

# Windows System Setup

## Table of contents

1 Introduction.....	2
2 Visual C++ 2005 Express Edition.....	2
3 QT.....	2
4 CMake.....	3
5 ITK.....	3
6 Coin.....	3
7 SimVoleon.....	4
8 SoQt.....	4
9 boost.....	4
10 iMEDgine.....	4

## 1. Introduction

This document describes the installation procedure to compile everything from scratch under windows which is needed for a successful installation. We are using Visual C++ 2005 Express Edition for compiling. At the moment we are trying to get imedgine to run on windows, here is what works so far:

## 2. Visual C++ 2005 Express Edition

Download and install the Visual C++ 2005 Express Edition from the Microsoft homepage. This will install the compiler and the nmake tool as well as a light-weight but free development environment. Further, you need the Microsoft Platform SDK, in this guide we were using the MS Platform SDK for Windows Server 2003 R2.

Install both items on your system. The most important thing from now on is the "Visual Studio 2005 CommandLine Prompt", since there the environment variables are set accordingly. Note that you might have to adapt the environment variables INCLUDE and LIB to point to the MS Platform SDK before the Visual Studio 8 directories. We will refer to this again in the following section.

## 3. QT

Download QT from [Trolltech](#) and extract it to its final directory. Note that you should not move this directory later on, since the subsequent configuration processes depend on qmake, which needs to remain in the directory it was built. You can remove the demos and examples directory if you want. for this guide we were using qt-win-opensource-4.1.4.zip.

General reading on QT4 installation on Windows can be found [here](#), while the guide to compile Qt with Visual C++ 2005 can be found [here](#).

Download the Visual C++ 2005 patch for Qt, in order to be able to compile the open-source edition of Qt with this compiler. It can be found [here](#)

Extract the patch into the same directory where Qt's configure.exe program lives. Open a command prompt (cmd) and execute:

```
installpatch41.bat
```

Now we have to open a Visual Studio command prompt. Refer to the file Readme414.txt that was distributed with the patch, to set up your INCLUDE and LIB paths in vsvars.bat accordingly. On the command prompt execute:

```
qconfigure.bat msvc2005
```

Have a look at the configure output to make sure that opengl support is enabled! Finally execute:

```
nmake sub-src sub-tools
```

to build the debug and release versions of the Qt library, as well as the required tools. Note that the nmake targets above will prevent the demos and examples to be built, since this is rather time-consuming.

## 4. CMake

Get the Windows binary from [here](#) and install it. We were working with version 2.4.3.

Furthermore go to the online cvs repository of the cmake distribution and go into the directory 'Modules'. From there download the current versions of 'FindQt4.cmake' and 'UseQt4.cmake'. Replace these two files in your installation of CMake (folder: share/CMake/Modules).

## 5. ITK

Download ITK from [here](#), extract it, move it to desired directory.

Open a Visual Studio command prompt. Go into the ITK directory (e.g. c:\libraries\Insight) and create a directory called 'build'. Execute CMakeSetup.exe. Now enter the source and the build directory of the ITK in the according boxes. Use 'NMake makefiles' as build environment. Switch off the BUID\_TESTING and BUILD\_EXAMPLES to save time and hard drive space. Use Release as the build type.

Finish the cmake setup program by clicking on 'OK', this generates the makefiles. Run 'nmake' on the Visual Studio command prompt to compile the ITK library.

## 6. Coin

Download the current release version of Coin from [here](#). At the time of this writing this was version 2.4.5. Take the zip file that is marked as "Windows binaries, no installer (VC8)".

Extract the binary release into the final directory, e.g. c:\Libraries\coin. Set the environment variable COINDIR to point to the install directory. Put the bin directory of COINDIR into the system path.

## 7. SimVoleon

Download the current release version of SimVoleon from [here](#). At the time of this writing this was version 2.0.1. Take the file that is marked as Windows installer.

Install the SimVoleon package into a temporary directory.

Copy simvoleon2.dll, simvoleon2d.dll to \$COINDIR\bin.

Copy bin\simvoleon2.dll, bin\simvoleon2d.dll to \$COINDIR\bin.

Copy directory include\VolumeViz to \$COINDIR\include.

Copy lib\simvoleon2.lib, bin\simvoleon2d.lib to \$COINDIR\lib.

## 8. SoQt

Download the current cvs snapshot version of SoQt from [here](#). We cannot yet use a release version due to erroneous integration of SoQt with Qt 4! Select the latest snapshot from [here](#).

Unzip the latest cvs snapshot to a temporary directory. Go into the build/msvc8 directory. Open the Visual Studio project.

Here it is important that the COINDIR variable is set accordingly. Choose the Release DLL version of SoQt and compile it within Visual Studio.

Open a command window and go into the directory build/msvc8 of the SoQt source directory. Execute the install-dll-release.bat script to install headers, dlls and libs into the COINDIR installation of Coin.

## 9. boost

Download the boost library from [here](#). Unzip it into its final directory, e.g.# C:\Libraries\boost. This will create a sub-directory boost\_version\_number, where the library resides.

Note that it is not necessary to build the boost library.

## 10. iMEDgine

Finally we come to the heart of our efforts. We will suppose that you have a subversion client installed, if not you might have a look at [TortoiseSVN](#).

Checkout the cvs version of iMEDgine by doing a:

```
svn co https://svn.sourceforge.net/svnroot/imedgine
```

or a similar operation e.g. using the Windows explorer plugin TortoiseSVN.

Open the CMakeLists.txt file in the imedgine/src directory. if there are still remaining hard-coded include and library paths, replace these paths by your specific installation paths. We are working on getting rid of these hard-coded settings!

From a Visual Studio command prompt, execute CMakeSetup.exe and enter the source and the build directory of iMEDgine. Choose NMake makefiles for compilation. Configure and generate the makefiles. Leave CMakeSetup.exe.

Now there are still some unresolved issues which will lead to troubles during compilation. Go into the imedgine\_build/src/imedgine.dir directory and open the file build.make. There will be a line for the linker down there somewhere. You might have to manually include the directory where the Coin libraries are lying. You might have to remove two libraries called optimized.lib, they are an error from the CMake creation process. You might have to manually add QtXml4.lib, this is another error. Note that these errors might occur or they might already be resolved, we are working on it!

Now the compilation should be working, go into the directory imedgine\_build and do a nmake.