capsule@.service

- Description
- Examples
- See Also

systemd

capsule@.service

5

capsule@.service

System unit for the capsule service manager

`capsule@NAME.service`

Description

Service managers for capsules run in `capsule@NAME.service` system units, with the capsule name as the instance identifier. Capsules are way to run additional instances of the service manager, under dynamic user IDs, i.e. UIDs that are allocated when the capsule service manager is started, and released when it is stopped.

In many ways `capsule@.service` is similar to the per-user `user@.service` service manager, but there are a few important distinctions:

- The capsule service manager utilizes `DynamicUser=` (see `systemd.exec(5)`) to allocate a new UID dynamically on invocation. The user name is automatically generated from the capsule name, by prefixing `c-`. The UID is released when the service is terminated. The user service manager on the other hand operates under a statically allocated user ID that must be pre-existing, before the user service manager is invoked.
- User service managers register themselves with `pam(8)`, capsule service managers do not.

- User service managers typically read their configuration from a `$HOME` directory below `/home/`, capsule service managers from a `$HOME` directory below `/var/lib/capsules/`.
- User service managers are collectively contained in the `user.slice` unit, capsule service managers in `capsule.slice`. Also see `systemd.special(7)`.
- User service managers start the user unit `default.target` initially. Capsule service managers invoke the user unit `capsule@.target` instead.

The capsule service manager and the capsule's bus broker can be reached via the `--capsule=` switch to `systemctl(1)`, `systemd-run(1)` and `busctl(1)`.

New capsules can be started via a simple `systemctl start capsule@NAME.service` command, and stopped via `systemctl stop capsule@NAME.service`. Starting a capsule will implicitly create a home directory `/var/lib/capsules/NAME/`, if missing. A runtime directory is created as `/run/capsules/NAME/`. To remove these resources use `systemctl clean capsule@NAME.service`, for example with the `--what=all` switch.

The `capsule@.service` unit invokes a `systemd --user` service manager process. This means unit files are looked for according to the same rules as for regular user service managers, for example in `/var/lib/capsules/NAME/.config/systemd/user/`.

Capsule names may be chosen freely by the user, however, they must be suitable as UNIX filenames (i.e. 255 characters max, and contain no `/`), and when prefixed with `c-` be suitable as a user name matching strict POSIX rules, see User/Group Name Syntax for details.

Examples

Create a new capsule, invoke two programs in it (one interactively), terminate it, and clean everything up

```
# systemctl start capsule@tatze.service
# systemd-run --capsule=tatze --unit=sleeptest.service sleep 999
# systemctl --capsule=tatze status sleeptest.service
# systemd-run -t --capsule=tatze bash
# systemctl --capsule=tatze stop sleeptest.service
# systemctl stop capsule@tatze.service
# systemctl clean --all capsule@tatze.service
```

See Also

`systemd(1)`, `user@.service(5)`, `systemd.service(5)`, `systemd.slice(5)`, `systemd.exec(5)`, `systemd.special(7)`, `systemctl(1)`, `systemd-run(1)`, `busctl(1)`, `pam(8)`

□□□ #2
□ □□□ □□□ 2026-10-31 14:45:56 UTC
□ □□□ □□□ 2026-10-31 14:45:57 UTC