

```
# $Id: README.debian,v 1.2 2017/06/12 20:50:59 pm Exp $
```

QUICK START:

1. Install packages that BSU depends on.

```
BLAS
LAPACK
GSL
PLPLOT
```

(see details below if a problem here)

2. Install BSU Binary Packages:

For 64 Bit machines:

```
sudo dpkg -i bsu_3.0.0-1_amd64.deb
OR
```

```
sudo gdebi bsu_3.0.0-1_amd64.deb
```

For 32 Bit machines:

```
sudo dpkg -i bsu_3.0.0-1_i386.deb
```

=====DETAILS=====

Tools for downloading and/or installation of packages include:

```
dpkg
apt-get
aptitude (like yum on CentOs or other Redhat)
synaptic (like yumex on CentOs or other Redhat, GUI Tool)
```

IMPORTANT NAMING ISSUE:

Exact package names will change with each new release of a distribution. The following examples will give you an idea of what to look for.

In general, Debian tends toward many sub-packages (compared to Redhat). To compile code, you will need development packages. These typically end in *-dev for Debian (compared to *-devel in Redhat).

RECOMMENDATION: Use the synaptic tool and search button to look for all the plplot packages, or all the lapack packages, etc., then install all of them.

=====BLAS and LAPACK (Linear Algebra)

IMPORTANT Fortran Notice: You will need libraries compiled with the same Fortran compiler as you plan on using with BSU. At the time of this writing, g77 is being replaced by gfortran.

```
libblas3gf (the gf suffix is for gfortran)
refblas3 (this is f77 or g77 compiled)
liblapack3gf (this is gfortran)
lapack3 (this is f77 or g77)
```

NOTE: It is ok to have all of above on same system. You will also need the development packages:

```
On Ubuntu
libblas-dev
liblapack-dev
```

```
On Debian (development packages end in -dev)
liblapack-dev
liblapack-doc
```

```
liblapack-pic
liblapack-test
liblapack3gf
libblas-dev
libblas-doc
libblas-test
libblas3gf
```

=====
=====GSL (Gnu Scientific Library)

For GSL, Debian names packages as the following:

```
gsl-bin
libgsl0-dbg
libgsl0-dev
libgsl0ldbl
gsl-ref-html
```

You may also want to install documentation packages for all of the above in case you wish to modify BSU or extend it.

=====
=====PLPLOT (Plotting Libraries for C, Fortran, etc.)

On a Ubuntu machine, there are many packages:

```
libplplot-c++9c2
libplplot-dev
libplplot-fortran9
libplplot9
libplplot9-java
octave-plplot
plplot-bin
plplot-doc
plplot-tcl
plplot-tcl-dev
plplot9-driver-cairo
plplot9-driver-gd
plplot9-driver-gnome2
plplot9-driver-wxwidgets
plplot9-driver-xwin
python-plplot
```

=====
=====SPECIFIC SOURCE PACKAGING DETAILS:

Debian (and Debian based distributions, like Ubuntu)

WHAT IS A SOURCE PACKAGE?

In Debian distributions, this would typically be several files:

```
bsu_3.0.0.orig.tar.gz
bsu_3.0.0-1.dsc
bsu_3.0.0-1.diff.gz
```

TIP: If you need a source package from a repository, you can do it with the apt-get command. Here is an example to get the blas package:

```
apt-get source blas
```

Assuming a working internet connection, apt-get would download the following files (or ones like these) into your current directory.

```
blas_1.2.orig.tar.gz
blas_1.2-1.3ubuntu4.dsc
blas_1.2-1.3ubuntu4.diff.gz
```

With the BSU package, you can download the files from <http://cgiss.boisestate.edu/~pm/downloads.php>

BUILDING A SOURCE PACKAGE:

You would unpack the source with a command like this:

```
dpkg-source -x bsu_3.0.0-1.dsc
```

This will untar the archive file into a directory, here named bsu-3.0.0. Change into that directory, then into the debian directory:

```
cd bsu-3.0.0/debian
```

You then edit what files you need (for example, at least edit the changelog file, use your email and make the release number name specific to yourself).

Then, change back one directory,

```
cd ../
```

and issue the command,

Builds both binary and source:

```
dpkg-buildpackage -rfakeroot -uc -us
```

Builds binary:

```
dpkg-buildpackage -rfakeroot -b -uc
```

And the package should start building. If all goes well, it will build a new package for your architecture, and place it just above the bsu-3.0.0 directory.

Of course, a lot of things can go wrong. If you are missing dependencies on your system, sometimes a good command to issue is

```
apt-get build-dep bsu
```

(where bsu would be the root name of whatever package you are building). This may seek out and install whatever you need.

TO BUILD A PACKAGE FROM SCRATCH

You will want to learn about dh_make and related programs. Too big a topic for this readme.