Brief Rosetta Installation Guide

Rosetta is a library based object-oriented software suite which provides a robust system for predicting and designing protein structures, protein folding mechanisms, and protein-protein interactions.

The latest release of Rosetta, version 3.2, may be downloaded at http://www.rosettacommons.org/software. The software is free to academic users, who may get a license (and download link) by clicking on the appropriate link on the left side of the page. The user manual, containing more detailed installation and usage information, is available online at http://www.rosettacommons.org/manuals/archive/rosetta3.2_user_guide.

Rosetta is distributed as source code; individual applications must be compiled. Rosetta has been successfully built on the following platforms:

Fully Support Platforms (extensively tested):

- Darwin Kernel Version 9.6.0: root:xnu-1228.9.59~1/RELEASE_I386 i386
- Darwin Kernel Version 9.5.2: root:xnu-1228.8.59~2/RELEASE_I386 i386
- Linux 2.6.27-9-generic #1 SMP x86_64 GNU/Linux
- Linux 2.6.10-1.741_FC3smp #1 SMP i686 athlon i386 GNU/Linux
- Linux 2.6.9-78.0.8.ELsmp #1 SMP Wed i686 i686 i386 GNU/Linux

Partial Support (built but limited testing was performed)

- Fedora Core Linux v3 or greater, x86, 32-bit, GCC 4.0 (*)(**)
- RedHat Enterprise Linux v3 or greater, x86, 32-bit, Intel C++ 9.1 (*)
- RedHat Enterprise Linux v3 or greater, x86, 32-bit, GCC 4.0(**)
- MinGW on Microsoft Windows XP, 32-bit, GCC 3.4

After downloading Rosetta, the next step is building the Rosetta executables. Rosetta uses 'scons' as its build system; a local version of scons is included in the Rosetta distribution. This does require Python version 2.2 or greater is installed. To use this version of Scons, call the file external/scons-local/scons.py within your Rosetta installation as your SCons executable.

For the following build commands you need to be in the home directory of Rosetta source. In order to build a release version of Rosetta executables, simply run scons:

Release Mode --This command will build the defaults: libraries devel, protocols, core, numeric, utility and all applications listed in src/apps.src.settings or src/pilot_apps.src.settings.all.

```
scons bin mode=release
```

Debug Mode -- In order to build a debug version of executables, remove the flag 'mode=release'.

```
scons bin
```

To display more rosetta-specific build options call

```
scons -h
```

Automatic location of other compilers (assuming they are already in your path) such as Intel C/C++ may be enabled by uncommenting the "program_path" line in 'tools/build/user.settings'.

After running 'scons', links to executables are placed in the folder 'bin/'. Rosetta is run through the command line, with simulation parameters controlled through command line parameters, or (optionally) a flags file. Options can be listed with the command. Options, and arguments to the options, are separated by whitespace. A colon is used to clarify options when there are multiple separate options with the same name. Multiple layers of colons may be needed.

```
        fixbb.macgccrelease -in:file:s myinput.pdb -database mypath
```

Options can also be written in a flag file. In this file, put one option on each line, still using the colon to specify the layers. An example options file appears below.

```
        -database /home/yiliu/Programing/branches/rosetta_database
        -in:file:s 1l2y_centroid.pdb
        -in:centroid_input
        -score:weights centroid_des.wts
```

If this file were called “option”, then it would be used like this (notice the @ symbol):

```
        fixbb.macgccrelease @option
```

Options are program-specific. For the options, one may find more information in two places: first the application documentation (http://www.rosettacommons.org/manuals/archive/rosetta3.2_user_guide/); second, if you pass -help as a flag on the command line, Rosetta will spit out all existing options and then quit (ignoring other flags).