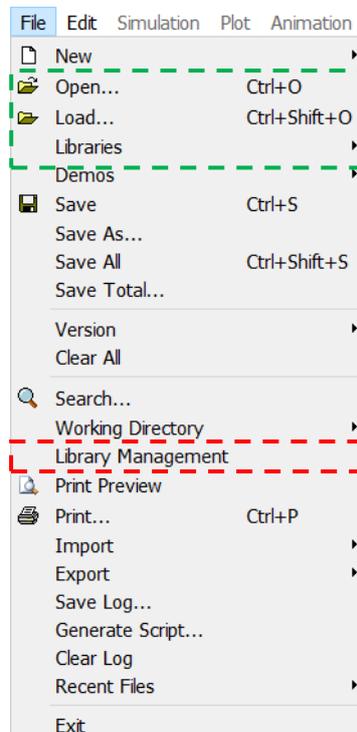


Figure 1– Dymola File Menu.

### *The methods:*

1. **File > Open...** helps to manually browse a **package.mo** file which is normally stored within a Modelica library folder. This **Open** command reads the contents of a file and opens this library in **Dymola Package Browser**. By the end of the reading process, Dymola will automatically change the current working directory to the directory of the opened file.
2. **File > Load...** works in the same way as the first option above but keeps the current working directory.
3. **File > Libraries** display a menu with shortcuts to libraries configured with **Library Management** option marked with a red box in Figure 1.

Using any of the three methods above, we can open an individual and/or multiple libraries in **Dymola Package Browser** manually. If any loaded library uses a class from another library, i.e. it is dependent on it, Dymola will search for those additional libraries in the directories of DYMOLAPATH or MODELICAPATH.

It can be time-consuming to manually load a preferred set of libraries using any of the three methods above. You may also experience version conflicts as Dymola does not permit multiple versions of the same library within a **Package Browser**. Therefore, the rest of this blog post is dedicated to presenting a methodology for opening or loading a customised set of Modelica libraries into **Dymola Package Browser** with a single click.

## Customise Dymola Menus to add a Library Opening Function

When we start a non-customised Dymola session, it typically opens in Modeling tab as shown in Figure 2.

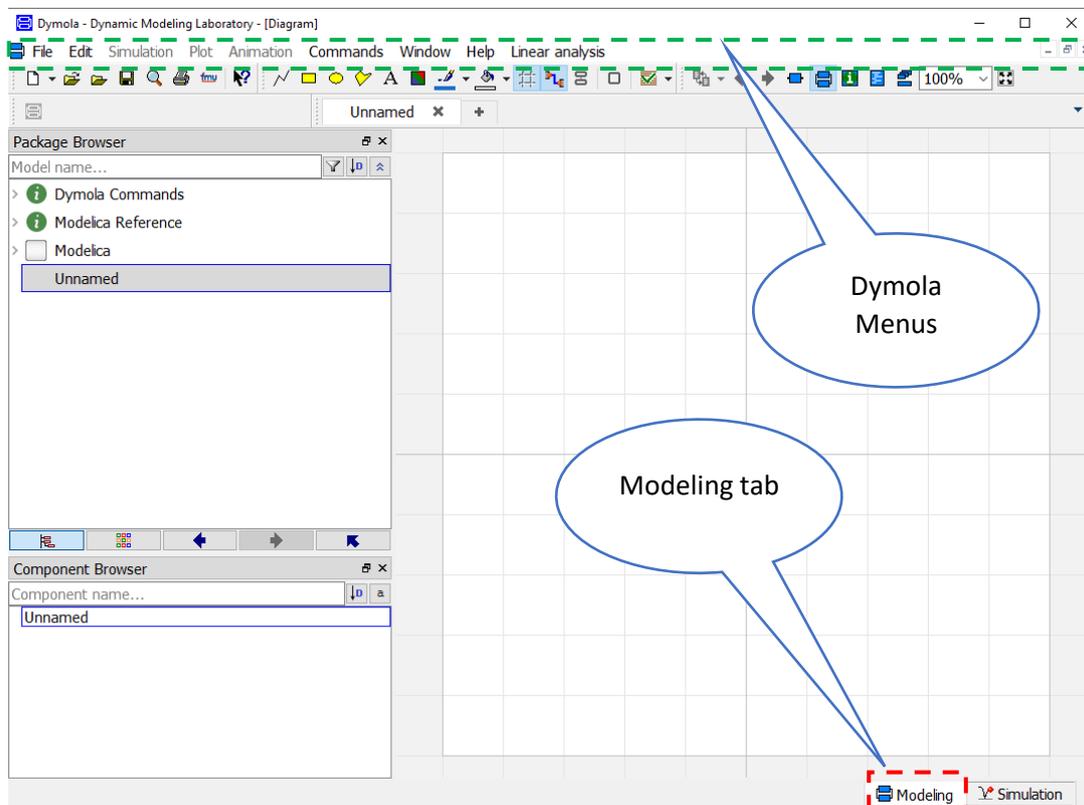


Figure 2– Dymola 2019 FD01 GUI Modeling tab.

However, it is possible to extend this graphical user interface (GUI) of Dymola by configuring your own menus. I will show you how to create a function to open a set of libraries and add this to a custom menu.

The example code provided below will load a selected set of libraries with the help of Modelica functions. It builds upon the previous blog post [Creating a function for loading libraries](#) and is adapted to create a package structure called **CommercialLibraries**.

```
package CommercialLibraries "Commercial libraries"

  extends Modelica.Icons.Package;

  package Tags "Tags"
    function Open = CommercialLibraries.Opener (libraryNames={
      "C:\Claytex\Modelica\Tags\2018.3\Claytex\Modelica\Claytex\package.mo",
      "C:\Claytex\Modelica\Tags\2018.3\Claytex\Modelica\ClaytexFluid\package.mo",
      "C:\Claytex\Modelica\Tags\2018.3\Suspensions\Modelica\Kinematics\package.mo",
      "C:\Claytex\Modelica\Tags\2018.3\VeSyMA\Modelica\VeSyMA\package.mo",
      "C:\Claytex\Modelica\Tags\2018.3\Engines\package.mo",
      "C:\Claytex\Modelica\Tags\2018.3\PTDynamics\Modelica\PTDynamics\package.mo",
      "C:\Claytex\Modelica\Tags\2018.3\Suspensions\Modelica\Suspensions\package.mo",
      "C:\Claytex\Modelica\Tags\2018.3\Suspensions\Modelica\Motorsports\package.mo",
      "C:\Claytex\Modelica\Tags\2018.3\Simulator\Modelica\TerrainServer\package.mo",
      "C:\Claytex\Modelica\Tags\2018.3\Simulator\Modelica\DiL\package.mo"})
      "Open";
    annotation ();
  end Tags;

  function Opener "Opens some of the commercial libraries"

    input String libraryNames[:];
    "Commercial library names";

  algorithm

    // SCRIPT TO OPEN SOME OF THE COMMERCIAL LIBRARIES
    for i in 1:size(libraryNames, 1) loop
      DymolaCommands.SimulatorAPI.openModel(libraryNames[i],
      changeDirectory=false);
    end for;

    //ENSURE THE CORRECT WORKING DIRECTORY IS VALID
    DymolaCommands.System.cd("C:\Users\Raees\Documents\Dymola");

    annotation ();
  end Opener;
  annotation (__Dymola_menu=true,
  Protection(hideFromBrowser=false),
  uses(Modelica(version="3.2.2"),
  DymolaCommands(version="1.6")));
end CommercialLibraries;
```

Note: The above Modelica code can be downloaded [here](#) and saved to your preferred location.

This package has a subpackage, **Tags** and function, **Opener**. The **Tags** package has its own subpackage named as **Open**. The `function` `Open` provides the path of the libraries, and the `function` `Opener` executes the opening task.

By using one of the aforementioned library opening methods (say, **File > Open...**), the above downloaded code (**CommercialLibraries**) can be loaded into **Dymola Package Browser** and it will look like the one shown in Figure 3.

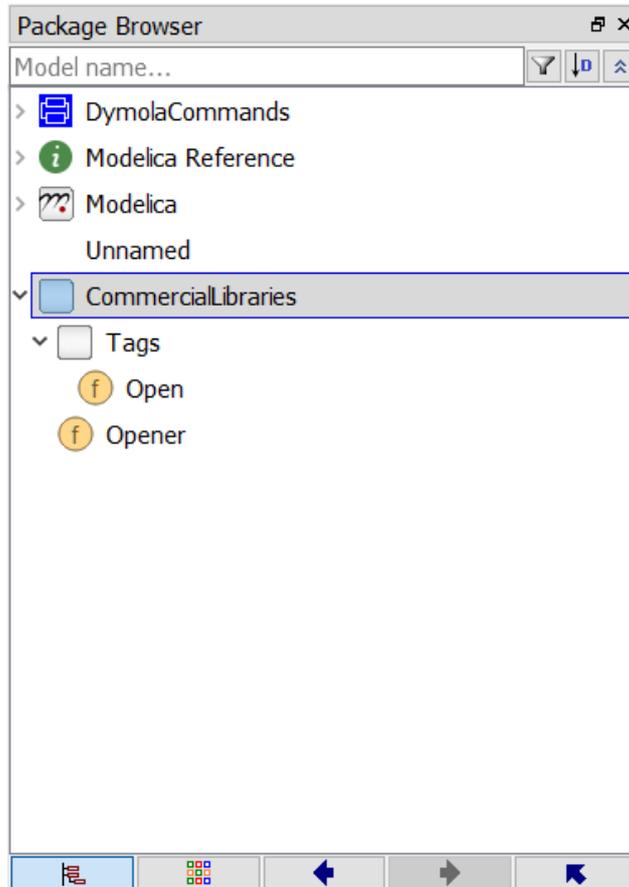


Figure 3 – The fully expanded view of the **CommercialLibraries** package in **Dymola Package Browser**.

Comparing to Figure 2, at the end of the provided code, `'annotation (__Dymola_menu=true, '` helps to include an additional menu in the Main window Modeling tab of Dymola as highlighted in Figure 4.

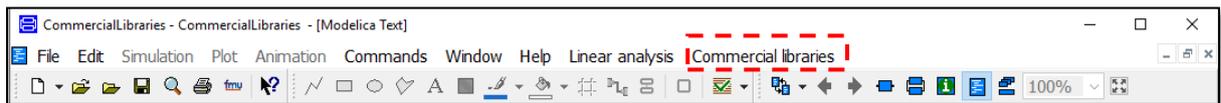


Figure 4 – A cropped & customised version of Figure 2.

Figure 4 depicts cropped and customised Dymola menus after opening the newly created **CommercialLibraries** package in the **Dymola Package Browser**.

Moreover, in the code, by changing `'annotation (. . . , Protection (hideFromBrowser=false), '` to `'annotation (. . . , Protection (hideFromBrowser=true), '` and then clicking anywhere within **Dymola Modeling Window**, the recently manually opened **CommercialLibraries** package will hide from the **package browser** as in Figure 5, but it will continue to exist in the Dymola Modeling Window until the closing of this Dymola session.

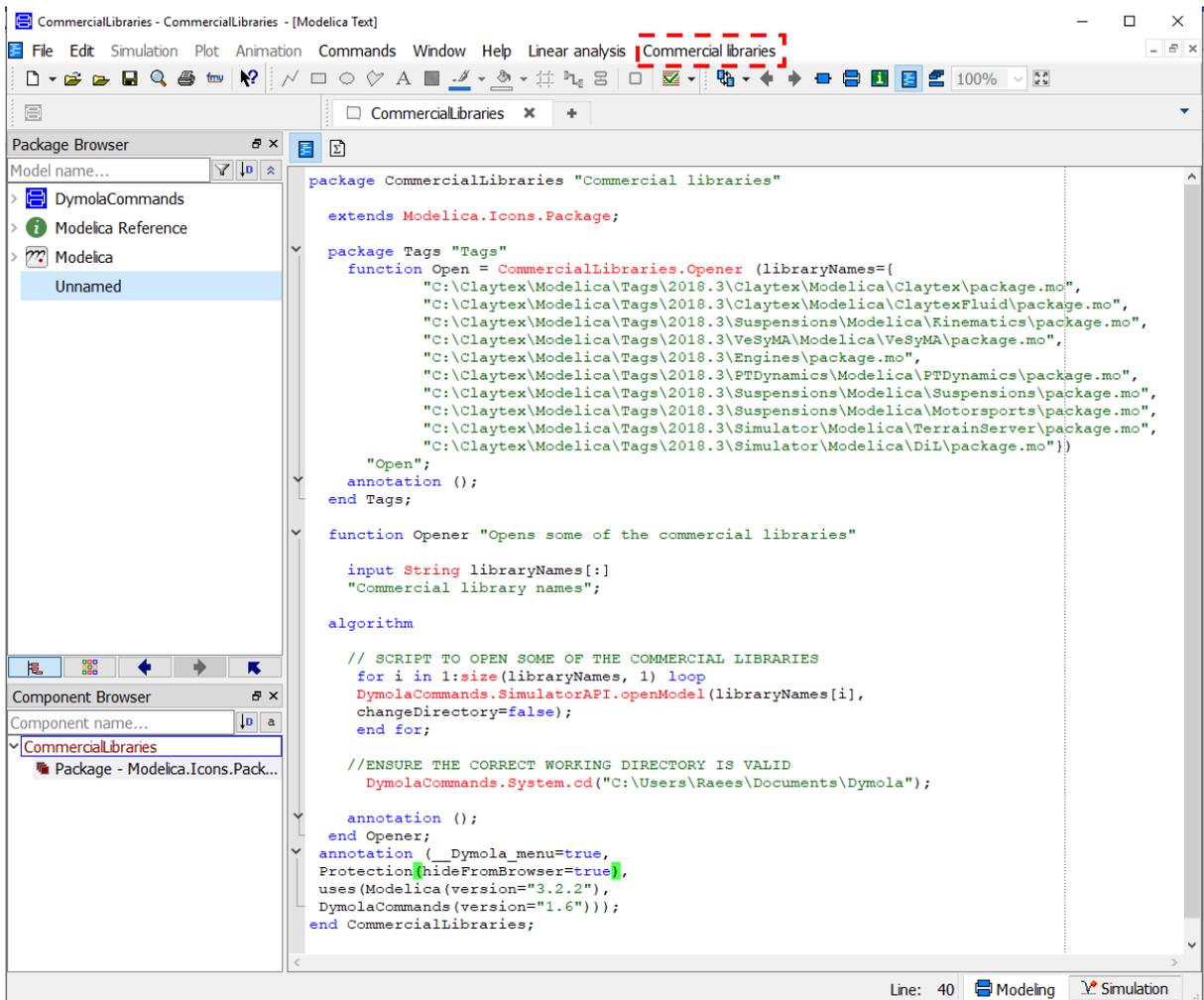


Figure 5 – Edited *CommercialLibraries* code and the package is no longer visible in the *Dymola Package Browser*.

To be able to view this hidden package in the **Dymola Package Browser**, without editing the code, please refer to Appendix A. Optionally, you can save the above changes in *CommercialLibraries* using File > Save option in Dymola, before navigating through the rest of the blog.

Figure 6 shows selection options available under *CommercialLibraries* package.

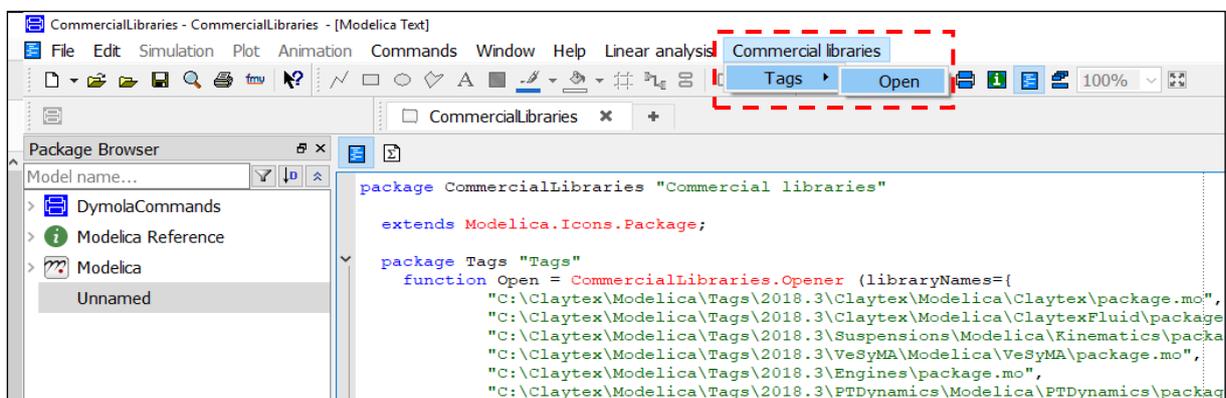


Figure 6 – Selection options of *CommercialLibraries* package.

By clicking on **Open**, in Figure 6, the following dialog box will appear:

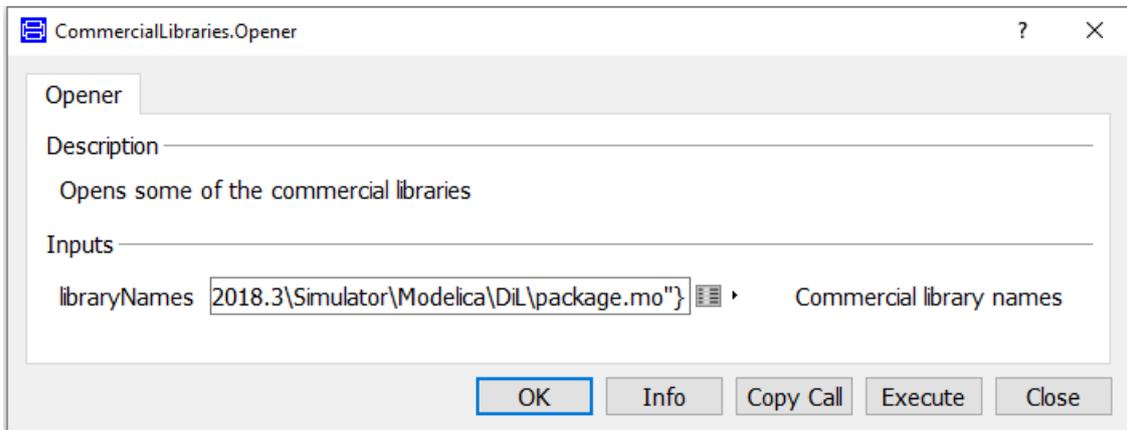


Figure 7 – Dialog box generated by clicking on **Open** in Figure 6.

If we press **Execute** or **OK**, Dymola will call the **Open** function which in turn calls **Opener** function in the provided code. This action will automatically switch Dymola from the Modeling tab to the Simulation tab and the following outputs will be printed in the **Commands** section of the Dymola simulation tab.

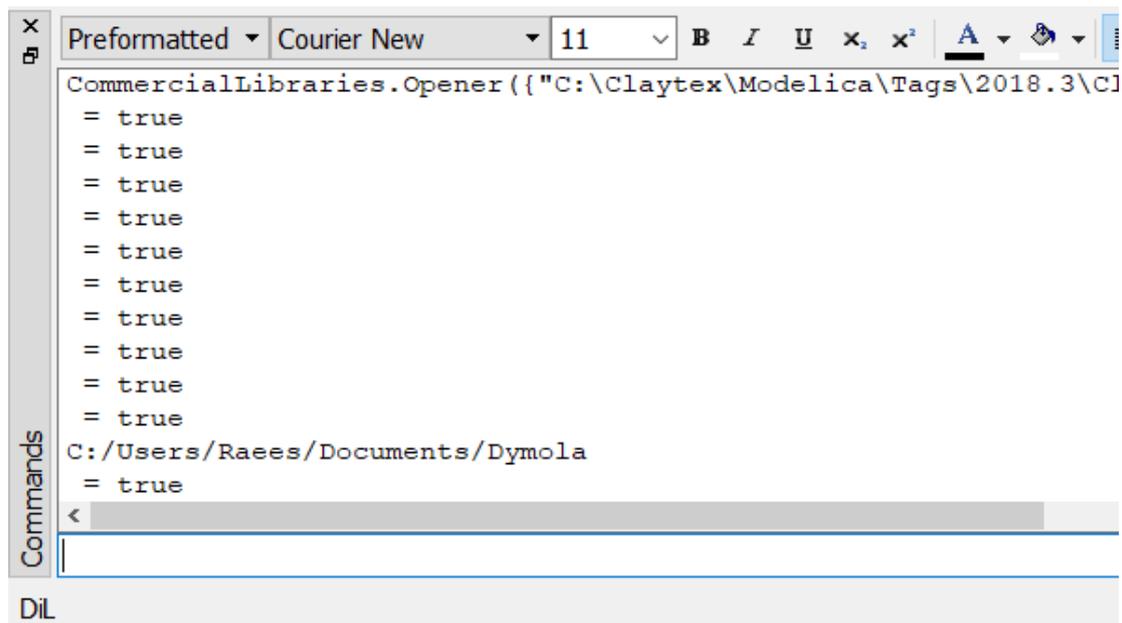


Figure 8 – Command log pane of the Dymola command window.

By switching back to the Dymola Modeling tab, the following list of commercial libraries will be opened in the **Package Browser**.

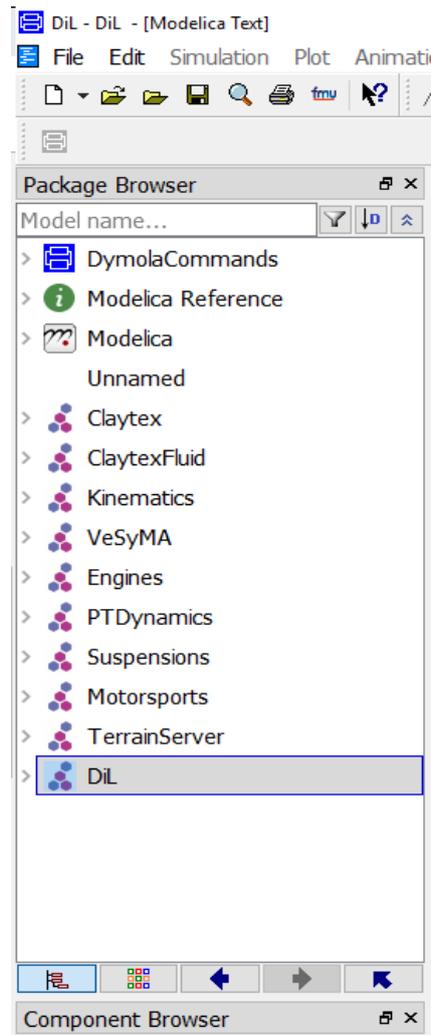


Figure 9 – Requested commercial libraries opened in **Dymola Package Browser** with a single click.

## Customise Dymola Startup Configuration

To make the **CommercialLibraries** package appear by default in the Dymola menus for each new Dymola session, the **dymola.mos** file needs to be modified. Prior to doing this, it is recommended to create a backup copy of the '<Dymola installation directory>\insert\dymola.mos' file, the original folder look like the one shown below.

Name	Date modified	Type	Size
displayunit	18/05/2018 10:40	MOS File	4 KB
displayunit_us	15/03/2017 10:34	MOS File	3 KB
dymola	28/11/2018 17:07	MOS File	1 KB
dymola5_ja.qm	26/09/2018 05:51	QM File	250 KB
dymosim	04/12/2002 18:20	Configuration sett...	8 KB
qtbases_ja.qm	26/05/2017 20:18	QM File	127 KB
splash	04/03/2004 00:21	PNG File	28 KB
ss2dymAll.cx	27/09/2018 12:15	CX File	1,318 KB
style1	12/10/2015 18:38	Cascading Style S...	3 KB
tire	14/07/2003 16:07	GIF File	3 KB

Figure 10 – Dymola 2019 FD01 the original ‘insert’ folder.

This can be done by the following steps;

- Make a copy of **dymola.mos** from the folder <Dymola installation directory>\insert and save it in a non-write-protected area of your hard drive.
- Right-click the copied **dymola.mos** and select Open with > Notepad.

By default, the script should look something like the one below:

```

dymola - Notepad
File Edit Format View Help
RunScript("$DYMOLA/insert/displayunit.mos", true);
// RunScript("$DYMOLA/insert/displayunit_us.mos", true);
definePostProcessing("SDF output", "Convert result file to SDF format",

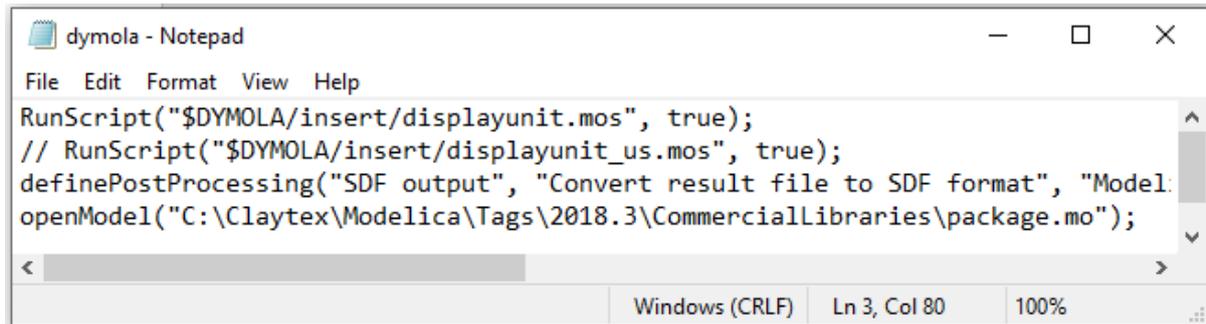
```

Figure 11 – The original *dymola.mos* file.

Add the following line to the end of the script, but please do not forget to change the path of the folder depending on where you saved the downloaded **CommercialLibraries** package,

```
openModel("C:\Claytex\Modelica\Tags\2018.3\CommercialLibraries\package.mo");
```

Then the modified script looks like the figure below.



```
File Edit Format View Help
RunScript("$DYMOLA/insert/displayunit.mos", true);
// RunScript("$DYMOLA/insert/displayunit_us.mos", true);
definePostProcessing("SDF output", "Convert result file to SDF format", "Model:
openModel("C:\Claytex\Modelica\Tags\2018.3\CommercialLibraries\package.mo");
Windows (CRLF) Ln 3, Col 80 100%
```

Figure 12 – The modified *dymola.mos* file.

Followed by save and close **dymola.mos**. Then, go back to the **insert** directory, rename the existing **dymola.mos** file, then manually copy the edited **dymola.mos** file back to the folder <Dymola installation directory>\insert.

Note: To perform the above steps you must have administrator rights.

Now open a new Dymola session, a window similar to Figure 5 will be opened, but the code may be hidden if you were previously saved *CommercialLibraries package after editing the code*.

## Conclusion

This blog post presented a methodology for opening/loading a customised set of Modelica libraries into the **Package Browser** of Dymola with a single click and demonstrated a method to customise Dymola startup configuration.

## Appendix A:

### How to View Protected Modelica Packages in Dymola Package Browser

In order to see the hidden Modelica package in the **Dymola Package Browser**, go to the Package Browser option available in Edit > Options... command in the Main window Modeling tab of Dymola. After clicking on the Edit > Options..., the following popup window will appear.

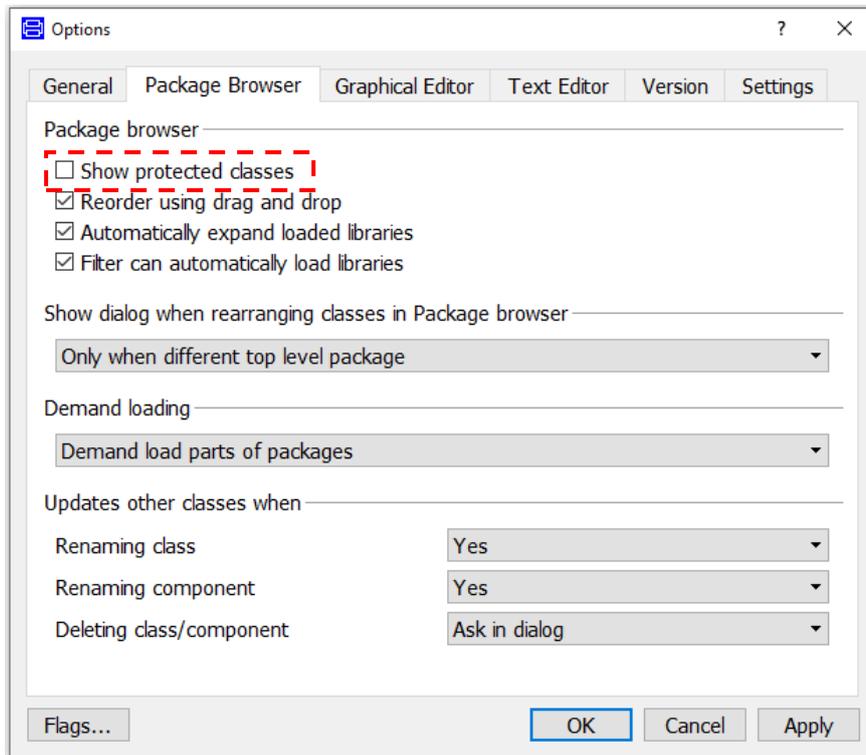


Figure A – Edit > Options... popup window.

In Figure A, tick **Show protected classes** and then click **OK**. This will open all the protected classes that include *CommercialLibraries* and other packages in **Dymola Package Browser**.