Cheatography

RPM Build Basics Cheat Sheet

by Craig Gardner via cheatography.com/23935/cs/5382/

tar archives

tar xzf { source-code-archive.tgz }	extract a gzip-compressed file
tar xjf { source-code-archive.tar.bz2 }	extract a bz2-compressed file
tar cjf { compressed-archive.tar.bz2 } { directory-to-archive }	create a tar archive from the contents of a directory

autotools

autoscan	Scans directory for source files to be used to build software
autoheader	Helps to manage C header files
aclocal	creates macros that help the other autotools
autoconf	Writes the "./configure" script
libtool	Sets up the build environment for application libraries
automake	Reads rules to create an all-purpose Makefile
autoreconf	A shortcut that sequentially runs the appropriate all previous autotools tools
./configure	setup the build scripts according the discovered environment
./configure --libdir=/usr/lib	set some important variables from the commandline without manually editing config files

Manual Build

make	compile and link according to rules in Makefile
make install	put the built binaries and other software files into their runtime location

RPM database queries

rpm -qa	Show all packages installed on the system
rpm -qi { packagename }	Show high-level information about an installed package
rpm -qf { filename }	Show which package owns filename
rpm -ql { packagename }	List the files controlled by a package
rpm -qlv { packagename }	Verbose listing of files

RPM database queries (cont)

rpm -qlvc { packagename }	Verbose listing of only the configuration files
rpm -qlvd { packagename }	Verbose listing of only the documentation files
rpm -qV { packagename }	Verify the permissions of the controlled files
rpm -qVv { packagename }	Verify, verbosely

RPM installation commands

rpm -i { filename.rpm }	Install the filename RPM package
rpm -Uvh { filename.rpm }	Update (will also install) the RPM package, verbosely, showing progess
rpm -Fvh { filename.rpm }	Freshen the RPM package, i.e.: update the installed software if the file contains a newer version/release of the package
rpm -U { http://some.website.com/filename.rpm }	Update a package using a URL
rpm -U --test { filename.rpm }	Test run of installation without actually installing anything
rpm -e { packagename }	Remove a package from the system
rpm -e --test { packagename }	Test the removal of a package without actually removing it

RPM build

rpmbuild -bb { packagename.spec }	Build the standard package RPM file using the information contained in the specfile
rpmbuild -bs { packagename.spec }	Build only the source RPM
rpmbuild -ba { packagename.spec }	Build both the binary and source RPM

By **Craig Gardner**
cheatography.com/craig-gardner/

Published 29th October, 2015.
Last updated 29th October, 2015.