

Prerequisites

1. Python 2 (2.6 or 2.7) or 3 (3.5 - 3.9) to run Spack
2. A C/C++ compiler for building
3. The `make` executable for building
4. The `tar`, `gzip`, `bzip2`, `xz` and optionally `zstd` executables for extracting source code
5. The `patch` command to apply patches
6. The `git` and `curl` commands for fetching
7. If using the `gpg` subcommand, `gnupg2` is required

Specs & dependencies

@ is optional version specifier (@1.2:1.4)	Ex: <code>amdfft w@2.2</code>
% is optional compiler specifier	Ex: <code>amdfft w@2.2 %aocc@2.2</code>
+ or - or ~ are optional variant specifiers for boolean variants	Ex: <code>+debug, -qt, or ~qt</code>
name=< val ue> is optional variant specifier	Ex: <code>amdbli s@2.2 thread s=o penmp</code>
name=< val ue> is optional compiler flag specifiers	Valid flag names are <code>cflags</code> , <code>cxxflags</code> , <code>fflags</code> , <code>cppflags</code> , <code>ldflags</code> , and <code>ldlibs</code>
target =<v alu e> os=<va lue> are optional architecture specifier	Ex: <code>target =zen2 os=aocc</code>
^ Dependency specs	Ex: <code>amdfftw ^openm pi@ 4.0.3</code>

Configuration

<code>spack config edit [-h] [--pri nt- file]</code> section	get and set configuration options, section: <code>compil ers /mi rro rs/ rep os/ pac kag es/ mod ule s/c o nf ig/ ups treams</code>
<code>spack config update [-hy] section</code>	update configuration options

Query packages

<code>spack info package</code>	get detailed information on a particular packages are on disk as installed
<code>spack depend encies package/package_specs</code>	show dependencies of a package
<code>spack dependents package</code>	show packages that depend on another
<code>spack find [-ldvf] [package]</code>	list and search installed packages
<code>spack graph package/package_specs</code>	generate graphs of package dependency relationships
<code>spack list</code>	list and search available packages
<code>spack location [-i, --inst all -dir] [-p, --pack age -dir] package</code>	print out locations of packages and spack directories
<code>spack providers virtual_package</code>	list packages that provide a particular virtual package



Build packages

<code>spack clean [-a, --all] [-s, --stage] [-p, --python-cache] [package(s)]</code>	remove temporary build files and/or downloaded archives
<code>spack gc [-y, --yes-to-all]</code>	remove specs that are now no longer needed
<code>spack install [-j, --jobs JOBS] [-v, --verbose] [--cache-only] [--source] package_specs</code>	build and install packages
<code>spack setup [-v, --verbose] package_specs</code>	create a configuration script and module, but don't build
<code>spack spec [-l, --long] [-I, --install-all-status] package(s)</code>	show what would be installed, given a spec
<code>spack uninstall [-R, --dependencies] [-y, --yes-to-all] package_specs</code>	remove installed packages

Create packages

<code>spack create [-hfb] [--keep-stage] [-n NAME] [url]</code>	create a new package file, url:url of package archive
<code>spack edit [-h] [package]</code>	open package files in \$EDITOR
<code>spack versions [-hs] [-c CONCURRENCY] package</code>	list available versions of a package

Environments

<code>spack add [-h] [-l LIST_NAME] package_specs</code>	add a spec to an environment
<code>spack concretize [-f, --force]</code>	concretize an environment and write a lockfile
<code>spack env create [--with-view] env [envfile]</code>	create virtual environments
<code>spack env list</code>	list all virtual environments
<code>spack env activate [-v, --with-view]</code>	manage virtual environments
<code>spack env remove [-hy] env [envfile]</code>	remove virtual environments

System

<code>spack arch [-h] [--known-targets] [-p -o -t] [-f -b]</code>	print architecture information about this machine
<code>spack compiler find add_paths</code>	find compilers from given path
<code>spack compiler remove compiler_spec</code>	remove given compiler
<code>spack compiler list/spack compilers</code>	list compilers available
<code>spack compiler info [-h] compiler_spec</code>	show compiler information

Developer

<code>spack cd [-i, --install-all-dir] [-p, --package-dir] package_spec</code>	cd to spack directories in the shell
<code>spack flake8 specific_files_to_check</code>	runs source code style checks on Spack. requires flake8



By [mohan2](#)
cheatography.com/mohan2/

Published 2nd December, 2020.
 Last updated 4th December, 2020.
 Page 2 of 3.

Sponsored by [Readable.com](#)
 Measure your website readability!
<https://readable.com>

User environment

<code>spack load package_specs</code>	add installed package to the user environment
<code>spack unload package_specs</code>	remove package from the user environment

Extensions

<code>spack activate installed_spec</code>	activate a package extension on installed package spec
<code>spack extensions [-s {packages, installed, activated, all}]</code>	list extensions for package package_spec

More help

<code>spack docs [-h]</code>	open spack documentation in a web browser
<code>spack help [-ha] [--spec] help_command</code>	get help on spack and its commands

References

Spack commands	https://spack.readthedocs.io/en/latest/command_index.html#command-reference
Prerequisites	https://spack.readthedocs.io/en/latest/getting_started.html#prerequisites
Specs & dependencies	https://spack.readthedocs.io/en/latest/basic_usage.html#specs-dependencies



By **mohan2**
cheatography.com/mohan2/

Published 2nd December, 2020.
 Last updated 4th December, 2020.
 Page 3 of 3.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>