ARMINES

OMOptim developer documentation

Hubert Thieriot – CEP ARMINES

21/08/2012

(Windows version) How to download code ? How to compile it ? How to create an installer ?

Sommaire

Prepare your workstation for OMOptim development

Getting source code

OMOptim source code is stored on a subversion repository (cf. Annex).

In below instructions, MyDevFolder refers to a newly created folder where source code will be stored. Please use a folder without space character within its path.

Create a new folder for source code, without space character.

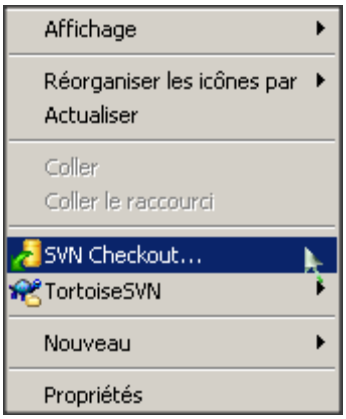
How to get it ?

OMOptim can be found on a dedicated subversion server.

The repository address is https://openmodelica.org/svn/OpenModelica/trunk/

A login and a password should first be obtained through registration. Or one could use *Anonymous* as user and *none* as password.

Right click in MyDevFolder and ask for a Checkout.



Do a check out of https://openmodelica.org/svn/OpenModelica/trunk/ in MyDevFolder

OMDev

OMDev is a folder containing needed libraries to compile OMOptim. It is available through subversion here https://openmodelica.org/svn/OpenModelica/installers/windows/OMDev/ . Login: *anonymous*. Pass : *none*.

- Do a check out of https://openmodelica.org/svn/OpenModelica/installers/windows/OMDev/ in MyDevFolder/OMDev.
- Add an environment variable called OMDEV with MyDevFolder/OMDev as a value

Getting Qt

Download Qt sdk : http://qt.nokia.com/products/qt-sdk/

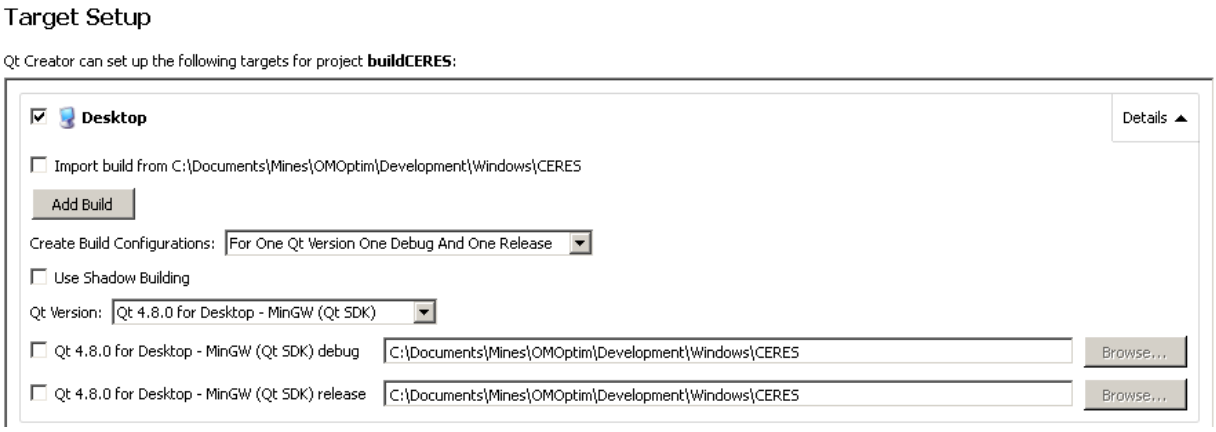
Select custom install and choose **Qt 4.8.0 Mingw libraries** (or newer version)

Download and install Qt sdk

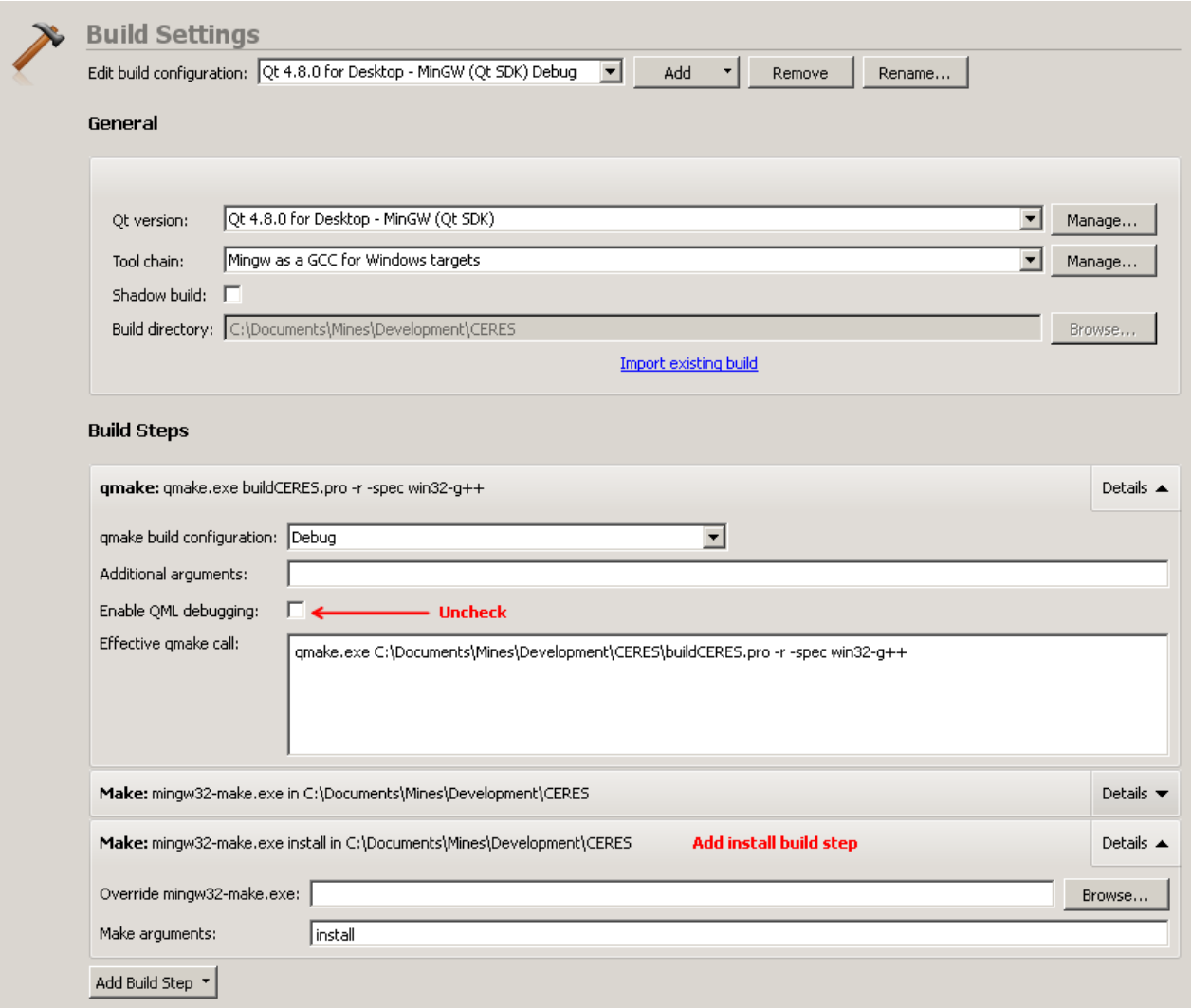
Setting up Qt project

Launch QtCreator and open OMOptim/buildOMOptim.pro

Select as following building options (be sure to unselect *Use Shadow Building)*



In *Projects* tab (on left pane), disable QML debugging and add an install build step (click on *Add build step* -> *Make* and set *install* as Make arguments).



Build documentation

Code documentation can be build from code itself and its commentaries. Download *Doxygen* from here: http://www.stack.nl/~dimitri/doxygen/download.html. To build inheritance diagrams, it will also require graphviz package, available at www.graphviz.org.

With doxygen application (doxywizard), open OMOptim/Doc/doxyfile and run documentation generation. An html folder will then be created in OMOptim /Doc folder. Documentation is now available starting from index.html. Within this documentation, one could find every function available in different libraries.

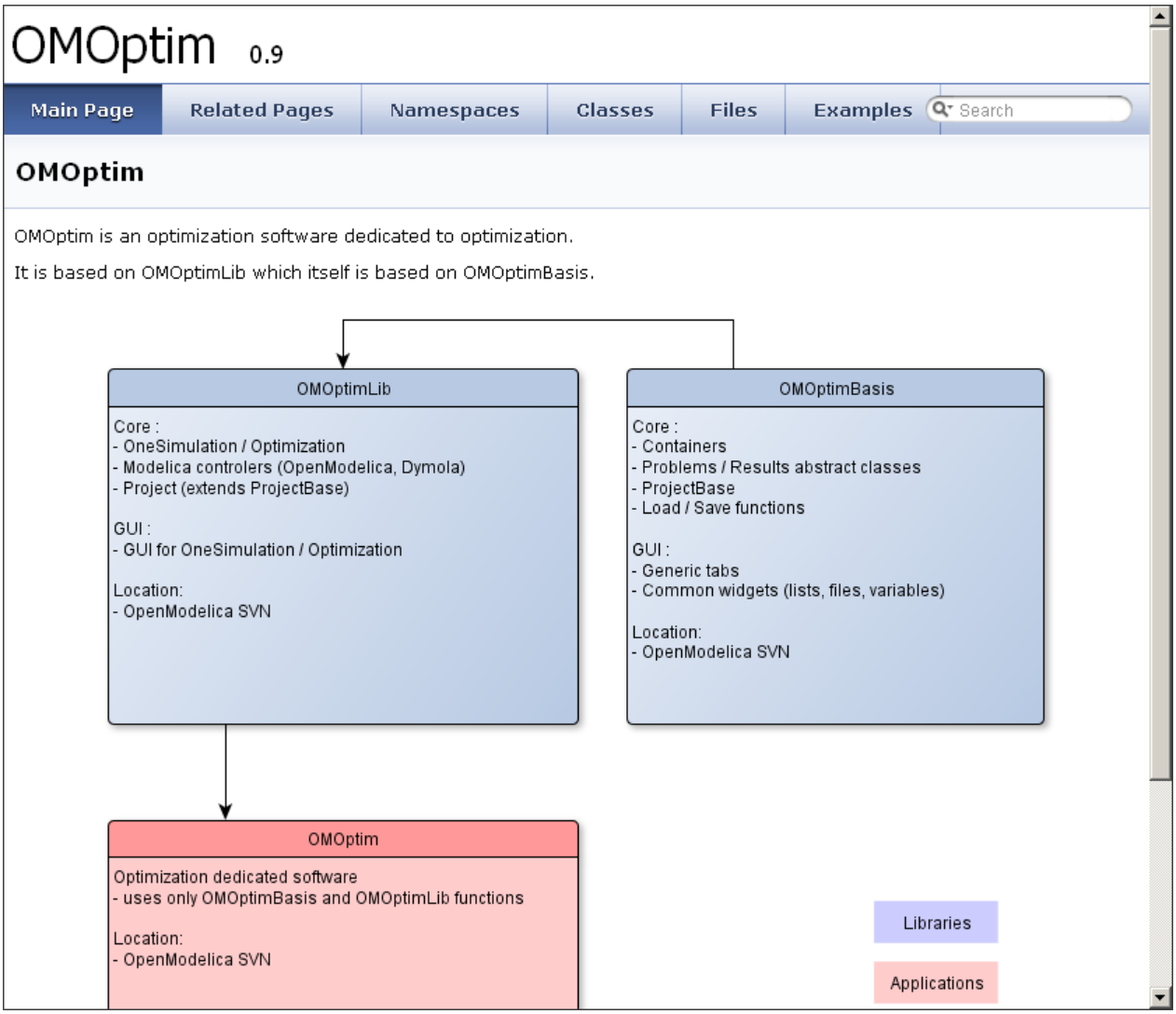


Figure 1 - Main page of CERES code documentation

Building an update

OMOptim is installed with OpenModelica. However, you can build a separate installer for OMOptim[1]. Download and install InnoSetup Quickstart pack (http://www.jrsoftware.org/isdl.php). **Be sure to install the QuickStart pack, not the standard version.**

You now just have to open OMOptim/Tools/Installer/Update.iss . Then, press Build/Compile. It will create a new setup wizard available in Output folder.

When executed, the wizard will replace current OMOptim version (in OpenModelica folder) with the new one.

[1] OMOptim requires OpenModelica. But you can replace an older version of OMOptim using the built wizard.


Setup - OMOptim
Welcome to the OMOptim Setup Wizard
This will install OMOptim 0.9.12606 on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
Next >
Cancel

How code is structured ?

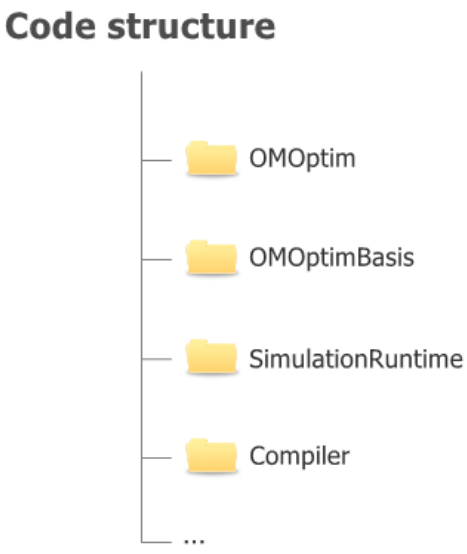


Figure 2 - Folders structure

- OMOptimBasis contains basic functions:
  - Data management
  - Basic widgets, tabs, dialogs
  - File controls
  - ProjectBase class
- SimulationRuntime and Compiler are part of larger OpenModelica project. Those are required for OMOptim compilation.
- OMDev is a folder containing other needed libraries (for OMOptim compilation). **One should set an environment variable (called OMDEV) referring to OMDev folder (e.g. C:\OMDev).**

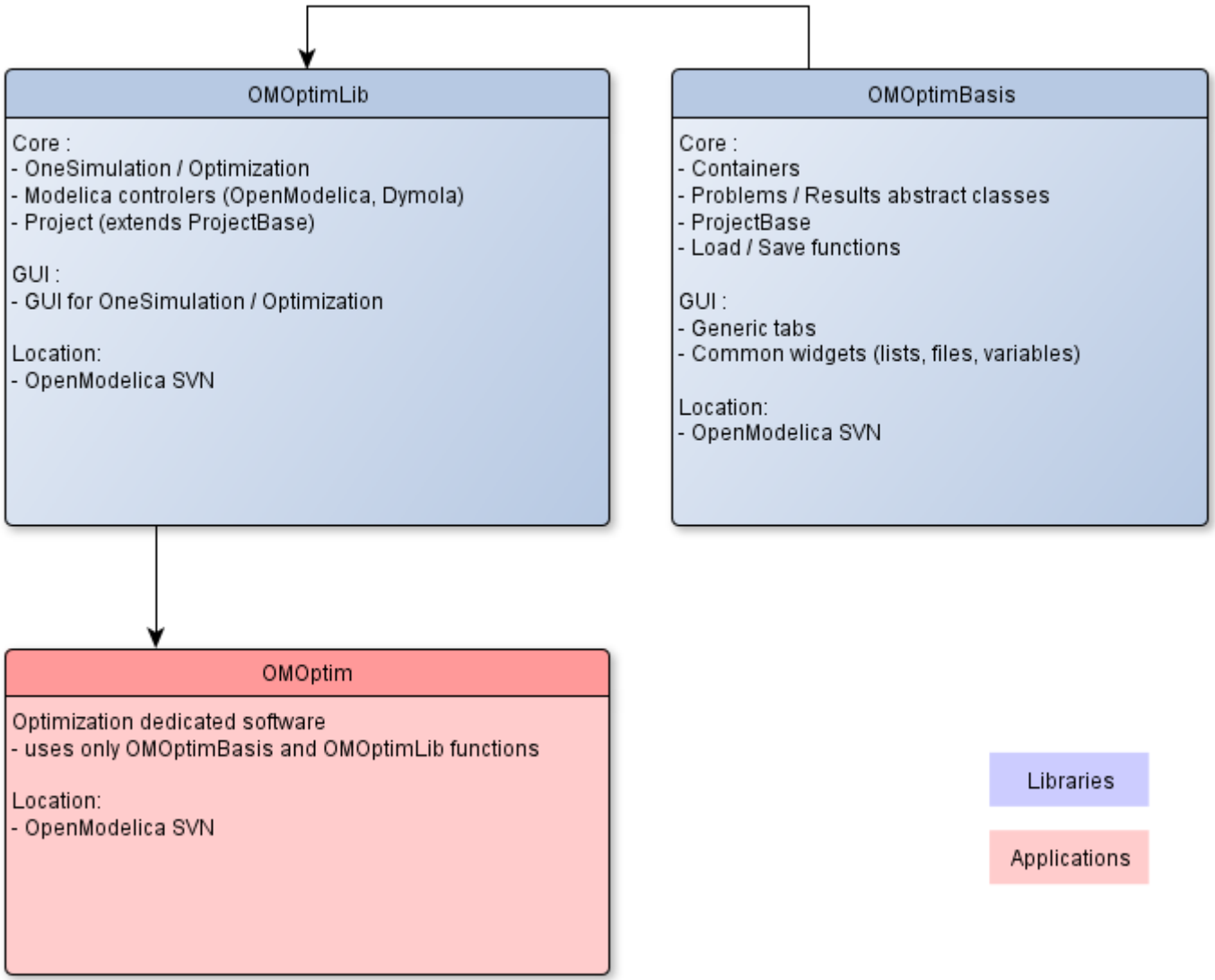


Figure 3 - Code structure: libraries and applications

Annex

Using subversion (svn)

What is subversion ?

(From svnbook.red-bean.com)

Subversion is a free/open-source version control system. That is, Subversion manages files and directories, and the changes made to them, over time. This allows you to recover older versions of your data, or examine the history of how your data changed. In this regard, many people think of a version control system as a sort of "time machine".

Subversion can operate across networks, which allows it to be used by people on different computers. At some level, the ability for various people to modify and manage the same set of data from their respective locations fosters collaboration. Progress can occur more quickly without a single conduit through which all modifications must occur. And because the work is versioned, you need not fear that quality is the trade-off for losing that conduit—if some incorrect change is made to the data, just undo that change.

How to use it ?

On Windows, Tortoise SVN is an ergonomic and powerful tool for subversioning. It can be downloaded here : http://tortoisesvn.tigris.org

Basic commands :

- Checkout : use *checkout* command to download a folder from a svn *repository*. It will create a *working copy* on your local computer.
- Update : Using *Update* command, source code modifications from other users will be applied to your local source code.
- Commit : Using *Commit* command, all the modifications you've made locally will be applied to server folder, and therefore, to other connected users (after they have execute *Update* command).

Here are more detailed *how-to*:
http://svn-ref.assembla.com/subversion-how-tos.html