



Compiling V-REP and related items

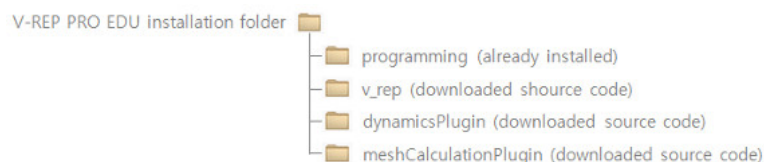
The full source code of V-REP, the DYNAMICS PLUGIN and the MESH CALCULATION PLUGIN may be downloaded from the [coppelia robotics website](http://www.coppeliarobotics.com). While V-REP is GNU GPL licensed, the DYNAMICS PLUGIN and the MESH CALCULATION PLUGIN have a specific license attached. In short, only EDUCATIONAL ENTITIES (students, teachers, professors, schools or Universities) may download and use those plugins without explicit permission. Make sure you understand and comply with the [licensing conditions](#) before doing so.

Other source code items (e.g. various projects, plugins, interfaces, etc.) that are included in the installation packages of V-REP PRO and V-REP PRO EDU are not discussed here.

V-REP, the DYNAMICS PLUGIN and the MESH CALCULATION PLUGIN are [Qt](#) projects, and will require you to have Qt installed on your computer. Try to comply with following list of requirements in order to encounter least problems during compilation:

- if possible, try using the same Qt version as the one that was used to compile the binary V-REP PRO EDU (click [Help --> About V-REP PRO EDU...] to display the Qt version)
- if possible, try compiling the projects using [Qt Creator](#).
- under Windows, use the MinGW or the MSVC compiler. Under Mac OSX and Linux, use the GCC compiler.
- Ubuntu is the preferred Linux distribution.
- you will need to download and compile [QScintilla2](#) (C++ editor control).
- you will need to download and install the [Boost C++ library](#).
- you will need to download the [Lua5.1](#) binaries and headers.

Download and install V-REP PRO EDU. Then download the V-REP source code (and optionally the plugin source codes) into the V-REP PRO EDU installation folder. You should have following folder structure:



When using V-REP in conjunction with a plugin, make sure to use the source code / binaries that come from the same release, in order to avoid incompatibilities. Finally, before being able to compile the V-REP project, you will have to adjust various paths in the `v_rep.pro` project file, such as:

- the path to your Boost library folder
- the path to your QScintilla/Qt4 folder
- the path to your QScintilla folder
- the path to your Lua5.1 folder
- the location of your lua5.1 library file
- the location of your libqscintilla2 library file

V-REP compiles to a shared library. The default client application that loads and runs the library is "vrep" or "vrep.exe". You can use the precompiled one, or recompile it yourself (refer to following project files: `programming/v_repClientApplication` or `programming/windowsOnlyProjects/v_repClientApplication`). The load operation of the library is successful only if:

- the client application could find all needed functions in the shared library
- the shared library could find all shared libraries that itself relies on (e.g. the Lua shared library)

The easiest for you to do is to copy the compiled libraries into the V-REP PRO or V-REP PRO EDU folder, then you will be sure that V-REP should start and won't complain about missing dependencies.

It is very important that you do not mix various Qt versions, or binaries from various compilers. If your main V-REP library was compiled with Qt X and compiler Y, then all plugins related to V-REP should also have been compiled with Qt X and compiler Y, otherwise, you will experience strange behaviours (library that cannot load, sudden crashes, etc.). This requirement can be relaxed if your plugin is not making use of any Qt function.

Since release 3.3.0, you can also compile V-REP for headless mode operation, without any Qt dependencies. In that case however, the vision sensors will only generate an image content when the rendering is handled via a plugin (e.g. POV-Ray or and external renderer).

Windows 8, V-REP V3.0.5, Qt V5.1.1

Try following next few points in order to correctly install Qt 5.1.1, VS2010 and the debuggers:

- Install VS2010
- Install VS2010 service pack1

- Install debugging tools for Windows as part of the Windows SDK
- Install Qt5.1.1 via the Qt online installer for windows
- Set-up the debugger in Qt Creator. Basically: **Projects**-button --> Manage Kits... --> select **Desktop Qt 5.1.1 MSVC2010 32bit**. Then click **Auto-detect**. The exclamation mark next to **Desktop Qt 5.1.1 MSVC2010 32bit** should have disappeared.
- Refer to [this page](#) for more information about launching the debugger.
- You might also have to: **Tools** --> **options** --> **Debugger** --> **CDB** --> **Symbol paths** and click **Edit...**, then **Insert** and select **Symbol Server...** then **Ok**.
- Specify the executable for the debug sessions: **Projects** --> **Run** --> **Executable:** ...\\minGwDebug\\debug\\vrep.exe

Ubuntu Precise 12.04 LTS 64-bit, V-REP V3.0.3, Qt V4.8.1

Following instructions are courtesy of David Butterworth:

If you can, run V-REP once, from within its installation directory:

```
$ ./vrep.sh
```

Check the *About* menu to see what version of Qt is used. Then download a fresh copy of V-REP PRO EDU. Download the V-REP source, and extract it to an installation directory (e.g. "V-REP_PRO_EDU"). Then check what version of Qt4 is available from the Ubuntu repo:

```
$ sudo aptitude show libqt4-core
```

e.g. v4.8.1, which is what we want. Now install the dependencies:

```
$ sudo apt-get install qt4-dev-tools libqt4-dev libqt4-core libqt4-gui
$ sudo apt-get install lua5.1 lua5.1-doc lua5.1-lgi lua5.1-lgi-dev lua5.1-policy lua5.1-policy-dev
$ sudo apt-get install liblua5.1-0 liblua5.1-0-dbg liblua5.1-0-dev liblua5.1-dev
$ sudo apt-get install libboost-all-dev
$ sudo apt-get install qtcreator qtcreator-dbg qtcreator-doc
```

Don't install QScintilla2 from the repo, it doesn't include the header files we need. Instead, download *QScintilla-gpl-2.7.1.tar.gz* from [here](#). Fix the directory permissions and compile:

```
$ chown -R username:username QScintilla-gpl-2.7.1/
$ cd Qt4Qt5
$ qmake qscintilla.pro
$ make
$ sudo make install
```

Check what version of Qt that QMake is using:

```
$ qmake -v
QMake version 2.01a
Using Qt version 4.8.1 in /usr/lib/x86_64-linux-gnu
```

which is what we want. In the V-REP source directory, edit *v_rep.pro* to something similar to:

```
unix:!macx {
    INCLUDEPATH += /usr/lib                                # Boost dir.
    INCLUDEPATH += /usr/include/qt4                        # QScintilla "Qt4" dir.
    INCLUDEPATH += /usr/include/qt4/Qt5                   # QScintilla "include" dir.
    INCLUDEPATH += /home/USERNAME/Downloads/QScintilla-gpl-2.7.1/include # for ScilExer.h
    LIBS += /usr/lib/x86_64-linux-gnu/libqscintilla2.so     # QScintilla2 lib.
    contains(QMAKE_HOST.arch, x86_64):{                    # 64 Bit
        INCLUDEPATH += /usr/include/lua5.1                # lua5.1 64 "include" dir.
        LIBS += -llua5.1_4_Linux26g4_64_lib/ -llua5.1
        DEFINES += QT_VREP_64BIT
    }
    !contains(QMAKE_HOST.arch, x86_64):{                   # 32 Bit
        INCLUDEPATH += /usr/include/lua5.1                # lua5.1 32 "include" dir.
        LIBS += -llua5.1_4_Linux26g4_lib/ -llua5.1
    }
    DEFINES += LIN_VREP
}
```

Now compile V-REP: run Qt Creator, load the project file *v_rep.pro*, or go to the directory where *v_rep.pro* is located, and type:

```
$ qmake v_rep.pro
$ make
$ sudo make install
```

Which should install */usr/lib/libv_rep.so***