Build Linux

Installing and Building Emergent

- Install from packages
  - Mac
  - Ubuntu
  - Windows
- Build from source (expert)
  - Mac
  - Linux
  - Windows
  - Build everything from source
- Packaging for release (dev team)
  - Mac
  - Ubuntu
  - Windows

Build from source on Ubuntu 14.04/16.04 (32 or 64 bit)

Step 1: Update system and gather dependencies

First, become root:

```html
sudo su
```

And install aptitude:

```html
apt -y install aptitude
```

And update your system:

```html
aptitude update -y
```
And install upgrades:

`aptitude upgrade -y`

Next, install libraries:

`aptitude install -y qt5-default qttools5-dev libqt5webkit5-dev qtlocation5-dev libqt5designer5 qtdesigner5-dev libqt5sensors5-dev qtmultimedia5-dev libqt5svg5-dev libcoin80-dev cmake g++ libreadline6-dev libgs10-dev zlib1g-dev libpng12-dev libjpeg-dev libncurses5-dev libsvn-dev libsndfile1-dev mercurial python-pip subversion devscripts csh pkg-config libode-dev`

Finally, exit the root user:

`exit`

**Step 2: Build the quarter library and install the package**

First, grab emergent's source code:

`svn checkout https://grey.colorado.edu/svn/emergent/emergent/trunk ~/.emergent`

Next, run a script to build Quarter and turn it into a Debian package:
Build Linux

Step 3: Build and install emergent

Start with an optimized build:

```bash
cd ~/emergent
./configure --qt5 --clean --webkit --suffix=_svn
cd build_svn
make -j4
```

And install it to `/usr/local/bin/emergent`

```bash
sudo make install
```

If you have time, also do a debug build:

```bash
cd ~/emergent
./configure --qt5 --clean --webkit --enable-debug
cd build_dbg
make -j4
```

And install it to /usr/local/bin/emergent_dbg

```html
sudo make install
```

**Step 4: Update your path**

Open up ~/.bashrc and make sure that your PATH and LD_LIBRARY_PATH environment variables look something like this:

```bash
export PATH=/bin:/usr/bin:/usr/local/bin
export LD_LIBRARY_PATH=/lib:/usr/lib:/usr/local/lib
```

And update your environment:

```bash
source ~/.bashrc
```

**Step 5 - Run emergent**

If you installed emergent using our apt repo you can run that optimized version of emergent with:

```bash
emergent
```

The optimized version of emergent we built from source here can be run via:

```bash
emergent_svn
```
And the debug version can be run via:
emergent_dbg

Or more likely:
gdb -ex=run emergent_dbg

Troubleshooting the Build Process

If you are running Ubuntu Trusty (14.04) run this:
aptitude install -y libode-sp-dev

If you encounter errors regarding libserf linking against symbols found in libssl, set this environment variable before running configure, which will tell the linker to go ahead and link in the static version of libssl:
export CMAKE_SHARED_LINKER_FLAGS=-l:libssl.a

If you have issues with configure finding libGL.so, perhaps you installed the nvidia native binary driver, in which case you need to manually make symlinks from the libGL.so.* files in /usr/lib/ to /usr/lib/x86_64-linux-gnu.
If you get error: ld terminated with signal 9 [Killed], use a system with more memory.
If you get error: CPU you selected does not support x86-64 instruction set, this can be solved by re-running configure in non-native mode:
cd ~/emergent && ./configure --not-native && cd build && make

If you get warning: inline function '<some function>' used but never defined [enabled by default] or Wno-undefined-inline you can ignore this error.

If a file that was present during the last time cmake configured itself is subsequently removed, you'll get this kind of install error:

CMake Error at prog_lib/cmake_install.cmake:31 (file):
  file INSTALL cannot find
   "~/Users/oreilly/emergent/prog_lib/ObjectsEnv.prog".
Call Stack (most recent call first):
  cmake_install.cmake:105 (include)

To fix this, in your build directory, do:

ccmake ./
then hit c (configure) and then g (generate) then e (exit)
sudo make install

Reference Information on Prerequisite Libraries

These are all the libraries that emergent depends upon -- on most Build procedures, these libraries are available as pre-compiled, easy-to-use packages of one form or another. This information is just for general reference and in case you want to update or otherwise do something special with a particular library.

Current versions of each relevant package are always available on our ftp site:
ftp://grey.colorado.edu/pub/emergent or https://grey.colorado.edu/emergent_ftp/

The prerequisites are generally the same across operating systems but always check the operating system specific build instructions for version info.

<table>
<thead>
<tr>
<th>Package</th>
<th>Cur Version</th>
<th>Main URL</th>
<th>Download URL</th>
<th>Docs URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qt (gui and OS api)</td>
<td>5.6.1 (LTS)</td>
<td><a href="https://www.qt.io/">https://www.qt.io/</a></td>
<td>[1]</td>
<td>[2]</td>
</tr>
<tr>
<td></td>
<td>(also works with old versions, back to 4.8 or so)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coin &amp; Quarter (3D scene graph &amp; Qt api)</td>
<td>3.1.3</td>
<td>We have our own patched version that is required: Coin3d</td>
<td>Coin3d</td>
<td></td>
</tr>
<tr>
<td>ODE (Physics sim)</td>
<td>0.14</td>
<td><a href="http://ode.org/">http://ode.org/</a></td>
<td>[3]</td>
<td>[4]</td>
</tr>
<tr>
<td>subversion (required from v6.2.0)</td>
<td>1.9.4</td>
<td><a href="http://subversion.apache.org">http://subversion.apache.org</a></td>
<td>[8] (use WANdisco, full package for mac, installs in /opt/subversion)</td>
<td></td>
</tr>
<tr>
<td>libsndfile (sound file load / save -- optional, needed for auditory)</td>
<td>1.0.26</td>
<td>[9]</td>
<td>[10]</td>
<td>n/a</td>
</tr>
</tbody>
</table>
References

Article Sources and Contributors


Image Sources, Licenses and Contributors