

NAME

schroot — securely enter a chroot environment

SYNOPSIS

```
schroot [-h|--help | -V|--version | -l|--list | -i|--info | --config | --location | --automatic-session
| -b|--begin-session | --recover-session | -r|--run-session | -e|--end-session] [-f|--force] [-n session-name|--session-name=session-name] [-d directory|--directory=directory] [-u user|--user=user]
[-p|--preserve-environment] [-q|--quiet | -v|--verbose] [-c chroot|--chroot=chroot | --all |
--all-chroots | --all-sessions] [COMMAND [ ARG1 [ ARG2 [ ARGn]]]]
```

DESCRIPTION

schroot allows the user to run a command or a login shell in a chroot environment. If no command is specified, a login shell will be started in the user's current working directory inside the chroot.

The command is a program, plus as many optional arguments as required. Each argument may be separately quoted.

The directory the command or login shell is run in depends upon the context. See `--directory` option below for a complete description.

If the user is not an allowed user, or a member of the allowed groups (or if changing to root, the allowed root users or allowed root groups) for the specified chroot(s), the user will be required to authenticate themselves (typically with a password, but this depends upon the PAM configuration). All chroot usage will be logged in the system logs.

If no chroot is specified, the chroot name or alias 'default' will be used as a fallback. This is equivalent to "`--chroot=default`".

OPTIONS

schroot accepts the following options:

Actions

-h, --help

Show help summary.

-V, --version

Print version information.

-l, --list

List all available chroots.

-i, --info

Print detailed information about the specified chroots.

--config

Print configuration of the specified chroots. This is useful for testing that the configuration in use is the same as the configuration file. Any comments in the original file will be missing.

--location

Print location (path) of the specified chroots. Note that chroot types which can only be used within a session will not have a location until they are active.

General options

-q, --quiet

Print only essential messages.

-v, --verbose

Print all messages.

Chroot selection

-c, --chroot=*chroot*

Specify a chroot or active session to use. This option may be used multiple times to specify more than one chroot, in which case its effect is similar to `--all`.

-a, --all

Select all chroots and active sessions. When a command has been specified, the command will be run in all chroots and active sessions. If `--info` has been used, display information about all chroots. This option does not make sense to use with a login shell (when no command has been specified). This option is equivalent to “`--all-chroots --all-sessions`”.

--all-chroots

Select all chroots. Identical to `--all`, except that active sessions are not considered.

--all-sessions

Select all active sessions. Identical to `--all`, except that chroots are not considered.

Chroot environment**-d, --directory=*directory***

Change to *directory* inside the chroot before running the command or login shell. If *directory* is not available, schroot will exit with an error status.

The default behaviour is as follows (all directory paths are inside the chroot). A login shell is run in the current working directory. If this is not available, it will try \$HOME (when `\--preserve-environment` is used), then the user’s home directory, and / inside the chroot in turn. A command is always run in the current working directory inside the chroot. If none of the directories are available, schroot will exit with an error status.

-u, --user=*user*

Run as a different user. The default is to run as the current user. If required, the user may be required to authenticate themselves with a password.

-p, --preserve-environment

Preserve the user’s environment inside the chroot environment. The default is to use a clean environment; this option copies the entire user environment and sets it in the session. The environment variables allowed are subject to certain restrictions; see the section “*Environment*”, below.

Session actions**--automatic-session**

Begin, run and end a session automatically. This is the default action, so does not require specifying in normal operation.

-b, --begin-session

Begin a session. A unique session identifier (session ID) is returned on standard output. The session ID is required to use the other session options. Note that the session identifier may be specified with the `--session-name` option.

--recover-session

Recover an existing session. If an existing session has become unavailable, for example becoming unmounted due to a reboot, this option will make the session available for use again, for example by remounting it.

-r, --run-session

Run an existing session. The session ID is specified with the `--chroot` option.

-e, --end-session

End an existing session. The session ID is specified with the `--chroot` option.

Session options**-n, --session-name=*session-name***

Name a session. The specified *session-name* replaces the default session name containing an automatically-generated session ID.

-f, --force

Force a session operation, even if it would otherwise fail. This may be used to forcibly end a session, even if it has active users. This does not guarantee that the session will be ended cleanly; filesystems may not be unmounted, for example.

EXAMPLES**List available chroots**

```
$ schroot -l
default
etch
sid
testing
unstable
```

Get information about a chroot

```
$ schroot -i -c sid
— Chroot —
Name                sid
Description         Debian sid (unstable)
Type                plain
Priority            3
Users               rleigh
Groups              sbuild
Root Users
Root Groups         sbuild
Aliases             unstable unstable-sbuild unstable-p
owerpc-sbuild
Environment Filter  (BASH_ENV|CDPATH|ENV|HOSTALIASES|IFS|KRB5_CONFIG|KRBCONFDIR|KRBTKFILE|KRB_CONF|LD_*|LOCALDOMAIN|NLSPATH|PATH_LOCALE|RES_OPTIONS|TERMINFO|TERMINFO_DIRS|TERMPATH)$
Run Setup Scripts   true
Run Execution Scripts false
Script Configuration script-defaults
Session Managed     true
Personality         linux32
Location            /srv/chroot/sid
```

Use `--all` or `-c` multiple times to use all or multiple chroots, respectively.

Running commands in a chroot

```
$ schroot -c sid /bin/ls
[sid chroot] Running command: "/bin/ls"
CVS                sbuild-chroot.c    sbuild-session.h  schroot.conf.5
Makefile           sbuild-chroot.h    schroot.1         schroot.conf.5.in
Makefile.am        sbuild-config.c    schroot.1.in
Makefile.in        sbuild-config.h    schroot.c
pam                sbuild-session.c   schroot.conf
```

```
$ schroot -c sid -- ls -l | head -n 5
[sid chroot] Running command: "ls -l"
ABOUT-NLS
AUTHORS
COPYING
ChangeLog
INSTALL
```

Use `--` to allow options beginning with `'-` or `'--` in the command to run in the chroot. This prevents them being interpreted as options for schroot itself. Note that the top line was echoed to standard error, and the remaining lines to standard output. This is intentional, so that program output from commands run in the chroot may be piped and redirected as required; the data will be the same as if the command was run directly on the host system.

Switching users

```
$ schroot -c sid -u root
```

```
Password:
```

```
[sid chroot] (rleigh→root) Running login shell: "/bin/bash"
```

```
#
```

If the user 'rleigh' was in *root-users* in */usr/local/etc/schroot/schroot.conf*, or one of the groups he belonged to was in *root-groups*, they would be granted root access without authentication, but the PAM authorisation step is still applied.

Sessions

A chroot may be needed to run more than one command. In particular, where the chroot is created on the fly from an LVM LV or a file on disc, there is a need to make the chroot persistent while a given task (or set of tasks) is performed. Sessions exist for this purpose. For simple chroot types such as 'plain' and 'directory', sessions may be created but are not strictly necessary.

Let's start by looking at a session-capable chroot:

```
% schroot -i -c sid-snap
```

```
— Chroot —
Name                sid-snap
Description         Debian sid snapshot
Type                lvm-snapshot
Priority             3
Users               maks rleigh
Groups              sbuild
Root Users
Root Groups         root sbuild
Aliases
Environment Filter  (BASH_ENV|CDPATH|ENV|HOSTALIASES|IFS|KRB5_CONFIG|KRBCONFDIR|KRBTKFILE|KRB_CONF|LD_*|LOCALDOMAIN|NLSPATH|PATH_LOCALE|RES_OPTIONS|TERMINFO|TERMINFO_DIRS|TERMPATH)$
Run Setup Scripts   true
Run Execution Scripts true
Script Configuration script-defaults
Session Managed     true
Personality         linux
Device              /dev/hda_vg/sid_chroot
Mount Options       -o atime,async,user_xattr
Source Users
Source Groups       root rleigh
Source Root Users
Source Root Groups  root rleigh
LVM Snapshot Options --size 2G -c 128
```

Note that the *Session Managed* option is set to 'true'. This is a requirement in order to use session management, and is supported by most chroot types. Next, we will create a new session:

```
% schroot -b -c sid-snap
```

```
sid-snap-46195b04-0893-49bf-beb8-0d4ccc899f0f
```

The session ID of the newly-created session is returned on standard output. It is common to store it like this:

```
% SESSION=$(schroot -b -c sid-snap)
```

```
% echo $SESSION
```

```
sid-snap-46195b04-0893-49bf-beb8-0d4ccc899f0f
```

The session may be used just like any normal chroot. This is what the session looks like:

```
% schroot -i -c sid-snap-46195b04-0893-49bf-beb8-0d4ccc899f0f
  — Session —
  Name                sid-snap-46195b04-0893-49bf-beb8-0d
4ccc899f0f
  Description         Debian sid snapshot
  Type                lvm-snapshot
  Priority             3
  Users               maks rleigh
  Groups              sbuild
  Root Users
  Root Groups         root sbuild
  Aliases
  Environment Filter  (BASH_ENV|CDPATH|ENV|HOSTALIASES|IFS|
S|KRB5_CONFIG|KRB5CONFDIR|KRBTKFILE|KRB_CONF|LD_*|LOCALDOMAI
N|NLSPATH|PATH_LOCALE|RES_OPTIONS|TERMINFO|TERMINFO_DIRS|TER
MPATH)$
  Run Setup Scripts   true
  Run Execution Scripts true
  Script Configuration script-defaults
  Session Managed     true
  Personality         linux
  Mount Location      /var/lib/schroot/mount/sid-snap-461
95b04-0893-49bf-beb8-0d4ccc899f0f
  Path                /var/lib/schroot/mount/sid-snap-461
95b04-0893-49bf-beb8-0d4ccc899f0f
  Mount Device        /dev/hda_vg/sid-snap-46195b04-0893-
49bf-beb8-0d4ccc899f0f
  Device              /dev/hda_vg/sid_chroot
  Mount Options       -o atime,async,user_xattr
  Source Users
  Source Groups       root rleigh
  Source Root Users
  Source Root Groups root rleigh
  LVM Snapshot Device /dev/hda_vg/sid-snap-46195b04-0893-
49bf-beb8-0d4ccc899f0f
  LVM Snapshot Options --size 2G -c 128
```

Now the session has been created, commands may be run in it:

```
% schroot -r -c sid-snap-46195b04-0893-49bf-beb8-0d4ccc899f0f -- uname -sr
I: [sid-snap-46195b04-0893-49bf-beb8-0d4ccc899f0f chroot] Running com-
mand: "uname -sr"
Linux 2.6.18-3-powerpc
% schroot -r -c $SESSION -- uname -sr
I: [sid-snap-fe170af9-d9be-4800-b1bd-de275858b938 chroot] Running com-
mand: "uname -sr"
Linux 2.6.18-3-powerpc
```

When all the commands to run in the session have been performed, the session may be ended:

```
% schroot -e -c sid-snap-46195b04-0893-49bf-beb8-0d4ccc899f0f
% schroot -e -c $SESSION
```

Finally, the session names can be long and unwieldy. A name may be specified instead of using the automatically generated session ID:

```
% schroot -b -c sid-snap -n my-session-name
```

my-session-name

TROUBLESHOOTING

If something is not working, and it's not clear from the error messages what is wrong, try using the `--debug=level` option to turn on debugging messages. This gives a great deal more information. Valid debug levels are 'none', and 'notice', 'info', 'warning' and 'critical' in order of increasing severity. The lower the severity level, the more output.

If you are still having trouble, the developers may be contacted on the mailing list:

Debian buildd-tools Developers
<buildd-tools-devel@lists.alioth.debian.org>

BUGS

On the **mips** and **mipsel** architectures, Linux kernels up to and including at least version 2.6.17 have broken **personality(2)** support, which results in a failure to set the personality. This will be seen as an "Operation not permitted" (EPERM) error. To work around this problem, set *personality* to 'undefined', or upgrade to a more recent kernel.

ENVIRONMENT

By default, the environment is not preserved, and the following environment variables are defined: HOME, LOGNAME, PATH, SHELL, TERM (preserved if already defined), and USER. The environment variables SCHROOT_COMMAND, SCHROOT_USER, SCHROOT_GROUP, SCHROOT_UID and SCHROOT_GID are set inside the chroot specifying the command being run, the user name, group name, user ID and group ID, respectively.

The following, potentially dangerous, environment variables are removed for safety by default: BASH_ENV, CDPATH, ENV, HOSTALIASES, IFS, KRB5_CONFIG, KRBCONFDIR, KRBTKFILE, KRB_CONF, LD_.*, LOCALDOMAIN, NLSPATH, PATH_LOCALE, RES_OPTIONS, TERMINFO, TERMINFO_DIRS, and TERMPATH. If desired, the *environment-filter* configuration key will allow the exclusion list to be modified; see **schroot.conf(5)** for further details.

FILES

/usr/local/etc/schroot/schroot.conf

The system-wide chroot definition file. This file must be owned by the root user, and not be writable by other.

/usr/local/etc/schroot/setup.d

/usr/local/etc/schroot/exec.d

The system-wide chroot setup and execution directories. See **schroot-setup(5)**.

/etc/pam.d/schroot

PAM configuration.

AUTHORS

Roger Leigh.

COPYRIGHT

Copyright © 2005–2007 Roger Leigh <rleigh@debian.org>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

SEE ALSO

dchroot(1), **sbuid(1)**, **chroot(2)**, **schroot-setup(5)**, **schroot.conf(5)**.

NAME

schroot.conf — chroot definition file for schroot

DESCRIPTION

schroot.conf is a plain UTF-8 text file, describing the chroots available for use with sbuild.

Comments are introduced following a '#' ("hash") character at the beginning of a line, or following any other text. All text right of the '#' is treated as a comment.

The configuration format is an INI-style format, split into groups of key-value pairs separated by section names in square brackets.

General options

A chroot is defined as a group of key-value pairs, which is started by a name in square brackets on a line by itself. The file may contain multiple groups which therefore define multiple chroots.

A chroot definition is started by the name of the chroot in square brackets. For example,

```
[sid]
```

This is then followed by several key-value pairs, one per line:

type=type

The type of the chroot. Valid types are 'plain', 'directory', 'file', 'block-device' and 'lvm-snapshot'. If empty or omitted, the default type is 'plain'.

description=description

A short description of the chroot. This may be localised for different languages; see the section "*Localisation*" below.

priority=number

Set the priority of a chroot. *number* is a positive integer indicating whether a distribution is older than another. For example, "oldstable" and "oldstable-security" might be '0', while "stable" and "stable-security" are '1', "testing" is '2' and "unstable" is '3'. The values are not important, but the difference between them is. This is used by sbuild when using the schroot chroot mode.

users=user1,user2,...

A comma-separated list of users which are allowed access to the chroot. If empty or omitted, no users will be allowed access (unless a group they belong to is also specified in *groups*).

groups=group1,group2,...

A comma-separated list of groups which are allowed access to the chroot. If empty or omitted, no groups of users will be allowed access.

root-users=user1,user2,...

A comma-separated list of users which are allowed **password-less** root access to the chroot. If empty or omitted, no users will be allowed root access without a password (but if a user or a group they belong to is in *users* or *groups*, respectively, they may gain access with a password). See the section "*Security*" below.

root-groups=group1,group2,...

A comma-separated list of groups which are allowed **password-less** root access to the chroot. If empty or omitted, no users will be allowed root access without a password (but if a user or a group they belong to is in *users* or *groups*, respectively, they may gain access with a password). See the section "*Security*" below.

aliases=alias1,alias2,...

A comma-separated list of aliases (alternate names) for this chroot. For example, a chroot named "sid" might have an 'unstable' alias for convenience.

run-setup-scripts=true|false

Set whether chroot setup scripts will be run. The default is not to run setup scripts ('false'), for safety reasons (so it won't clobber your passwd and other critical files). The default for session-managed chroots ('file' and 'lvm-snapshot') is to run setup scripts. If *type* is set to a value other

than ‘plain’ or ‘directory’, setup scripts are **required** to mount and configure the chroot environment. If enabled for a ‘plain’ or ‘directory’ chroot, the chroot will support simple session management (not true session management, because it does not make a copy of the chroot). If your chroots are exclusively controlled by schroot, set to ‘true’.

run-exec-scripts=true|false

Set whether chroot execution scripts will be run. The default is the same as the default for the *run-setup-scripts* key. This option was called *run-session-scripts* in versions prior to 0.2.5.

script-config=filename

The behaviour of the chroot setup and execution scripts may be customised on a per-chroot basis by providing a shell script which the scripts will source. The filename is relative to */usr/local/etc/schroot*. The default filename is ‘script-defaults’. The recommended method of customisation is to copy this script and amend the copy; or alter the original to set the defaults for all chroots. Settings for specific chroots may also be set in a single script by using conditionals checking the chroot name and/or type. Note that the setup script will be sourced once for each and every setup and exec script invocation, and must be idempotent.

command-prefix=command,option1,option2,...

A comma-separated list of a command and the options for the command. This command and its options will be prefixed to all commands run inside the chroot.

personality=persona

Set the personality (process execution domain) to use. This option is useful when using a 32-bit chroot on 64-bit system, for example. Valid options on Linux are ‘bsd’, ‘hpux’, ‘irix32’, ‘irix64’, ‘irixn32’, ‘iscr4’, ‘linux’, ‘linux32’, ‘linux_32bit’, ‘osf4’, ‘osr5’, ‘riscos’, ‘scorvr3’, ‘solaris’, ‘sunos’, ‘svr4’, ‘uw7’, ‘wysev386’, and ‘xenix’. The default value is ‘linux’. There is also the special option ‘undefined’ (personality not set). For a 32-bit chroot on a 64-bit system, ‘linux32’ is the option required. The only valid option for non-Linux systems is ‘undefined’. The default value for non-Linux systems is ‘undefined’.

environment-filter=refex

The environment to be set in the chroot will be filtered in order to remove environment variables which may pose a security risk. Any environment variable matching the specified POSIX extended regular expression will be removed prior to executing any command in the chroot.

Potentially dangerous, environment variables are removed for safety by default using the following regular expression: “^(BASH_ENV|CDPATH|ENV|HOSTALIASES|IFS|KRB5_CONFIG|KRBCONFDIR|KRBTKFILE|KRB_CONF|LD_*|LOCALDOMAIN|NLSPATH|PATH_LOCALE|RES_OPTIONS|TERMINFO|TERMINFO_DIRS|TERMPATH)\$”.

Plain and directory chroots

Chroots of type ‘plain’ or ‘directory’ are directories accessible in the filesystem. The two types are equivalent except for the fact that if *run-setup-scripts* is set to ‘true’ for ‘plain’ chroots, filesystem mounting is disabled.

These chroot types have an additional (mandatory) configuration option:

location=directory

The directory containing the chroot environment. This is where the root will be changed to when executing a login shell or a command. The directory must exist and have read and execute permissions to allow users access to it. Note that on Linux systems it will be bind-mounted elsewhere for use as a chroot; the directory for ‘plain’ chroots is mounted with the *--rbind* option to **mount(8)**, while for ‘directory’ chroots *--bind* is used instead so that sub-mounts are not preserved.

File chroots

Chroots of type ‘file’ are files on the current filesystem containing an archive of the chroot files. They implement the **source chroot** options (see “*Source chroot options*”, below) and have an additional (mandatory) configuration option:

file=filename

The file containing the archived chroot environment. This must be a tar (tape archive), optionally compressed with gzip or bzip2, or a zip archive. The file extensions used to determine the type are *.tar*, *.tar.gz*, *.tar.bz2*, *.tgz*, *.tbz* and *.zip*. This file must be owned by the root user, and not be writable by other.

Block device chroots

Chroots of type ‘block–device’ are a filesystem available on an unmounted block device. The device will be mounted and unmounted on demand. They have additional options:

device=device

This is the device name of the block device, including the absolute path. For example, “/dev/sda5”.

mount-options=mount_options Mount options for the block

device. These are additional options to pass to **mount(8)**. For example, “–o atime, sync, user_xattr”.

location=path

This is the path to the chroot *inside* the filesystem on the device. For example, if the filesystem contains a chroot in */chroot/sid*, you would specify “/chroot/sid” here. If the chroot is the only thing on the filesystem, i.e. / is the root filesystem for the chroot, this option should be left blank, or omitted entirely.

LVM snapshot chroots

Chroots of type ‘lvm–snapshot’ are a filesystem available on an LVM logical volume (LV). A snapshot LV will be created from this LV on demand, and then the snapshot will be mounted. At the end of the session, the snapshot LV will be unmounted and removed. For each chroot of this type, a corresponding ‘block–device’ chroot will be created, with a *–source* suffix appended to the chroot name and all its aliases; this is for convenient access to the source device.

They implement the **source chroot** options (see “*Source chroot options*”, below), and all the options for ‘block–device’, plus an additional option:

lvm-snapshot-options=snapshot_options

Snapshot options. These are additional options to pass to **lvcreate(8)**. For example, “–L 2g” to create a snapshot 2 GiB in size. **Note:** the LV name (*–n*), the snapshot option (*–s*) and the original LV path may not be specified here; they are set automatically by schroot.

Source chroot options

Some chroots implement source chroots. These are chroots which automatically create a copy of themselves before use, and are usually session managed. These chroots also provide an additional chroot with a *–source* suffix added to their name, to allow access to the original data, and to aid in chroot maintenance. These chroots provide the following additional options:

source-users=user1, user2, ...

A comma-separated list of users which are allowed access to the source chroot. If empty or omitted, no users will be allowed access. This will become the *users* option in the source chroot.

source-groups=group1, group2, ...

A comma-separated list of groups which are allowed access to the source chroot. If empty or omitted, no users will be allowed access. This will become the *groups* option in the source chroot.

source-root-users=user1, user2, ...

A comma-separated list of users which are allowed **password-less** root access to the source chroot. If empty or omitted, no users will be allowed root access without a password (but if a user is in *users*, they may gain access with a password). This will become the *root-users* option in the source chroot. See the section “*Security*” below.

source-root-groups=group1,group2,...

A comma-separated list of groups which are allowed **password-less** root access to the source chroot. If empty or omitted, no users will be allowed root access without a password (but if a user's group is in *groups*, they may gain access with a password). This will become the *root-groups* option in the source chroot. See the section “*Security*” below.

Localisation

Some keys may be localised in multiple languages. This is achieved by adding the locale name in square brackets after the key name. For example:

```
description[en_GB]=British English translation
```

This will localise the *description* key for the en_GB locale.

```
description[fr]=French translation
```

This will localise the *description* key for all French locales.

SECURITY

Note that giving untrusted users root access to chroots is a **serious security risk!** Although the untrusted user will only have root access to files inside the chroot, in practice there are many obvious ways of breaking out of the chroot and of disrupting services on the host system. As always, this boils down to *trust*.

Don't give chroot root access to users you would not trust with root access to the host system.

EXAMPLE

```
# Sample configuration

[sid]
type=plain
description=Debian unstable
description[fr_FR]=Debian instable
location=/srv/chroot/sid
priority=3
users=jim
groups=sbuild
root-users=rleigh
aliases=unstable,default

[etch]
type=block-device
description=Debian testing (32-bit)
priority=2
groups=users
#groups=sbuild-security
aliases=testing
device=/dev/hda_vg/etch_chroot
mount-options=-o atime
personality=linux32
run-setup-scripts=true
run-exec-scripts=true

[sid-file]
type=file
description=Debian sid file-based chroot
priority=3
groups=sbuild
file=/srv/chroots/sid.tar.gz
run-setup-scripts=true
run-exec-scripts=true
```

```
[sid-snapshot]
type=lvm-snapshot
description=Debian unstable LVM snapshot
priority=3
groups=sbuild
users=rleigh
source-root-users=rleigh
source-root-groups=admin
device=/dev/hda_vg/sid_chroot
mount-options=-o atime, sync, user_xattr
lvm-snapshot-options=--size 2G
run-setup-scripts=true
run-exec-scripts=true
```

FILES

/usr/local/etc/schroot/schroot.conf

The system-wide chroot definition file. This file must be owned by the root user, and not be writable by other.

AUTHORS

Roger Leigh.

COPYRIGHT

Copyright © 2005–2007 Roger Leigh <rleigh@debian.org>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

SEE ALSO

sbuid(1), **schroot(1)**, **mount(8)**.

NAME

schroot-setup — schroot chroot setup scripts

DESCRIPTION

schroot uses scripts to to set up and then clean up the chroot environment. The directories */usr/local/etc/schroot/setup.d* and */usr/local/etc/schroot/exec.d* contain scripts run when a chroot is created and destroyed, and entered and left, respectively. Several environment variables are set while the scripts are being run, which allows their behaviour to be customised, depending upon, for example, the type of chroot in use.

The scripts are run in name order, like those run by **init**(8), by using the same style of execution as **run-parts**(8).

The setup scripts are all invoked with two options:

- 1 The action to perform.

When a session is first started, the chroot is set up by running the scripts in */usr/local/etc/schroot/setup.d* with the ‘setup–start’ option. When a command or shell is run in the configured chroot, the scripts in */usr/local/etc/schroot/exec.d* are run with the ‘exec–start’ option. After the command or command has completed, the scripts are run in reverse order with the ‘exec–stop’ option. Finally, when the session is ended, the scripts in */usr/local/etc/schroot/setup.d* are run in reverse order with the ‘setup–stop’ option.

- 2 The chroot status.

This is either ‘ok’ if there are no problems, or ‘fail’ if something went wrong. For example, particular actions may be skipped on failure.

Note that the scripts should be *idempotent*. They **must** be idempotent during the ‘exec–stop’ and ‘setup–stop’ phases, because they may be run more than once.

ENVIRONMENT**General variables**

AUTH_USER

The username of the user the command in the chroot will run as.

AUTH_VERBOSITY

Set to ‘quiet’ if only error messages should be printed, ‘normal’ if other messages may be printed as well, and ‘verbose’ if all messages may be printed.

MOUNT_DIR

The directory under which non-filesystem chroots are mounted (e.g. block devices and LVM snapshots).

LIBEXEC_DIR

The directory under which helper programs are located.

PID The process ID of the schroot process.

SESSION_ID

The session identifier.

CHROOT_TYPE

The type of the chroot. This is useful for restricting a setup task to particular types of chroot (e.g. only block devices or LVM snapshots).

CHROOT_NAME

The name of the chroot. This is useful for restricting a setup task to a particular chroot, or set of chroots.

CHROOT_DESCRIPTION

The description of the chroot.

CHROOT_MOUNT_LOCATION

The location to mount the chroot. It is used for mount point creation and mounting.

CHROOT_LOCATION

The location of the chroot inside the mount point. This is to allow multiple chroots on a single filesystem.

CHROOT_PATH

The absolute path to the chroot. This is typically **CHROOT_MOUNT_LOCATION** and **CHROOT_LOCATION** concatenated together. This is the path which should be used to access the chroots.

CHROOT_MOUNT_DEVICE

The device containing the chroot root filesystem. This is only set for chroots which require mounting.

Plain and directory chroot variables

These chroot types use only general variables.

File variables**CHROOT_FILE**

The file containing the chroot files.

CHROOT_FILE_REPACK

Set to 'true' to repack the chroot into an archive file on ending a session, otherwise 'false'.

Block device variables**CHROOT_DEVICE**

The device containing the chroot root filesystem. This is usually, but not necessarily, the device which will be mounted. For example, an LVM snapshot this will be the original logical volume.

CHROOT_MOUNT_OPTIONS

Options to pass to **mount(8)**.

LVM snapshot variables**CHROOT_LVM_SNAPSHOT_NAME**

Snapshot name to pass to **lvcreate(8)**.

CHROOT_LVM_SNAPSHOT_DEVICE

The name of the LVM snapshot device.

CHROOT_LVM_SNAPSHOT_OPTIONS

Options to pass to **lvcreate(8)**.

FILES***/usr/local/etc/schroot/setup.d***

The system-wide chroot setup directory.

00check

Print debugging diagnostics and perform basic sanity checking.

05file

Unpack, clean up, and repack file-based chroots.

05lvm

Create and remove LVM snapshots.

10mount

Mount and unmount filesystems.

20network

Configure networking by copying *resolv.conf* from the host system.

30passwd

Configure *passwd*, *shadow* and *groups* in the chroot. This copies the files from the host system.

50chrootname

Set the chroot name (*/etc/debian_chroot*) in the chroot. This may be used by the shell prompt to display the current chroot.

50sbuid

If **sbuid**(1) is installed, set up the build directories, including ownership and permissions, inside the chroot for use by sbuid.

/usr/local/etc/schroot/exec.d

The system-wide chroot execution directory

00check

Print debugging diagnostics and perform basic sanity checking.

AUTHORS

Roger Leigh.

COPYRIGHT

Copyright © 2005–2007 Roger Leigh <rleigh@debian.org>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

SEE ALSO

sbuid(1), **schroot.conf**(5), **run-parts**(8).

NAME

dchroot — enter a chroot environment

SYNOPSIS

dchroot [~~-h~~|--help | ~~-V~~|--version | ~~-l~~|--list | ~~-i~~|--info | --config | --location] [~~--directory=directory~~] [~~-d~~|--preserve-environment] [~~-q~~|--quiet | ~~-v~~|--verbose] [~~-c chroot~~|--chroot=chroot | --all] [COMMAND [ARG1 [ARG2 [ARGn]]]]

DESCRIPTION

dchroot allows the user to run a command or a login shell in a chroot environment. If no command is specified, a login shell will be started in the user's home directory inside the chroot.

The command is one or more arguments which will be run in the user's default shell using its `-c` option. As a result, shell code may be embedded in this argument. If multiple command options are used, they are concatenated together, separated by spaces. Users should be aware of the shell quoting issues this presents, and should use **schroot** if necessary, which does not have any quoting issues.

The directory the command or login shell is run in depends upon the context. See `--directory` option below for a complete description.

This version of dchroot is a compatibility wrapper around the **schroot**(1) program. It is provided for backward compatibility with the dchroot command-line options, but schroot is recommended for future use. See the section "*Migration*" below for help migrating an existing dchroot configuration to schroot. See the section "*Incompatibilities*" below for known incompatibilities with older versions of dchroot.

If no chroot is specified, the chroot name or alias 'default' will be used as a fallback. If using the configuration in `/usr/local/etc/dchroot.conf`, the first chroot in the file is the default.

OPTIONS

dchroot accepts the following options:

Basic options

-h, --help

Show help summary.

-a, --all

Select all chroots.

-c, --chroot=chroot

Specify a chroot to use. This option may be used multiple times to specify more than one chroot, in which case its effect is similar to `--all`.

-l, --list

List all available chroots.

-i, --info

Print detailed information about the specified chroots. Note that earlier versions of dchroot did not include this option.

-p, --path

Print location (path) of the specified chroots.

--config

Print configuration of the specified chroots. This is useful for testing that the configuration in use is the same as the configuration file. Any comments in the original file will be missing. Note that earlier versions of dchroot did not include this option.

-d, --directory=directory

Change to *directory* inside the chroot before running the command or login shell. If *directory* is not available, dchroot will exit with an error status.

The default behaviour is as follows (all directory paths are inside the chroot). Unless the `--preserve-environment` option is used to preserve the environment, the login shell or command will run in the user's home directory, or `/` if the home directory is not available. When the

`--preserve-environment` option is used, it will attempt to use the current working directory, again falling back to `/` if it is not accessible. If none of the directories are available, `dchroot` will exit with an error status.

-d, --preserve-environment

Preserve the user's environment inside the `chroot` environment. The default is to use a clean environment; this option copies the entire user environment and sets it in the session.

-q, --quiet

Print only essential messages.

-v, --verbose

Print all messages. Note that earlier versions of `dchroot` did not include this option.

-V, --version

Print version information.

Note that earlier versions of `dchroot` did not provide long options.

CONFIGURATION

The `dchroot` configuration file, `/usr/local/etc/dchroot.conf`, used by earlier versions of `dchroot`, has the following format:

- `#` starts a comment line.
- Blank lines are ignored.
- `Chroot` definitions are a single line containing an *identifier*, *path*, and an optional *personality* separated by whitespace.
- The first `chroot` is also the default `chroot`.

An example file:

```
# Example comment

sarge /srv/chroot/sarge
sid /srv/chroot/sid linux32
```

This file defines a `chroot` called 'sarge', located at `/srv/chroot/sarge`, and a second `chroot` called 'sid', located at `/srv/chroot/sid`. The second `chroot` uses the 'linux32' personality, which allows a 32-bit `chroot` to be used on a 64-bit system. 'sarge' is the default `chroot`, because it was listed first, which means if the `-c` option is omitted this `chroot` will be used.

INCOMPATIBILITIES

Debian `dchroot` prior to version 0.99.0

- Log messages are worded and formatted differently.
- The parsing of `/usr/local/etc/dchroot.conf` uses a smaller list of allowed whitespace characters (space and tab), which may cause a parse error during tokenising if the file contains odd characters as separators, such as carriage returns, vertical tabs and form feeds.
- `su(1)` is no longer used to run commands in the `chroot`; this is done by `dchroot` internally. This change may cause subtle differences. If you find an incompatibility, please report it so it may be corrected.
- `dchroot` provides a restricted subset of the functionality implemented by `schroot`, but is still `schroot` underneath. Thus `dchroot` is still subject to `schroot` security checking, including PAM authentication and authorisation, and session management, for example, and hence may behave slightly differently to older `dchroot` versions in some circumstances.

DSA `dchroot`

Machines run by the Debian System Administrators for the Debian Project have a `dchroot-dsa` package which provides an alternate `dchroot` implementation.

- All the above incompatibilities apply.
- This version of dchroot has incompatible command-line options, and while some of those options are supported or have equivalent options by a different name, the `-c` option is not required to specify a chroot, and this version of dchroot cannot implement this behaviour in a backward-compatible manner (because if `-c` is omitted, the default chroot is used). DSA dchroot uses the first non-option as the chroot to use, only allowing one chroot to be used at once.
- This version of dchroot has an incompatible format for `dchroot.conf`. While the first two fields are the same, the remaining fields are an optional **users**, a list of users permitted to access the chroot, instead of the `personality` field allowed by this version. If access restrictions are needed, please use `/usr/local/etc/schroot/schroot.conf` and add the allowed users there, as shown in “Migration” below.

MIGRATION

To migrate an existing **dchroot** configuration to **schroot**, perform the following steps:

- 1 Dump the dchroot configuration in schroot keyfile format to `/usr/local/etc/schroot/schroot.conf`.

```
# dchroot --config >> /usr/local/etc/schroot/schroot.conf
```
- 2 Edit `/usr/local/etc/schroot/schroot.conf` to add access to the users and/or groups which are to be allowed to access the chroots, and make any other desired changes to the configuration. See **schroot.conf(5)**.
- 3 Remove `/usr/local/etc/dchroot.conf`, so that dchroot will subsequently use `/usr/local/etc/schroot/schroot.conf` for its configuration.

EXAMPLES

```
$ dchroot -l
Available chroots: sarge [default], sid

$ dchroot -p sid
/srv/chroot/sid

$ dchroot -q -c sid -- uname -smr
Linux 2.6.16.17 ppc
$ dchroot -q -c sid -- "uname -smr"
Linux 2.6.16.17 ppc

$ dchroot -q -c sid "ls -l / | tac | head -n 4"
var
usr
tmp
sys

$ dchroot -c sid
I: [sid chroot] Running login shell: "/bin/bash"
$
```

Use `--` to allow options beginning with `'-` or `'--'` in the command to run in the chroot. This prevents them being interpreted as options for dchroot itself. Note that the top line was echoed to standard error, and the remaining lines to standard output. This is intentional, so that program output from commands run in the chroot may be piped and redirected as required; the data will be the same as if the command was run directly on the host system.

TROUBLESHOOTING

If something is not working, and it's not clear from the error messages what is wrong, try using the `--debug=level` option to turn on debugging messages. This gives a great deal more information. Valid debug levels are `'none'`, and `'notice'`, `'info'`, `'warning'` and `'critical'` in order of increasing severity. The

lower the severity level, the more output.

If you are still having trouble, the developers may be contacted on the mailing list:

Debian build-tools Developers
<build-tools-devel@lists.alioth.debian.org>

BUGS

On the **mips** and **mipsel** architectures, Linux kernels up to and including at least version 2.6.17 have broken **personality**(2) support, which results in a failure to set the personality. This will be seen as an “Operation not permitted” (EPERM) error. To work around this problem, set *personality* to ‘undefined’, or upgrade to a more recent kernel.

FILES

/usr/local/etc/dchroot.conf

The system-wide **dchroot** chroot definition file. This file must be owned by the root user, and not be writable by other. If present, this file will be used in preference to */usr/local/etc/schroot/schroot.conf*.

/usr/local/etc/schroot/schroot.conf

The system-wide **schroot** definition file. This file must be owned by the root user, and not be writable by other. It is recommended that this file be used in preference to */usr/local/etc/dchroot.conf*, because the chroots can be used interchangeably with schroot, and the user and group security policies provided by schroot are also enforced.

AUTHORS

Roger Leigh.

This implementation of dchroot uses the same command-line options as the original **dchroot** by David Kimdon <dwhedon@debian.org>, but is an independent implementation.

COPYRIGHT

Copyright © 2005–2007 Roger Leigh <rleigh@debian.org>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

SEE ALSO

schroot(1), **sbuid**(1), **chroot**(2), **schroot-setup**(5), **schroot.conf**(5).

NAME

`dchroot-dsa` — enter a chroot environment

SYNOPSIS

`dchroot-dsa` [`-h`|`--help` | `-V`|`--version` | `-l`|`--list` | `-i`|`--info` | `--config` | `-p`|`--listpaths`] [`-d` *directory*|`--directory=directory`] [`-q`|`--quiet` | `-v`|`--verbose`] [`-c` *chroot*|`--chroot=chroot` | `--all` | **CHROOT**] [**COMMAND**]

DESCRIPTION

`dchroot-dsa` allows the user to run a command or a login shell in a chroot environment. If no command is specified, a login shell will be started in the user's home directory inside the chroot.

The user's environment will be preserved inside the chroot.

The command is a single argument which must be an absolute path to the program. Additional options are not permitted.

The directory the command or login shell is run in depends upon the context. See `--directory` option below for a complete description.

This version of `dchroot-dsa` is a compatibility wrapper around the `schroot(1)` program. It is provided for backward compatibility with the `dchroot-dsa` command-line options, but `schroot` is recommended for future use. See the section "*Migration*" below for help migrating your existing `dchroot-dsa` configuration to `schroot`. See the section "*Incompatibilities*" below for known incompatibilities with older versions of `dchroot-dsa`.

OPTIONS

`dchroot-dsa` accepts the following options:

Basic options

`-h, --help`

Show help summary.

`-a, --all`

Select all chroots. Note that earlier versions of `dchroot-dsa` did not include this option.

`-c, --chroot=chroot`

Specify a chroot to use. This option may be used multiple times to specify more than one chroot, in which case its effect is similar to `--all`. If this option is not used, the first non-option argument specified the chroot to use. Note that earlier versions of `dchroot-dsa` did not include this option.

`-l, --list`

List all available chroots.

`-i, --info`

Print detailed information about the available chroots. Note that earlier versions of `dchroot-dsa` did not include this option.

`-p, --listpaths`

Print absolute locations (paths) of the available chroots.

`--config`

Print configuration of the available chroots. This is useful for testing that the configuration in use is the same as the configuration file. Any comments in the original file will be missing. Note that earlier versions of `dchroot-dsa` did not include this option.

`-d, --directory=directory`

Change to *directory* inside the chroot before running the command or login shell. If *directory* is not available, `dchroot-dsa` will exit with an error status.

The default behaviour (all directory paths are inside the chroot) is to run the login shell or command in the user's home directory, or / if the home directory is not available. If none of the directories are available, `dchroot-dsa` will exit with an error status.

-q, --quiet

Print only essential messages. Note that earlier versions of `dchroot-dsa` did not include this option.

-v, --verbose

Print all messages. Note that earlier versions of `dchroot-dsa` did not include this option.

-V, --version

Print version information.

CONFIGURATION

The original **dchroot-dsa** configuration file, `/usr/local/etc/dchroot.conf`, used by earlier versions of `dchroot-dsa`, has the following format:

- `#` starts a comment line.
- Blank lines are ignored.
- Chroot definitions are a single line containing an **identifier**, **path**, and **users**, an optional user list separated by whitespace (space and tab), or a colon (`:`), semicolon (`;`), or comma (`,`).

An example file:

```
# Example comment

sarge /srv/chroot/sarge
sid /srv/chroot/sid rleigh,fred
```

This file defines a chroot called ‘sarge’, located at `/srv/chroot/sarge`, and a second chroot called ‘sid’, located at `/srv/chroot/sid`. The second chroot specifies that it may only be used by the users “rleigh” and “fred”.

INCOMPATIBILITIES**DSA dchroot**

- Log messages are worded and formatted differently.
- `dchroot-dsa` provides a restricted subset of the functionality implemented by **schroot**, but is still **schroot** underneath. Thus `dchroot-dsa` is still subject to **schroot** security checking, including PAM authentication and authorisation, and session management, for example, and hence may behave slightly differently to earlier versions of `dchroot-dsa` in some circumstances.

Debian dchroot

A **dchroot** package provides an alternative `dchroot` implementation.

- All the above incompatibilities apply.
- This version of `dchroot` has incompatible command-line options, and while some of those options are supported or have equivalent options by a different name, the `-c` option is required to specify a chroot. It also allows a shell script to be used as the option instead of a single absolute path, and allows multiple command options instead of a single option.
- This version of `dchroot` has an incompatible format for `dchroot.conf`. While the first two fields are the same, the third field is an optional **personality**, instead of the `users` user list permitted to access the chroot allowed by this version. If personality support is needed, please use `schroot.conf` and add the allowed users there, as shown in “*Migration*” below.

MIGRATION

To migrate an existing **dchroot-dsa** configuration to **schroot**, perform the following steps:

- 1 Dump the `dchroot-dsa` configuration in `schroot` keyfile format to `/usr/local/etc/schroot/schroot.conf`.

```
# dchroot-dsa --config >> /usr/local/etc/schroot/schroot.conf
```

- 2 Edit */usr/local/etc/schroot/schroot.conf* to add access to the users and/or groups which are to be allowed to access the chroots, and make any other desired changes to the configuration. See **schroot.conf**(5).
- 3 Remove */usr/local/etc/dchroot.conf*, so that `dchroot-dsa` will subsequently use */usr/local/etc/schroot/schroot.conf* for its configuration.

EXAMPLES

```
$ dchroot-dsa -l
Available chroots: sarge, sid

$ dchroot-dsa --listpaths
/srv/chroot/sarge
/srv/chroot/sid

$ dchroot-dsa -q sid -- /bin/uname
Linux

$ dchroot-dsa sid
I: [sid chroot] Running login shell: "/bin/bash"
$
```

Note that the top line was echoed to standard error, and the remaining lines to standard output. This is intentional, so that program output from commands run in the chroot may be piped and redirected as required; the data will be the same as if the command was run directly on the host system.

TROUBLESHOOTING

If something is not working, and it's not clear from the error messages what is wrong, try using the `--debug=level` option to turn on debugging messages. This gives a great deal more information. Valid debug levels are 'none', and 'notice', 'info', 'warning' and 'critical' in order of increasing severity. The lower the severity level, the more output.

If you are still having trouble, the developers may be contacted on the mailing list:
 Debian builddd-tools Developers
[<builddd-tools-devel@lists.alioth.debian.org>](mailto:builddd-tools-devel@lists.alioth.debian.org)

BUGS

None known at this time.

FILES

/usr/local/etc/dchroot.conf

The system-wide **dchroot-dsa** chroot definition file. This file must be owned by the root user, and not be writable by other. If present, this file will be used in preference to */usr/local/etc/schroot/schroot.conf*.

/usr/local/etc/schroot/schroot.conf

The system-wide **schroot** definition file. This file must be owned by the root user, and not be writable by other. It is recommended that this file be used in preference to */usr/local/etc/dchroot.conf*, because the chroots can be used interchangeably with **schroot**, and the user and group security policies provided by **schroot** are also enforced.

AUTHORS

Roger Leigh.

This implementation of `dchroot-dsa` uses the same command-line options as the **dchroot** found on machines run by the Debian System Administrators for the Debian Project. These machines have a **dchroot-dsa** source package which provides a **dchroot-dsa** package, written by Ben Collins [<bcollins@debian.org>](mailto:bcollins@debian.org) and Martin Schulze [<joey@debian.org>](mailto:joey@debian.org).

COPYRIGHT

Copyright © 2005–2007 Roger Leigh <rleigh@debian.org>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

SEE ALSO

schroot(1), **sbuid(1)**, **chroot(2)**, **schroot-setup(5)**, **schroot.conf(5)**.