Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'dracut.8' command

$ man dracut.8

DRACUT(8) dracut DRACUT(8)

NAME

dracut - low-level tool for generating an initramfs/initrd image

SYNOPSIS

dracut [OPTION...] [<image> [<kernel version>]]

DESCRIPTION

Create an initramfs <image> for the kernel with the version <kernel version>. If <kernel version> is omitted, then the version of the actual running kernel is used. If <image> is omitted or empty, depending on bootloader specification, the default location can be /efi/<machine-id>/<kernel-version>/initrd, /boot/<machine-id>/<kernel-version>/initrd, /boot/efi/<machine-id>/<kernel-version>/initrd, /lib/modules/<kernel-version>/initrd or /boot/initramfs-<kernel-version>.img.

dracut creates an initial image used by the kernel for preloading the block device modules (such as IDE, SCSI or RAID) which are needed to access the root filesystem, mounting the root filesystem and booting into the real system.

At boot time, the kernel unpacks that archive into RAM disk, mounts and uses it as initial root file system. All finding of the root device happens in this early userspace.

Initramfs images are also called "initrd".

For a complete list of kernel command line options see dracut.cmdline(7).

If you are dropped to an emergency shell, while booting your initramfs, the file /run/initramfs/rdsosreport.txt is created, which can be saved to a (to be mounted by hand) partition (usually /boot) or a USB stick.

Additional debugging info can be produced by adding rd.debug to the kernel command line. /run/initramfs/rdsosreport.txt contains all logs and the output of some tools. It should be attached to any report about dracut problems.

USAGE

To create a initramfs image, the most simple command is:

```
# dracut
```

This will generate a general purpose initramfs image, with all possible functionality resulting of the combination of the installed dracut modules and system tools. The image, depending on bootloader specification, can be /efi/<machine-id>/<kernel-version>/initrd, /boot/<machine-id>/<kernel-version>/initrd, /boot/efi/<machine-id>/<kernel-version>/initrd, /lib/modules/<kernel-version>/initrd or /boot/initramfs-<kernel-version>.img and contains the kernel modules of the currently active kernel with version <kernel-version>.

If the initramfs image already exists, dracut will display an error message, and to overwrite the existing image, you have to use the --force option.

```
# dracut --force
```

If you want to specify another filename for the resulting image you would issue a command like:

```
# dracut foobar.img
```

To generate an image for a specific kernel version, the command would be:

```
# dracut foobar.img 2.6.40-1.rc5.f20
```

A shortcut to generate the image at the default location for a specific kernel version is:

dracut --kver 2.6.40-1.rc5.f20

If you want to create lighter, smaller initramfs images, you may want to specify the --hostonly or -H option. Using this option, the resulting image will contain only those dracut modules, kernel modules and filesystems, which are needed to boot this specific machine. This has the drawback, that you can?t put the disk on another controller or machine, and that you can?t switch to another root filesystem, without recreating the initramfs image. The usage of the --hostonly option is only for experts and you will have to keep the broken pieces. At least keep a copy of a general purpose image (and corresponding kernel) as a fallback to rescue your system.

Inspecting the Contents

To see the contents of the image created by dracut, you can use the lsinitrd tool.

lsinitrd | less

To display the contents of a file in the initramfs also use the lsinitrd tool:

lsinitrd -f /etc/ld.so.conf

include ld.so.conf.d/*.conf

Adding dracut Modules

Some dracut modules are turned off by default and have to be activated manually. You can do this by adding the dracut modules to the configuration file /etc/dracut.conf or /etc/dracut.conf.d/myconf.conf. See dracut.conf(5). You can also add dracut modules on the command line by using the -a or --add option:

dracut --add module initramfs-module.img

To see a list of available dracut modules, use the --list-modules option:

dracut --list-modules

Omitting dracut Modules

Sometimes you don?t want a dracut module to be included for reasons of speed, size or functionality. To do this, either specify the omit_dracutmodules variable in the dracut.conf or

/etc/dracut.conf.d/myconf.conf configuration file (see dracut.conf(5)),

or use the -o or --omit option on the command line:

dracut -o "multipath lvm" no-multipath-lvm.img

Adding Kernel Modules

If you need a special kernel module in the initramfs, which is not automatically picked up by dracut, you have the use the --add-drivers option on the command line or the drivers variable in the /etc/dracut.conf or /etc/dracut.conf.d/myconf.conf configuration file (see dracut.conf(5)):

dracut --add-drivers mymod initramfs-with-mymod.img

Boot parameters

An initramfs generated without the "hostonly" mode, does not contain any system configuration files (except for some special exceptions), so the configuration has to be done on the kernel command line. With this flexibility, you can easily boot from a changed root partition, without the need to recompile the initramfs image. So, you could completely change your root partition (move it inside a md raid with encryption and LVM on top), as long as you specify the correct filesystem LABEL or UUID on the kernel command line for your root device, dracut will find it and boot from it.

The kernel command line can also be provided by the dhcp server with the root-path option. See the section called ?Network Boot?.

For a full reference of all kernel command line parameters, see dracut.cmdline(7).

To get a quick start for the suitable kernel command line on your system, use the --print-cmdline option:

dracut --print-cmdline

root=UUID=8b8b6f91-95c7-4da2-831b-171e12179081 rootflags=rw,relatime,discard,data=ordered rootfstype=ext4

Specifying the root Device

This is the only option dracut really needs to boot from your root partition. Because your root partition can live in various environments, there are a lot of formats for the root= option. The most basic one is root=<path to device node>:

root=/dev/sda2

Because device node names can change, dependent on the drive ordering, you are encouraged to use the filesystem identifier (UUID) or filesystem label (LABEL) to specify your root partition:

root=UUID=19e9dda3-5a38-484d-a9b0-fa6b067d0331

or

root=LABEL=myrootpartitionlabel

To see all UUIDs or LABELs on your system, do:

ls -l /dev/disk/by-uuid

or

ls -l /dev/disk/by-label

If your root partition is on the network see the section called ?Network Boot?.

Keyboard Settings

If you have to input passwords for encrypted disk volumes, you might want to set the keyboard layout and specify a display font.

A typical german kernel command line would contain:

rd.vconsole.font=eurlatgr rd.vconsole.keymap=de-latin1-nodeadkeys rd.locale.LANG=de_DE.UTF-8

Setting these options can override the setting stored on your system, if you use a modern init system, like systemd.

Blacklisting Kernel Modules

Sometimes it is required to prevent the automatic kernel module loading of a specific kernel module. To do this, just add rd.blacklist=<kernel module name>, with <kernel module name> not containing the .ko suffix, to the kernel command line. For example:

rd.driver.blacklist=mptsas rd.driver.blacklist=nouveau

The option can be specified multiple times on the kernel command line.

Speeding up the Boot Process

If you want to speed up the boot process, you can specify as much information for dracut on the kernel command as possible. For example, you can tell dracut, that you root partition is not on a LVM volume or not on a raid partition, or that it lives inside a

specific crypto LUKS encrypted volume. By default, dracut searches
everywhere. A typical dracut kernel command line for a plain
primary or logical partition would contain:

rd.luks=0 rd.lvm=0 rd.md=0 rd.dm=0

This turns off every automatic assembly of LVM, MD raids, DM raids
and crypto LUKS.

Of course, you could also omit the dracut modules in the initramfs
creation process, but then you would lose the possibility to turn
it on on demand.

Injecting custom Files

To add your own files to the initramfs image, you have several
possibilities.

The --include option let you specify a source path and a target path.

For example

```
# dracut --include cmdline-preset /etc/cmdline.d/mycmdline.conf initramfs-cmdline-pre.img
```

will create an initramfs image, where the file cmdline-preset will be
copied inside the initramfs to /etc/cmdline.d/mycmdline.conf. --include
can only be specified once.

```
# mkdir -p rd.live.overlay/etc/cmdline.d
# mkdir -p rd.live.overlay/etc/conf.d
# echo "ip=dhcp" >> rd.live.overlay/etc/cmdline.d/mycmdline.conf
# echo export FOO=testtest >> rd.live.overlay/etc/conf.d/testvar.conf
# echo export BAR=testtest >> rd.live.overlay/etc/conf.d/testvar.conf
# tree rd.live.overlay/
rd.live.overlay/
`-- etc
    |-- cmdline.d
    |   `-- mycmdline.conf
    `-- conf.d
        `-- testvar.conf
# dracut --include rd.live.overlay / initramfs-rd.live.overlay.img
```

This will put the contents of the rd.live.overlay directory into the
root of the initramfs image.

The --install option let you specify several files, which will get installed in the initramfs image at the same location, as they are present on initramfs creation time.

```
# dracut --install 'strace fsck.ext3 ssh' initramfs-dbg.img
```

This will create an initramfs with the strace, fsck.ext3 and ssh executables, together with the libraries needed to start those. The --install option can be specified multiple times.

Network Boot

If your root partition is on a network drive, you have to have the network dracut modules installed to create a network aware initramfs image.

If you specify ip=dhcp on the kernel command line, then dracut asks a dhcp server about the ip address for the machine. The dhcp server can also serve an additional root-path, which will set the root device for dracut. With this mechanism, you have static configuration on your client machine and a centralized boot configuration on your TFTP/DHCP server. If you can?t pass a kernel command line, then you can inject /etc/cmdline.d/mycmdline.conf, with a method described in the section called ?Injecting custom Files?.

Reducing the Image Size

To reduce the size of the initramfs, you should create it with by omitting all dracut modules, which you know, you don?t need to boot the machine.

You can also specify the exact dracut and kernel modules to produce a very tiny initramfs image.

For example for a NFS image, you would do:

```
# dracut -m "nfs network base" initramfs-nfs-only.img
```

Then you would boot from this image with your target machine and reduce the size once more by creating it on the target machine with the --host-only option:

```
# dracut -m "nfs network base" --host-only initramfs-nfs-host-only.img
```

This will reduce the size of the initramfs image significantly.

If the boot process does not succeed, you have several options to debug the situation. Some of the basic operations are covered here. For more information you should also visit:

https://www.kernel.org/pub/linux/utils/boot/dracut/dracut.html

Identifying your problem area

1. Remove 'rhgb' and 'quiet' from the kernel command line
2. Add 'rd.shell' to the kernel command line. This will present a shell should dracut be unable to locate your root device
3. Add 'rd.shell rd.debug log_buf_len=1M' to the kernel command line so that dracut shell commands are printed as they are executed
4. The file /run/initramfs/rdsosreport.txt is generated, which contains all the logs and the output of all significant tools, which are mentioned later.

If you want to save that output, simply mount /boot by hand or insert an USB stick and mount that. Then you can store the output for later inspection.

Information to include in your report

All bug reports

In all cases, the following should be mentioned and attached to your bug report:

- ? The exact kernel command-line used. Typically from the bootloader configuration file (e.g. /boot/grub2/grub.cfg) or from /proc/cmdline.
- ? A copy of your disk partition information from /etc/fstab, which might be obtained booting an old working initramfs or a rescue medium.
- ? Turn on dracut debugging (see the debugging dracut section), and attach the file /run/initramfs/rdsosreport.txt.
- ? If you use a dracut configuration file, please include /etc/dracut.conf and all files in /etc/dracut.conf.d/*.conf

Network root device related problems

This section details information to include when experiencing problems on a system whose root device is located on a network

attached volume (e.g. iSCSI, NFS or NBD). As well as the information from the section called ?All bug reports?, include the following information:

? Please include the output of

/sbin/ifup <interfacename>

ip addr show

Debugging dracut

Configure a serial console

Successfully debugging dracut will require some form of console logging during the system boot. This section documents configuring a serial console connection to record boot messages.

1. First, enable serial console output for both the kernel and the bootloader.
2. Open the file /boot/grub2/grub.cfg for editing. Below the line 'timeout=5', add the following:

```
serial --unit=0 --speed=9600
terminal --timeout=5 serial console
```

3. Also in /boot/grub2/grub.cfg, add the following boot arguments to the 'kernel' line:

```
console=tty0 console=ttyS0,9600
```

4. When finished, the /boot/grub2/grub.cfg file should look similar to the example below.

```
default=0
timeout=5
serial --unit=0 --speed=9600
terminal --timeout=5 serial console
title Fedora (2.6.29.5-191.fc11.x86_64)
  root (hd0,0)
                    kernel /vmlinuz-2.6.29.5-191.fc11.x86_64 ro root=/dev/mapper/vg_uc1-lv_root console=tty0
console=ttyS0,9600
  initrd /dracut-2.6.29.5-191.fc11.x86_64.img
```

5. More detailed information on how to configure the kernel for console output can be found at

http://www.faqs.org/docs/Linux-HOWTO/Remote-Serial-Console-HOWTO.html#CONFIGURE-KERNEL.

6. Redirecting non-interactive output

Note

You can redirect all non-interactive output to /dev/kmsg

and the kernel will put it out on the console when it

reaches the kernel buffer by doing

exec >/dev/kmsg 2>&1 </dev/console

Using the dracut shell

dracut offers a shell for interactive debugging in the event dracut

fails to locate your root filesystem. To enable the shell:

1. Add the boot parameter 'rd.shell' to your bootloader

configuration file (e.g. /boot/grub2/grub.cfg)

2. Remove the boot arguments 'rhgb' and 'quiet'

A sample /boot/grub2/grub.cfg bootloader configuration file is

listed below.

default=0

timeout=5

serial --unit=0 --speed=9600

terminal --timeout=5 serial console

title Fedora (2.6.29.5-191.fc11.x86_64)

root (hd0,0)

kernel /vmlinuz-2.6.29.5-191.fc11.x86_64 ro root=/dev/mapper/vg_uc1-lv_root console=tty0 rd.shell

initrd /dracut-2.6.29.5-191.fc11.x86_64.img

3. If system boot fails, you will be dropped into a shell as seen

in the example below.

No root device found

Dropping to debug shell.

#

4. Use this shell prompt to gather the information requested above

(see the section called ?All bug reports?).

Accessing the root volume from the dracut shell

From the dracut debug shell, you can manually perform the task of

locating and preparing your root volume for boot. The required

steps will depend on how your root volume is configured. Common scenarios include:

? A block device (e.g. /dev/sda7)

? A LVM logical volume (e.g. /dev/VolGroup00/LogVol00)

? An encrypted device (e.g. /dev/mapper/luks-4d5972ea-901c-4584-bd75-1da802417d83)

? A network attached device (e.g. netroot=iscsi:@192.168.0.4::3260::iqn.2009-02.org.example:for.all)

The exact method for locating and preparing will vary. However, to continue with a successful boot, the objective is to locate your root volume and create a symlink /dev/root which points to the file system. For example, the following example demonstrates accessing and booting a root volume that is an encrypted LVM Logical volume.

1. Inspect your partitions using parted

```
# parted /dev/sda -s p
Model: ATA HTS541060G9AT00 (scsi)
Disk /dev/sda: 60.0GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Number  Start   End     Size    Type      File system  Flags
1      32.3kB  10.8GB  107MB   primary   ext4         boot
2      10.8GB  55.6GB  44.7GB  logical               lvm
```

2. You recall that your root volume was a LVM logical volume. Scan and activate any logical volumes.

```
# lvm vgscan
# lvm vgchange -ay
```

3. You should see any logical volumes now using the command blkid:

```
# blkid
/dev/sda1: UUID="3de247f3-5de4-4a44-afc5-1fe179750cf7" TYPE="ext4"
/dev/sda2: UUID="Ek4dQw-cOtq-5MJu-OGRF-xz5k-O2l8-wdDj0l" TYPE="LVM2_member"
/dev/mapper/linux-root: UUID="def0269e-424b-4752-acf3-1077bf96ad2c" TYPE="crypto_LUKS"
/dev/mapper/linux-home: UUID="c69127c1-f153-4ea2-b58e-4cbfa9257c5e" TYPE="ext3"
/dev/mapper/linux-swap: UUID="47b4d329-975c-4c08-b218-f9c9bf3635f1" TYPE="swap"
```

4. From the output above, you recall that your root volume exists on an encrypted block device. Following the guidance disk encryption guidance from the Installation Guide, you unlock your encrypted root volume.

```
# UUID=$(cryptsetup luksUUID /dev/mapper/linux-root)
# cryptsetup luksOpen /dev/mapper/linux-root luks-$UUID
Enter passphrase for /dev/mapper/linux-root:
Key slot 0 unlocked.
```

5. Next, make a symbolic link to the unlocked root volume

```
# ln -s /dev/mapper/luks-$UUID /dev/root
```

6. With the root volume available, you may continue booting the system by exiting the dracut shell

```
# exit
```

Additional dracut boot parameters

For more debugging options, see dracut.cmdline(7).

Debugging dracut on shutdown

To debug the shutdown sequence on systemd systems, you can rd.break on pre-shutdown or shutdown.

To do this from an already booted system:

```
# mkdir -p /run/initramfs/etc/cmdline.d
# echo "rd.debug rd.break=pre-shutdown rd.break=shutdown" > /run/initramfs/etc/cmdline.d/debug.conf
# touch /run/initramfs/.need_shutdown
```

This will give you a dracut shell after the system pivot?ed back in the initramfs.

OPTIONS

--kver <kernel version>

Set the kernel version. This enables to specify the kernel version, without specifying the location of the initramfs image. For example:

```
# dracut --kver 3.5.0-0.rc7.git1.2.fc18.x86_64
```

-f, --force

Overwrite existing initramfs file.

<output file> --rebuild

Append the current arguments to those with which the input initramfs image was built. This option helps in incrementally building the initramfs for testing. If optional <output file> is not provided, the input initramfs provided to rebuild will be used as output file.

-a, --add <list of dracut modules>

Add a space-separated list of dracut modules to the default set of modules. This parameter can be specified multiple times.

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

```
# dracut --add "module1 module2"  ...
```

--force-add <list of dracut modules>

Force to add a space-separated list of dracut modules to the default set of modules, when -H is specified. This parameter can be specified multiple times.

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

```
# dracut --force-add "module1 module2"  ...
```

-o, --omit <list of dracut modules>

Omit a space-separated list of dracut modules. This parameter can be specified multiple times.

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

```
# dracut --omit "module1 module2"  ...
```

-m, --modules <list of dracut modules>

Specify a space-separated list of dracut modules to call when building the initramfs. Modules are located in /usr/lib/dracut/modules.d. This parameter can be specified multiple times. This option forces dracut to only include the specified dracut modules. In most cases the "--add" option is what you want

to use.

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

dracut --modules "module1 module2" ...

-d, --drivers <list of kernel modules>

Specify a space-separated list of kernel modules to exclusively include in the initramfs. The kernel modules have to be specified without the ".ko" suffix. This parameter can be specified multiple times.

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

dracut --drivers "kmodule1 kmodule2" ...

--add-drivers <list of kernel modules>

Specify a space-separated list of kernel modules to add to the initramfs. The kernel modules have to be specified without the ".ko" suffix. This parameter can be specified multiple times.

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

dracut --add-drivers "kmodule1 kmodule2" ...

--force-drivers <list of kernel modules>

See add-drivers above. But in this case it is ensured that the drivers are tried to be loaded early via modprobe.

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

dracut --force-drivers "kmodule1 kmodule2" ...

--omit-drivers <list of kernel modules>

Specify a space-separated list of kernel modules not to add to the initramfs. The kernel modules have to be specified without the ".ko" suffix. This parameter can be specified multiple times.

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

dracut --omit-drivers "kmodule1 kmodule2" ...

--filesystems <list of filesystems>

Specify a space-separated list of kernel filesystem modules to exclusively include in the generic initramfs. This parameter can be specified multiple times.

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

dracut --filesystems "filesystem1 filesystem2" ...

-k, --kmoddir <kernel directory>

Specify the directory, where to look for kernel modules.

--fwdir <dir>[:<dir>...]++

Specify additional directories, where to look for firmwares. This parameter can be specified multiple times.

--libdirs <list of directories>

Specify a space-separated list of directories to look for libraries to include in the generic initramfs. This parameter can be specified multiple times.

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

dracut --libdirs "dir1 dir2" ...

--kernel-cmdline <parameters>

Specify default kernel command line parameters.

--kernel-only

Only install kernel drivers and firmware files.

--no-kernel

Do not install kernel drivers and firmware files.

--early-microcode

Combine early microcode with ramdisk.

--no-early-microcode

Do not combine early microcode with ramdisk.

--print-cmdline

Print the kernel command line for the current disk layout.

--mdadmconf

Include local /etc/mdadm.conf file.

--nomdadmconf

Do not include local /etc/mdadm.conf file.

--lvmconf

Include local /etc/lvm/lvm.conf file.

--nolvmconf

Do not include local /etc/lvm/lvm.conf file.

--fscks <list of fsck tools>

Add a space-separated list of fsck tools, in addition to dracut.conf's specification; the installation is opportunistic (non-existing tools are ignored).

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

```
# dracut --fscks "fsck.foo barfsck"  ...
```

--nofscks

Inhibit installation of any fsck tools.

--strip

Strip binaries in the initramfs (default).

--aggresive-strip

Strip more than just debug symbol and sections, for a smaller initramfs build. The --strip option must also be specified.

--nostrip

Do not strip binaries in the initramfs.

--hardlink

Hardlink files in the initramfs (default).

--nohardlink

Do not hardlink files in the initramfs.

--prefix <dir>

Prefix initramfs files with the specified directory.

--noprefix

Do not prefix initramfs files (default).

-h, --help

Display help text and exit.

--debug

Output debug information of the build process.

-v, --verbose

Increase verbosity level (default is info(4)).

--version

Display version and exit.

-q, --quiet

Decrease verbosity level (default is info(4)).

-c, --conf <dracut configuration file>

Specify configuration file to use.

Default: /etc/dracut.conf

--confdir <configuration directory>

Specify configuration directory to use.

Default: /etc/dracut.conf.d

--tmpdir <temporary directory>

Specify temporary directory to use.

Default: /var/tmp

-r, --sysroot <sysroot directory>

Specify the sysroot directory to collect files from. This is useful

to create the initramfs image from a cross-compiled sysroot

directory. For the extra helper variables, see ENVIRONMENT below.

Default: empty

--sshkey <sshkey file>

SSH key file used with ssh-client module.

--logfile <logfile>

Logfile to use; overrides any setting from the configuration files.

Default: /var/log/dracut.log

-l, --local

Activates the local mode. dracut will use modules from the current working directory instead of the system-wide installed modules in /usr/lib/dracut/modules.d. This is useful when running dracut from a git checkout.

-H, --hostonly

Host-only mode: Install only what is needed for booting the local host instead of a generic host and generate host-specific configuration.

Warning

If chrooted to another root other than the real root device, use "--fstab" and provide a valid /etc/fstab.

-N, --no-hostonly

Disable host-only mode.

--hostonly-mode <mode>

Specify the host-only mode to use. <mode> could be one of "sloppy" or "strict". In "sloppy" host-only mode, extra drivers and modules will be installed, so minor hardware change won?t make the image unbootable (e.g. changed keyboard), and the image is still portable among similar hosts. With "strict" mode enabled, anything not necessary for booting the local host in its current state will not be included, and modules may do some extra job to save more space. Minor change of hardware or environment could make the image unbootable.

Default: sloppy

--hostonly-cmdline

Store kernel command line arguments needed in the initramfs.

--no-hostonly-cmdline

Do not store kernel command line arguments needed in the initramfs.

--no-hostonly-default-device

Do not generate implicit host devices like root, swap, fstab, etc. Use "--mount" or "--add-device" to explicitly add devices as needed.

--hostonly-i18n

Install only needed keyboard and font files according to the host configuration (default).

--no-hostonly-i18n

Install all keyboard and font files available.

--hostonly-nics <list of nics>

Only enable listed NICs in the initramfs. The list can be empty, so other modules can install only the necessary network drivers.

--persistent-policy <policy>

Use <policy> to address disks and partitions. <policy> can be any directory name found in /dev/disk. E.g. "by-uuid", "by-label"

--fstab

Use /etc/fstab instead of /proc/self/mountinfo.

--add-fstab <filename>

Add entries of <filename> to the initramfs /etc/fstab.

--mount "<device> <mountpoint> <filesystem type> [<filesystem options> [<dump frequency> [<fsck order>]]]"

Mount <device> on <mountpoint> with <filesystem type> in the initramfs. <filesystem options>, <dump options> and <fsck order> can be specified, see fstab manpage for the details. The default <filesystem options> is "defaults". The default <dump frequency> is "0". The default <fsck order> is "2".

--mount "<mountpoint>"

Like above, but <device>, <filesystem type> and <filesystem options> are determined by looking at the current mounts.

--add-device <device>

Bring up <device> in initramfs, <device> should be the device name. This can be useful in host-only mode for resume support when your swap is on LVM or an encrypted partition. [NB --device can be used for compatibility with earlier releases]

-i, --include <SOURCE> <TARGET>

Include the files in the SOURCE directory into the TARGET directory in the final initramfs. If SOURCE is a file, it will be installed

to TARGET in the final initramfs. This parameter can be specified multiple times.

-I, --install <file list>

Install the space separated list of files into the initramfs.

Note

If the list has multiple arguments, then you have to put these in quotes. For example:

dracut --install "/bin/foo /sbin/bar" ...

--install-optional <file list>

Install the space separated list of files into the initramfs, if they exist.

--gzip

Compress the generated initramfs using gzip. This will be done by default, unless another compression option or --no-compress is passed. Equivalent to "--compress=gzip -9".

--bzip2

Compress the generated initramfs using bzip2.

Warning

Make sure your kernel has bzip2 decompression support compiled in, otherwise you will not be able to boot. Equivalent to "--compress=bzip2 -9".

--lzma

Compress the generated initramfs using lzma.

Warning

Make sure your kernel has lzma decompression support compiled in, otherwise you will not be able to boot. Equivalent to "--compress=lzma -9 -T0".

--xz

Compress the generated initramfs using xz.

Warning

Make sure your kernel has xz decompression support compiled in, otherwise you will not be able to boot. Equivalent to "--compress=xz --check=crc32 --lzma2=dict=1MiB -T0".

--lzo

Compress the generated initramfs using lzop.

Warning

Make sure your kernel has lzo decompression support compiled in, otherwise you will not be able to boot. Equivalent to "--compress=lzop -9".

--lz4

Compress the generated initramfs using lz4.

Warning

Make sure your kernel has lz4 decompression support compiled in, otherwise you will not be able to boot. Equivalent to "--compress=lz4 -l -9".

--zstd

Compress the generated initramfs using Zstandard.

Warning

Make sure your kernel has zstd decompression support compiled in, otherwise you will not be able to boot. Equivalent to "--compress=zstd -15 -q -T0".

--compress <compressor>

Compress the generated initramfs using the passed compression program. If you pass it just the name of a compression program, it will call that program with known-working arguments. If you pass a quoted string with arguments, it will be called with exactly those arguments. Depending on what you pass, this may result in an initramfs that the kernel cannot decompress. The default value can also be set via the INITRD_COMPRESS environment variable.

--squash-compressor <compressor>

Compress the squashfs image using the passed compressor and compressor specific options for mksquashfs. You can refer to mksquashfs manual for supported compressors and compressor specific options. If squash module is not called when building the initramfs, this option will not take effect.

--no-compress

Do not compress the generated initramfs. This will override any other compression options.

--reproducible

Create reproducible images.

--no-reproducible

Do not create reproducible images.

--list-modules

List all available dracut modules.

-M, --show-modules

Print included module?s name to standard output during build.

--keep

Keep the initramfs temporary directory for debugging purposes.

--printsize

Print out the module install size.

--profile

Output profile information of the build process.

--ro-mnt

Mount / and /usr read-only by default.

-L, --stdlog <level>

[0-6] Specify logging level (to standard error).

0 - suppress any messages

1 - only fatal errors

2 - all errors

3 - warnings

4 - info

5 - debug info (here starts lots of output)

6 - trace info (and even more)

--regenerate-all

Regenerate all initramfs images at the default location with the kernel versions found on the system. Additional parameters are passed through.

-p, --parallel

Try to execute tasks in parallel. Currently only supported with

--regenerate-all (build initramfs images for all kernel versions simultaneously).

--noimageifnotneeded

Do not create an image in host-only mode, if no kernel driver is needed and no /etc/cmdline/*.conf will be generated into the initramfs.

--loginstall <directory>

Log all files installed from the host to <directory>.

--uefi

Instead of creating an initramfs image, dracut will create an UEFI executable, which can be executed by an UEFI BIOS. The default output filename is <EFI>/EFI/Linux/linux-$kernel$-<MACHINE_ID>-<BUILD_ID>.efi. <EFI> might be /efi, /boot or /boot/efi depending on where the ESP partition is mounted. The <BUILD_ID> is taken from BUILD_ID in /usr/lib/os-release or if it exists /etc/os-release and is left out, if BUILD_ID is non-existant or empty.

--no-uefi

Disables UEFI mode.

--no-machineid

Affects the default output filename of --uefi and will discard the <MACHINE_ID> part.

--uefi-stub <file>

Specifies the UEFI stub loader, which will load the attached kernel, initramfs and kernel command line and boots the kernel. The default is $prefix/lib/systemd/boot/efi/linux<EFI-MACHINE-TYPE-NAME>.efi.stub.

--uefi-splash-image <file>

Specifies the UEFI stub loader?s splash image. Requires bitmap (.bmp) image format.

--kernel-image <file>

Specifies the kernel image, which to include in the UEFI executable. The default is /lib/modules/<KERNEL-VERSION>/vmlinuz or

/boot/vmlinuz-<KERNEL-VERSION>.

--enhanced-cpio

Attempt to use the dracut-cpio binary, which optimizes archive creation for copy-on-write filesystems by using the copy_file_range(2) syscall via Rust?s io::copy(). When specified, initramfs archives are also padded to ensure optimal data alignment for extent sharing. To retain reflink data deduplication benefits, this should be used alongside the --no-compress and --no-strip parameters, with initramfs source files, --tmpdir staging area and destination all on the same copy-on-write capable filesystem.

ENVIRONMENT

INITRD_COMPRESS

sets the default compression program. See --compress.

DRACUT_LDCONFIG

sets the ldconfig program path and options. Optional. Used for --sysroot.

Default: ldconfig

DRACUT_LDD

sets the ldd program path and options. Optional. Used for --sysroot.

Default: ldd

DRACUT_TESTBIN

sets the initially tested binary for detecting library paths. Optional. Used for --sysroot. In the cross-compiled sysroot, the default value (/bin/sh) is unusable, as it is an absolute symlink and points outside the sysroot directory.

Default: /bin/sh

DRACUT_INSTALL

overrides path and options for executing dracut-install internally. Optional. Can be used to debug dracut-install while running the main dracut script.

Default: dracut-install

Example: DRACUT_INSTALL="valgrind dracut-install"

DRACUT_COMPRESS_BZIP2, DRACUT_COMPRESS_BZIP2, DRACUT_COMPRESS_LBZIP2,

DRACUT_COMPRESS_LZMA, DRACUT_COMPRESS_XZ, DRACUT_COMPRESS_GZIP,

DRACUT_COMPRESS_PIGZ, DRACUT_COMPRESS_LZOP, DRACUT_COMPRESS_ZSTD,

DRACUT_COMPRESS_LZ4, DRACUT_COMPRESS_CAT

overrides for compression utilities to support using them from

non-standard paths.

Default values are the default compression utility names to be

found in PATH.

DRACUT_ARCH

overrides the value of uname -m. Used for --sysroot.

Default: empty (the value of uname -m on the host system)

SYSTEMD_VERSION

overrides systemd version. Used for --sysroot.

SYSTEMCTL

overrides the systemctl binary. Used for --sysroot.

NM_VERSION

overrides the NetworkManager version. Used for --sysroot.

DRACUT_INSTALL_PATH

overrides PATH environment for dracut-install to look for binaries

relative to --sysroot. In a cross-compiled environment (e.g.

Yocto), PATH points to natively built binaries that are not in the

host?s /bin, /usr/bin, etc. dracut-install still needs plain /bin

and /usr/bin that are relative to the cross-compiled sysroot.

Default: PATH

DRACUT_INSTALL_LOG_TARGET

overrides DRACUT_LOG_TARGET for dracut-install. It allows running

dracut-install* to run with different log target that dracut** runs

with.

Default: DRACUT_LOG_TARGET

DRACUT_INSTALL_LOG_LEVEL

overrides DRACUT_LOG_LEVEL for dracut-install. It allows running

dracut-install* to run with different log level that dracut** runs

with.

Default: DRACUT_LOG_LEVEL

FILES

/var/log/dracut.log

logfile of initramfs image creation

/tmp/dracut.log

logfile of initramfs image creation, if /var/log/dracut.log is not writable

/etc/dracut.conf

see dracut.conf5

/etc/dracut.conf.d/*.conf

see dracut.conf5

/usr/lib/dracut/dracut.conf.d/*.conf

see dracut.conf5

Configuration in the initramfs

/etc/conf.d/

Any files found in /etc/conf.d/ will be sourced in the initramfs to set initial values. Command line options will override these values set in the configuration files.

/etc/cmdline

Can contain additional command line options. Deprecated, better use /etc/cmdline.d/*.conf.

/etc/cmdline.d/*.conf

Can contain additional command line options.

AVAILABILITY

The dracut command is part of the dracut package and is available from https://dracut.wiki.kernel.org

AUTHORS

Harald Hoyer

Victor Lowther

Amadeusz ?o?nowski

Hannes Reinecke

Daniel Molkentin

Will Woods

Philippe Seewer

Warren Togami

SEE ALSO

dracut.cmdline(7) dracut.conf(5) lsinitrd(1)