- Artix Main Site
- Artix Forum
- Artix Wiki Home
- Page list
- Random page
- WikiSandbox

Basic tasks

- Migration
- Installation
- Configuration
- Troubleshooting

Advanced tasks

- Installation using buildiso

Artix components

- OpenRC
- Runit

External links

- OpenRC at Gentoo Wiki

PmWiki

- Initial Setup Tasks
- Basic Editing
- PmWiki FAQ

edit SideBar

# Installation {Main}

Login View Edit History Print

It is possible to use runit iso to install OpenRC-based system, and vice-versa.

Contents [hide]

## Fresh Artix installation from bootable media

Artix can either be installed through the console or the GUI installer. The GUI install is quite straightforward, we'll focus on the console installation procedure here. The installation media is confirmed to work on both BIOS and UEFI systems.

### Partition your disk (BIOS)

Partition your hard drive ( `/dev/sda` will be used in this guide) with **fdisk** or **cfdisk**, the partition numbers and order are at your discretion:

```
cfdisk /dev/sda
```

If you want to install side-by-side with other operating systems, you must make some space on the disk by resizing the existing partitions. You may use **gparted** for this, however detailed instructions are out of the scope of this guide. Next, format the new partitions, for example as ext4:

```
mkfs.ext4 -L ROOT /dev/sda2        <- root partition
mkfs.ext4 -L HOME /dev/sda3        <- home partition, optional
mkfs.ext4 -L BOOT /dev/sda4        <- boot partition, optional
mkswap -L SWAP /dev/sda1           <- swap partition
```

The -L switch assigns labels to the partitions, which helps referring to them later through `/dev/disk/by-label` without having to remember their numbers. Now, mount your partitions:

```
swapon /dev/sda1
mount /dev/sda2 /mnt
mount /dev/sda3 /mnt/home  (if created)
mount /dev/sda4 /mnt/boot  (if created)
```

### Install base system

A working internet connection is required and assumed. A wired connection is setup automatically, if found. Wireless ones must be configured by the user. Verify your connection and update the repositories:

```
ping artixlinux.org
pacman -Syy
```

Use **basestrap** to install the **base** and optionally the **base-devel** package groups and your preferred init (currently available: `openrc` and `runit` ):

```
basestrap /mnt base base-devel openrc
```

or

```
basestrap /mnt base base-devel runit
```

Use **fstabgen** to generate `/etc/fstab` , use -U for UUIDs and -L for partition labels:

```
fstabgen -L /mnt >>/mnt/etc/fstab
```

Check the resulting fstab for errors before rebooting. Now, you can chroot into your new Artix system with:

```
artools-chroot /mnt
```

### Configure the base system

First, install **grub** and **os-prober** (for detecting other installed operating systems):

```
pacman -S grub os-prober
grub-install --recheck /dev/sda
grub-mkconfig -o /boot/grub/grub.cfg
```

Create a user and password:

```
useradd -m user
passwd user
```

Set root password:

```
passwd
```

Install **networkmanager**:

```
pacman -S networkmanager networkmanager-openrc network-manager-applet
rc-update add NetworkManager default
```

Generate locales:

```
nano /etc/locale.gen  <- uncomment your locale
locale-gen
```

To set the locale systemwide, edit `/etc/locale.conf` (which is sourced by `/etc/profile` ) or `/etc/bash/bashrc.d/artix.bashrc` or

`/etc/bash/bashrc.d/local.bashrc`; user-specific changes may be made to their respective `~/.bashrc`, for example:

```
export LANG="en_US.UTF-8"
export LC_COLLATE="C"
```

### Post installation configuration

Now, you can reboot and enter into your new installation:

```
exit   <- exit chroot environment
umount -R /mnt
reboot
```

Once shutdown is complete, remove your installation media. If all went well, you should boot into your new system. Log in as your root to complete the post-installation configuration. To get a graphical environment you need to install the xorg group:

```
pacman -S xorg
```

Choose your packages, or just install all of them. For the closed-source nvidia drivers you can use the **nvidia-lts** package, as our default kernel is **linux-lts**:

```
pacman -S nvidia-lts
```

Older nvidia cards work with the legacy series, **nvidia-340xx-lts** and **nvidia-304xx-lts**. If you want to run a custom kernel, you can install the respective nvidia dkms package which ensures all installed kernels get their nvidia modules. AMD and Intel cards enjoy excellent (or near-excellent) 3D support with open-source drivers. Read the Arch wiki, for information on how Xorg chooses the best available video driver and which one is optimal for your hardware.

Install a desktop environment, for example **MATE**, **LXQt** or **XFCE4**:

```
pacman -S mate mate-extra xfce4 xfce4-goodies lxqt
```

And optionally a display manager, like **LXDM** or **SDDM**:

```
pacman -S lxdm displaymanager-openrc
rc-update add xdm default
nano /etc/conf.d/xdm   <- edit and set DISPLAYMANAGER="lxdm"
```

Or you can use .xinitrc to launch your DE manually; edit (or create) `~/.xinitrc` and add **exec mate-session**. Notice: **mate-session** and quite a few other packages from the Arch repositories are compiled against systemd even if they don't actually use it, at least not as PID1; to satisfy the library link you may install **elogind** and its services files **elogind-openrc** or **elogind-runit**.

```
pacman -S elogind
```