🡐 Back (/tutorials?cat=install)

Windows

Version: 6.0 ▾ Edit (https://bitbucket.org/osrf/gazebo_tutorials/src/default/install_on_windows/tutorial_6-0.md)

Install on Windows

This documentation describes how to set up a workspace for trying to compile Gazebo on Windows. It does not actually work yet.

Important requirement: 30Gb free disk space

It is recommended to have at least 30 Gigabytes of disk space to host Gazebo sources together with all the dependencies and compilation artifacts.

Supported compilers

At this moment, compilation has been tested on Windows 7 and 8.1 and is supported when using Visual Studio 2013 (https://www.visualstudio.com/downloads/). Patches for other versions are welcome.

Installation

This installation procedure uses pre-compiled binaries in a local workspace. To make things easier, use a MinGW shell for your editing work (such as the Git Bash Shell (https://msysgit.github.io/) with Mercurial (http://tortoisehg.bitbucket.org/download/index.html)), and only use the Windows `cmd` for configuring and building. You might also need to disable the Windows firewall (http://windows.microsoft.com/en-us/windows/turn-windows-firewall-on-off#turn-windows-firewall-on-off=windows-7).

1. Make a directory to work in, e.g.:

   ```
   mkdir gz-ws
   cd gz-ws
   ```

2. Download the following dependencies into that directory:

   1. freeImage 3.x, slightly modified to build on VS2013 (http://packages.osrfoundation.org/win32/deps/FreeImage-vc12-x64-release-debug.zip)
   2. boost 1.56.0 (http://packages.osrfoundation.org/win32/deps/boost_1_56_0.zip)
   3. bzip2 1.0.6 (http://packages.osrfoundation.org/win32/deps/bzip2-1.0.6-vc12-x64-release-debug.zip)
   4. dlfcn-win32 (http://packages.osrfoundation.org/win32/deps/dlfcn-win32-vc12-x64-release-debug.zip)
   5. libcurl HEAD (http://packages.osrfoundation.org/win32/deps/libcurl-vc12-x64-release-debug-static-ipv6-sspi-winssl.zip)
   6. OGRE 1.9.0 rc1 (http://packages.osrfoundation.org/win32/deps/ogre_src_v1-8-1-vc12-x64-release-debug.zip)
   7. protobuf 2.6.0 (http://packages.osrfoundation.org/win32/deps/protobuf-2.6.0-win64-vc12.zip)
   8. TBB 4.3 (http://packages.osrfoundation.org/win32/deps/tbb43_20141023oss_win.zip)
   9. zziplib 0.13.62 (http://packages.osrfoundation.org/win32/deps/zziplib-0.13.62-vc12-x64-release-debug.zip)
   10. zlib (http://packages.osrfoundation.org/win32/deps/zlib-1.2.8-vc12-x64-release-debug.zip)

3. Unzip each of them in gz-ws.

4. Also download Qt 4.8, using the link below, and unzip it into `C:\Qt\4.8.6\x64\msvc2013`:

   > Qt 4.8.6 (http://packages.osrfoundation.org/win32/deps/qt-4.8.6-x64-msvc2013-rev1.zip)

5. Install cmake, make sure to select the "Add CMake to system path for all users" option in the install dialog box

   > Cmake (http://www.cmake.org/download/)

6. Install Ruby 1.9 or greater. During the install process make sure add Ruby to your system paths.

   > Ruby (http://rubyinstaller.org/downloads/)

7. Clone Ignition Math, Sdformat, and Gazebo:

   ```
   hg clone https://bitbucket.org/ignitionrobotics/ign-math
   hg clone https://bitbucket.org/osrf/sdformat
   hg clone https://bitbucket.org/osrf/gazebo
   ```

8. Open a regular Windows shell (Start->Run->"cmd"->enter), and load your compiler setup by copying and pasting the following line:

   ```
   "C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\vcvarsall.bat" x86_amd64
   ```

9. In a Windows shell, configure and build Ignition Math

   ```
   cd ign-math
   mkdir build
   cd build
   ..\configure
   nmake
   nmake install
   ```

   You should now have an installation of Ignition Math in gz-ws/ign-math/build/install/Release.

10. In the same Windows shell, configure and build Sdformat

    ```
    cd ..\..\sdformat
    mkdir build
    cd build
    #if you want debug, run ..\configure Debug
    ..\configure
    nmake
    nmake install
    ```

    You should now have an installation of Sdformat in gz-ws/sdformat/build/install/Release or gz-ws/sdformat/build/install/Debug.

11. In the same Windows shell, configure and build Gazebo:

    ```
    cd ..\..\gazebo
    mkdir build
    cd build
    #if you want debug, run ..\configure Debug
    ..\configure
    nmake gzclient
    nmake gzserver
    ```

    Once this all works you should now have an installation of Gazebo in gz-ws/gazebo/build/install/Release or gz-ws/gazebo/build/install/Debug.

Running

gzserver

1. Adjust all paths to load dll

   1. if in Debug

      cd gz-ws\gazebo\build ..\win_addpath.bat Debug

   2. if in Release

      cd gz-ws\gazebo\build ..\win_addpath.bat Release

2. Create an ogre plugins.cfg file

   1. `cd gz-ws\gazebo\build\gazebo`
   2. If in Debug: Copy in the following into `plugins.cfg and replace MYUSERNAME with your actual username`

      ```
      # Define plugin folder
      PluginFolder=C:\Users\MYUSERNAME\gz-ws\ogre_src_v1-8-1-vc12-x64-release-debug\build\install\Debug\bin\Debug

      # Define plugins
      Plugin=RenderSystem_GL_d
      Plugin=Plugin_ParticleFX_d
      Plugin=Plugin_BSPSceneManager_d
      Plugin=Plugin_PCZSceneManager_d
      Plugin=Plugin_OctreeZone_d
      Plugin=Plugin_OctreeSceneManager_d
      ```

   3. If in Release: Copy in the following into `plugins.cfg`

      ```
      # Define plugin folder
      PluginFolder=C:\Users\MYUSERNAME\gz-ws\ogre_src_v1-8-1-vc12-x64-release-debug\build\install\Release\bin\Release

      # Define plugins
      Plugin=RenderSystem_GL
      Plugin=Plugin_ParticleFX
      Plugin=Plugin_BSPSceneManager
      Plugin=Plugin_PCZSceneManager
      Plugin=Plugin_OctreeZone
      Plugin=Plugin_OctreeSceneManager
      ```

   4. Copy this file into the `gui` directory

      ```
      cp plugins.cfg gui/
      ```

3. Run gzserver

   ```
   gzserver.exe ..\..\worlds\empty.world
   ```

Debugging

Just in case that you need to debug problems on Gazebo

Running gzserver

If you run into issues, use –verbose to get more information.

Running gzclient

If you run into issues, use –verbose to get more information. A known issue is that it does not run on VirtualBox 3.4, with Ubuntu 15.04 Host. The current theory is that it does not support off-screen frame buffering. It has been confirmed to work on VMWare Player with windws 7 guest and Ubuntu 14.04 Host. More details will be added as testing continues.

Building Ogre Examples

1. Download OIS

   http://sunet.dl.sourceforge.net/project/wgois/Source%20Release/1.3/ois-v1-3.zip (http://sunet.dl.sourceforge.net/project/wgois/Source%20Release/1.3/ois-v1-3.zip)

2. Compile OIS in Visual Studio Use the project in Win32/ folder

3. Place OIS headers and libs into

   ogre-...\Dependencies\include ogre-...\Dependencies\lib ogre-...\Dependencies\bin

4. Patch configure.bat inside ogre-1.8 to use

   `-DOGRE_BUILD_SAMPLES:BOOL=TRUE ..`

5. Compile as usual

   ```
   ..\configure.bat
   nmake
   ```

6. Run the demo browser using:

   # copy OIS_*.dll into the bin directory ogre-...\build\bin\SampleBrowser.exe

(//plus.google.com/u/0/115981436296571800301?prsrc=3)

(https://www.youtube.com/channel/UCJyqf9XJpDoM9XnpAwW6WxA)