

dinit

dinit is a service supervisor with dependency support which can also act as the system "init" program. It was created with the intention of providing a portable init system with dependency management, that was functionally superior to many extant inits.

Contents

[Installation](#)

[Installation of services](#)

[Programs](#)

[Files/Directories](#)

[Usage](#)

[tty handling](#)

[Arch compatibility](#)

[Service file structure](#)

[User services](#)

[User Services needing "Display"](#)

[User Services needing "Display" Example](#)

Installation

Install the `dinit` package.

Installation of services

Dinit service packages are named `package_name-dinit` and, when installed, will be available in `/etc/dinit.d`.

Programs

- `dinit`
- `dinitcheck`
- `dinitctl`

Files/Directories

- `/etc/dinit.d/` - user-installed dinit service files
- `/usr/lib/dinit.d/` - system dinit service files
- `/etc/dinit.d/config/` - configurations for dinit services
- `/etc/dinit.d/boot.d/` - dinit services that run on boot
- `/usr/lib/dinit/` - dinit wrapper scripts
- `/var/log/dinit/` - logs for dinit services
- `/etc/dinit.d/service_name-pre` - preparation service, in some cases, this is needed because the "preparation" must be run as root but the service itself must be run as its own user
- `/etc/dinit.d/user` - user services (see [User services](#))

Usage

Nearly every interaction with dinit are done with the `dinitctl` program.

- Start service: `# dinitctl start service_name`
- Stop service: `# dinitctl stop service_name`
- Restart service: `# dinitctl restart service_name`
- Enable service: `# dinitctl enable service_name`
- Disable service: `# dinitctl disable service_name`
- List services: `# dinitctl list`

`dinitcheck` is usually used if to check if there is any problem with the dinit service directory, it will check for any syntax errors, invalid parameter values, and dependency cycles.

tty handling

dinit handles tty through `getty` service. You can configure active ttys through

`/etc/dinit.d/config/console.conf`

To modify individual ttys, you can copy (do **NOT** delete) `/etc/dinit.d/config/agetty-default.conf` to your desired tty (e.g. `/etc/dinit.d/config/agetty-tty1.conf`)

Inside, you'll see the contents like

```
# DO NOT REMOVE THIS FILE!  
# Note: You can copy and rename this file to the name of the tty you  
# want (e.g.: /etc/dinit.d/config/agetty-tty1.conf will make a  
# configuration specific to tty1)
```

```
GETTY_BAUD=38400
GETTY_TERM=linux
GETTY_ARGS=
```

You can modify the files according to [agetty\(1\)](#) , which allows one to autologin.

Arch compatibility

TBA

Service file structure

A typical service file looks like this:

```
type = process | bgprocess | scripted | internal
command = /path/to/servicename
restart = (boolean)
smooth-recovery = (boolean)
logfile = ...
depends-on = (service name)
waits-for = (service name)
```

There are four types of dinit services:

- [process](#) , for foreground daemons
- [bgprocess](#) , for forking daemons
- [scripted](#) , for oneshots
- [internal](#) , which is only useful inside dinit and is usually used to "gather" lots of dependencies into one big dependency, or similar to "bundle" in s6.

There are three types of dependencies:

- [depends-on](#) is hard dependency, so if any service is named here, it must run no matter what before the service is started
- [depends-ms](#) is a milestone dependency, so while the rules of depends-on apply when starting service, if the dependency stopped, the dependent will still run
- [waits-for](#) is soft dependency, while the service will wait for its dependency to run, it will still run if the dependency fails

For more details, see [dinit-service\(5\)](#) .

User services

Install the `turnstile` `turnstile-dinit` packages:

```
sudo pacman -S turnstile turnstile-dinit
```

Enable `turnstile`

```
dinitctl enable turnstiled
```

Relogin and check

```
$ dinitctl list
[[+] ] boot
[{} ] system
[{} ] dbus
```

User Services needing "Display"

Generally make sure

```
$.cat ~/.config/dinit.d/enviroment
DISPLAY=:0
```

Explicitly reference the enviroment file in the service, where `<USER>` is the user path

```
env-file = /home/<USER>/.config/dinit.d/enviroment
```

User Services needing "Display" Example

For example we create a service for installed `volumeicon` :

We make sure `~/.config/dinit.d/enviroment` contains `DISPLAY=:0`

Edit as root

```
/etc/dinit.d/user/volumeicon
```

to

```
type = process
command = /usr/bin/volumeicon
logfile = $HOME/.local/state/dinit/volumeicon.log
env-file = /home/<USER>/.config/dinit.d/enviroment
restart = true
```

Check User service

```
#: dinitcheck volumeicon
```

```
Checking service: volumeicon...
```

```
dinitcheck: Warning: Variable substitution performed by dinitcheck for file paths may not match dinit daemon (environment may differ); use --online to avoid this warning
```

```
Performing secondary checks...
```

```
Secondary checks complete.
```

```
No problems found.
```

Enable

```
#:dinitctl enable volumeicon
```

```
Service 'volumeicon' has been enabled.
```

```
Service 'volumeicon' started.
```

Check list

```
#:dinitctl list
```

```
[[{+} ] boot
```

```
[[{+} ] system
```

```
[[{+} ] dbus (pid: 875)
```

```
[[{+} ] pipewire (pid: 879)
```

```
[[{+} ] pipewire-pulse (pid: 881)
```

```
[[{+} ] wireplumber (pid: 880)
```

```
[[{+} ] volumeicon (pid: 16945)
```