Replicator 3.1 for Debian GNU/Linux Sarge
User's Guide

Sébastien Chaumat

08-19-2004

Contents

1 Conventions **3**

2 Disclaimer **3**
2.1 Authors 3
2.2 License 3
2.3 Greetings 3
2.4 Support 3

3 Introduction **4**
3.1 Where to find it 4
3.2 Purpose 4
3.3 Overview of the replication 4
3.4 Warnings 4
3.5 What's new since woody version 5
3.6 TODO 5

4 Using Replicator **6**
4.1 Roadmap (READ THIS COMPLETELY FIRST!) 6
4.2 Installing Replicator (warning) 6
4.3 Upgrading from prior version 6
4.4 Configuration of *replicator.conf* 6
4.4.1 the general section 6
4.4.2 the miniroot section 6
4.4.3 the bootdisk section 7
4.4.4 the target network section 7
4.4.5 the target partitioning section 7
4.4.6 the update rules section 8
4.4.7 the miscellaneous section 8
4.4.8 the interactivity section 8
4.4.9 the Experts Only section 9
4.5 Site preparation for the replication 9
4.5.1 Compiling a boot kernel for the installation floppy (VERY IMPORTANT) 9
4.5.2 Preparing the miniroot 10
4.5.3 Authorising targets to access model with rsyncd 10
4.5.4 Preparing the custom pre-replication script 11
4.5.5 Preparing the custom post-replication script 11
4.6 Installing the target 11
4.6.1 Creation of the boot floppy 11
4.6.2 Using the boot floppy 11
4.7 The light classes mechanism (aka the strategy guide) 12

4.7.1 Installing computers in more than one network 12
4.7.2 Installing a special computer among a set of identical others 12

5 Internals : how it works **14**

6 Doing more with Replicator **14**
6.1 Modifying the behaviour of replicator with the command line 14
6.2 Keeping 2 computers synchronised 15
6.3 Replicate faster 15
6.4 Installing a diskless client 15
6.5 Installing a client with */usr* mounted by NFS 15
6.6 model being different of miniroot-server 15

A Default update rules **16**

B replicator.conf.example **17**

1 Conventions

- This is a **computer**
- This is a command
- This is a $perl_scalar_variable
- This is a @perl_list_variable
- This is a *file*

2 Disclaimer

2.1 Authors

Replicator is an original work by Loïc Prylli (lprylli@lhpca.univ-lyon1.fr). It originated as some custom scripts to help quickly installing computer classrooms and or clusters of computers. As I (Sébastien Chaumat (schaumat@debian.org)) needed to do the same kind of work (having the same Debian/GNU Linux system on a lot of different computers), we started to transform the initial scripts to make a general-purpose system allowing to quickly and non-interactively install computers in any networked environment. It allows you to choose for each station whether it will be a full install, a "/usr nfs-mounted" install or a diskless install. It can handle automatically the main steps of hardware detection (hard disk, Xfree86), and partitioning (either a completely automatic partitioning with the option of preserving your DOS/Windows partition, or a manual partitioning preserving some of the hard disk filesystems).

Loïc Prylli stopped working on Replicator in 2002. All requests to the authors should be sent to Sébastien Chaumat.

2.2 License

Copyright (C) 1999-2004 by Sébastien Chaumat (schaumat@debian.org) and Loïc Prylli (lprylli@lhpca.univ-lyon1.fr).

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

A copy of the GNU General Public License is available as */usr/share/common-licences/GPL* in the Debian GNU/Linux distribution or on the World Wide Web at http://www.gnu.org/copyleft/gpl.html. You can also obtain it by writing to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

2.3 Greetings

Pascal Degiovani convinced me to look for an auto-install tool for Linux. He introduced Loïc Prylli to me. I'm very grateful to him.

I thanks the École Normale Supérieure de Lyon for its support.

Bertrand Louis-Lucas proof-read this documentation.

Dominique Ponsard and Hervé Brunet made beta-testing.

Lots of debugging was done by Bernd Harmsen and Jerome Warnier.

Thanks to all users who reported bugs and made suggestion.

2.4 Support

Feel free to contact the author (schaumat@debian.org) and to use the Debian Bug Tracking System (please try to contact me before).

3 Introduction

3.1 Where to find it

You can find the latest development version of Replicator at :

http://replicator.sourceforge.net

Check out the cvs for the very last improvements. Releases are available in Debian unstable.

3.2 Purpose

If you are the system administrator of a networked site, how can you (re)install quickly a computer in generally less than fives minutes (even less if you install a set of similar machines)?

You want to customise the configuration of a computer only once (on the **model** computer) for your site. Moreover you want any new **target** computer to have almost the same configuration as the **model**. The **target** will be mostly identical to the **model** except:

- some specific configuration files (network, fstab, X configuration, hostname),
- the way it is partitioned,
- the possibility of being diskless, having /usr NFS-mounted instead of local, or having specific filesystems (local homes for instance),
- ...

The present software, called “Replicator”, will automates the installation of **target**. More precisely it consists of a set of easy-to-use scripts to prepare the installation, the result of the preparation being a floppy that when booted on the target, will almost automatically install it (by default, it asks confirmation first!).

3.3 Overview of the replication

The copying process is composed of the following steps:

1. Installation and configuration of Replicator on the computer **miniroot-server** which usually is also the **model** you want to replicate,,
2. Creation and setup of a tiny NFS root filesystem onto **miniroot-server**,
3. Creation of bootfloppy onto **miniroot-server**,
4. Configuration of remote access to the **model**,
5. Quick install of the computer **target** with the floppy.

3.4 Warnings

1. The whole directory tree of your **model** machine will be duplicated according to the “update rules” mention in 4.4.6. If you created non-standard directories (e.g. mountpoints) which you don’t want to duplicate, then edit the update rules accordingly.
2. This software may be dangerous for your data, be sure that for every machine where you execute one of the scripts or for every machine that you include in one the configurations files, all the volumes accessible from the machine (either local or NFS-mounted) are properly backup’ed,
3. Understanding a bit of Perl and sh scripts will help to diagnose potential problems.

3.5 What's new since woody version

- Replicator will use the bootloader (lilo or grub) which is installed onto the model and fall back to grub if necessary.
- You can skip the bootloader installation ($skip_bootloader).
- Lilo configuration is now always done using a template file (default is */etc/lilo.conf* from **model**).
- You can tune the level of interactivity during the replication. You can even create a completely non-interactive process.
- The file */etc/resolv.conf* is copied from the **miniroot-server** inside the miniroot and thus is available during the replication.
- The content of the $hosts_supp variable is added to the file */etc/hosts* in the miniroot to allow resolving names not in the DNS during the replication.
- You can define the filesystem type you want on the target *on each partition independently* .
- The bootdisk now has a 5 seconds timeout and boot a dhcp target by default.
- This version is designed for Sarge but still supports Potato and Woody.

3.6 TODO

I am still working on:

- read-only miniroot with ramdisk support for parallel replication
- add rsyncd password / ssh encryption for replication of sensitive data
- ultra fast replication using an archive on model
- faster than light replication for clusters using codafs
- switch to full class mechanism.
- make congelator: put the model on a CD.
- PXE helper
- something better

4 Using Replicator

The computer you want to replicate is called your **model**. All along this documentation we refer to a computer called **miniroot-server**. For basic use of replicator, **miniroot-server** *IS THE SAME* than **model**. This computer is the one where you want to install replicator.

4.1 Roadmap (READ THIS COMPLETELY FIRST!)

To replicate your **model** you will follow this roadmap :

1. Install replicator on the **model**
2. Configure replicator by editing */etc/replicator/replicator.conf* and other files on the **model**
3. Create a miniroot on the **model** , tweak it, and export it with a nfs server
4. Create a custom kernel and put it in */etc/replicator/* on the **model**
5. Create a bootdisk using repli-bootdisk on the **model**
6. Launch a rsync daemon on the **model**
7. Boot the **target** with the just created bootdisk

4.2 Installing Replicator (warning)

It's a standard Debian package :-). Just use dpkg or apt.

Warning: The kernel running on the computer where you will create the bootdisk (see 4.6.1) must support loop device. Standard kernels shipped with Debian are ok.

4.3 Upgrading from prior version

If you are upgrading from prior version, I strongly suggest that you create a new miniroot using repli-miniroot.

You should also check carefully the new */usr/share/doc/replicator/replicator.conf.example*.

4.4 Configuration of *replicator.conf*

Edit the file */etc/replicator/replicator.conf* (it is heavily commented). Change the value of the Perl variables to fit your needs.

The following sections are presented here in the order they are used during the replication.

4.4.1 the general section

- $verbose: set this to 1 and replicator will be more verbose than usual.
- $debian_version : which Debian version will you replicate ("potato", "woody" or "sarge")?

4.4.2 the miniroot section

- $nfsroot: where, on miniroot-server, will you create the miniroot.
- $debian_mirror: where will you fetch the Debian packages from to build the miniroot (http://, ftp:// or file:///).

4.4.3 the bootdisk section

- $bootkernel: full path to the *bzImage* you prepared as explained in 4.5.1.
- if you need to boot one target on a serial console:
  - $grub: set to 0 to use Lilo instead of Grub on the bootdisk. This is compulsory if you want to boot on a serial console (grub serial boot is not supported in replicator yet).
  - $serial: definition of serial port for Lilo, only if you want to boot on serial console,
  - $serialcons: definition or serial port for the kernel, only if you want to boot on serial console.
- @networks: define one network per line. For each network, choose a name for the target list then create the corresponding Perl variable. Look at the following example:

```
#List of networks/targets the bootdisk will handle.
#you don't need to fill this variable if you only
#plan to install dhcp clients.

@networks =
#  network         domain              netmask          gateway          broadcast          targets list
(["192.168.32.0", "prof.my.domain",   "255.255.255.0", "192.168.32.1", "192.168.32.255",  "teachers"     ],
 ["192.168.33.0", "my.domain",        "255.255.255.0", "192.168.33.1", "192.168.33.255",  "sysadm_net"   ],
 ["192.168.34.0", "maths.my.domain",  "255.255.255.0", "192.168.34.1", "192.168.34.255",  "maths"        ],
 );

#for each network create the corresponding targets list
@teachers=qw(teach1 teach2);
@sysadm_net=qw(adm1 adm2 adm3);
@maths=qw(maths1 maths2 maths3 maths4);
```

4.4.4 the target network section

- $hosts_supp: lines to add to the */etc/hosts* of **target** machine. It is also added inside the miniroot and is thus available during the replication.
- $nfsopts: for diskless stations or NFS-mounted */usr* partition only. See 6.4.

4.4.5 the target partitioning section

- $target_disk: disk of **target** on which you want to install. By default Replicator use the first detected disk (hda or sda).
- $mkfs : default command to create a filesystem on the **target**. The kernel you use on the replication floppy must support this filesystem. The miniroot contains needed packages for reiserfs, xfs and jfs. e.g.

  $mkfs="mkfs -text3"
- $custom_part: set to 1 if you want to manually partition the hard disk of **target**,
- @autopart_specs: specification of the partitioning and formating scheme of $target_disk (min and max size for each partition). See *replicator.conf.example*.
- @manualpart_specs: describe the existing partitions and filesystems to reuse. Only used if $custom_part=1. See *replicator.conf.example*.
- $fstab_supp: lines to add to the */etc/fstab* of **target**.

4.4.6 the update rules section

Nothing but a reminder. To fine tune the list of files/directories which are replicated, edit */etc/replicator/update_rules* AND set $modify_rules_file to 1. you delete this file, the default rules are in */usr/lib/replicator/update_rules.default*. In this files are defined 3 variables:

- @slash_exclude
- @usr_exclude
- @var_include

This is how Replicator proceeds to copy the content of the filesystems of **model** (see 5 for explanation about the whole process):

- It replicates the root (/) filesystem except for what is mentioned in @slash_exclude. If you use cfengine (a very useful program for network administration), Replicator parses */etc/cfengine/cfengine.conf* and does not copy files handled by cfengine copying facility.
- It replicates the */usr* directory (not the filesystem) except for what is mentioned in @usr_exclude. Be aware that it may cross filesystems to achieve this.
- It replicates the directories structure of */var* (only directories, no files).
- It replicates the contents of the */var* subdirectories mentioned in @var_include.

To define an empty update rule use the syntax: @usr_exclude=qw("");

4.4.7 the miscellaneous section

- $usr_is_nfs: set this variable to 1 if */usr* must be a NFS-mounted filesystem on **model** instead of being on a local partition (see 6.4).
- @extra_keymaps: this is the list of keymaps you will be asked to choose from when booting the target at replication time. Note that the keymap of the **model** is automatically added. This list of keymaps is not used if $keymap is defined in the next section. For a list of valid keymaps names look at */usr/share/keymaps/*/*/*.

4.4.8 the interactivity section

The variable in this section will allow you to limit interactivity during the replication. Be aware that disabling warnings raise the risk of loosing data or worse.

- $keymap: if set, the corresponding keymap will be installed on the **target**.
  Look at */usr/share/keymaps/*/*/* for the possible values. If this variable is set, you won't be prompted to choose a keymap at the beginning of the replication.
- @nowarn: if set to 1, Replicator won't warn you about the partition scheme and will erase the disk immediately, according to the settings in */etc/replicator/replicator.conf*. So if you made a mistake in the configuration you will loose data.
- @nologin:if set to 1, you won't be asked to login before starting the replication. So be sure that nobody else can boot on the exported miniroot. Otherwise someone can install a target without knowing the installation password.
- @autoreboot: if set to 1, the **target** will reboot as soon as the replication has finished. The drawback is that you must have removed the bootfloppy first. Otherwise the replication can restart automatically.

Note : to have a completely non interactive replication you need to have the 4 above variables defined. Anyway if you choose to install a dhcp client target you will always be prompted for the hostname at the beginning of the replication.

4.4.9 the Experts Only section

- $model : if the **model** is NOT the same as the **miniroot-server** set this to the name of the **model** and read 6.6.
- $lilo_template : instead of using as a template the */etc/lilo.conf* file from the **model** , Replicator can use the one you provide here. Only the boot and root statements will be changed by Replicator according to the detected medium on the target.
- $skip_lilo : if you don't want Replicator to generate *lilo.conf*, then set this to 1. To force Replicator to copy the */etc/lilo.conf* from the model to the target, you must modify the update rules (*lilo.conf* is excluded by default). See 4.4.6.
- $skip_bootloader : if you want to completely prevent Replicator from installing a bootloader, then set this to 1. You must then install a bootloader on the **target** either manually or using /repli-postint (see 4.5.5).
- $skip_fstab : same as $skip_lilo but for */etc/fstab*.
- $sharedir, $scriptsdir, $sbindir : used to run Replicator from the cvs .
  See */usr/share/doc/replicator.conf.example*.

4.5 Site preparation for the replication

4.5.1 Compiling a boot kernel for the installation floppy (VERY IMPORTANT)

Compile a kernel with all the necessary driver for the computer **target**. Do not use modules for the drivers required at boot time (typically network and SCSI). You should even completely disable modules support in this kernel.

Add the following option during configuration of the kernel:

kernel 2.4.X	kernel 2.6.X	Option to set
Networking options / IP: kernel level autoconfiguration	Devices Drivers / Networking Support / Networking options	IP: kernel level autoconfiguration DHCP support
Filesystems / Network File Systems	Filesystems / Network File Systems	NFS filesystem support Root file system on NFS
Filesystems/ *	Filesystems/ *	See below

- Filesystems/ * : add the filesystem you want to put onto the partitions of the targets (see the $mkfs var in 4.4.5)

You *must* create such a kernel because the standard Debian kernels do not have nfsroot nor dhcp support enabled. This kernel will only be used for the installation floppy and will not be installed on **target**.

Compile this kernel with classical make dep ; make bzImage (don't use make-kpkg). This kernel will be use to boot the computer **target**. Put this new kernel in a safe place. In the configuration file */etc/replicator/replicator.conf*, the variable $bootkernel should point to this kernel.

Notes :

- if you cannot make a small enough kernel with linux 2.6, use version 2.4 of the kernel sources. Remember this kernel won't be installed on the **target**.

4.5.2 Preparing the miniroot

As root, execute repli-miniroot. You need access to a Debian mirror. It can be a CD set, a remote mirror somewhere on the network or a local mirror.

This will create a special filesystem (e.g. */export/install/miniroot*) with a basic Debian/GNU Linux on it.

VERY IMPORTANT: Each time you modify the configuration in */etc/replicator/* you must run the command:

```
repli-miniroot –update-config
```

or

```
repli-miniroot -u
```

Don't forget to export *miniroot-dir* for **target** *with the no_root_squash* option. To do that, edit */etc/export* and add a line like:

```
/export/install/miniroot target.my.domain(rw,no_root_squash)
```

Then tell the nfs-server to reload it's configuration:

```
/etc/init.d/nfs-kernel-server reload
```

Warning: Every line of */etc/export* file must end with a newline character otherwise you will get silly error messages at boot time.

4.5.3 Authorising targets to access model with rsyncd

Replicator comes with a */etc/replicator/rsyncd.conf* file:

```
#
#this is rsyncd.conf for replicator
#
[replicator]
path=/
use chroot=no
read only=yes
uid=0
#max connections = 20
#
#in case you don't use tcpwrapper (bad:)
hosts allow=192.168.0.0/255.255.0.0
hosts deny=*
```

You are responsible for launching rsyncd on the **model** with this config files. That means either moving this file to */etc/rscynd.conf* or editing your existing */etc/rscynd.conf* to add the rsync module [replicator] as defined into */etc/replicator/rsyncd.conf*.

You can either :

- configure inetd to allow rsyncd.
- edit */etc/default/rscync* (for Debian Sarge) or make a script to launch rsyncd as a standalone server at boot time (for Debian potato or woody).
- launch rsyncd by hand with the command rsync –daemon.

Don't forget to change the "hosts allow" and "hosts deny" settings.

4.5.4 Preparing the custom pre-replication script

You can create a shell script /etc/replicator/repli-pretinst. It will be executed at the very begining of the copying process (before any attempt to partition a disk). It is *your* preinst script. Again run repli-miniroot –update-config to copy this file to the miniroot.

4.5.5 Preparing the custom post-replication script

You can create a shell script /etc/replicator/repli-postinst. It will be executed at the very end of the copying process. It is *your* postinst script. Again run repli-miniroot –update-config to copy this file to the miniroot.

Note: This script is run chrooted in the root filesystem of **target** (exactly as it would run after rebooting).

Here are some suggestions of what can you put in this script:

- dpkg-reconfigure ssh: will create a new public/private keys pair.
- dpkg-reconfigure xserver-xfree86: will replay the xserver configuration (useful if video hardware is different).

4.6 Installing the target

4.6.1 Creation of the boot floppy

Use the script repli-bootdisk.

You can put a floppy right now in your floppy drive. If you interrupt repli-bootdisk with ctrl-C you can copy by hand the file */tmp/name-of-the-target.img* on a floppy with the command cp.

4.6.2 Using the boot floppy

1. sit down on front of the **target** computer,
2. insert the floppy
3. turn the power on,
4. if you did not set $nologin, login as root with the password you set at the creation of the miniroot,
5. answer the questions if needed,
6. If you choose the manual installation, launch the following scripts yourself. You must add the –real option to make the scripts actually do something.
   (a) repli-install hdsetup (–real)
   (b) repli-install hostconf (–real)
   (c) repli-install netcopy (–real)
   (d) repli-install configure(–real)
   (e) repli-postint
7. remove the floppy,
8. reboot.

4.7 The light classes mechanism (aka the strategy guide)

This is very important and useful. One easy way to define replication classes is to create specific configuration files.

To summarise:

- repli-bootdisk –config /etc/replicator/my_specific_file first loads */etc/replicator/replicator.conf* and then loads the file *my_specific_file*. Whatever you put into *my_specific_file* overrides the content of */etc/replicator/replicator.conf*.
- *after booting the target* the file */etc/replicator/replicator.conf* is read first and, if it exist, the file */etc/replicator/replicator.conf_targetname* is read.
- */etc/replicator/replicator.conf_targetname* can be a symbolic link to */etc/replicator/replicator.conf_myclass*.

The following examples should help you understand how replicator deals with its configurations files.

4.7.1 Installing computers in more than one network

Starting with version 2.0.1 of replicator, you only need to fill the variable @networks and the corresponding lists of targets in *replicator.conf*. See 4.4.3.

4.7.2 Installing a special computer among a set of identical others

Lets suppose you are installing a classroom: all computers are identical (call them **student1**, **student2**, ...) except the teacher's one (just call it **teach1**).

For example, students computers, have a 1GB harddisk with /home nfs-mounted and boot vith dhcp, whereas **teach1** has hardcoded network configuration, one 4GB disk for the OS and a second ide disk with local /home in /dev/hdb1.

In such a case we use */etc/replicator/replicator.conf* to put defaults for the identical computers and we make a file */etc/replicator/replicator.conf_teach1* for the unique computer **teach1**.

In */etc/replicator/replicator.conf* :

```
@networks =
#  network      domainname          netmask        gateway       broadcast        name of targets list
(
[ "192.168.32.0" , "prof.my.domain", "255.255.255.0", "192.168.32.1", "192.168.32.255 ", "Teachers"]
);
@Teachers=qw(teach1);
@autopart_specs =
#mount point          min_size       max_size    mkfs
   ([ "/" ,            "100Mo",      "1000Mo",        ”mkfs -text3”],
   [ "swap",           "64Mo",        "128Mo",          ””            ]
   );
$fstab_supp = "/dev/fd0 /floppy auto defaults,user,noauto 0 0
homeserver:/exports/home /home nfs rw,defaults,rsize=8192,wsize=8192 0 0
/dev/cdrom /cdrom auto ro,defaults,user,noauto 0 0\n";
```

In file */etc/replicator/replicator.conf_teach1*:

```
#mount point           min_size        max_size mkfs
   ([ "/" ,             "100Mo",       "400Mo",   ”mkfs -treiserfs” ],
   [ "swap",            "64Mo",        "128Mo",  ””                ],
   [ "/usr" ,          "800Mo",      "3000Mo",  ”mkfs -text3 ”     ],
   [ "/var" ,          "200Mo",      "1000Mo", ”mkfs -t ext3”      ]
  );
$fstab_supp = "/dev/fd0 /floppy auto defaults,user,noauto 0 0
/dev/hdb1 /home auto defaults 0 0
/dev/cdrom /cdrom auto ro,defaults,user,noauto 0 0\n";
```

Now create the bootdisk with :

```
repli-bootdisk –config /etc/replicator/replicator.conf_teach1
```

To install a student computer, boot it with the bootdisk and choose ”Boot with DHCP and Install a DHCP target” in grub’s menu. To install **teach1**, use the same bootdisk and choose ”Boot teach1 ...” in grub’s menu.

5 Internals : how it works

This section describe precisely how replicator works. This is useful to understand common errors and misconfigurations.

What appends just after booting the target with the replication bootdisk

1. The kernel (see 4.5.1) is loaded from the floppy.
2. The hard disks and the network card are detected because you compiled appropriate drivers in the kernel.
3. The root filesystem is mounted from the exported miniroot. This is possible because you added the appropriate stuffs in the kernel (see 4.5.1 again) and because you allowed the **target** to access the exported miniroot (see 4.5.2).
4. The inittab is read, if you defined $nologin, the system enters runlevel 3 and the next stage automatically starts, otherwise this will be runlevel 2 and you are prompted for a password before steeping in next stage.
5. If you are installing a dhcp target, you are prompted for the hostname.
6. If $keymap is not defined you are asked to choose the keymap in @extra_keymaps. If the later is also not defined, the installed keymap is the same as the one of the **model**.
7. $nowarn is not defined you are asked to confirm you partitioning and formating scheme.

- If you created a preinstallation script */etc/replicator/repli-preinst*, it is run.

1. The actual replication starts :
   (a) the disk is partioned and or formated, partitions are mounted.
   (b) */*, */usr* and */var* are rsynced according to the update rules (see 4.4.6); This is possible because you set up the rsync daemon on the **model** (see 4.5.3)
   (c) The files */etc/hosts* and */etc/fstab* are created on the **target**
   (d) The bootloader is installed on the **target**. If you use lilo or grub on the **model**, this process does not require any specific configuration. Nevertheless you can fine tune it in the ”Experts Only ” section of the configuration file */etc/replicator/replicator.conf*.
2. The script repli-postint is executed on the **target**, chrooted in the newly created root filesystem (see 4.5.5).
3. If $autoreboot is not set you are asked to confirm the reboot of the **target**.

6 Doing more with Replicator

6.1 Modifying the behaviour of replicator with the command line

You can also give the arguments on the command line of the different scripts repli-command:
instead of "$var = val" in *replicator.conf* you can use the option –var val. Example:

```
repli-bootdisk –bootkernel /path/to/the/boot/kernel
```

Command-line arguments take precedence over the configuration file *replicator.conf*.

6.2 Keeping 2 computers synchronised

Suppose you change something (e.g. you add a package) onto the computer **model**. To keep **target** a real copy of **model** you can either make the same operation by hand onto the computer **target** or use the script repli-sync.

To use this script you must install replicator onto the target (it is already on the target if it was installed onto the model).

Before using repli-sync [–real], you have to check that the file */etc/replicator/reply-sync.conf* exists on the **target**.

It should contains something similar to:

```
$model="mymodel";
@usr_exclude=qw(/var /local /src);
$noboot=1; #option : do not update /boot
$norcd=1;  #option : do not update /etc/rc?.d
$nousr=0;  #option : do not update /usr
#do not remove the following line..
$the_end=1;
```

6.3 Replicate faster

If the disk on the **target** is IDE then you can make the replication faster by using DMA access for this disk. To allow this, on the **miniroot-server**, in the miniroot, edit */etc/hdparm.conf*.

6.4 Installing a diskless client

If you want to easily install a diskless client, **target**, with its filesystem on the computer **miniroot-server**, follow these instruction:

- export with NFS, read-only, the directory */usr* of **miniroot-server** for itself and for **target**,
- edit the variable $targetbase into repli-diskless or create a directory */export/diskless*,
- create the NFS root filesystem of **target** by executing the command: sh repli-diskless,
- export with NFS */export/diskless/**target*** for **target**,
- edit *repli-postint.diskless* if necessary,
- create the bootdisk of **target** by executing the command:
  repli-bootdisk –nfsroot /export/diskless/target –host target

Boot **target** with this floppy. That's all!

6.5 Installing a client with */usr* mounted by NFS

Add the option $usr_is_nfs=1 into *replicator.conf* and export the directory */usr* of **model** for **target**.

6.6 model being different of miniroot-server

You may want to have a dedicated **miniroot-server**, for security reason or because you just don't want to install replicator onto your **model**.

To do so, first install replicator on **miniroot-server**. Then in */etc/replicator/replicator.conf* customise the variable $model.

Then read again this documentation without considering **miniroot-server=model**.

A Default update rules

You can find them in */usr/lib/replicator/update_rules.default*.

```
#
#update_rules
#
#In this file you can tune the behaviour of replicator
#
#WARNING: to define an empty rule use the syntax: @usr_exclude=qw("");
#####################
#                   #
# "/" exclude rules #
#                   #
#####################
#The following files/directoriess are NOT copied from $server to $target

@slash_exclude=qw(
/etc/my
/etc/network/interfaces
/etc/.rootkey
/etc/adjtime
/etc/isapnp.conf
/etc/minirc*
/etc/gpm.conf
/etc/fstab
/etc/lilo.conf
/etc/hostname
/etc/hosts
/etc/aboot.conf
/etc/exports
/etc/dhcpd.conf
/etc/fmirror
/etc/apache/*
/etc/squid.conf
/etc/chatscripts/*
/etc/init.d/network
/etc/ppp/*
/etc/named.conf
/etc/named.boot
/etc/ssh/ssh_host_*key*
/etc/ssh/ssh_random_seed
/etc/rmtab
/etc/mtab
/etc/kbd/*
/etc/console-tools/*
/etc/X11/XF86Config*
/etc/X11/Xserver
/etc/ioctl.save
/etc/mon/
/etc/ssh2/hostkey
*.cfsaved
*.dpkg-old
/tmp/*
/var
/usr
/bigpart
/boot
/*old*
/cdrom/*
/local
/localhd
/fatboot
/home*
/proc/*
/tchich
/ramdisk/
/linuxppc/
/oldexports/
/amd/
/tftpboot/
/*mnt*/*
/floppy/*
/export*/
/exports/*
/opt*
/net*
/vol*
/ufs*
/lost+found/
);

#####################
#                   #
#/usr exclude rules #
#                   #
#####################
#The following files/subdirectories of /usr are NOT copied from $server to $target

@usr_exclude=qw(local src);

#####################
#                   #
#/var include rules #
#                   #
#####################
#ONLY the content of the following subdirectories of /var will be copied from $server to $target
#NOTE: the complete directories structure will be replicated (not the content of those directories)
@var_include = qw(lib);
#NOTE: it is not possible to always include yp/nicknames here. If this file
#does not exist, rsync will fail.
```

B replicator.conf.example

You can find this file in */usr/share/doc/replicator/* .

```
#/etc/replicator/replicator.conf
#
#This is the place to configure replicator

#Warning this file is an example. Customize it and check it twice
#before using replicator.

############################################################
#                                                          #
# Section: general configuration                           #
#                                                          #
############################################################

#Be very verbose
$verbose=1;

# Debian version to install both in the miniroot and on the target
#("potato", "woody", or "sarge")
#comment out to let replicator guess for you

$debian_version="sarge";

############################################################
```

```
#                                                          #
# Section: miniroot configuration                          #
#                                                          #
############################################################

#Where will you create the miniroot
$nfsroot= "/export/miniroot";

#prefered debian mirror (http://, ftp:// or file:///).
#$debian_mirror= "http://ftp.debian.org/debian";

###########################################################
#                                                         #
# Section: bootdisk configuration                         #
#                                                         #
###########################################################

# If for the bootdisk you want to use a special kernel rather
# than the Debian default.
# Debian default is not yet supported
$bootkernel = "/etc/replicator/do-not-for-get-to-compile-a-bzImage-for-replicator";

#not used yet
#$initrd = '';

#to boot on a console on a serial port customize this:
#$grub=0;  #we need to use lilo instead
#$serial='0,9600n8';
#$serialcons='console=ttyS0,9600 CONSOLE=/dev/ttyS0';

#List of networks/targets the bootdisk will handle.
#you don't need to fill this variable if you only
#plan to install dhcp clients.

@networks =
 #   network         domainname             netmask          gateway         broadcast        targets list
  (
#   [ "192.168.32.0" , "prof.my.domain",  "255.255.255.0", "192.168.32.1", "192.168.32.255", "teachers"],
#   [ "192.168.33.0" , "my.domain",       "255.255.255.0", "192.168.33.1", "192.168.33.255", "sysadm_net"],
#   [ "192.168.34.0" , "maths.my.domain", "255.255.255.0", "192.168.34.1", "192.168.34.255", "maths"],
  );

#for each network create the corresponding targets list
#@teachers=qw(teach1 teach2);
#@sysadm_net=qw(adm1 adm2 adm3);
#@maths=qw(maths1 maths2 maths3 maths4);


###########################################################
#                                                         #
# Section: network configuration of the target            #
#                                                         #
###########################################################

#Lines to add to the /etc/hosts of the target computer
#Also added inside the miniroot /etc/hosts.
#$hosts_supp = "192.168.32.1  host1.myfake.domain host1 \n
#192.168.32.45 mynfsserver.myfake.domain mynfsserver \n";

# NFS options for diskless stations or NFS-monted /usr partition
#$nfsopts = "nolock,rsize=8192,wsize=8192";

###########################################################
#                                                         #
# Section: partitioning of the target                     #
#                                                         #
```

```
##########################################################

# If $nowarn (see the interactivity section)
# is defined, you will not be prompted to confirm
# the partioning of the target at install time.

# To install on a specific disk rather than the first detected disk,
# modify and uncomment next line
#$target_disk = "hdb" ;

# For MANUAL partionning set $custom_part to 1
$custom_part = 0;

# Default command to create filesystem (not for swap)
# The filesystem type must be compiled in the kernel you
# put on the bootdisk ($bootkernel)
#
# If you want different filesystem types among the different partition,
# fill the optionnal entry in @autopart_specs or in @manualpart_specs
$mkfs="mkfs -text3"; #used to make the filesystem


#For AUTOMATIC PARTITIONING (not used if $custom_part = 1)
@autopart_specs =
  #mount point     min_size max_size     mkfs(optional)
  ([ "/" ,  "100Mo","100Mo"                       ],
   [ "swap",  "64Mo","128Mo"                       ],
   [ "/usr" ,  "800Mo",        "3000Mo"                     ],
   [ "/var" ,  "200Mo",        "500Mo",      "mkfs -text2 "  ],
   [ "/export/home",  "remaining",    "",           "mkfs -txfs"    ]);


#For MANUAL partionning (not used if $custom_part = 0)
# for instance if you want to keep an existing partition
# or to define the partition sizes manually before starting the install
# process, please gives the affectation of filesystems to partition numbers
# and which ones need to be allocated


@manualpart_specs =
#mount_point partition_number    mkfs (0 to avoid formating)
  ([ "/" ,   1,  "mkfs -treiserfs" ],
   [ "swap" ,  2,  0                 ],
   [ "/usr" ,  3,  0                 ],
   [ "/var" ,  5,  "mkfs -text3"     ],
   [ "/export/home",  6,    0                 ]
  );

#Lines to add to the /etc/fstab of the target machine
#This will not be used if $skip_fstab is used

$fstab_supp = "/dev/fd0 /floppy auto defaults,user,noauto 0 0
#mynfsserver:/usr/local /usr/local nfs ro,defaults,rsize=8192,wsize=8192 0 0
/dev/cdrom /cdrom auto ro,defaults,user,noauto 0 0\n";

##########################################################
#                                                        #
# Section: update rules                                  #
#                                                        #
##########################################################

#To fine tune the list of files/directories which are
#replicated, edit /etc/replicator/update_rules (default rules
#are in /usr/share/replicator/update_rules.default) and uncomment
#the following line
```

```
#$modify_rules_file = 1;

##########################################################
#                                                        #
# Section: miscellaneous                                 #
#                                                        #
##########################################################

# To have an NFS /usr-mounted partition on the target
# $usr_is_nfs=1;

# Choosing the keyboard mapping of the target :
# (see in /usr/share/keymaps/*/*/)

# If you don't set the variable ($keymap) in next section, you can define
# a list of keymaps to choose from at replication time.
# Use the syntax : "keymap-name","description"
# @extra_keymaps=("fr-latin0","french keyboard","us","us keyboard","uk","uk keyboard");

# If you just want the same keymap as on the miniroot-server
# then just leave $keymap and @extra_keymaps undefined.

##########################################################
#                                                        #
# Section: interactivity                                 #
#                                                        #
##########################################################
# If you already know it you can define the keymap of the
# target here. If you want the same keymap as on the model
# put $keymap="same". If $keymap is defined you won't be
# prompted to choose a keymap at the begining of the
# replication
# $keymap="fr-pc";

# If $nowarn is defined, you will not be prompted to confirm
# the partioning of the target at install time.
#$nowarn=1;

# If $no_login is defined, you will not be prompted to login
# into the target. The replication will start immediately.
# WARNING : anybody that can boot on the exported miniroot
# can then start a replication.
#$nologin=1;

# If $autoreboot is defined, the target will automatically reboot
# at the end of the replication process
#$autoreboot=1;

# Note : to have a completely non interactive replication you
# need to have the 4 above variables defined . Anyway if you
# choose to install a dhcp client target you will always be
# prompted for the hostname at the beginning of the replication.

##########################################################
#                                                        #
# Section: Experts Only                                  #
#                                                        #
##########################################################
# Name of the model machine. Change this only if
# the model machine is not the miniroot server
# $model= "mymodel";

# if you want another lilo template than /etc/lilo.conf
#(which will just be prefixed by the
# right boot= and root= command, put it here)
# $lilo_template = "/somewhere/a-lilo-file";
# if you want to manage lilo.conf manually (or in repli-postinst)
# $skip_lilo = 1;

# if you want to completely prevent replicator from installing a bootloader
# (you must do this in repli-postinst then)
# $skip_bootloader = 1;

# if you want to manage /etc/fstab manually (or in repli-postinst)
# $skip_fstab = 1;

# to run from cvs, put the configuration file in /etc/replicator/
# and uncomment this
# $sharedir='./';
# $scriptsdir='./';
# $sbindir='./';

#Do NOT modify the following line AND leave it at the end of this file
$the_end=1;
```