

Early creation of mapped devices

It is possible to configure a device-mapper device to act as the root device for your system in two ways.

The first is to build an initial ramdisk which boots to a minimal userspace which configures the device, then `pivot_root(8)` in to it.

The second is to create one or more device-mappers using the module parameter `"dm-mod.create="` through the kernel boot command line argument.

The format is specified as a string of data separated by commas and optionally semi-colons, where:

- a comma is used to separate fields like name, uuid, flags and table (specifies one device)
- a semi-colon is used to separate devices.

So the format will look like this:

```
dm-mod.create=<name>,<uuid>,<minor>,<flags>,<table>[,<table>+][; <name>,<uuid>,<minor>,<flags>,<table>[,<table>+]
```

Where:

```
<name>          ::= The device name.
<uuid>          ::= xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx | ""
<minor>         ::= The device minor number | ""
<flags>         ::= "ro" | "rw"
<table>         ::= <start_sector> <num_sectors> <target_type> <target_args>
<target_type>   ::= "verity" | "linear" | ... (see list below)
```

The dm line should be equivalent to the one used by the `dmsetup` tool with the `--concise` argument.

Target types

Not all target types are available as there are serious risks in allowing activation of certain DM targets without first using userspace tools to check the validity of associated metadata.

<i>cache</i>	constrained, userspace should verify cache device
<i>crypt</i>	allowed
<i>delay</i>	allowed
<i>era</i>	constrained, userspace should verify metadata device
<i>flakey</i>	constrained, meant for test
<i>linear</i>	allowed
<i>log-writes</i>	constrained, userspace should verify metadata device
<i>mirror</i>	constrained, userspace should verify main/mirror device
<i>raid</i>	constrained, userspace should verify metadata device
<i>snapshot</i>	constrained, userspace should verify src/dst device
<i>snapshot-origin</i>	allowed
<i>snapshot-merge</i>	constrained, userspace should verify src/dst device
<i>striped</i>	allowed
<i>switch</i>	constrained, userspace should verify dev path
<i>thin</i>	constrained, requires dm target message from userspace
<i>thin-pool</i>	constrained, requires dm target message from userspace
<i>verity</i>	allowed
<i>writecache</i>	constrained, userspace should verify cache device
<i>zero</i>	constrained, not meant for rootfs

If the target is not listed above, it is constrained by default (not tested).

Examples

An example of booting to a linear array made up of user-mode linux block devices:

```
dm-mod.create="lroot,,rw, 0 4096 linear 98:16 0, 4096 4096 linear 98:32 0" root=/dev/dm-0
```

This will boot to a rw dm-linear target of 8192 sectors split across two block devices identified by their major:minor numbers. After boot, udev will rename this target to `/dev/mapper/lroot` (depending on the rules). No uuid was assigned.

An example of multiple device-mappers, with the `dm-mod.create="..."` contents is shown here split on multiple lines for readability:

```
dm-linear,,1,rw,
 0 32768 linear 8:1 0,
 32768 102400 linear 8:2 0;
dm-verity,,3,ro,
 0 163840 verity 1 /dev/sdc1 /dev/sdc2 4096 4096 204800 1 sha256
 ac87db56303c9c1da433d7209b5a6ef3e4779df141200cbd7c157dcb8dd89c42
 5ebfe87f7df3235b80a117ebc4078e44f55045487ad4a96581d1adb564615b51
```

Other examples (per target):

"crypt":

[illegible]

"delay":

```
dm-delay,,4,ro,0 409600 delay /dev/sda1 0 500
```

"linear":

```
dm-linear,,,rw,
0 32768 linear /dev/sda1 0,
32768 1024000 linear /dev/sda2 0,
1056768 204800 linear /dev/sda3 0,
1261568 512000 linear /dev/sda4 0
```

"snapshot-origin":

```
dm-snap-orig,,4,ro,0 409600 snapshot-origin 8:2
```

"striped":

```
dm-striped,,4,ro,0 1638400 striped 4 4096
/dev/sda1 0 /dev/sda2 0 /dev/sda3 0 /dev/sda4 0
```

"verity":

```
dim-verity,,4,ro,  
0 1638400 verity 1 8:1 8:2 4096 4096 204800 1 sha256  
fb1a5a0f00deb908d8b53cb270858975e76cf64105d412ce764225d53b8f3cfd  
51934789604d1b92399c52e7cb149d1b3a1b74bbcb103b2a0aaacbed5c08584
```