If you have any questions/corrections/contributions about this how-to, you're welcome on this OpenELEC forum topic

Table of contents:

Introduction

> Playing with iSCSI requires some basic knowledge on UNIX/Linux systems and networks. If you have never heard of iSCSI before, or don't really know what it is, chances are that you don't really want to use it.

Goals

- Having a diskless (noiseless) HTPC no drive, so less noise (even SSD can make the device louder, because it will make the whole device warmer, and the fans – if you have fans – will make more noise).
- On a Gigabit Ethernet network, iSCSI should be faster than any USB device.
- To host everything on your lovely network storage device (for your media, I do not recommend to store them on an iSCSI target, NFS or CIFS shares are far better options in order to share them between several clients).

- an iSCSI target it could be an end-user NAS device (Synology, Qnap, Drobo... offers iSCSI on every devices they sell, even the end-user ones, Netgear offers iSCSI on enterprise class models), a home-made server using FreeNAS, OpenFiler, or any GNU/Linux distribution.
- A little knowledge on how to configure your iSCSI target (creating LUNs, targets, etc.), as it will not be covered in this HowTo.
- A DHCP/TFTP server or an SD card (or an USB device), more on this later.

FAQ

- **Do I really need to do that?**
  Of course not! But you could, so you should, you know that!

- **Why not using a traditionnal PXE + NFS boot?**
  You can do that, yes, the goals are exactly the same. The difference is that NFS is file-oriented, iSCSI is block-oriented. Using iSCSI, you will have access to a standard block device (`/dev/sdX`), as if it was a local disk, which you could find easier (or not) to use than NFS. ISCSI can also you file extents, which are basically just a disk image. ISCSI should be a little faster than NFS, cause there is far less overhead (no lock controls, no FS code on the target side.)

- **Why not using PXE + NBD boot?**
  You can use that too, yes, the goals are still the same but NBD is not an adopted standard, it only exists on UNIX/Linux systems, and you probably won't find it on any commercial NAS/SAN devices...

Create the LUN/target on the iSCSI Target

This part is really dependent on your iSCSI target device, so it can't be covered in this HowTo

The LUN that you create will be used to store the OpenELEC System and Storage partitions, so size it as you want/need (4GB should be enough for starting, and there are chance that your iSCSI system allows you to resize your LUNs later – that's one good reason to use iSCSI, but you can give your system more or less space if you want). Information That You Will Need

- iSCSI target IP address (in my example `192.168.1.14`)
- The target Name (IQN) (in my example `iqn.2000-01.com.synology:pandora.openelec`)
- If you have enabled the CHAP authentication username and password

> Depending on your iSCSI system, you could have to add authorizations (or LUN masquerading) for your initiators (your building system and your HTPC).

Installing OpenELEC on the iSCSI LUN

Accessing the LUN

So, your LUN is created, now you have to access it from your building system. First you need to have iscsi support

```
aptitude install open-iscsi
```

Now you have to discover your iscsi portal

```
iscsiadm -m discovery -t st -p 192.168.1.14
```

That should list the available targets. In my example

```
192.168.1.14:3260,0 iqn.2000-01.com.synology:pandora.openelec
```

It should have created a config file in `/etc/iscsi/nodes/iqn.2000-01.com.synology\:pandora.openelec/192.168.1.14\,3260\,0/default`. If you have configured CHAP authentication (and reverse CHAP), you should edit this file (adapt the path to your needs) and add

```
node.session.auth.authmethod = CHAP
node.session.auth.username = username
node.session.auth.password = password
node.session.auth.username_in = username_in
node.session.auth.password_in = password_in
```

```
iscsiadm -m node -T iqn.2000-01.com.synology:pandora.openelec -p 192.168.1.14:3260 -l
```

If you have some errors like

```
Logging in to [iface: default, target: iqn.2000-01.com.synology:pandora.openelec, portal: 192.168.1.14,3260]
iscsiadm: Could not login to [iface: default, target: iqn.2000-01.com.synology:pandora.openelec, portal: 192.168.1.14,3260].
iscsiadm: initiator reported error (19 - encountered non-retryable iSCSI login failure)
```

Then, you should review your authentication configuration. If everything went fine, you should see some logs on the terminal

```
Logging in to [iface: default, target: iqn.2000-01.com.synology:pandora.openelec, portal: 192.168.1.14,3260]>
Login to [iface: default, target: iqn.2000-01.com.synology:pandora.openelec, portal: 192.168.1.14,3260] successful.
```

Cool, now you have a new block device on your system.

```
dmesg | tail -n 8
```

```
[ 9648.232779] scsi4 : iSCSI Initiator over TCP/IP
[ 9649.246417] scsi 4:0:0:0: Direct-Access SYNOLOGY iSCSI Storage    3.1  PQ: 0 ANSI: 5
[ 9649.247062] sd 4:0:0:0: [sdc] 16777216 512-byte logical blocks: (8.58 GB/8.00 GiB)
[ 9649.248907] sd 4:0:0:0: [sdc] Write Protect is off
[ 9649.248915] sd 4:0:0:0: [sdc] Mode Sense: 3b 00 00 00
[ 9649.249371] sd 4:0:0:0: [sdc] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
```

So for me it's `/dev/sdc` (there are already partitions on it, with a brand new LUN, that should of course not be the case). Remember what device it is for the next step (I will use `/dev/sdX` to refer to it, so people who does copy/paste will not destroy any existing data in the next steps).

Installing OpenELEC

Partitioning

First you have to create a partition table on your device. You can use your preferred tool for that (`fdisk`, `cfdisk`, `parted`, ...). My favourite tool is cfdisk

```
cfdisk /dev/sdX
```

- Create two partitions:
  - a 256MB (or more) one, that will hold the OpenELEC `KERNEL` and `SYSTEM`. Partition type should be Linux (83, which is the default). Make it bootable.
  - all the remaining space will be used as storage. Partition type should be Linux (83, which is the default)

Check that the partition table is fine

```
fdisk -l /dev/sdX
```

```
Disk /dev/sdX: 8589 MB, 8589934592 bytes
133 heads, 62 sectors/track, 2034 cylinders, total 16777216 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000eece8
```

```
/dev/sdX2          503006    16777215     8137105   83  Linux
```

Formatting

Now you have to create the two filesystems, System and Storage (the labels are important to have OpenELEC booting fine, if you want/need to change them, you'll have to adapt the `extlinux.conf` file later)

```
mkfs.ext4 -L System /dev/sdX1
mkfs.ext4 -L Storage /dev/sdX2
```

Installing Openelec

Mount the System partition on your system

```
mkdir -p /mnt/openelec/system
mount LABEL=System /mnt/openelec/system
```

Copy the OpenELEC KERNEL and SYSTEM files in it :

```
cp /path/to/SYSTEM /mnt/openelec/system
cp /path/to/KERNEL /mnt/openelec/system
```

First, create the `extlinux.conf` file.

```
nano /mnt/openelec/system/extlinux.conf
```

```
DEFAULT linux
PROMPT 0

LABEL linux
  KERNEL /KERNEL
  APPEND ip=dhcp boot=ISCSI='''ISCSI_OPTIONS''',LABEL=System disk=LABEL=Storage quiet
```

The `ISCSI_OPTIONS` will depend on how you will boot. You can edit it later. Basically, there will be two options :

- You will use an iBFT capable boot rom (IPXE), `ISCSI_OPTIONS` will be: `auto`
- You will not use an iBFT capable boot rom, `ISCSI_OPTIONS` will be something like (all on one line) : `iscsi_initiator=iqn.2010-04.org.whatever:mybox,iscsi_target_name=iqn.2000-01.com.synology,iscsi_target_ip=192.168.1.14,iscsi_target_port=3260,iscsi_target_group=pandora.openelec iscsi_username=username,iscsi_password=password,iscsi_in_username=username_in,iscsi_in_password=password_in`

If you are a sane person, you probably already know that you will use an iBFT capable ROM like IPXE

> You don't have to specify the `ISCSI_OPTIONS` for every device, even if you both have System and Storage on iSCSI. Just use it on the boot definition, as in the example Now install extlinux on the device

Now install extlinux on the device

Now unmount the device

```
umount /mnt/openelec/system
```

Now write the mbr (or gptmbr)

```
dd if=/usr/share/syslinux/mbr.bin of=/dev/sdx bs=440 count=1
```

And log out from the target

```
iscsiadm -m node -T iqn.2000-01.com.synology:pandora.openelec -p 192.168.1.14:3260 -u
```

> There are chances that your target will allow you to access to your device from several computers at the same time. **YOU SHOULD NEVER MOUNT IT FROM MORE THAN ONE COMPUTER AT A TIME**. Writing on a filesystem from more than one computer at the same time will corrupt your filesystem, you don't want to do that. **NEVER** !

Configuring the iSCSI Boot

Ok, so now you have your LUN and OpenELEC is installed on it, but how will you make your HTPC boot from it.

There is several answers to that question.

If you have the chance to have an HTPC which have an iSCSI compatible network adapter (most intel network adapters) it will be really easy just go into your network card BIOS/ROM, and configure the iSCSI target, and tell it to boot from it. You should be done!

You probably don't have such a network adapter (it's more an entreprise-class feature), but no problem, IPXE (http://ipxe.org) is there for you!

IPXE is an OpenSource Boot Firmware, it's a maintained fork of Etherboot/GPXE. It does PXE, is capable to access files from TFTP, FTP, HTTP servers, and more important it is also capable to boot from iSCSI (it also supports san boot on AoE and FCoE) !

Installing a DHCP and/or a TFTP server is out of the scope of this howto. You'll find good tutorials about that on the Net. There are some good docs on the IPXE homepage about that on http://ipxe.org/howto/dhcpd and http://ipxe.org/howto/msdhcp .

If you don't already have a DHCP server nor a TFTP server, I recommend you to look at dnsmasq, which is a lightweight DHCP/TFTP server.

So basically, you just have to make the ipxe ROM available on your TFTP server, and configure the DHCP server to make your box boot on it (next-server...). Then, configure a DHCP chainloader (http://ipxe.org/howto/chainloading) to boot a script like

```
echo Ready for iscsi boot !
set username username
set password password
set reverse-username username_in
set reverse-password password_in
:retry_sanboot
sanboot iscsi:192.168.1.14::::iqn.2000-01.com.synology:pandora.openelec || goto retry_sanboot
```

Of course you will have to adapt it to your needs (ip address, target name, usernames and passwords).

Install the ROM on a Removable Device

You don't want to (or you can't) install and configure a DHCP server and a TFTP server.

And you have that old SD card of 128MB that you earned with your first digital camera years ago, that is of no use now... great! It's a perfect candidate for becoming a booting ROM device! (of course, if your HTPC doesn't allow you to boot from SD card, you'll need to find something else, like an old USB key (the smaller, the better).

Create a new text file, that will be your ipxe embeded boot script

```
:retry_dhcp
dhcp || goto retry_dhcp
echo Ready for iscsi boot !
set username username
set password password
set reverse-username username_in
set reverse-password password_in
:retry_sanboot
sanboot iscsi:192.168.1.14::::iqn.2000-01.com.synology:pandora.openelec || goto retry_sanboot
```

Once again, adapt the values to your needs!

Building the ROM

- rom-o-matic

  You can use a service like rom-o-matic to automatically generate the IPXE rom for you. Select either (depending on your method)

  - USB Keychain disk image (.usb)
  - UNDI only (.kpxe)

  Paste your script into the text box and select Proceed.

- Build from Source

```
git clone git://git.ipxe.org/ipxe.git
cd ipxe/src
make bin/ipxe.usb EMBED=~/my_script.ipxe
```

```
make bin/undionly.kpxe EMBED=~/my_script.ipxe
```

- Writing the boot ROM

  Just put it on your SD card/USB device (plug it on your building system, look at dmesg to find on which /dev/sdX it is available).

  ```
  dd if=ipxe.usb of=/dev/sdX
  ```

  > Don't miss this one, you don't want to blow up your other disks!

  Just put the SD card/USB key in your HTPC box, configure it to boot from it, and enjoy!