



Gaussian 16 Source Code Installation Instructions, Rev. C.01

If you will be using Linda, build the regular Gaussian 16 program first, and then build the Linda version as described on page 2. Building a version with Linda from source requires the new revision of Linda (9.2).

1. Check that you have the correct versions of the OS, compiler, and libraries for your machine, as listed in the platform list file **platform_rev.pdf** on the CD (*rev* represents the revision of the program; replace with the revision you actually use). The latest version of this file is always available on our website at gaussian.com/g16/g16_plat.pdf.

Building G16 C.01 from source for x86_64 requires version 18.10 of the PGI compiler. Earlier versions are not supported, and some later versions have problems.

2. Select a group which will own the Gaussian files. Users who will run *Gaussian* should either already be in this group, or should have this added to their list of groups. Consult your system administrator if you need help with this process.
3. Mount the DVD, or determine where the system has mounted it automatically.
4. Change to the C shell, and set the **g16root** and **mntpnt** environment variables:

```
$ /bin/csh
% setenv mntpnt "/mnt/dvd"          # Set to wherever CD is mounted.
% setenv g16root "dir"             # dir=install location for G16.
% cd $g16root
```

5. Read the DVD, and run the script which sets group ownership::

```
% bzip2 -d -c $mntpnt/tar/*.tbz | tar xvf -
% chgrp -R grp g16                 # grp=group from step 1.
% cd g16
% ./bsd/install
```

6. Compile the program:

```
% source $g16root/g16/bsd/g16.login
% bsd/bldg16 >& make.log            # Will take between 0.5 and 3 hours.
```

See *Build Notes for Specific Versions* below for additional information about building various Gaussian 16 versions.

7. Check the end of **make.log** for successful completion. Confirm that the executables have been built. There should be 85 files:

```
% ls $g16root/g16/*.exe
```

8. You are now ready to run. Users will want to add the following to their **.login** or **.bash_profile** file:

```
# .login commands                                # .profile commands
setenv g16root "dir"                             g16root="dir"
setenv GAUSS_SCRDIR "sdir"                       GAUSS_SCRDIR="sdir"
source $g16root/g16/bsd/g16.login                 export g16root GAUSS_SCRDIR
                                                    . $g16root/g16/bsd/g16.profile
```

The *dir* in the first command is the location of the **g16** directory. For example, if the path to this directory is **/usr/local/g16**, then set **g16root** to **/usr/local**. The *sdir* in the second command is a directory to be used by default for *Gaussian 16* scratch files. There should be plenty of disk space available at this location.

Building G16 with Linda

Note: Successfully building and running Gaussian 16 with Linda requires Linda version 9.1. Although the executables will build with earlier versions of Linda, they will not run.

1. First install G16 from source
2. Mount the Linda CD using the same location as above. Now install Linda:

```
$ /bin/csh
% setenv mntpnt "/mnt/dvd"           # Set to wherever CD is mounted.
% setenv g16root "dir"              # dir=install location for G16.
% cd $g16root/g16
% bzip2 -d -c $mntpnt/tar/*.tbz | tar xvf -
% bsd/fixlinda                      # fixes paths and soft links
% source $g16root/g16/bsd/g16.login  # repeat command for Linda
% mg linda
```

3. You are now ready to run. Users will need to set the **%LindaWorkers** Link 0 command in their input files to specify the list of nodes to use for the parallel Gaussian calculation. You can also use **-W-** in **Default.Route** or the **-w** command line option.

Build Notes for Specific Versions

Mac OS X systems: A case-sensitive file system is required for building from source on Mac OS X systems.

IBM Power systems: When building from source on IBM Power systems, the default is to build for the current (Power8) processors. In order to build the versions for Power 5, 6, or 7 machines, you must specify the desired version as the second argument to the build script. For example, this command builds Gaussian 16 for Power 5 systems:

```
% bsd/bldg16 all ibmp5 >&make.log # or ibmp6 or ibmp7
```

Use **ibmp6** or **ibmp7** for Power 6 or Power 7 versions (respectively), In order to build for little-endian Power 8 Linux systems, use the keyword **ibmp8le**.

x86 and x86_64 systems: On x86 and x86_64 machines, the default for the build script is to builds for the current CPU type (or the closest supported CPU), use the current level 3 cache size and does not include support for GPUs. A specific CPU type can be specified as an argument to the build script, e.g.:

```
% bsd/bldg16 all sandybridge
```

to build executables compiled for Intel Sandy Bridge processors even if the current machine is a different type of x86_64. Similarly, this command:

```
% bsd/bldg16 all volta
```

will build with NVIDIA K40, K80, P100 and V100 GPU support and the current type of x86_64 processor. Use a command like this one:

```
% bsd/bldg16 all volta sandybridge
```

to turn on both GPU support and a particular CPU type.

By default, the build scripts check the size of level 3 cache on each chip and the number of CPUs per chip and set the parameter for the amount of cache each thread should try to use as L3 size divided by number of CPUs. If the executables are being built on one type of systems but will be used primarily on a different machine, then before building one should set the CACHESIZE environment variable to the amount of cache in bytes that each CPU/thread should use.

On x86_64 machines the supported machine types are: **ia32p4** (32-bit Pentium), **amd64** (legacy AMD machines), **em64t** (legacy Intel x86_64 machines), **istanbul** (Istanbul and later AMD processors), **nehalem** (Nehalem and later Intel processors), **sandybridge** (Intel Sandybridge and later processors) and **haswell** (Intel Haswell and Broadwell processors). Note that there is not a specific keyword for every processor. These keyword effectively bracket processor releases. For example, **sandybridge** executables will work on

subsequent Intel processors and should be used on later ones for which there is no specific build keyword. They will also work on Haswell processors, although there will be some performance advantages in building for Haswell/Broadwell specifically.

NEC Ace systems: Use the following command to build executables for NEC Ace systems:

```
% bsd/bldg16 all necsxace
```