systemd-id128

- Description
- Options
- Exit status
- Examples
- See Also

systemd

systemd-id128

1

systemd-id128

Generate and print sd-128 identifiers

systemd-id128

OPTIONS

new

systemd-id128

OPTIONS

machine-id

systemd-id128

OPTIONS

boot-id

systemd-id128

OPTIONS

invocation-id

systemd-id128

OPTIONS

var-partition-uuid

systemd-id128

OPTIONS

show

NAME|UUID

Description

`id128` may be used to conveniently print `sd-id128(3)` UUIDs. What identifier is printed depends on the specific verb.

With `new`, a new random identifier will be generated.

With `machine-id`, the identifier of the current machine will be printed. See `machine-id(5)`.

With `boot-id`, the identifier of the current boot will be printed.

With `invocation-id`, the identifier of the current service invocation will be printed. This is available in systemd services. See `systemd.exec(5)`.

With `show`, well-known IDs are printed (for now, only GPT partition type UUIDs), along with brief identifier strings. When no arguments are specified, all known IDs are shown. When arguments are specified, they may be the identifiers or ID values of one or more known IDs, which are then printed with their name, or arbitrary IDs, which are then printed with a placeholder name. Combine with `--uuid` to list the IDs in UUID style, i.e. the way GPT partition type UUIDs are usually shown.

`machine-id`, `boot-id`, and `show` may be combined with the `--app-specific=app-id` switch to generate application-specific IDs. See `sd_id128_get_machine(3)` for the discussion when this is useful. Support for `show --app-specific=` was added in version 255.

`var-partition-uuid` prints a UUID which, following the UAPI.2 Discoverable Partitions Specification, should be used as the GPT partition UUID for `/var/`, being derived from the GPT partition type, keyed by the local `/etc/machine-id`. Added in version 257.

Options

The following options are understood:

`-p`; `--pretty`
Generate output as programming language snippets.

`-P`; `--value`
Only print the value. May be combined with `-u`/`--uuid`.

`-a app-id`; `--app-specific=app-id`
With this option, identifiers will be printed that are the result of hashing the application identifier <app-id> and another ID. The <app-id> argument must be a valid sd-id128 string identifying the application. When used with `machine-id`, the other ID will be the machine ID as described in `machine-id(5)`, when used with `boot-id`, the other ID will be the boot ID, and when used with `show`, the other ID or IDs should be specified via the positional arguments.

`-u`; `--uuid`
Generate output as a UUID formatted in the "canonical representation", with five groups of digits separated by hyphens. See the Wikipedia entry for Universally Unique Identifiers for more discussion.

Exit status

On success 0 is returned, and a non-zero failure code otherwise.

Examples

Show a well-known UUID

```
$ systemd-id128 show --value user-home
773f91ef66d449b5bd83d683bf40ad16

$ systemd-id128 show --value --uuid user-home
773f91ef-66d4-49b5-bd83-d683bf40ad16

$ systemd-id128 show 773f91ef-66d4-49b5-bd83-d683bf40ad16
```

```
NAME      ID
user-home 773f91ef66d449b5bd83d683bf40ad16
```

Generate an application-specific UUID

```
$ systemd-id128 machine-id -u
3a9d668b-4db7-4939-8a4a-5e78a03bffb7

$ systemd-id128 new -u
1fb8f24b-02df-458d-9659-cc8ace68e28a

$ systemd-id128 machine-id -u -a 1fb8f24b-02df-458d-9659-cc8ace68e28a
47b82cb1-5339-43da-b2a6-1c350aef1bd1

$ systemd-id128 -Pu show 3a9d668b-4db7-4939-8a4a-5e78a03bffb7 \
    -a 1fb8f24b-02df-458d-9659-cc8ace68e28a
47b82cb1-5339-43da-b2a6-1c350aef1bd1
```

On a given machine with the ID 3a9d668b-4db7-4939-8a4a-5e78a03bffb7, for the application 1fb8f24b-02df-458d-9659-cc8ace68e28a, we generate an application-specific machine ID (47b82cb1-5339-43da-b2a6-1c350aef1bd1). If we want to later recreate the same calculation on a different machine, we need to specify both IDs explicitly as parameters to `show`.

See Also

`systemd(1)`, `sd-id128(3)`, `sd_id128_get_machine(3)`

□□□ #1
□ □□□ □□□ 2026-10-31 14:59:54 UTC
□ □□□ □□□ 2026-10-31 14:59:54 UTC