how to install R from source code and configure text based mode in linux centos 7

[**20 Practical Examples of RPM Commands in Linux - Tecmint**](http://www.tecmint.com/20-practical-examples-of-rpm-commands-in-linux/)

*www.tecmint.com/20-practical-examples-of-rpm-commands-in-****linux****/*

Apr 10, 2013 - RPM (Red Hat Package Manager) is an default open source and most ... utility for Red Hat *based* systems like (RHEL, *CentOS* and Fedora). ... RPM is the only way to *install* packages under *Linux* systems, if you've *installed* packages using *source code*, then ... There are five basic *modes* for RPM command.

RPM is free and released under **GPL** (**General Public License**).

RPM keeps the information of all the installed packages under **/var/lib/rpm** database.

RPM is the only way to install packages under Linux systems, if you’ve installed packages using source code, then rpm won’t manage it.

RPM deals with **.rpm** files, which contains the actual information about the packages such as: **what it is**, **from where it comes**, **dependencies info**, **version info** etc.

1. **Install** : It is used to install any RPM package.
2. **Remove** : It is used to erase, remove or un-install any RPM package.
3. **Upgrade** : It is used to update the existing RPM package.
4. **Verify** : It is used to query about different RPM packages.
5. **Query** : It is used for the verification of any RPM package.

how to install r on linux centos 7 text mode

[https://paulmellorsblog.wordpress.com/.../**centos-7**-**text**-**mode**-**install**er/](https://paulmellorsblog.wordpress.com/.../centos-7-text-mode-installer/)

[**How to setup cent os 7 in text mode - YouTube**](https://www.youtube.com/watch?v=eplStIEmtlA)

### [How to install CentOS in text mode - Xmodulo](http://xmodulo.com/how-to-install-centos-in-text-mode.html)

[**How Install R / R Studio on CentOS 7 - LinOxide**](http://linoxide.com/linux-how-to/install-r-rstudio-centos-7/)

*linoxide.com/****linux****-how-to/****install****-****r****-rstudio-****centos-7****/*

[**"X startup failed, falling back to text mode" during CentOS-7 ...**](http://www.centos.org/forums/viewtopic.php?t=58815)

<http://unix.stackexchange.com/questions/149451/install-r-in-my-own-directory>

The easiest way to do this is to install R [from source](http://cran.rstudio.com):

$ wget http://cran.rstudio.com/src/base/R-3/R-3.1.1.tar.gz

$ tar xvf R-3.1.1.tar.gz

$ cd R-3.1.1

$ ./configure --prefix=$HOME/R

$ make && make install

The second-to-last step is the critical one. It configures R to be installed into a subdirectory of your own home directory.

To run it on Linux and similar systems, add $HOME/R/bin to your PATH. Then, commands like R and Rscript will work.

On OS X, the bin dir is buried in $HOME/R/R.framework/Versions/Current/Resources instead, due to the way OS X packaging works.

You can give --prefix=$HOME instead if you don't wish to have R and all its dependencies hidden in its own subdirectory. If you are on Linux and have $HOME/bin in your PATH already, doing so would avoid the need to edit the PATH. The downside is that it makes later uninstallation harder, since R would be intermingled among your other $HOME contents.

This general pattern applies to a large amount of Unix software you can install from source code. If the software has a configure script, it probably understands the --prefix option, and if not, there is usually some alternative with the same effect.

These features are common for a number of reasons. Your case — where you can't get root to install the software into a typical location so you install into $HOME instead — is actually one of the least common uses for this sort of feature. Here are some more common use cases:

* Circumstances may dictate a better location than the standard default (/usr/local) such as /usr, /opt/$PKGNAME, etc.
* Binary package building systems (RPM, DEB, PKG, Cygport...) typically build and install the package into a special staging directory, then pack that up in such a way that it expands into the desired installation location.

would recommend ./configure --prefix=$HOME/R --enable-R-shlib to make sure you compile R shared library. Otherwise, RStudio will complain

Build from source with ./configure --prefix=/local/data/project/behi ; make ; make install

If you're installing from an RPM package and it was created relocatable, you could use

rpm ... --prefix /local/data/project/behi

But not all packages are built with relocatable binaries, and I don't think debian packages have this option (although you might succeed with dpkg --instdir)

<http://stackoverflow.com/questions/30158570/install-r-from-source>

|  |  |
| --- | --- |
|  | I'm trying to install R from source in my home directory on a server running CentOS.  **I do not have root rights**, and I'm not permitted to write to `usr/local/include/'. |

wget http://cran.rstudio.com/src/base/R-3/R-3.2.0.tar.gz

tar xvf R-3.2.0.tar.gz

cd R-3.2.0

./configure --prefix=$HOME/R

In configuration step, I get error

configure:error:--with-readline=yes (default) and headers/libs are not available

In my understanding, it tells me that readline library is not available. So I try to install readline. I downloaded tar.gz file. and then I use the following command

tar xvf readline-6.3.tar.gz

cd readline-6.3

./configure --prefix=$HOME/readline

make

make install

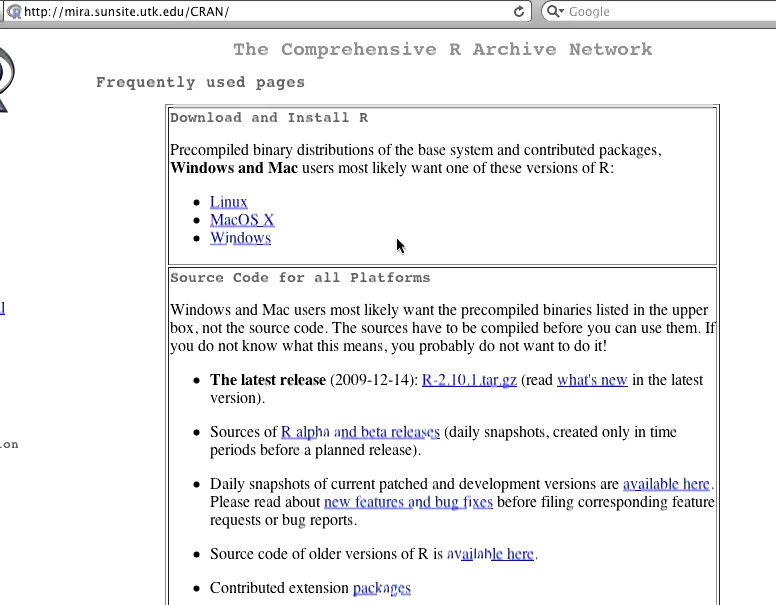
Things are fine, and there's an additional folder in my home directory named "readline".

When I go back and try to configure R again, I still get the same error message. How can I fix it?

<http://mira.sunsite.utk.edu/CRAN>

<http://cran.mirrors.hoobly.com/>

https://mirrors.nics.utk.edu/cran/

s