

debootstrap(8) - Linux man page

Name

debootstrap - Bootstrap a basic Debian system

Synopsis

debootstrap [**OPTION...**] *SUITE* *TARGET* [*MIRROR* [*SCRIPT*]]

debootstrap [**OPTION...**] --second-stage

Description

debootstrap bootstraps a basic Debian system of *SUITE* (eg, lenny, squeeze, sid) into *TARGET* from *MIRROR* by running *SCRIPT*. *MIRROR* can be an `http://` URL, a `file:///` URL, or an `ssh:///` URL. Notice that `file:/` URLs are translated to `file:///` (correct scheme as described in RFC1738 for local filenames), and `file://` will **not** work. `ssh://USER@HOST/PATH` URLs are retrieved using **scp**; use of **ssh-agent** or similar is strongly recommended.

Debootstrap can be used to install Debian in a system without using an installation disk but can also be used to run a different Debian flavor in a **chroot** environment. This way you can create a full (minimal) Debian installation which can be used for testing purposes (see the **EXAMPLES** section). If you are looking for a chroot system to build packages please take a look at **pbuilder**.

Options

--arch=ARCH

Set the target architecture (use if dpkg isn't installed). See also --foreign.

--include=alpha,beta

Comma separated list of packages which will be added to download and extract lists.

--exclude=alpha,beta

Comma separated list of packages which will be removed from download and extract lists. WARNING: you can and probably will exclude essential packages, be careful using this option.

--components=alpha,beta

Use packages from the listed components of the archive.

--no-resolve-deps

By default, debootstrap will attempt to automatically resolve any missing dependencies, warning if any are found. Note that this is not a complete dependency resolve in the sense of dpkg or apt, and that it is far better to specify the entire base system than rely on this option. With this option set, this behaviour is disabled.

--variant=minbase|buldd|fakechroot|scratchbox

Name of the bootstrap script variant to use. Currently, the variants supported are minbase, which only includes essential packages and apt; buldd, which installs the build-essential packages into *TARGET*; and fakechroot, which installs the packages without root privileges. Finally

there is variant scratchbox, which is for creating targets for scratchbox usage. The default, with no **--variant=X** argument, is to create a base Debian installation in *TARGET*.

--keyring=KEYRING

Download signatures for retrieved Release files and check them against *KEYRING*. By default, Release file signatures are not checked.

--verbose

Produce more info about downloading.

--print-debs

Print the packages to be installed, and exit. Note that a *TARGET* directory must be specified so debootstrap can download Packages files to determine which packages should be installed, and to resolve dependencies. The *TARGET* directory will be deleted unless **--keep-debootstrap-dir** is specified.

--download-only

Download packages, but don't perform installation.

--foreign

Do the initial unpack phase of bootstrapping only, for example if the target architecture does not match the host architecture. A copy of debootstrap sufficient for completing the bootstrap process will be installed as */debootstrap/debootstrap* in the target filesystem. You can run it with the **--second-stage** option to complete the bootstrapping process.

--second-stage

Complete the bootstrapping process. Other arguments are generally not needed.

--second-stage-target=DIR

Run second stage in a subdirectory instead of root. (can be used to create a foreign chroot) (requires **--second-stage**)

--keep-debootstrap-dir

Don't delete the */debootstrap* directory in the target after completing the installation.

--unpack-tarball=FILE

Acquire .debs from tarball *FILE* instead of downloading via *http*.

--make-tarball=FILE

Instead of bootstrapping, make a tarball (written to *FILE*) of the downloaded packages. The resulting tarball may be passed to a later

--unpack-tarball.

--boot-floppies

Used for internal purposes by boot-floppies

--debian-installer

Used for internal purposes by the debian-installer

Example

To setup a *squeeze* system:

```
#  
debootstrap squeeze ./squeeze-chroot http://ftp.us.debian.org/debian
```

```
# debootstrap squeeze ./squeeze-chroot file:///PATH_TO_LOCAL_MIRROR/debian
```

Full process to create a complete Debian installation of *sid* (unstable):

```
main # cd / ; mkdir /sid-root
main # debootstrap sid /sid-root http://ftp.us.debian.org/debian/
[ ... watch it download the whole system ]
main # echo "proc /sid-root/proc proc defaults 0 0" >> /etc/fstab
main # mount proc /sid-root/proc -t proc
main # echo "sysfs /sid-root/sys sysfs defaults 0 0" >> /etc/fstab
main # mount sysfs /sid-root/sys -t sysfs
main # cp /etc/hosts /sid-root/etc/hosts
main # chroot /sid-root /bin/bash
chroot # dselect
[ you may use aptitude, install mc and vim ... ]
main # echo "8:23:respawn:/usr/sbin/chroot /sid-root " \
"/sbin/getty 38400 tty8" >> /etc/inittab
[ define a login tty that will use this system ]
main # init q
[ reload init ]
```

Note that you will generally need a recent version of debootstrap to do this; the version currently in stable will generally have stopped working due to changes to unstable shortly after the last release.

Author

debootstrap was written by Anthony Towns <ajt@debian.org>. This manpage was written by Matt Kraai <kraai@debian.org>.

Referenced By

[**fakechroot**](#)(1), [**febootstrap**](#)(8)