

Extra Notes

Some things to note...

20210620 Slackware's CPU frequency scaling works (checkout `rc.cpufreq`) - just to remind myself, no need to look into this!

Admin-stuff for non-root user

To allow non-root users basic admin (poweroff, reboot, etc.), add them to power group and insert the following to sudoers.

```
%power ALL=(ALL) NOPASSWD:/sbin/poweroff
%power ALL=(ALL) NOPASSWD:/sbin/reboot
%power ALL=(ALL) NOPASSWD:/usr/sbin/pm-suspend
%power ALL=(ALL) NOPASSWD:/usr/sbin/pm-hibernate
%power ALL=(ALL) NOPASSWD:/usr/sbin/pm-powersave
```

Note: The `pm-*` binaries (`pm-utils`) are no longer available on Slackware 15.0

Note20250112: use `loginctl` on Slackware 15.0 - no need to add sudoers

Multicast DNS @ Zeroconf

- install `nss-mdns` (requires `avahi`, which requires `libdaemon`)
- save nss config (in etc)
 - `mv nsswitch.conf nsswitch.conf-orig`
- use provided nss config
 - `cp nsswitch.conf-mdns nsswitch.conf`
- run daemon at startup `rc.avahidaemon` and `rc.avahidsconfd`

Note20260720: latest install seems to NOT need to switch/edit `nssswitch.conf`

Using kolourpaint in XFCE without KDE

Simply install these packages

[pkgs4_kolourpaint.txt](#)

```
ark
karchive
kauth
kbookmarks
kcodecs
```

```
kcompletion
kconfig
kconfigwidgets
kcoreaddons
kcrash
kdbusaddons
kglobalaccel
kguiaddons
kil8n
kiconthemes
kinit
kio
kitemviews
kjobwidgets
kolourpaint
kservice
ktextwidgets
kwallet
kwidgetsaddons
kwin
kwindowsystem
kxmlgui
libksane
oxygen
solid
sonnet
```

Note: Can use `getslack --reset --slack $(cat pkgs4_kolourpaint.txt)`

NIS on Slackware

Got this from [here](#).

Slackware-NIS-mini-HOWTO

How to set up a NIS server and configure NIS clients on Slackware Linux

14-Dec-2000

David Cantrell <david@slackware.com>

```
=====
THE NIS SERVER
=====
```

STEP 1: Decide which machine will be the NIS server

This is usually not a very involving process, but does require some thought. You'll want to use a machine that is on most of the time and

does not need to reboot into other operating systems. If you have an existing fileserver, that would be a good choice for your NIS server.

STEP 2: Decide on what you want NIS to serve

NIS is fairly complex. It allows you to distribute all kinds of information. Things like usernames, passwords, groups, hostnames, and network services that should be running. The most common use is to use NIS as a common user logon authentication system. The second most common use is to have it share NFS automount maps so that users logging in on a system will have their home directory automounted for them.

In this mini-HOWTO, I will explain how to set up your NIS server and clients for passwd/group sharing and home directory sharing.

STEP 3: Move home directories to sharing location

In order to make the home directory NFS shares work seamlessly across the NIS server and clients, I move them to /export/home. This is the location they will be mounted on the client, so I make them exist there on the server as well. You might not have /export on your server, so let's make it:

```
mkdir -p /export
```

Now make the home directory location:

```
mkdir /export/home
```

Now move all your home directories over to /export/home. You should leave "ftp" in /home, as that's where Slackware installs it and it is sometimes site specific. These commands should move all home directories to /export/home and leave ftp in /home:

```
mv /home/* /export/home
mv /export/home/ftp /home
```

STEP 4: Edit home directory paths on the server

Since home directories are now in /export/home, you need to edit /etc/passwd on the server to reflect those locations. If you have a user called "david", you will want to change his home directory to /export/home/david. Do this for all users on the system.

STEP 5: Write the automount maps

I usually make two automount maps for the home directories. One is called `auto.master` and controls how the automounter works. The other is called `auto.export` and will describe the `/export` map.

Edit `/etc/auto.master` and add this to it:

```
#
# auto.master - Master map file for NFS automounter
#

/export auto.export --timeout 60
```

Now edit `/etc/auto.export` and add this to it:

```
#
# auto.export - NFS automounter configuration file
#

# Home directories
home -fstype=nfs <NFS/NIS server>:/export/home
```

Of course, replace `<NFS/NIS server>` with the hostname of your NFS/NIS server.

STEP 6: Edit the NFS exports file on the server

Now we need to make sure that `/export` is shared from the server. Edit `/etc/exports` and make sure this line is present:

```
/export (rw,no_root_squash)
```

Save the file and restart the NFS server daemons:

```
killall -HUP rpc.nfsd rpc.mountd
```

STEP 7: Pick a name for your NIS domain

Each NIS domain needs to have a unique name. Do not use the same name as

your DNS domain. This should be something different. I will use "mynis" in this example.

Edit /etc/defaultdomain and add the name of your NIS domain:

```
mynis
```

Save and exit. The name of your NIS domain should be the only thing in that file. Once you've done that, set the NIS domain with this command:

```
nisdomainname `cat /etc/defaultdomain`
```

STEP 8: Edit /var/yp/Makefile

The NIS server has a Makefile that controls what maps are served out by NIS. Each time you make a change (such as adding a user), you should go into /var/yp and type "make" to update the NIS information.

In this file we need to edit the all: target and specify the stuff we want to share. Here's what your all: target should look like:

```
all: passwd group hosts rpc services netid protocols netgrp mail \
    shadow auto.master auto.export
```

Now find the line that looks like this:

```
AUTO_LOCAL = $(YPSRCDIR)/auto.local
```

We want to add this line just after that line:

```
AUTO_EXPORT = $(YPSRCDIR)/auto.export
```

Now find the auto.local: target. We need to add this target just after that target:

```
auto.export: $(AUTO_EXPORT) $(YPDIR)/Makefile
@echo "Updating $@"
-@sed -e "/^#/d" -e s/#.*$$// $(AUTO_EXPORT) | $(DBLOAD) \
    -i $(AUTO_EXPORT) -o $(YPMAPDIR)/$@ - $@
-@$(NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@
```

Save the Makefile and exit.

STEP 9: Edit /var/yp/securenets

Edit /var/yp/securenets and change the "0.0.0.0 0.0.0.0" line to something like this:

```
255.255.255.0 192.168.1.0
```

You should specify your network address and submask there, this is only an example. This example restricts NIS access to any machine on the 192.168.1 class C subnet.

STEP 10: Create an /etc/ypserv.conf file

The only line you need in this file is:

```
dns: yes
```

This line tells ypserv to resolve hostnames using a DNS server instead of relying on NIS hostname maps (which you probably aren't using). Save and exit.

STEP 11: Start ypbind

Start ypbind:

```
/usr/sbin/ypbind
```

Use ps to make sure it started and is running properly. You will probably see a master and slave process for it.

STEP 12: Start ypserv

Start ypserv:

```
/usr/sbin/ypserv
```

Use ps to make sure it started and is running properly.

STEP 13: Initialize the NIS maps

We need to initialize the NIS maps with this command:

```
/usr/lib/yp/ypinit -m
```

Fill in the blanks for all machines on your network that will be acting as

NIS servers. At this point, you probably want to fire up `rpc.yppasswdd` to make sure that users on NIS clients can run `chfn` and `passwd`.

STEP 14: Done, making sure NIS stuff loads at bootup

Your NIS server is now up and running. You should edit `/etc/rc.d/rc.inet2` and make sure the following block is uncommented:

```
# Setting up NIS:
# (NOTE: For detailed information about setting up NIS, see the
documentation
# in /usr/doc/yp-tools, /usr/doc/ypbind, and /usr/doc/ypserv)
#
# First, we must set the NIS domainname. NOTE: this is not
# necessarily the same as your DNS domainname, set in
# /etc/resolv.conf! The NIS domainname is the name of a domain
# served by your NIS server.

if [ -r /etc/defaultdomain ]; then
    nisdomainname `cat /etc/defaultdomain`
fi

# Then, we start up ypbind. It will use broadcast to find a server.

if [ -d /var/yp ] ; then
    echo -n " ypbind"
    ${NET}/ypbind
fi

# If you are the NIS master server for the NIS domain, then
# you must run rpc.yppasswdd, which is the RPC server that
# lets users change their passwords.

if [ -x ${NET}/rpc.yppasswdd ]; then
    echo -n " yppasswdd"
    ${NET}/rpc.yppasswdd
fi
```

It's probably advisable to run `rpc.yppasswdd` with this line:

```
${NET}/rpc.yppasswdd -e chsh -e chfn
```

This ensures that `chfn` and `chsh` work. You also need to add this after the `ypbind` if block:

```
if [ -x ${NET}/ypserv ]; then
    echo -n " ypserv"
    ${NET}/ypserv
fi
```

The ypserv program needs to run on the NIS server.

```
=====
THE NIS CLIENT
=====
```

STEP 1: Set the NIS domain

Edit /etc/defaultdomain just like you did on the server and enter the name of your NIS domain.

STEP 2: Edit /etc/yp.conf

Open /etc/yp.conf in a text editor and make sure this line is present:

```
ypserver <IP or hostname of the NIS server>
```

Specify either the IP address or hostname of your NIS server. If you don't have working hostnames, you must use the IP address here.

STEP 3: Set the NIS domain

Set the NIS domain by typing this command:

```
nisdomainname `cat /etc/defaultdomain`
```

STEP 4: Run ypbind

Start ypbind with this command:

```
/usr/sbin/ypbind
```

STEP 5: Test the NIS connection

See if the NIS server is working with this command:

```
ypcat passwd.byname
```

If you see the /etc/passwd entries from the NIS server, you're good to go.

Otherwise, go back to the NIS server section and see if you missed any steps.

STEP 6: Edit /etc/nsswitch.conf

You need to edit /etc/nsswitch.conf and tell it to use NIS for passwd, shadow, and group. Change those entries to:

```
db files nis
```

Change the automount entry to:

```
files nis
```

STEP 7: Enable NIS logins in /etc/passwd and /etc/group

Edit /etc/passwd and add this line to the end:

```
+:::
```

Edit /etc/group and add this line to the end:

```
+:::
```

STEP 8: Create the /export directory for home directory mounting

Make sure you have that directory:

```
mkdir -p /export
```

STEP 9: Create an rc.autofs script

The included rc.autofs script should be moved into /etc/rc.d. Edit rc.local and add this line:

```
/etc/rc.d/rc.autofs
```

STEP 10: Done, enable NIS stuff at bootup

Your NIS client is now configured. Edit /etc/rc.d/rc.inet2 and uncomment this block:

```
# Setting up NIS:
# (NOTE: For detailed information about setting up NIS, see the
documentation
# in /usr/doc/yp-tools, /usr/doc/ypbind, and /usr/doc/ypserv)
#
# First, we must set the NIS domainname. NOTE: this is not
# necessarily the same as your DNS domainname, set in
# /etc/resolv.conf! The NIS domainname is the name of a domain
# served by your NIS server.

if [ -r /etc/defaultdomain ]; then
    nisdomainname `cat /etc/defaultdomain`
fi

# Then, we start up ypbind. It will use broadcast to find a server.

if [ -d /var/yp ] ; then
    echo -n " ypbind"
    ${NET}/ypbind
fi
```

From:

<https://www.azman.my1matrix.cc/> - Azman @MY1

Permanent link:

https://www.azman.my1matrix.cc/doku.php?id=linux:slack_9notes&rev=1790524774Last update: **2026/09/27 15:59**