

Operating principle

To create a live Slackware system, two Slackware systems are needed:

- the first one, with "**SysLinux**", "**Slackware-Live**", "**Unionfs-FUSE**" (if You don't plan to use AUFS) and "**SquashFS-Tools**" will be used to build the live system;
- the second one is the system to make live; this system must have a **kernel** (Slackware stock huge SMP kernel is recommended), **modules** and the "**MkInitRD**" utility; the utilities "**SysLinux**" and/or "**LiLo**" are needed if You want the live system able to be copied to an USB key and/or or installable on a disk partition (respectively); "**KTSUSS**" and "**Xdialog**" are needed for installation program GUI.

Once the system has been setup, the building of the live system needs three commands:

- "**build-slackware-live.sh --init ...**" to create the initrd and copy the kernel;
- "**build-slackware-live.sh --module ...**" to create a SquashFS module of the system;
- "**build-slackware-live.sh --iso ...**" to create an ISO image of the live-system or "**build-slackware-live.sh --usb ...**" to copy it on USB key.

Example

- Setup your system:
 - From a Slackware Linux system — the build system —, install wanted packages into the **"/mnt/system"** — the live system root — directory:

```
installpkg -root /mnt/system /mnt/cdrom/slackware/a/*.t?z
```
 - Add a user (if needed):

```
chroot /mnt/system #live system
useradd -m -g users -s /bin/bash liveuser
usermod -G floppy,cdrom,netdev,plugdev,scanner,lp,audio,video,power liveuser
passwd liveuser
exit #chroot
```
- Build the live system:

```
export SLTITLE="Custom distribution name and version"
build-slackware-live.sh --init /mnt/system /tmp/live
build-slackware-live.sh --module /mnt/system /tmp/live 0-slackware-live
build-slackware-live.sh --iso /tmp/live /tmp/slackware-live.iso
```

Remarks

- Build and live system don't have to be the same Slackware version.
- It is possible to have a single system that transforms itself into a live system: just use "/" for the system root directory parameter ("**/mnt/system**" in the example above).
Be careful: all directories but **"/sys"**, **"/proc"**, **"/dev"** and **"/tmp"** are included in the live system; think to umount removable devices for example.

System setup

Live system is just an ordinary Slackware

Install the live system like any other Slackware Linux (or derived distribution):

- make a regular install from the distribution install media, then, access this system by mounting its partition from the build system:

```
mkdir /mnt/system
```

```
mount /dev/installation_device /mnt/system
```

- or install packages into a subdirectory of the build system (the live system root directory) — with **"installpkg -root /mnt/system ..."** for example —.

Installing from a list of selected packages

The build system includes an option to install a list of selected packages into a subdirectory:

- usage: **"build-slackware-live.sh --add packages_dir system_root_dir pkg_list_file"**
- example: **"build-slackware-live.sh --add /mnt/cdrom /mnt/system packages-list.txt"**

packages-list.txt example

```
slackware/a/*
slackware/n/dhccpd
slackware/n/iputils
slackware/n/net-tools
slackware/n/network-scripts
postinstall=ln -sf ifconfig usr/bin/ifcfg
postinstall=echo "live.slackware.org" > etc/HOSTNAME
```

Divided system: setup GUI (fonts, icons ...)

The live system can consist of several modules ('core', 'gui', 'apps', 'drivers'... for example) build from multiple root directories; in this case it has to be unified for GUI setup to success; GUI setup consists on commands like **"mkfontdir"**, ..., **"update-pango-querymodules"** — this last one is done by stock Slackware startup scripts, but not by LiNomad, Salix or ZenWalk ones —.

- usage: **"build-slackware-live.sh --guiprep root_dir_1(rw) root_dir_2(ro) root_dir_3(ro)..."**
list needed directories to recompose a working system; only first branch is writable
- example: **"build-slackware-live.sh --guiprep /mnt/system-gui /mnt/system-core"**

Environment variables

Live system build depends on two environment variables:

- **"SLTITLE"**, the live-CD/DVD label, the boot menu title and GUI installation program title; by default, it is set to **"Slackware 13.1 Live"**;
- **"SLMODLIST"**, the module list needed for the live InitRD; by default, it is set to **"squashfs:fuse"** the two only modules needed if You use Slackware stock huge kernel; if not, here is an example with an almost exhaustive list of modules needed to boot a live system:

```
export SLMODLIST="squashfs:fuse:loop:isofs:nls_utf8:\
ehci-hcd:uhci-hcd:ohci-hcd:usb-storage"
```

Live system build commands

Setup kernel and initrd

- usage: **"build-slackware-live.sh --init system_root_dir live_dir [-linomad|-nosli]"**
*the **"-linomad"** option enables LiNomad startup scripts*
*the **"-nosli"** option prevents the use of Slackware-Live installer*
- example: **"build-slackware-live.sh --init /mnt/system /tmp/live"**

Create a SquashFS module for the system

- usage:
"build-slackware-live.sh --module system_root_dir live_dir module_name [-optional]"

the module name is also the SquashFS file name

with the **"-optional"** option, the module is stored in the optional directory

- example: **"build-slackware-live.sh --module /mnt/system /tmp/live 0-slackware-live"**
- remark: you can put your own modules inside **"live_system_dir/boot/modules/"** or **"live_system_dir/boot/optional/"**, or move them between 'modules' and 'optional'

Copy live system on USB device

- usage: **"build-slackware-live.sh --usb live_dir device [-linomad]"**
the **"-linomad"** option enables the home directory stored on USB key and GUI auto-detection for LiNomad startup scripts
- example - after initialization and module creation:
"build-slackware-live.sh --usb /tmp/live /dev/sdx1"
- example - from a running live system:
"build-slackware-live.sh --usb /live/livemedia /dev/sdx1"

Create a live CD/DVD ISO from live system

- usage: **"build-slackware-live.sh --iso live_dir iso_file_name [-linomad]"**
the **"-linomad"** option enables GUI auto-detection for LiNomad startup scripts
- example - after initialization and module creation:
"build-slackware-live.sh --iso /tmp/live /tmp/slackware-live.iso"

Boot parameters

System language and keymap layout

- **"locale"**: system language; example: **"locale=fr_FR.UTF-8"**
- **"keymap"**: keymap layout; example: **"keymap=fr"**; the two first characters are used for Xorg keymap layout

modules loading

- **"include=module1:module2:..."**: to load selected modules from **"/boot/optional"** directory (module names are the SquashFS file names)
- **"exclude=module1:module2:..."**: to specify the main modules (from **"/boot/modules"** directory) not to load; example: **"exclude=1-gui"**

Misc

- **"runlevel=[1-5]"**: to override default startup runlevel (cf **"/etc/inittab"**)
- **"copy2ram=yes"**: to ask copy of live system content to memory (allow its removal)

LiNomad

- **"usbhome=yes|no"**: in case of live-USB system, activates the home directory storage on USB key (if not, it is stored in RAM)
- **"gui=auto|yes|no"**: activates or not the GUI, with auto-detection (allows the activation of the NVidia drivers) or not

Live system installation

The live system can be installed into a hard disk partition.

Install live system

- usage: **"build-slackware-live.sh --install system_root_dir partition_device [-auto|-expert]"**

*the **"-auto"** option enables LiLo installation into the MBR
with the **"-expert"** option LiLo is not installed - see expert setup*

- example - from a running live system (typically):
"build-slackware-live.sh --install /live/system /dev/sdx2 -auto"
- example - system cloning:
"build-slackware-live.sh --install /mnt/system /dev/sdx2"

LiLo expert setup

- usage: **"build-slackware-live.sh --loadersetup partition_device"**
- example - after a system install:
"build-slackware-live.sh --loadersetup /dev/sdx2"