Remastering the Install ISO

Related articles

Archiso

Remastering the official Arch Linux install ISO image is not necessary for most applications. However, it may be desirable in some cases.

- Basic hardware is not supported by the core install. (rare)
- Installation on a non-internet capable machine.
- Deployment of Arch Linux on many similar machines, requiring the same installation procedure.

As these ISOs are bootable, they can also be used for system rescue, testing, project demos, and more.

Contents

Archiso

It is often preferable to rebuild the installation ISO with **Archiso**, instead of remastering an existing ISO.

Manually

How it works

The root filesystems of the live system is stored in `arch/x86_64/airootfs.sfs` in the ISO. The kernel and initramfs are in `arch/boot/x86_64` .

When booting, the initramfs will search for the device it was booted from via its label, `ARCH_201410` for example, and will mount the root filesystem for the architecture.

Extracting the ISO

To remaster the Arch Linux ISO, you will need a copy of the original ISO image. Download it from the **download page (https://www.archlinux.org/download/)**

Now, create a new directory to mount the ISO:

```
# mkdir /mnt/archiso
```

Mount the ISO to this directory (due to the nature of ISOs, the result is read-only):

```
# mount -t iso9660 -o loop /path/to/archISO /mnt/archiso
```

Copy the contents to another directory, where they can be edited:

```
$ cp -a /mnt/archiso ~/customiso
```

Note: Make sure `customiso` does not exist beforehand, otherwise this will create a subdirectory called `archiso` inside `customiso`

Customization

Modifying the x86_64 system

Change into the directory of the x86_64 system:

```
$ cd ~/customiso/arch/x86_64
```

Unsquash `airootfs.sfs` (to `squashfs-root`):

```
$ unsquashfs airootfs.sfs
```

Note: You need **`squashfs-tools (https://www.archlinux.org/packages/?name=squashfs-tools)`** in order to do that.

If you will need to run `mkinitcpio` within `arch-chroot`, you need to temporarily copy the kernel over:

```
$ cp ../boot/x86_64/vmlinuz squashfs-root/boot/vmlinuz-linux
```

Now you can modify the content of the system in `squashfs-root`. You can also chroot into this system to install packages etc.:

```
# arch-chroot squashfs-root /bin/bash
```

Note: `arch-chroot` is part of the package **`arch-install-scripts (https://www.archlinux.org/packages/?name=arch-install-scripts)`**

Note: If the `arch-chroot` script is not available in your system (e.g, when remastering arch-based distros), mount the api file systems and copy over your DNS details. See **Chroot#Using chroot**.

To be able to install package, you have to initialise the pacman keyring:

```
(chroot) # pacman-key --init
(chroot) # pacman-key --populate archlinux
```

Note: This step can take quite a while, be patient. (see **Pacman-Key**)

If the kernel or initrd is updated, additional steps are required. In this case you have to install archiso (https://www.archlinux.org/packages/?name=archiso) inside the chroot and change the content of `/etc/mkinitcpio.conf`:

```
(chroot) # pacman -Syu --force archiso linux
(chroot) # nano /etc/mkinitcpio.conf
```

Change the line that says `HOOKS="...` to:

```
HOOKS="base udev memdisk archiso_shutdown archiso archiso_loop_mnt archiso_pxe_common archiso_pxe_nbd archiso_pxe_http archiso_pxe_nfs archiso_kms block pcmcia filesystems keyboard"
```

Now update the initramfs:

```
(chroot) # mkinitcpio -p linux
```

When you are done, create a list of all installed packages, clean the pacman cache and exit the chroot:

```
(chroot) # LANG=C pacman -Sl | awk '/\[installed\]$/ {print $1 "/" $2 "-" $3}' > /pkglist.txt
(chroot) # pacman -Scc
(chroot) # exit
```

If you updated the kernel or the initramfs, move them over to the system, and remove the fallback initramfs (the install ISO doesn't use this):

```
$ mv squashfs-root/boot/vmlinuz-linux ~/customiso/arch/boot/x86_64/vmlinuz
$ mv squashfs-root/boot/initramfs-linux.img ~/customiso/arch/boot/x86_64/archiso.img
$ rm squashfs-root/boot/initramfs-linux-fallback.img
```

Move the list of packages:

```
$ mv squashfs-root/pkglist.txt ~/customiso/arch/pkglist.x86_64.txt
```

Now recreate `airootfs.sfs`:

```
$ rm airootfs.sfs
$ mksquashfs squashfs-root airootfs.sfs
```

Note: The monthly install ISO uses the `-comp xz` option on `mksquashfs` to significantly reduce size, but it also takes longer

Cleanup:

```
# rm -r squashfs-root
```

Now update the MD5 checksum of `airootfs.sfs`:

```
$ md5sum airootfs.sfs > airootfs.md5
```

Modifying the EFI boot image

If you have updated the kernel or the initramfs and wish to boot on EFI systems, update the EFI boot image. You will need **dosfstools (https://www.archlinux.org/packages/?name=dosfstools)** as the EFI boot image is a `FAT16` filesystem.

```
$ mkdir mnt
# mount -t vfat -o loop ~/customiso/EFI/archiso/efiboot.img mnt
# cp ~/customiso/arch/boot/x86_64/vmlinuz mnt/EFI/archiso/vmlinuz.efi
# cp ~/customiso/arch/boot/x86_64/archiso.img mnt/EFI/archiso/archiso.img
```

If you see `No space left on device` errors, you might need to resize `efiboot.img`. You can also create a new `efiboot.img` and copy the old files (replace `50` with the required size).

```
$ dd if=/dev/zero bs=1M count=50 of=efiboot-new.img
$ mkfs.fat -n "ARCHISO_EFI" efiboot-new.img
$ mkdir new
# mount -t fat -o loop efiboot-new.img new
$ cp -r mnt/* new/
# umount new mnt
$ mv efiboot-new.img ~/customiso/EFI/archiso/efiboot.img
```

And use the new `efiboot.img` as above.

Create a new ISO

Create a new ISO image with `genisoimage`, which is part of **cdrtools (https://www.archlinux.org/packages/?name=cdrtools)**, as a symlink to `mkisofs`.

```
$ genisoimage -l -r -J -V "ARCH_201209" -b isolinux/isolinux.bin -no-emul-boot -boot-load-size 4 -boot-info-table -c isolinux/boot.cat -o ../arch-custom.iso ./
```

Note: The ISO label must remain the same as the original label (in this case `ARCH_201209`) for the image to boot successfully.

Note: The `-b` and `-c` options expect paths relative to the root of the ISO

The resulting ISO image will boot only from CD, DVD or BD. For booting from USB stick or hard disk, it needs the **isohybrid (http://www.syslinux.org/wiki/index.php/Isohybrid)** feature. This can be achieved by postprocessing the ISO by program isohybrid included in **syslinux (https://www.archlinux.org/packages/?name=syslinux)**. Officially, the version of installed SYSLINUX has to be the same as the version of /isolinux/isolinux.bin in the ISO. It is not known whether really incompatible version combinations exist.

An alternative to genisoimage plus isohybrid can be derived from the xorriso run of mkarchiso.

```
$ iso_label="ARCH_201209"
$ xorriso -as mkisofs \
       -iso-level 3 \
       -full-iso9660-filenames \
       -volid "${iso_label}" \
       -eltorito-boot isolinux/isolinux.bin \
       -eltorito-catalog isolinux/boot.cat \
       -no-emul-boot -boot-load-size 4 -boot-info-table \
       -isohybrid-mbr ~/customiso/isolinux/isohdpfx.bin \
       -output arch-custom.iso \
       ~/customiso
```

Option -isohybrid-mbr needs an **MBR** template file. Most probably there is already such a file /isolinux/isohdpfx.bin in the original ISO, which matches the SYSLINUX version used in the ISO. Only if this file is missing in the copied ISO content, it has to be cut out of the original ISO image file, before above xorriso run is performed:

```
$ dd if=/path/to/archISO bs=512 count=1 of=~/customiso/isolinux/isohdpfx.bin
```

If the original ISO supports bootability via EFI, this can be activated in the new ISO by inserting the following options between the lines "-isohybrid-mbr ..." and "-output ...":

```
       -eltorito-alt-boot \
       -e EFI/archiso/efiboot.img \
       -no-emul-boot -isohybrid-gpt-basdat \
```

The file /EFI/archiso/efiboot.img is a FAT filesystem image file. If it is missing in the original ISO, then there was no EFI support in that ISO.

The newly created ISO image `arch-custom.iso` is found in the home directory. You can write the ISO image to a USB stick as explained in **USB Installation Media**. Alternatively you can burn the ISO image on a CD, DVD, or BD with your preferred software. On Arch, that is covered in the **article about burning an ISO image**.

See also

- **http://www.knoppix.net/wiki/KnoppixRemasteringHowto**
- **http://syslinux.zytor.com/iso.php**
- **http://busybox.net/**
- **Linux Live Kit (http://www.linux-live.org/)**

Retrieved from "https://wiki.archlinux.org/index.php?title=Remastering_the_Install_ISO&oldid=557386"

This page was last edited on 26 November 2018, at 13:57.