

# OSes and Prerequisites

You will often need to install some additional packages before you can build the **LSST Stack** from source.

## Official

The latest "factory" platform is currently CentOS 7. You should expect few if any problems on this platform, and bugs are likely to be promptly attended to. See the Linux section below for prerequisites.

**New for 11.0:** The minimum gcc version required to compile the stack is **gcc 4.8**. If you using our previous factory platform, RedHat/CentOS 6, and you are unable to upgrade to version 7 (which comes with gcc 4.8 as default) consult the Linux (legacy) section below

## Linux (current)

While we cannot support the **LSST Stack** on arbitrary OSes, we do endeavor to try different platforms and fix bugs on best efforts basis, You can get an idea of what is working for us from <https://ls.st/faq>

There are a number of prerequisite packages, depending on your Unix/Linux distribution and version, that you may need to install in order to build the LSST Stack from source. If you are building on a VM from a cloud provider such as AWS or Digital Ocean, the VM usually comes with core packages only and quite a few additional packages will be needed. If you have a more fully-featured desktop distribution, you may only need a few "-devel" packages. Unofficial Linux support is on a best-effort basis.

The link below is the Puppet manifest we use to provision various operating systems for trying out builds, so it is guaranteed to be current and tested. The 'Debian' section can be used for "apt-get" distros like Ubuntu; the 'RedHat' section is for "rpm"-based RedHat clones such as CentOS.

<https://github.com/lsst-sqre/puppet-lsststack/blob/master/manifests/params.pp>

Here are the prerequisites as of 11.0 for readability, but consult the canonical file above if you experience issues with this list.

<b>Debian /Ubuntu</b>  (use apt-get)	bison cmake curl flex g++ gettext git libbz2-dev libcurl4-openssl-dev libfontconfig1 libglib2.0-dev libncurses5-dev libreadline6-dev libx11-dev libxrender1 libxt-dev m4 make openjdk-7-jre perl-modules zlib1g-dev
<b>RedHat /CentOS</b>  (use yum)	blas bison bzip2 bzip2-devel cmake curl flex fontconfig freetype-devel gcc-c++ gcc-gfortran gettext git glib2-devel java libXext libXrender libXt-devel libcurl-devel libuuid-devel make ncurses-devel openssl-devel patch perl perl-ExtUtils-MakeMaker readline-devel zlib-devel

## Linux (legacy)

The minimum gcc version required to compile the stack is gcc 4.8. This comes as standard in the LSST "factory" platform, Red Hat / CentOS 7. On our previous factory platform, Red Hat / CentOS 6, you will need to use a more current version of gcc that what is available with your system. If you can go to Red Hat 7, we recommend that you do; if you cannot, we recommend that you use a newer gcc version for the stack by using a Software Collection (SCL) with a different version of devtoolset. This will enable you to safely use a different version of gcc (4.9) for the stack than that used by your operating system (4.4)

Here is how to do this (after the pre-requisite setup mentioned in the current Linux section but before you attempt to build the stack with newinstall.sh or eups distrib):

```
# you (or your IT administrator) only need to do this once:

sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
sudo yum install -y https://www.softwarecollections.org/en/scls/rhscl/rh-java-common/epel-6-x86_64/download/rhscl-rh-java-common-epel-6-x86_64.noarch.rpm
sudo yum install -y https://www.softwarecollections.org/en/scls/rhscl/devtoolset-3/epel-6-x86_64/download/rhscl-devtoolset-3-epel-6-x86_64.noarch.rpm
sudo yum install -y scl-utils
sudo yum install -y devtoolset-3

# you can do this any time you need to work with the stack, or exec it in your bash_profile
scl enable devtoolset-3 bash
```

## OS X

We have tested the stack on Yosemite (10.10). You will need the Xcode command line tools to build from source.

The stack is not currently supported on El Capitan OSX (11.\*). See [Known Installation issues for 11.0](#)

Versions prior to 10.9 have not been tested recently and may not work.

## Python

You can use your own 2.7.\* Python install or let the stack install *anaconda* in your local directory.

## Additional Science Software

*matplotlib* and *ds9* are used to display debugging from within Python code and to display results in various example iPython notebooks. *scipy* is also optionally used by some of our code. We do not distribute these directly, but *matplotlib* and *scipy* are included with *anaconda* and are readily available for most other Python distributions. Binary distributions of *ds9* are readily available from the official site.