ParaView Plugin Development

Table of Contents:

ParaView "SuperBuild"

When you wish to deploy a plugin that works with the KitWare release of ParaView, you must build a special version of ParaView: the Super Build.

SuperBuild Prerequisites

I typically use the GNU compilers (gcc, g++, gfortran or g77) for everything. You also need a recent copy of cmake. On Mac OS-X systems, you need to install a recent version of XCode.

Building the SuperBuild

Here's a cheat sheet on how to build the SuperBuild.

1. Download SuperBuild via git:

   ```
   git clone git://paraview.org/ParaViewSuperbuild.git
   ```

   If you'd like to get a specific version (eg. 4.0.1), find the tag identifier via GitWeb. Edit versions.cmake and set GIT_TAG "v4.0.1"
2. Create a build directory:
   a. Mac OS-X

      ```
      mkdir ~/ParaView-4.0.1_OSX-10.7
      cd ~/ParaView-4.0.1_OSX-10.7
      ```

   b. Linux

      ```
      mkdir ~/ParaView-4.0.1_Linux-x86_64
      cd ~/ParaView-4.0.1_Linux-x86_64
      ```

3. Configure

   ```
   ccmake ../src/ParaViewSuperbuild/ -DENABLE_boost=on -DENABLE_diy=ON -DENABLE_freetype=ON -
   DENABLE_hdf5=ON -DENABLE_libxml2=ON -DENABLE_matplotlib=ON -DENABLE_numpy=ON -DENABLE_paraview=ON -
   DENABLE_png=ON -DENABLE_python=ON -DENABLE_qhull=ON -DENABLE_qt=ON -DENABLE_szip=ON -DENABLE_zlib=ON
   ```

4. Once the CMake Curses GUI comes up, set a few common settings:
   a. BUILD_TESTING: ON
   b. ENABLE[boost, diy, freetype, hdf5, libxml2, matplotlib, numpy, paraview, png, python, qhull, qt, szip, zlib] = ON
   c. ENABLE[accusolve, cgns, cosmologytools, ffmpeg, manta, silo, visitbridge, vistrails] = OFF
   d. We recommend USE_SYSTEM_qt and USE_SYSTEM_mpi = OFF
   e. For Mac OS-X, CMAKE_OSX_DEPLOYMENT_TARGET and CMAKE_OSX_SYSROOT should match version of OS-X (Click Apple menu -> About This Mac... Should be 10.7, 10.8, etc.)
5. Press "g" to Generate build scripts.
6. Compile with gmake (do not use parallel "gmake -j2"!) . . . and grab a cup of coffee. . . Took 4 hours on 2012 MacBook Pro
7. For Mac OS-X
   a. ctest to link binary & libraries
   b. cpack -G"DragNDrop" to build distributable binary
   c. Use fixup_plugin.py to fix library paths in plugins.

Debugging problems with the ParaView SuperBuild

- You might get errors about Python library incompatibility.
    - Solution: Make sure your `$PYTHONPATH` environment variable is unset prior to running `cmake` & `gmake`. You may need to wipe your build directory prior to re-configuring (with cmake) and compiling (with gmake) to get around the error(s).

- If you get an error at the build stage with

```
[ 20%] Performing install step for 'szip'
Making install in src
 /bin/sh ../libtool --mode=install /opt/share/idl/idl/bin/install -c  'libsz.la' '/export/data1/schmitt
/paraview/opt/ParaView-4.0.1-Linux_x86_64/build/install/lib/libsz.la'
/opt/share/idl/idl/bin/install -c .libs/libsz.so.2.0.0 /export/data1/schmitt/paraview/opt/ParaView-4.0.1-
Linux_x86_64/build/install/lib/libsz.so.2.0.0

    The current directory must be set to the ITT directory.
    Change the default to the ITT directory and re-run
    this script.

gmake[5]: *** [install-libLTLIBRARIES] Error 1
gmake[4]: *** [install-am] Error 2
gmake[3]: *** [install-recursive] Error 1
gmake[2]: *** [szip/src/szip-stamp/szip-install] Error 2
gmake[1]: *** [CMakeFiles/szip.dir/all] Error 2
```

    - Solution: Whenever you ran CMake to configure the ParaView build, your $PATH contained the directory where IDL is installed. The problem is that IDL creates its own install tool which is incompatible. Remove IDL from your search $PATH and re-configure & recompile the code. (Note: you may have to wipe your build directory at start from scratch).

- If you get an error building Numpy in Linux:

```
/usr/bin/g77 -g -Wall -L/export/data1/schmitt/paraview/opt/ParaView-4.0.1-Linux_x86_64/build/install/lib
build/temp.linux-x86_64-2.7/numpy/linalg/lapack_litemodule.o build/temp.linux-x86_64-2.7/numpy/linalg
/python_xerbla.o -L/usr/lib64 -L/export/data1/schmitt/paraview/opt/ParaView-4.0.1-Linux_x86_64/build
/install/lib -Lbuild/temp.linux-x86_64-2.7 -llapack -lblas -lpython2.7 -lg2c -o build/lib.linux-x86_64-
2.7/numpy/linalg/lapack_lite.so
/usr/lib/gcc/x86_64-redhat-linux/3.4.6/libfrtbegin.a(frtbegin.o): In function `main':
(.text+0x1e): undefined reference to `MAIN__'
collect2: ld returned 1 exit status
error: Command "/usr/bin/g77 -g -Wall -L/export/data1/schmitt/paraview/opt/ParaView-4.0.1-Linux_x86_64
/build/install/lib build/temp.linux-x86_64-2.7/numpy/linalg/lapack_litemodule.o build/temp.linux-x86_64-
2.7/numpy/linalg/python_xerbla.o -L/usr/lib64 -L/export/data1/schmitt/paraview/opt/ParaView-4.0.1-
Linux_x86_64/build/install/lib -Lbuild/temp.linux-x86_64-2.7 -llapack -lblas -lpython2.7 -lg2c -o build
/lib.linux-x86_64-2.7/numpy/linalg/lapack_lite.so" failed with exit status 1
/usr/lib/gcc/x86_64-redhat-linux/3.4.6/libfrtbegin.a(frtbegin.o): In function `main':
(.text+0x1e): undefined reference to `MAIN__'
collect2: ld returned 1 exit status
CMake Error at /export/data1/schmitt/paraview/opt/ParaView-4.0.1-Linux_x86_64/build/pv-numpy-build.cmake:
26 (message):
  Failed!!!


gmake[2]: *** [numpy/src/numpy-stamp/numpy-build] Error 1
gmake[1]: *** [CMakeFiles/numpy.dir/all] Error 2
gmake: *** [all] Error 2
```

    - Solution: According to the Numpy bug tracker, you can fix this by leaving LDFLAGS unset prior to building Numpy. However, the SuperBuild passes in LDFLAGS to configure Numpy. The fix is easy: edit pv-numpy-build.cmake and remove the part that sets LDFLAGS.
- If you get an error building numpy on MacOSX:

```
File "numpy/core/setup.py", line 41, in check_types
    out = check_types(*a, **kw)
  File "numpy/core/setup.py", line 271, in check_types
    "Cannot compile 'Python.h'. Perhaps you need to "\
SystemError: Cannot compile 'Python.h'. Perhaps you need to install python-dev|python-devel.
CMake Error at /Users/wiltbemj/src/ParaView-4.0.1_OSX-10.7/pv-numpy-build.cmake:26 (message):
  Failed!!!


gmake[2]: *** [numpy/src/numpy-stamp/numpy-build] Error 1
gmake[1]: *** [CMakeFiles/numpy.dir/all] Error 2
gmake: *** [all] Error 2
```

- Solution: toggle advanced option on `ccmake` and update `CMAKE_C_FLAGS` to be `-I/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.7.sdk/usr/include`
  - Note: version of SDK must match choice for CMAKE_OSX variables

- When building Qt, you get an error that looks like

```
81%] Performing build step for 'qt'
LD_LIBRARY_PATH = /export/data1/schmitt/paraview/opt/ParaView-4.0.1-Linux_x86_64/build/install/lib:/opt
/local/pgi-9.04/linux86-64/9.0/lib:/opt/local/pgi-9.04/linux86-64/9.0/mpi/mpich/lib:/home/schmitt/opt/64
/mplayer/lib::/home/schmitt/opt/64/lib:/home/schmitt/opt/64/lib64:/opt/local/pgi-9.04/linux86-64/9.0/lib:
/opt/local/pgi-9.04/linux86-64/9.0/mpi/mpich/lib:/home/schmitt/opt/64/mplayer/lib::/home/schmitt/opt/64
/lib:/home/schmitt/opt/64/lib64
/usr/bin/ld: warning: libQtCLucene.so.4, needed by /export/data1/schmitt/paraview/opt/ParaView-4.0.1-
Linux_x86_64/build/qt/src/qt-build/lib/libQtHelp.so, not found (try using -rpath or -rpath-link)
/export/data1/schmitt/paraview/opt/ParaView-4.0.1-Linux_x86_64/build/qt/src/qt-build/lib/libQtHelp.so:
undefined reference to `QCLuceneIndexWriter::setMergeFactor(int)'
...
```

- Solution: `find . libQtCLucene.so.4` and then edit pv-qt-build.cmake and append to LD_LIBRARY_PATH and LDFLAGS.

LFM Plugin (aka GHOST-kit)

The LFM plugin is distributed with Geospace/Heliosphere Observation & Simulation Tool-kit (GHOST-kit) developed by Josh Murphy & Peter Schmitt.

Prerequisites

The "newCMAKE" branch will automagically download & build all the prerequisites required by GHOST. However, this hasn't been merged to master (similar to trunk in SVN parlance). Until Josh has a stable version, you will need to manually build the following prerequisites for LFM and Enlil plugins:

- HDF4 and its prerequisites (jpeg, sz, z)
- NetCDF for Enlil plugin

You only need the static (**.a) not dynamic (**.so or *.dylib) versions of these libraries! If you only build the dynamic version, then other users might get run-time errors about missing libraries. It's easiest to package everything together with the static libraries.

Download & build the plugin

Here's how you build the plugin:

1. Obtain source code

```
git clone https://github.com/ghost-kit/GHOST.git
```

2. Setup build directory

```
mkdir GHOST/build
cd GHOST/build
```

3. Use CMake to Configure. Note you need to pass a variable ParaView_DIR which points to the direcotry containing the file ParaViewConfig.cmake:

```
cmake .. -DParaView_DIR=/Users/schmitt/paraview/opt/ParaView-3.98.1_OSX-10.7/paraview/src/paraview-build
-DGHOST_BUILD_CDAWEB=OFF
```

4. If there were no errors, compile the plugin:

```
gmake
```

   a. On Mac, build the dist target to fix library locations

```
gmake dist
```

   Try running `otool -L lib/libGHOST.dylib` before & after the dist target to see how hard-coded library paths change in the dylib.

Now you should be ready to load the plugin into ParaView.

1. Start ParaView
2. Go to Tools -> Manage Plugins
3. Click "Load..."
4. Select GHOST/build/libGHOST.dylib
   a. Check the "Always Load" button to use the plugin at startup.

Useful GHOST-kit links

- GHOST-kit repository on GitHub
- GHOST-kit website
- Old bug tracker for LFM & Enlil ParView development
  - Note: there's limited documentation on the old bug tracker... check out poorly organized the wiki pages.
- The ParaView Mailing Lists are a great resource, especially paraview-users!