PINT – Compilation instructions

Please note that this guide does not cover deployment of PINT nor does it cover how to modify Qt libraries and build the used custom libraries from Qt source code.

Requirements

- **Qt 5.7.0**
  PINT was built using Qt Creator 3.6.83 with Qt 5.7.0 libraries.
  Download link: https://download.qt.io/official_releases/qt/5.7/5.7.0/

  Note! Any Qt version greater or equal to Qt 5.7.0 should be able to compile PINT. Lesser versions lack the required Qt Data Visualization libraries. However, we recommend you to use Qt 5.7.0 as PINT uses a modified QtDataVisualization library that has been compiled for this version of Qt.

- **PINT source code**
  Download link: https://pint-nmr.github.io/PINT/
  The latest version of the source code will ALWAYS be updated to this page, for older versions please check the release page (https://github.com/PINT-NMR/PINT/releases).

- **OpenGL graphics support**
  Mac and Windows: This is natively supported.
  Linux: Type "ldd --version" in a terminal. You should be able to compile PINT if your GNU libc version is greater or equal to 2.12. Updating this version is not trivial and will not be covered in this guide.
  Ubuntu: PINT was successfully compiled on Ubuntu using the libgl1-mesa-dev package.

- **Compiler**
  The selected compiler must have c++11 support. There may be unlisted compilers that will work. The listed compilers are those that have been successfully used with PINT.
  Mac: Latest version of Xcode and clang
  Windows: mingw (e.g. mingw49_32) NOTE! MSVC is not compatible with PINT.
  Linux/Ubuntu: gcc (e.g. gcc 4.7.2-5)

Qt installation

During Qt installation it is required to include "Qt Data Visualization".
Please note that this is not selected by default on Qt 5.7.0.

After Qt installation, you need to replace the Qt5DataVisualization library with the one used in PINT. To do this do the following steps:

- Navigate to "Qt_Custom_Libraries" in the PINT source code directory.
- Copy one of the QtDataVisualization libraries:

  -Mac: Copy QtDataVisualization.framework
  -Windows: Copy Qt5DataVisualization.dll
  -Linux/Ubuntu: Copy libQt5DataVisualization.so.5.7.0

- Navigate to your Qt installation directory and find this directory (or similar):

  -Windows: Qt5.7.0\5.7\mingw49_32\bin
  -Linux/Ubuntu: Qt5.7.0/5.7/gcc_64/lib
  -Mac: Qt5.7.0/5.7/clang_64/lib

- Make a backup of the existing Qt Data Visualization library, e.g. add the extension .bak to the file, in this directory and paste the custom library.

Setting up Qt and building PINT

- Open Qt creator located in Qt5.7.0/Tools/QtCreator/.
- In the menu bar select Tools>Options
- In Kits, make sure that the selected compiler is the correct one; then click Ok.
- Select File>Open file or project and open pint.pro in the source code folder.
- After Qt has parsed the files, click the Project icon to the left.
- Change Edit build configuration from Debug to Release
- Make sure that Shadow build is not checked (otherwise changes made to .ui files will not be updated when building)
- Click Edit to the left, right-click pint.pro and select build.

If all goes well PINT will compile without any error codes.
However, the compiler may still have compiler specific warnings.
Right-click pint.pro and select Run to execute the built version of PINT.