bgclang (LLVM/clang on the BG/Q)

For usage information, and information specific to using bgclang on ALCF’s BG/Q machines (Vesta, Mira and Cetus), please visit: https://www.alcf.anl.gov/user-guides/bgclang-compiler

Other BG/Q systems

If your system administrators have not been kind enough to install bgclang on your system, you can either direct them to this page, or install the distribution yourself. RPMs are provided (see below), and these are *relocatable* RPMs, meaning that they can be installed by a non-root user in any directory.

Please note that, if you wish to use dynamic linking (which you must do when certain features, like address sanitizer, are enabled), you must install bgclang in a directory that is mounted from the compute nodes (read-only is sufficient).

Mailing List

If you’re using bgclang, please subscribe to the mailing list: http://lists.alcf.anl.gov/mailman/listinfo/llvm-bgq-discuss.

bgclang downloads (for installing on your own)

For those managing their own installs, note that the MPI wrappers are installed in the PREFIX/mpi/bgclang.bgclang.legacy/bin directories. The non-MPI compiler wrappers are located in the PREFIX/wbin directory.

RPMs, etc.

Note: To take full advantage of the C++11 support, you’ll likely need files from the GCC 4.7.2 toolchain. This is provided by IBM, but unfortunately, not all BG/Q systems have this installed. If your system does not have a /bgsys/drivers/ppcfloor/gnu-linux-4.7.2 or /bgsys/drivers/toolchain/V1R2M1_base_4.7.2 or /bgsys/drivers/toolchain/V1R2M2_base_4.7.2 directory, then you can download the necessary files from this archive, extract them, and then change the path in the bgclang wrapper script:

http://www.mcs.anl.gov/~hfinkel/bgclang/V1R2M2_base_4.7.2-files.tar.gz (for bgclang builds r217688-20140912 and later, you should use this archive)

http://www.mcs.anl.gov/~hfinkel/bgclang/V1R2M1_base_4.7.2-files.tar.gz (for older bgclang builds you may use this archive)

There is currently a line in the wrapper script:

```
new_floor=/bgsys/drivers/toolchain/${dvr}_base_4.7.2
```

that you’ll need to update to reflect to where you’ve extracted the above archive.

Also, in order for dynamic linking to function correctly, you’ll need to edit the files:

```
gnu-linux-4.7.2/powerpc64-bq-linux/lib/libc.so
gnu-linux-4.7.2/powerpc64-bq-linux/lib/libpthread.so
```

despite the appearance of being binary files, these are actually ASCII text linker scripts, and within them, you should replace the /bgsys/drivers/toolchain prefix to reflect the actual location of the V1R2M2_base_4.7.2 or V1R2M1_base_4.7.2 directory.
If you're compiling with bgclang++11, and receiving linking errors like, "undefined reference to `vtable for std::nested_exception'", your problem is likely the lack of these files.

Using ESSL and other libraries compiled using IBM's OpenMP

If you'd like to compile your application using bgclang, but use IBM's multi-threaded ESSL (libesslsmpbg.a), you need to work-around the following problem: To link with libesslsmpbg.a you need to link against IBM's OpenMP runtime library (libxlsmp.a). But both IBM's OpenMP runtime library and LLVM's OpenMP runtime library define the C-level OpenMP API functions (omp_get_thread_num, etc.) and linking both into your application directly will likely result in problems. Because ESSL does not use these functions directly (using the Fortran interface functions exported by libxlsmp.a instead), we can avoid incompatibilities by generating a version of libxlsmp.a with the C-level API functions renamed. This can be done using the objcopy utility, and here's a script which will do this for you: make-xlsm
nonconflicting.sh.

Nightly builds

Please visit the automatically-generated listing http://www.mcs.anl.gov/~hfinkel/bgclang/ (you'll need both the 'stage1' and the 'stage2' RPMs to install recent nightly builds).

Also, you might find Michael Schlottke's bgclang installation scripts useful, see https://github.com/sloede/install-bgq.

r284961-stable

http://www.mcs.anl.gov/~hfinkel/bgclang/#r284961-stable

r266865-stable

http://www.mcs.anl.gov/~hfinkel/bgclang/#r266865-stable

r220548-20141024

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-binutils-r220548-20141024-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-gdb-r220548-20141024-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-r220548-20141024-1-2.ppc64.rpm (updated)
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-sleef-r220548-20141024-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libcxx-r220548-20141024-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r220548-20141024-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r220548-20141024-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-3.4-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-libcxx-3.4-2.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/vpkg-bin-sh-1-1.ppc64.rpm

A non-root (regular) user can install these RPMs (because they are relocatable), but in addition to specifying the installation prefix (with the --prefix argument), an alternate RPM database directory needs to be specified (in a directory to which you actually have write permission). For example, to install bgclang into the
/tmp/bgclang directory using /tmp/bgclang/rpm as the RPM database directory, run:

```
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang "
   bgclang-binutils-r220548-20141024-1-1.ppc64.rpm \
   bgclang-gdb-r220548-20141024-1-1.ppc64.rpm \
   bgclang-r220548-20141024-1-2.ppc64.rpm \
   bgclang-compiler-rt-r220548-20141024-1-1.ppc64.rpm \
   bgclang-libcxx-r220548-20141024-1-1.ppc64.rpm \
   bgclang-libomp-r220548-20141024-1-1.ppc64.rpm \
   bgclang-sleef-r220548-20141024-1-1.ppc64.rpm"
```

If you've not installed the stage1 packages (which is a clang configured to build for the login nodes), you'll need to do that first:

```
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang bgclang-stage1-3.4-1.ppc64.rpm bgclang-stage1-libcxx-
```

If the installation fails with an error like:

```
error: Failed dependencies:
   /bin/sh is needed by bgclang-r220548-20141024-1-1.ppc64
```

Then first install the vpkg-bin-sh-1-1.ppc64.rpm package (which is a virtual package which exists only to satisfy this /bin/sh dependency):

```
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang vpkg-bin-sh-1-1.ppc64.rpm
```

After the install is complete, the prefix directory should look like this:

```
$ ls -l /tmp/bgclang | awk '{print $1, $9, $10, $11}'
   lrwxrwxrwx  bin -> current/bin
   lrwxrwxrwx  compiler-rt -> current/compiler-rt
   lrwxrwxrwx  current -> r189357-20130827
   lrwxrwxrwx  docs -> current/docs
   lrwxrwxrwx  include -> current/include
   lrwxrwxrwx  lib -> current/lib
   lrwxrwxrwx  libcpp -> current/libcpp
   lrwxrwxrwx  libstdc++fixup -> current/libstdc++fixup
   lrwxrwxrwx  mpi -> current/mpi
   lrwxrwxrwx  omp -> current/omp
drwxr-xr-x  r220548-20141024
   drwxr-xr-x  rpm
   lrwxrwxrwx  scan-build -> current/scan-build
   lrwxrwxrwx  scan-view -> current/scan-view
   lrwxrwxrwx  share -> current/share
   lrwxrwxrwx  sleef -> current/sleef
   drwxr-xr-x  stage1
   lrwxrwxrwx  wbin -> current/wbin
```
These symlinks are only created by the post-install scripts in the RPMs if they don't already exist. As a result, upon subsequent upgrades, you'll need to manually update the 'current' symlink as desired.

The corresponding source RPMs are here:

http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-stage1-3.4-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/vpkg-bin-sh-1-1.src.rpm

To rebuild these, first build and install (or just install) the stage1 packages, then build the bgclang-binutils RPM and install it, then build the bgclang RPM and install it. Then the others can be built. When building the bgclang RPM (and the other packages that are built with bgclang) specify the same --dbpath command-line argument provided to rpm when installing the built stage1 and bgclang RPMs (if any).

r217688-20140912

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-binutils-r217688-20140912-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-gdb-r217688-20140912-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-r217688-20140912-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-sleef-r217688-20140912-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libcxxx-r217688-20140912-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r217688-20140912-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r217688-20140912-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-3.4-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-libcxxx-3.4-2.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/vpkg-bin-sh-1-1.ppc64.rpm
A non-root (regular) user can install these RPMs (because they are relocatable), but in addition to specifying the installation prefix (with the --prefix argument), an alternate RPM database directory needs to be specified (in a directory to which you actually have write permission). For example, to install bgclang into the /tmp/bgclang directory using /tmp/bgclang/rpm as the RPM database directory, run:

```
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang \
   bgclang-binutils-r217688-20140912-1-1.ppc64.rpm \
   bgclang-gdb-r217688-20140912-1-1.ppc64.rpm \
   bgclang-r217688-20140912-1-1.ppc64.rpm \
   bgclang-compiler-rt-r217688-20140912-1-1.ppc64.rpm \
   bgclang-libcxx-r217688-20140912-1-1.ppc64.rpm \
   bgclang-libomp-r217688-20140912-1-1.ppc64.rpm \
   bgclang-sleef-r217688-20140912-1-1.ppc64.rpm
```

If you've not installed the stage1 packages (which is a clang configured to build for the login nodes), you'll need to do that first:

```
rpm -Uhv --dbpath /tmp/bgclang.rpm --prefix /tmp/bgclang \
   bgclang-stage1-3.4-1.ppc64.rpm bgclang-stage1-libcxx-
```

If the installation fails with an error like:

```
error: Failed dependencies:
   /bin/sh is needed by bgclang-r217688-20140912-1-1.ppc64
```

Then first install the vpkg-bin-sh-1-1.ppc64.rpm package (which is a virtual package which exists only to satisfy this /bin/sh dependency):

```
rpm -Uhv --dbpath /tmp/bgclang.rpm --prefix /tmp/bgclang vpkg-bin-sh-1-1.ppc64.rpm
```

After the install is complete, the prefix directory should look like this:

```
$ ls -l /tmp/bgclang | awk '{print $1, $9, $10, $11}'
lrwxrwxrwx bin -> current/bin
lrwxrwxrwx compiler-rt -> current/compiler-rt
lrwxrwxrwx current -> r189357-20130827
lrwxrwxrwx docs -> current/docs
lrwxrwxrwx include -> current/include
lrwxrwxrwx lib -> current/lib
lrwxrwxrwx libc++ -> current/libc++
lrwxrwxrwx libstdc++fixup -> current/libstdc++fixup
lrwxrwxrwx mpi -> current/mpi
lrwxrwxrwx omp -> current/omp
drwxr-xr-x r217688-20140912
drwxr-xr-x rpm
lrwxrwxrwx scan-build -> current/scan-build
lrwxrwxrwx scan-view -> current/scan-view
lrwxrwxrwx share -> current/share
lrwxrwxrwx sleef -> current/sleef
```
These symlinks are only created by the post-install scripts in the RPMs if they don't already exist. As a result, upon subsequent upgrades, you'll need to manually update the 'current' symlink as desired.

The corresponding source RPMs are here:

http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-r217688-20140912-1-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-sleef-r217688-20140912-1-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-stage1-3.4-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/vpkg-bin-sh-1-1.src.rpm

To rebuild these, first build and install (or just install) the stage1 packages, then build the bgclang-binutils RPM and install it, then build the bgclang RPM and install it. Then the others can be built. When building the bgclang RPM (and the other packages that are built with bgclang) specify the same --dbpath command-line argument provided to rpm when installing the built stage1 and bgclang RPMs (if any).

r209570-20140527

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-binutils-r209570-20140527-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-r209570-20140527-1-2.ppc64.rpm (updated)
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-sleef-r209570-20140527-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libcxx-r209570-20140527-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r209570-20140527-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r209570-20140527-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-3.4-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-libcxx-3.4-2.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/vpkg-bin-sh-1-1.ppc64.rpm
A non-root (regular) user can install these RPMs (because they are relocatable), but in addition to specifying the installation prefix (with the --prefix argument), an alternate RPM database directory needs to be specified (in a directory to which you actually have write permission). For example, to install bgclang into the /tmp/bgclang directory using /tmp/bgclang/rpm as the RPM database directory, run:

```
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang 
  bgclang-binutils-r209570-20140527-1-1.ppc64.rpm 
  bgclang-r209570-20140527-1-1.ppc64.rpm 
  bgclang-compiler-rt-r209570-20140527-1-1.ppc64.rpm 
  bgclang-libcxx-r209570-20140527-1-1.ppc64.rpm 
  bgclang-libomp-r209570-20140527-1-1.ppc64.rpm 
  bgclang-sleef-r209570-20140527-1-1.ppc64.rpm
```

If you've not installed the stage1 packages (which is a clang configured to build for the login nodes), you'll need to do that first:

```
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang 
  bgclang-stage1-3.4-1.ppc64.rpm bgclang-stage1-libcxx-
```

If the installation fails with an error like:

```
error: Failed dependencies: 
/bin/sh is needed by bgclang-r209570-20140527-1-1.ppc64
```

Then first install the vpkg-bin-sh-1-1.ppc64.rpm package (which is a virtual package which exists only to satisfy this /bin/sh dependency):

```
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang vpkg-bin-sh-1-1.ppc64.rpm
```

After the install is complete, the prefix directory should look like this:

```
$ ls -l /tmp/bgclang | awk '{print $1, $9, $10, $11}'
  lrwxrwxrwx bin -> current/bin
  lrwxrwxrwx compiler-rt -> current/compiler-rt
  lrwxrwxrwx current -> r189357-20130827
  lrwxrwxrwx docs -> current/docs
  lrwxrwxrwx include -> current/include
  lrwxrwxrwx lib -> current/lib
  lrwxrwxrwx libcxx -> current/libcxx
  lrwxrwxrwx libstdc++fixup -> current/libstdc++fixup
  lrwxrwxrwx mpi -> current/mpi
  lrwxrwxrwx omp -> current/omp
  drwxr-xr-x r209570-20140527
  drwxr-xr-x rpm
  lrwxrwxrwx scan-build -> current/scan-build
  lrwxrwxrwx scan-view -> current/scan-view
  lrwxrwxrwx share -> current/share
  lrwxrwxrwx sleef -> current/sleef
  drwxr-xr-x stage1
  lrwxrwxrwx wbin -> current/wbin
```
These symlinks are only created by the post-install scripts in the RPMs if they don't already exist. As a result, upon subsequent upgrades, you'll need to manually update the 'current' symlink as desired.

The corresponding source RPMs are here:


http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-stage1-3.4-1.src.rpm


http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/vpkg-bin-sh-1-1.src.rpm

To rebuild these, first build and install (or just install) the stage1 packages, then build the bgclang-binutils RPM and install it, then build the bgclang RPM and install it. Then the others can be built. When building the bgclang RPM (and the other packages that are built with bgclang) specify the same --dbpath command-line argument provided to rpm when installing the built stage1 and bgclang RPMs (if any).

r209570-20140524

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-binutils-r209570-20140524-1-1.ppc64.rpm

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-r209570-20140524-1-1.ppc64.rpm

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-sleef-r209570-20140524-1-1.ppc64.rpm

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libcxx-r209570-20140524-1-1.ppc64.rpm

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r209570-20140524-1-1.ppc64.rpm

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r209570-20140524-1-1.ppc64.rpm

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-3.4-1.ppc64.rpm

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-libcxx-3.4-2.ppc64.rpm

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/vpkg-bin-sh-1-1.ppc64.rpm

A non-root (regular) user can install these RPMs (because they are relocatable), but in addition to specifying the installation prefix (with the --prefix argument), an alternate RPM database directory needs to be specified (in a directory to which you actually have write permission). For example, to install bgclang into the /tmp/bgclang directory using /tmp/bgclang/rpm as the RPM database directory, run:
If you've not installed the stage1 packages (which is a clang configured to build for the login nodes), you'll need to do that first:

```bash
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang bgclang-binutils-r209570-20140524-1-1.ppc64.rpm
bgclang-r209570-20140524-1-1.ppc64.rpm
bgclang-compiler-rt-r209570-20140524-1-1.ppc64.rpm
bgclang-libcxx-r209570-20140524-1-1.ppc64.rpm
bgclang-libomp-r209570-20140524-1-1.ppc64.rpm
bgclang-sleef-r209570-20140524-1-1.ppc64.rpm
```

If the installation fails with an error like:

```
error: Failed dependencies:
/bin/sh is needed by bgclang-r209570-20140524-1-1.ppc64
```

Then first install the vpkg-bin-sh-1-1.ppc64.rpm package (which is a virtual package which exists only to satisfy this /bin/sh dependency):

```bash
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang vpkg-bin-sh-1-1.ppc64.rpm
```

After the install is complete, the prefix directory should look like this:

```bash
$ ls -l /tmp/bgclang | awk '{print $1, $9, $10, $11}'
lrwxrwxrwx bin -> current/bin
lrwxrwxrwx compiler-rt -> current/compiler-rt
lrwxrwxrwx current -> r189357-20130827
lrwxrwxrwx docs -> current/docs
lrwxrwxrwx include -> current/include
lrwxrwxrwx lib -> current/lib
lrwxrwxrwx libc++ -> current/libc++
lrwxrwxrwx libstdc++fixup -> current/libstdc++fixup
lrwxrwxrwx mpi -> current/mpi
lrwxrwxrwx omp -> current/omp
drwxr-xr-x r209570-20140524
drwxr-xr-x rpm
lrwxrwxrwx scan-build -> current/scan-build
lrwxrwxrwx scan-view -> current/scan-view
lrwxrwxrwx share -> current/share
lrwxrwxrwx sleef -> current/sleef
drwxr-xr-x stage1
lrwxrwxrwx wbin -> current/wbin
```

These symlinks are only created by the post-install scripts in the RPMs if they don't already exist. As a result, upon subsequent upgrades, you'll need to manually update the 'current' symlink as desired.
The corresponding source RPMs are here:


To rebuild these, first build and install (or just install) the stage1 packages, then build the bgclang-binutils RPM and install it, then build the bgclang RPM and install it. Then the others can be built. When building the bgclang RPM (and the other packages that are built with bgclang) specify the same --dbpath command-line argument provided to rpm when installing the built stage1 and bgclang RPMs (if any).

### r206161-20140414

- [http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-binutils-r206161-20140414-1-1.ppc64.rpm](http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-binutils-r206161-20140414-1-1.ppc64.rpm)
- [http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-r206161-20140414-1-1.ppc64.rpm](http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-r206161-20140414-1-1.ppc64.rpm)
- [http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-sleef-r206161-20140414-1-1.ppc64.rpm](http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-sleef-r206161-20140414-1-1.ppc64.rpm)
- [http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libcxx-r206161-20140414-1-1.ppc64.rpm](http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libcxx-r206161-20140414-1-1.ppc64.rpm)
- [http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r206161-20140414-1-1.ppc64.rpm](http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r206161-20140414-1-1.ppc64.rpm)
- [http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r206161-20140414-1-1.ppc64.rpm](http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r206161-20140414-1-1.ppc64.rpm)
- [http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-3.4-1.ppc64.rpm](http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-3.4-1.ppc64.rpm)
- [http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-libcxx-3.4-2.ppc64.rpm](http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-stage1-libcxx-3.4-2.ppc64.rpm)

A non-root (regular) user can install these RPMs (because they are relocatable), but in addition to specifying the installation prefix (with the --prefix argument), an alternate RPM database directory needs to be specified (in a directory to which you actually have write permission). For example, to install bgclang into the /tmp/bgclang directory using /tmp/bgclang/rpm as the RPM database directory, run:

```bash
rpm -Uvh --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang 
bgclang-binutils-r206161-20140414-1-1.ppc64.rpm 
bgclang-r206161-20140414-1-1.ppc64.rpm 
```

https://trac.alcf.anl.gov/projects/llvm-bgq
If you've not installed the stage1 packages (which is a clang configured to build for the login nodes), you'll need to do that first:

```
rm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang bgclang-stage1-3.4-1.ppc64.rpm bgclang-stage1-libcxx-
```

If the installation fails with an error like:

```
error: Failed dependencies:
/bin/sh is needed by bgclang-r206161-20140414-1-1.ppc64
```

Then first install the vpkg-bin-sh-1-1.ppc64.rpm package (which is a virtual package which exists only to satisfy this /bin/sh dependency):

```
rm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang vpkg-bin-sh-1-1.ppc64.rpm
```

After the install is complete, the prefix directory should look like this:

```
$ ls -l /tmp/bgclang | awk '{print $1, $9, $10, $11}'
lrwxrwxrwx  bin -> current/bin
lrwxrwxrwx  compiler-rt -> current/compiler-rt
lrwxrwxrwx  current -> r189357-20130827
lrwxrwxrwx  docs -> current/docs
lrwxrwxrwx  include -> current/include
lrwxrwxrwx  lib -> current/lib
lrwxrwxrwx  libc++ -> current/libc++
lrwxrwxrwx  libstdc++fixup -> current/libstdc++fixup
lrwxrwxrwx  mpi -> current/mpi
lrwxrwxrwx  omp -> current/omp
drwxr-xr-x  r206161-20140414
drwxr-xr-x  rpm
lrwxrwxrwx  scan-build -> current/scan-build
lrwxrwxrwx  scan-view -> current/scan-view
lrwxrwxrwx  share -> current/share
lrwxrwxrwx  sleef -> current/sleef
drwxr-xr-x  stage1
lrwxrwxrwx  wbin -> current/wbin
```

These symlinks are only created by the post-install scripts in the RPMs if they don't already exist. As a result, upon subsequent upgrades, you'll need to manually update the 'current' symlink as desired.

The corresponding source RPMs are here:

```
```
To rebuild these, first build and install (or just install) the stage1 packages, then build the bgclang-binutils RPM and install it, then build the bgclang RPM and install it. Then the others can be built. When building the bgclang RPM (and the other packages that are built with bgclang) specify the same --dbpath command-line argument provided to rpm when installing the built stage1 and bgclang RPMs (if any).

r205936-20140409

A non-root (regular) user can install these RPMs (because they are relocatable), but in addition to specifying the installation prefix (with the --prefix argument), an alternate RPM database directory needs to be specified (in a directory to which you actually have write permission). For example, to install bgclang into the /tmp/bgclang directory using /tmp/bgclang/rpm as the RPM database directory, run:

```
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang
gclang-binutils-r205936-20140409-1-1.ppc64.rpm
bgclang-r205936-20140409-1-1.ppc64.rpm
bgclang-compiler-rt-r205936-20140409-1-1.ppc64.rpm
bgclang-libcxx-r205936-20140409-1-1.ppc64.rpm
bgclang-libomp-r205936-20140409-1-1.ppc64.rpm
bgclang-sleef-r205936-20140409-1-1.ppc64.rpm
```
If you've not installed the stage1 packages (which is a clang configured to build for the login nodes), you'll need to do that first:

```bash
rpm -Uvh --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang bgclang-stage1-3.4-1.ppc64.rpm bgclang-stage1-libcxx-
```

If the installation fails with an error like:

```
error: Failed dependencies:
  /bin/sh is needed by bgclang-r205936-20140409-1-1.ppc64
```

Then first install the vpkg-bin-sh-1-1.ppc64.rpm package (which is a virtual package which exists only to satisfy this /bin/sh dependency):

```bash
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang vpkg-bin-sh-1-1.ppc64.rpm
```

After the install is complete, the prefix directory should look like this:

```bash
$ ls -l /tmp/bgclang | awk '{print $1, $9, $10, $11}'
lrwxrwxrwx bin -> current/bin
lrwxrwxrwx compiler-rt -> current/compiler-rt
lrwxrwxrwx current -> r189357-20130827
lrwxrwxrwx docs -> current/docs
lrwxrwxrwx include -> current/include
lrwxrwxrwx lib -> current/lib
lrwxrwxrwx libc++ -> current/libc++
lrwxrwxrwx libstdc++fixup -> current/libstdc++fixup
lrwxrwxrwx mpi -> current/mpi
lrwxrwxrwx omp -> current/omp
drwxr-xr-x r205936-20140409
drwxr-xr-x rpm
lrwxrwxrwx scan-build -> current/scan-build
lrwxrwxrwx scan-view -> current/scan-view
lrwxrwxrwx share -> current/share
lrwxrwxrwx sleef -> current/sleef
drwxr-xr-x stage1
lrwxrwxrwx wbin -> current/wbin
```

These symlinks are only created by the post-install scripts in the RPMs if they don't already exist. As a result, upon subsequent upgrades, you'll need to manually update the 'current' symlink as desired.

The corresponding source RPMs are here:

To rebuild these, first build and install (or just install) the stage1 packages, then build the bgclang-binutils RPM and install it, then build the bgclang RPM and install it. Then the others can be built. When building the bgclang RPM (and the other packages that are built with bgclang) specify the same --dbpath command-line argument provided to rpm when installing the built stage1 and bgclang RPMs (if any).

r203443-20140309

A non-root (regular) user can install these RPMs (because they are relocatable), but in addition to specifying the installation prefix (with the --prefix argument), an alternate RPM database directory needs to be specified (in a directory to which you actually have write permission). For example, to install bgclang into the /tmp/bgclang directory using /tmp/bgclang/rpm as the RPM database directory, run:

```
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang bgclang-binutils-r203443-20140309-1-1.ppc64.rpm
bgclang-r203443-20140309-1-1.ppc64.rpm
bgclang-compiler-rt-r203443-20140309-1-1.ppc64.rpm
bgclang-libcxxxx-r203443-20140309-1-1.ppc64.rpm
bgclang-sleef-r203443-20140309-1-1.ppc64.rpm
```

If you've not installed the stage1 packages (which is a clang configured to build for the login nodes), you'll need to do that first:
If the installation fails with an error like:

```
error: Failed dependencies:
    /bin/sh is needed by bgclang-r203443-20140309-1-1.ppc64
```

Then first install the vpkg-bin-sh-1-1.ppc64.rpm package (which is a virtual package which exists only to satisfy this /bin/sh dependency):

```
rpm -Uhv --dbpath /tmp/bgclang/rpm --prefix /tmp/bgclang vpkg-bin-sh-1-1.ppc64.rpm
```

After the install is complete, the prefix directory should look like this:

```
$ ls -l /tmp/bgclang | awk '{print $1, $9, $10, $11}'
lrwxrwxrwx bin -> current/bin
lrwxrwxrwx compiler-rt -> current/compiler-rt
lrwxrwxrwx current -> r189357-20130827
lrwxrwxrwx docs -> current/docs
lrwxrwxrwx include -> current/include
lrwxrwxrwx lib -> current/lib
lrwxrwxrwx libc++ -> current/libc++
lrwxrwxrwx libstdc++fixup -> current/libstdc++fixup
lrwxrwxrwx mpi -> current/mpi
lrwxrwxrwx omp -> current/omp
drwxr-xr-x r203443-20140309
   drwxr-xr-x rpm
   lrwxrwxrwx scan-build -> current/scan-build
   lrwxrwxrwx scan-view -> current/scan-view
   lrwxrwxrwx share -> current/share
   lrwxrwxrwx sleef -> current/sleef
   drwxr-xr-x stage1
   lrwxrwxrwx wbin -> current/wbin
```

These symlinks are only created by the post-install scripts in the RPMs if they don’t already exist. As a result, upon subsequent upgrades, you’ll need to manually update the ‘current’ symlink as desired.

The corresponding source RPMs are here:

To rebuild these, first build and install (or just install) the stage1 packages, then build the bgclang-binutils RPM and install it, then build the bgclang RPM and install it. Then the others can be built. When building the bgclang RPM (and the other packages that are built with bgclang) specify the same --dbpath command-line argument provided to rpm when installing the built stage1 and bgclang RPMs (if any).

The corresponding source RPMs are here:


See instructions below regarding how to install these (but also install the libomp package).
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r200401-20140129-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r200401-20140129-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/vpkg-bin-sh-1-1.ppc64.rpm

See instructions below regarding how to install these (but also install the libomp package).

The corresponding source RPMs are here:
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/vpkg-bin-sh-1-1.src.rpm

r193439-20131025
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-binutils-r193439-20131025-2-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-r193439-20131025-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-sleef-r193439-20131025-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libcxx-r193439-20131025-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r193439-20131025-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r193439-20131025-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/vpkg-bin-sh-1-1.ppc64.rpm

See instructions below regarding how to install these.

The corresponding source RPMs are here:
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-binutils-r193439-20131025-2-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-r193439-20131025-1-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMs/bgclang-libomp-r193439-20131025-1-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMs/vpkg-bin-sh-1-1.src.rpm

r192411-20131010

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-binutils-r192411-20131010-2-1.ppc64.rpm (updated)
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-r192411-20131010-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-sleef-r192411-20131010-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libcxx-r192411-20131010-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r192411-20131010-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r192411-20131010-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/vpkg-bin-sh-1-1.ppc64.rpm

See instructions below regarding how to install these.

The corresponding source RPMs are here:

http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMs/bgclang-r192411-20131010-1-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMs/vpkg-bin-sh-1-1.src.rpm

r192297-20131009

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-binutils-r192297-20131009-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-r192297-20131009-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-sleef-r192297-20131009-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libcxx-r192297-20131009-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r192297-20131009-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r192297-20131009-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/vpkg-bin-sh-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r192297-20131009-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/vpkg-bin-sh-1-1.ppc64.rpm

See instructions below regarding how to install these.

The corresponding source RPMs are here:
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-r192297-20131009-1-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/vpkg-bin-sh-1-1.src.rpm

r192001-20131004

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-binutils-r192001-20131004-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r192001-20131004-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-sleef-r192001-20131004-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libcxx-r192001-20131004-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r192001-20131004-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r192001-20131004-1-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/vpkg-bin-sh-1-1.ppc64.rpm

See instructions below regarding how to install these.

The corresponding source RPMs are here:
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-r192001-20131004-1-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-sleef-r192001-20131004-1-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-libomp-r192001-20131004-1-1.src.rpm
See instructions below regarding how to install these.

The corresponding source RPMs are here:

http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/bgclang-r190771-20130914-1-1.src.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/SRPMS/vpkg-bin-sh-1-1.src.rpm

r189357-20130827 (v3)

In order to make this process easier, I've converted the various build scripts and patches into relocatable RPMs. Relocatable RPMs can be installed under an arbitrary prefix directory. Download these RPMs:

http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-r189357-20130827-3-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-sleef-r189357-20130827-3-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libcxx-r189357-20130827-3-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-libomp-r189357-20130827-3-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/bgclang-compiler-rt-r189357-20130827-3-1.ppc64.rpm
http://www.mcs.anl.gov/~hfinkel/bgclang/RPMS/ppc64/vpkg-bin-sh-1-1.ppc64.rpm

A non-root (regular) user can install these RPMs (because they are relocatable), but in addition to specifying the installation prefix (with the --prefix argument), an alternate RPM database directory needs to be specified (in a directory to which you actually have write permission). For example, to install bgclang into the /tmp/bgclang directory using /tmp/rpm as the RPM database directory, run:
If the installation fails with an error like:

```
error: Failed dependencies:
    /bin/sh is needed by bgclang-r189357-20130827-3-1.ppc64
```

Then first install the vpkg-bin-sh-1-1.ppc64.rpm package (which is a virtual package which exists only to satisfy this /bin/sh dependency):

```
rpm -Uhv --dbpath /tmp/rpm --prefix /tmp/bgclang vpkg-bin-sh-1-1.ppc64.rpm
```

After the install is complete, the prefix directory should look like this:

```
$ ls -l /tmp/bgclang | awk '{print $1, $9, $10, $11}'
lrwxrwxrwx bin -> current/bin
lrwxrwxrwx compiler-rt -> current/compiler-rt
lrwxrwxrwx current -> r189357-20130827
lrwxrwxrwx docs -> current/docs
lrwxrwxrwx include -> current/include
lrwxrwxrwx lib -> current/lib
lrwxrwxrwx libc++ -> current/libc++
lrwxrwxrwx libstdc++fixup -> current/libstdc++fixup
lrwxrwxrwx mpi -> current/mpi
drwxr-xr-x r189357-20130827
lrwxrwxrwx scan-build -> current/scan-build
lrwxrwxrwx scan-view -> current/scan-view
lrwxrwxrwx share -> current/share
lrwxrwxrwx sleef -> current/sleef
lrwxrwxrwx wbin -> current/wbin
```

These symlinks are only created by the post-install scripts in the RPMs if they don't already exist. As a result, upon subsequent upgrades, you'll need to manually update the 'current' symlink as desired.

The corresponding source RPMs are here:

To rebuild these, first build the bgclang RPM, and install it. Then the others can be built, specifying the same --dbpath command-line argument provided rpm when installing the built bgclang RPM (if any).

r189357-20130827 (v2)
- LLVM, Clang, libc++, compiler-rt, SLEEF (patches, wrapper and build scripts): r189357-20130827-files-v2.tar.gz

r189357-20130827
- LLVM, Clang, libc++, compiler-rt, SLEEF (patches, wrapper and build scripts): r189357-20130827-files.tar.gz

r188569-20130816 (v3)
- LLVM, Clang, libc++, compiler-rt (patches, wrapper and build scripts): r188569-20130816-files-v3.tar.gz

r188569-20130816 (v2)
- LLVM, Clang, libc++, compiler-rt (patches, wrapper and build scripts): r188569-20130816-files-v2.tar.gz

r188569-20130816
- LLVM, Clang, libc++, compiler-rt (patches, wrapper and build scripts): r188569-20130816-files.tar.gz

r188410-20130814
- LLVM, Clang, libc++, compiler-rt (patches, wrapper and build scripts): r188410-20130814-files.tar.gz

r186563-20130718
- LLVM, Clang, libc++, compiler-rt (patches, wrapper and build scripts): r186563-20130718-files.tar.gz

r185769-20130706 (v3)
- LLVM, Clang, libc++, compiler-rt (patches, wrapper and build scripts): r185769-20130706-files-v3.tar.gz

r185769-20130706 (v2)
- LLVM, Clang, libc++, compiler-rt (patches, wrapper and build scripts): r185769-20130706-files-v2.tar.gz

r185769-20130706
- LLVM, Clang, libc++, compiler-rt (patches, wrapper and build scripts): r185769-20130706-files.tar.gz

r185415-20130701 (v3)
- LLVM, Clang, libc++, compiler-rt (patches, wrapper and build scripts): r185415-20130701-files-v3.tar.gz

r185415-20130701 (v2)
- LLVM, Clang, libc++, compiler-rt (patches, wrapper and build scripts): r185415-20130701-files-v2.tar.gz


Significant improvements include:

- Support for type-safety attributes (clang feature developed by Dmitri Gribenko for type checking calls to MPI functions).
- LLVM now generates pre-increment load/store instructions (only \(r+\text{imm}\) form supported so far, \(r+r\) form, necessary for QPX pre-increment, is still under development).

Note: This information applies only to building from the outdated tar.gz archives. Please use the source RPMs if you wish to build bgclang yourself (or install the binary RPMs if you don't).

Follow the normal directions for checking out LLVM and clang from the llvm.org repositories (using the revision number specified in the archive name). Then apply the provided patches. Run configure (I recommend building from a directory different from the source directory). See the build scripts for how to compile everything.

For a better C++11 environment, you'll want to use libc++ as the C++ standard library implementation. The attached build script patches and builds libc++ so that it can correctly interoperate (be linked with) the libstdc++ which the MPI/PAMI implementation requires. The bgclang++11 (and corresponding MPI wrappers) setup this environment.
You'll need to build compiler-rt to use the address sanitizer feature. Do not checkout compiler-rt into the llvm directory (as per the upstream instructions), it will not be correctly cross compiled for the compute nodes. Instead, checkout compiler-rt into its own top-level directory and use the provided build script.

The install now includes a SIMD math library based on SLEEF. To make use of the patches in the archive, you'll need the source code from: http://shibatch.sourceforge.net/ (the current patches are against SLEEF version 2.80).

Repository Information

I have yet to figure out how to best mirror my local repositories. Here's why:

- LLVM uses subversion, and I've been using git-svn to manage local BG/Q changes.
- git-svn uses rebasing to apply local changes on top of upstream changes; this amounts to history rewriting (dangerous but effective). I cannot simply push from the git-svn-managed repository to the public one because doing so would confuse copies cloned from the public version.
- LLVM uses separate repositories for LLVM and clang, it is not clear how to best combine them into one public repository.

Thus, for the time being, we'll just need to use patches (now contained in the source RPMs below). As time allows, I'll move to a different setup for version control.

Attachments