Making Easily Accessible Modelica Examples

Sometimes users have difficulties getting started with Dymola and Modelica, and a recurring theme is that they also had difficulties running the examples in the libraries.

The purpose of this document is to collect tips and tricks (“best practices” with a nice word), to improve the experience for the people who initially meet Modelica.

Whom do we write for?

A key question when writing any text or creating an example is who we intend it for, and what questions these people want to have answered. The purpose of an example is to give the potential user a favorable first impression, and once “hooked” to make the user productive.

The example typically serves two purposes for the potential user:

► To demonstrate that the tool and the library can solve meaningful problems for the user, in a manner that is more convenient and more efficient than any other tool can.

► To demonstrate best practices of using Modelica for both component design and application-level model design. An important aspect here is for the library to provide suitable interfaces (connector types).

In some respects, designing a good example is like designing a good graph summarizing a large amount of data†.

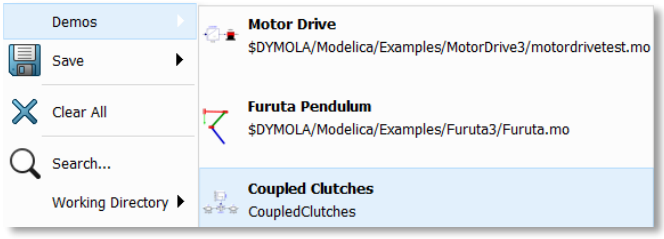
The users who evaluate Dymola have very different backgrounds. Some are Modelica experts, some are domain experts, and some are not experts at all.

The bottom line is this: the user may be an expert but most likely, he or she is much less experienced than you are; that makes your job harder.

Where to put examples

Example models are typically stored in a package, commonly called *Examples* or *Experiments,* one level below the library package itself. These conventions are easily recognized, so we recommend following them.

To highlight key examples, you may add them to the *File>Demos* menu, in addition to adding your library to *File>Libraries*.



The details of setting it up are described in the Dymola User Manual, Volume 1A.

Model documentation

The example documentation needs to answer several questions typically asked by the user.

► Why is this example relevant? What does it demonstrate that is relevant either for my application or for solving typical sub-tasks in my work?

► What can I do with it? What typical scenarios have been prepared for me (typically, using the Commands menu)?

► What can I experiment with? If I want to go deeper, what parameters are suitable to tune (or sweep) to show something interesting? What do I need to plot to see the effect?

► What can I expect the simulation result to be? Especially for those who just quickly browse the examples without running them, it is helpful to see at least one screenshot of typical results in the documentation. It also helps the user to verify that he or she understood and used the example correctly.

As the bare minimum, every example should provide

► A one-line description text, which shows up as a tooltip in the package browser.

► A short paragraph describing what it demonstrates, and what it doesn't.

► Short instructions to run the examples, even if it is as simple as pressing the Simulate button.

► A description of what parameter to change and what variables to plot, to make it interesting to more explorative users.

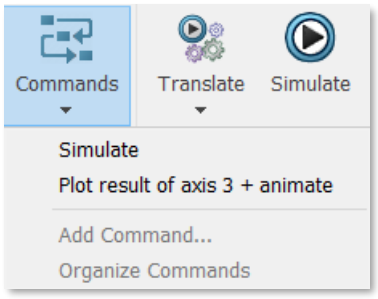
Simulation setup

We strongly recommend storing simulation setup attributes in the model, such as, stop time, integration algorithm and tolerance, and number of output points. Enough information so that just pressing the Simulate button produces a reasonable result.

If the model requires some unusual simulation setup, for example, a particular integration method or a very tight tolerance, the reason should be documented too.

Model commands

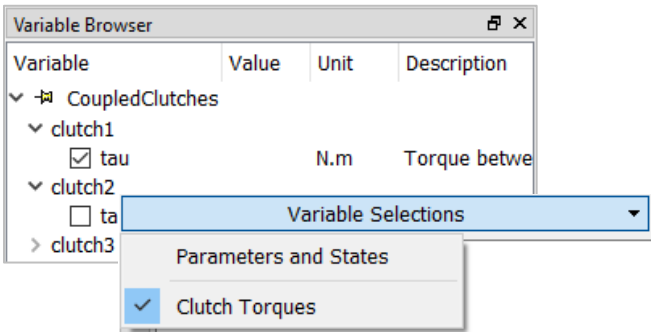
We suggest using the Commands menu to provide typical simulation, parameter sweep and result visualization scenarios. This gives the user a single place to find the most common operations, suitable for going through a number of examples before deep diving into the most interesting ones.



Note that for e.g. a plot operation, you can set a flag that ensures that Dymola will translate and simulate the model first. That avoids many user errors.

Variable selections

It is possible to create variable selections that greatly simplify navigation of the results after a simulation. Such selections highlight a group of variables while hiding others.



Variable selections are defined in a dedicated dialog in the Dymola editor, and stored as model annotations.

Conclusion

Good and easily accessible examples are key to illustrating the capabilities of a model library to new users. With a few simple rules, the quality and user impression can be improved.

First impressions matter!

† *Historical note*

A graph designed by Charles Joseph Minard in 1861 summarizes Napoleon's disastrous Russian campaign, and how the 422,000 men strong army was reduced to 10,000. It relates six variables against the background of the Russian map, and is sometimes called "the best statistical graph ever".

For more on the story of Charles Joseph Minard, see https://en.wikipedia.org/wiki/Charles_Joseph_Minard