

Microlinux Enterprise Desktop 14.1 Installation Guide

The following document provides step-by-step instructions to install and configure the Microlinux Enterprise Desktop 14.1. It assumes you already know how to install and configure Slackware Linux. Read everything carefully. The bits you skip will come back and bite you.



If you are not familiar with Slackware, head over to the *Slackware Documentation Project* at <http://docs.slackware.com> and read the following two articles in the section **GETTING**

STARTED WITH SLACKWARE :

- **SLACKWARE INSTALLATION**
- **CONFIGURE YOUR NEW SLACKWARE SYSTEM**

MLED installs on top of a carefully selected Slackware base and doesn't ship on its own installation ISO. All you need is a vanilla Slackware installation medium (DVD, CD-Rom, USB stick) and a working internet connection.

Install the base Slackware Linux system

Boot the Slackware installation medium : DVD, first CD or USB disk. Select your keyboard layout, login as root, partition your hard disk and start the Slackware installer.

```
root@slackware:/# setup
```

- **PACKAGE SERIES SELECTION** : unselect the **E**, **KDE** and **KDEI** package groups ;
- **SELECT PROMPTING MODE** : FULL OR TERSE ;
- **USE UTF-8 TEXT CONSOLE** : **Yes** ;
- **CONFIRM STARTUP SERVICES TO RUN** : accept the default selection ;
- **SELECT DEFAULT WINDOW MANAGER FOR X** : **XINITRC.WMAKER** OR **XINITRC.FLUXBOX**.
This choice is only temporary. I usually choose one of the more lightweight window managers like WindowMaker or Fluxbox to configure X.

Finish the base Slackware installation, exit the installer and reboot. Don't define a user yet. We'll wait until the MLED user profiles are installed.

Download the Microlinux scripts

Microlinux provides a few helper scripts to speed up the installation process. Download the file tree to your /root directory using the following command :

```
# cd  
# git clone https://github.com/kikinovak/microlinux
```

Configure slackpkg

In its default configuration, the `slackpkg` package manager only works with the official Slackware repositories. We need to install and configure Matteo Rossini's nifty `slackpkg+` plugin to enable third-party repositories. To avoid the hassle, Microlinux provides a custom `slackpkg+` package that comes preconfigured for the Microlinux repositories.

On the command line, use the Links browser to navigate through the repositories :

```
# links http://www.microlinux.fr/microlinux/
```

On a 32-bit system, navigate to the `desktop-14.1-32bit/slackware/ap/` subdirectory. If your system is 64-bit, go to `desktop-14.1-64bit/slackware64/ap/`. Grab the `slackpkg+` package from the repository (use the **[D]** key in the Links browser to download it) and install it :

```
# installpkg slackpkg+-1.3.2-noarch-5_microlinux.txz
```

Edit `/etc/slackpkg/mirrors` and choose a Slackware mirror according to your geographical location, for example :

```
# /etc/slackpkg/mirrors
...
# GERMANY (DE)
ftp://ftp.fu-berlin.de/unix/linux/slackware/slackware-14.1/
...
```

On a 64-bit system :

```
# /etc/slackpkg/mirrors
...
# GERMANY (DE)
ftp://ftp.fu-berlin.de/unix/linux/slackware/slackware64-14.1/
...
```

Make sure you choose only one single mirror for Slackware stable.¹

Update GPG keys :

```
# slackpkg update gpg
```

Update information about available packages :

```
# slackpkg update
```

¹ If you're using MLED in France, configure the ftp.fu-berlin.de or the mirror.switch.ch mirror. The admins at OVH can't seem to get their act together.

Trim the base installation

The `desktop-14.1-32bit/tools/` and `desktop-14.1-64bit/tools` subdirectories each provide a basic `trim.sh` script that takes care of two things :

- install needed base packages ;
- get rid of unneeded base packages.

Trim your Slackware installation :

```
# cd microlinux/desktop-14.1-32bit/tools/  
# ./trim.sh
```

Or :

```
# cd microlinux/desktop-14.1-64bit/tools/  
# ./trim.sh
```

On a side note, the `trim.sh` scripts and respective package selections in the 32bit and 64bit subdirectories are symlinked and thus identical.

Two packages are provided by the official `extra/` repository. Install them manually :

```
# slackpkg install mplayerplug-in recordmydesktop
```

A word on what's left out

MLED doesn't rely on a full Slackware installation. Here's what's not installed :

- everything related to KDE, since MLED is based on Xfce ;
- various exotic fonts, which clutter LibreOffice's font preview ;
- some X11 applications like Seamonkey, Blueman, GnuChess, etc.

Of course, you're free to override this selection by manually installing any package (exotic font, Seamonkey, Blueman, etc.) that's been left out with a simple `slackpkg install <package>`.

Upgrade the base installation

Now upgrade the base Slackware packages :

```
# slackpkg upgrade-all
```

You'll notice that some Slackware packages like the Xfce desktop, Audacious, MPlayer or HPLIP will be « upgraded », e. g. replaced by an MLED package. Don't worry, this is normal. MLED 14.1 replaces Xfce 4.10 by the latest Xfce 4.12, and the audio and video players have been rebuilt for enhanced multimedia capabilities. As for HPLIP, it has been upgraded to support recent HP printers.

Install the MLED package collection

Installing the full set of MLED packages is as simple as this :

```
# slackpkg install microlinux-desktop
```

Now we've completed the initial installation, there's still a few things left to do before we can start using our new system. The following explanations provide some quick help for the most crucial configuration steps. Again, you might want to check out the *Slackware Documentation Project* at <http://docs.slackware.com> for more in-depth information.

Set locales

You'll probably have to adjust your environment variables in `/etc/profile.d/lang.sh`. Default variables are set to `fr_FR.UTF8`, since MLED's main use is in France :

```
# /etc/profile.d/lang.sh
...
export LANG=fr_FR.utf8
export LC_COLLATE=fr_FR.utf8
```

English-speaking Slackware users will use something like this :

```
# /etc/profile.d/lang.sh
...
export LANG=en_US.utf8
export LC_COLLATE=en_US.utf8
```

Clean up the applications menu

The `desktop-14.1-32bit/tools/` and `desktop-14.1-64bit/tools/` subdirectories each feature the `cleanmenu.sh` utility, a small Bash script to clean up various desktop menu entries and make them Joe-Sixpack-friendly. Run this utility :

```
# cd microlinux/desktop-14.1-32bit/tools/
# ./cleanmenu.sh
```

Or :

```
# cd microlinux/desktop-14.1-64bit/tools/
# ./cleanmenu.sh
```

Two remarks :

- As above, the `cleanmenu.sh` scripts in the `32bit` and `64bit` subdirectories are symlinked and thus identical.
- The `cleanmenu.sh` script replaces many `*.desktop` files in `/usr/share/applications/` and similar locations by custom-made menu entry files. They're only localized in english, french and german, so you may not want to run the script if you use another language.

Switch to the GENERIC kernel

The HUGE kernel installed by the Slackware installer is essentially a kernel which has every hardware driver built in which we might have needed for a successful installation of our computer. It is recommended that we switch to the GENERIC kernel. This kernel has virtually no drivers built in and requires an initial ramdisk.

The first step will be to identify which kernel modules are needed at boot time. Our Slackware system is shipping a nifty little script to do just that :

```
# /usr/share/mkinitrd/mkinitrd_command_generator.sh
```

The result will vary depending on your hardware. Here's what I get on a Dell Optiplex 330 :

```
mkinitrd -c -k 3.10.17-smp -f ext4 -r /dev/sda3 -m mbcache:jbd2:ext4
-u -o /boot/initrd.gz
```

The highlighted interesting bit here is the `-m mbcache:jbd2:ext4` option. It means that my GENERIC kernel will require the `mbcache`, `jbd2` and `ext4` modules at boot time.

We can now provide this information to create and edit `/etc/mkinitrd.conf` :

```
# cd /etc
# cp mkinitrd.conf.sample mkinitrd.conf
```

Uncomment all options in the file and edit it according to your configuration. Here's an example, with the relevant bits highlighted :

```
# /etc/mkinitrd.conf
SOURCE_TREE="/boot/initrd-tree"
CLEAR_TREE="1"
OUTPUT_IMAGE="/boot/initrd.gz"
KERNEL_VERSION="$(uname -r)"
KEYMAP="fr-latin1"
MODULE_LIST="mbcache:jbd2:ext4"
ROOTDEV="/dev/sda3"
ROOTFS="ext4"
RESUMEDEV="/dev/sda2"
RAID="0"
LVM="0"
UDEV="1"
MODCONF="0"
WAIT="1"
```

The relevant options in a nutshell :

- `CLEAR_TREE` defaults to `0` and goes to `1` ;
- `KEYMAP` defines your console keyboard layout ;
- `MODULE_LIST` contains the list of all kernel modules inserted into the `initrd`, separated by `:` ;
- `ROOTDEV` is the root partition ;
- `ROOTFS` is the file system used on the root partition ;

-
- RESUMEDEV is the swap partition ;
 - You can safely removed all the LUKS* lines if you don't encrypt your partitions.

Now all our options are defined, we can build the initrd with this single command :

```
# mkinitrd -F
```

The last step will be to create a new stanza in our bootloader's configuration, pointing to the GENERIC kernel and the initrd. Here's an example :

```
# /etc/lilo.conf
...
image = /boot/vmlinuz-generic-smp-3.10.17-smp
  initrd = /boot/initrd.gz
  root = /dev/sda3
  label = Generic
  read-only
image = /boot/vmlinuz
  root = /dev/sda3
  label = Huge
  read-only
```

On a 64-bit system (or on a non-SMP system) the stanza would look like this :

```
image = /boot/vmlinuz-generic-3.10.17
  initrd = /boot/initrd.gz
  root = /dev/sda3
  label = Generic
  read-only
```

Don't forget to run LILO to install the new bootloader image :

```
# lilo
```

Cross your fingers and reboot. Once you're running the GENERIC kernel, you may eventually want to get rid of the HUGE stanza in `/etc/lilo.conf`.

Install VirtualBox Guest Additions

Some folks like to run MLED as a VirtualBox guest for testing purposes. You may want to install the *VirtualBox Guest Additions*, otherwise graphics will run very poorly. Run **DEVICES > INSTALL GUEST ADDITIONS** in your virtual machine's contextual menu. Then :

```
# mount /dev/cdrom /mnt/cdrom
# cd /mnt/cdrom
# ./VBoxLinuxAdditions.run
```

Once the modules have been built, reboot.

Add one or more users

For the time being, root is the only user defined on our system. Our next step will be to define a « mortal » user with the adduser script :

```
# adduser kikinovak
Login name for new user: kikinovak
```

On a Slackware system, every newly created user gets automatically added to the users group. Now, access to certain devices is only granted to members of specific groups. When you get to the additional group selection, press the **[UP]** key to preselect a coherent series of groups for desktop use and then confirm by pressing **[ENTER]** :

```
Initial group [ users ]:
Additional UNIX groups:
...
Press ENTER to continue without adding any additional groups
Or press the UP arrow key to add/select/edit additional groups
: audio cdrom floppy plugdev video power netdev lp scanner
```

Use the groups command to check your group membership :

```
# groups kikinovak
kikinovak : users lp floppy audio video cdrom plugdev power netdev scanner
```

X11 in a nutshell

The X11 graphical server doesn't require the /etc/X11/xorg.conf configuration file anymore, so configuring X11 is no more the ordeal it used to be. The three most common brands of video cards are Intel, AMD/ATi and NVidia. Here's a quick overview on how to configure these cards.

Check out the model of your video card. Here's an example :

```
$ /sbin/lspci | grep -i vga
00:02.0 VGA compatible controller: Intel Corporation 82G33/G31
Express Integrated Graphics Controller (rev 0a)
```

Intel video cards are quite common. They work with the i915 kernel module in *Kernel Mode Setting* (KMS), which means you have to add this module explicitly to your initrd in order for your video card to work correctly.

Edit /etc/mkinitrd.conf and add the i915 module to the list of required modules :

```
# /etc/mkinitrd.conf
...
KEYMAP="fr_CH-latin1"
MODULE_LIST="ext4:i915"
ROOTDEV="/dev/sda3"
...
```

Rebuild your initrd :

```
# mkinitrd -F
```

Don't forget to update your bootloader image :

```
# lilo
```

If you have an AMD/ATi card, things work similarly :

```
$ /sbin/lspci | grep -i vga
01:05.0 VGA compatible controller: AMD/ATI [Advanced Micro Devices, Inc.]
RS780C [Radeon 3100]
```

These cards can be configured either with the free `radeon` driver, or with the proprietary `fglrx` driver provided by AMD. The `radeon` driver often works very well, but like the `i915` module, *Kernel Mode Setting* (KMS) is required.

In order to use the `radeon` driver, edit `/etc/mkinitrd.conf` and add it explicitly to your `initrd` :

```
# /etc/mkinitrd.conf
...
KEYMAP="fr_CH-latin1"
MODULE_LIST="ext4:radeon"
ROOTDEV="/dev/sda3"
...
```

As above, rebuild your `initrd` and update your bootloader image.

The case of NVidia cards is a bit different :

```
$ /sbin/lspci | grep -i vga
01:00.0 VGA compatible controller: NVIDIA Corporation GF119 [GeForce GT
520]
(rev a1)
```

In theory, NVidia cards can be configured with the `nouveau` driver in *Kernel Mode Setting* (KMS), e. g. added explicitly to your `initrd`. In practice, I've found performance with these free drivers rather poor. On the other hand, the proprietary `nvidia` drivers provided by NVidia work very well, so I recommend using them.

The *Slackware Documentation Project* sports detailed step-by-step tutorials on how to install and configure the proprietary `fglrx` and `nvidia` drivers, so I won't reinvent the wheel here.

Once the video driver is installed properly, log in as a normal user and try to start X :

```
$ startx
```

Depending on your choice during the installation, this will start `WindowMaker` or `Fluxbox`. The following command brings back the window manager selection screen :

```
$ xwmconfig
```

In your graphical environment, open a terminal and test your video configuration :

```
$ glxinfo | head -n 3
name of display: :0
display: :0 screen: 0
direct rendering: Yes
```

Last but not least, you might want to define a different keyboard layout for X. In that case, copy over a sample X11 configuration file stub :

```
# cd /etc/X11/xorg.conf.d
# cp /usr/share/X11/xorg.conf.d/90-keyboard-layout.conf .
```

Edit this file according to your needs. Here's an example of my workstation, configured to default to a swiss french keyboard :

```
Option "xkbLayout" "ch"
Option "xkbVariant" "fr"
```

Run `startx` again to check your keyboard configuration. If things went well, you can now define Xfce as your main working environment :

```
$ xwmconfig
```

Now switch to runlevel 4 by editing `/etc/inittab` :

```
# /etc/inittab
...
# Default runlevel. (Do not set to 0 or 6)
id:4:initdefault:
...
```

Two important notes on runlevel 4 :

- The `lxdm` package comes with a custom `/etc/rc.d/rc.4` startup file. If you see the ugly XDM login manager instead of LXDM, then go to the `/etc/rc.d` directory and rename your `rc.4.new` file to `rc.4`.
- A newly created user must define Xfce in the LXDM desktop selector on the bottom panel before logging in the first time.

Basic maintenance

Here's a few tips & tricks to keep your MLED installation up-to-date.

Check out the various `ChangeLog.txt` files at the root of each package repository for new additions and/or updates.

If you want to know the state of your system, what's installed, what's not installed or what updates are available, you can do all this in a few commands :

```
# slackpkg update
# slackpkg search microlinux-desktop
```

Or :

```
# slackpkg search microlinux-extras
```

Last but not least, application updates may sometimes overwrite the custom menu entries with a default `*.desktop` file. If that is the case, simply rerun the `cleanmenu.sh` utility :

```
# cd
# cd microlinux
# git pull
# cd desktop-14.1-32bit/tools
# ./cleanmenu.sh
```

Or :

```
# cd desktop-14.1-64bit/tools
# ./cleanmenu.sh
```

Enjoy your shiny new Microlinux Enterprise Desktop !

– Niki Kovacs <info@microlinux.fr>