

<http://www.twitter.com/iXsystems>
<http://www.facebook.com/pages/San-Jose-CA/iXsystems-Inc/71393241507>
<http://www.linkedin.com/companies/ixsystems>
<https://plus.google.com/u/0/103239532448324979707>

search
 Support (<http://support.ixsystems.com/>)

FreeBSD Unattended Installation of Servers

[Storage \(/storage/\)](#)
[Servers \(/servers/\)](#)
[Solutions \(/solutions/\)](#)
[What's New \(/whats-new/\)](#)

About iX (/about/)

Posted in [BSD Magazine Article \(/whats-new/category/bsd-magazine-article/\)](#), [Tech Spotlight \(/whats-new/category/tech-spotlight/\)](#) on February 28, 2013

This article tries to show how to manage an important (in terms of size) computing park when talking about the unattended installation and upgrade of FreeBSD servers.

What you will learn

- How to manage the version of FreeBSD running on your machines
- The provisioning or upgrading of each machine in an advantageous way

What you should know

- What PXE means and how it behaves
- Some advanced FreeBSD admin skills

I think this topic is very interesting because it will help you deploy fast provisioning and upgrading of servers (without the need to go through the whole process: buildworld, mergemaster, etc.), having a homogeneous implementation of FreeBSD. For me, particularly, this has been extremely useful in several situations but mainly when I needed to perform an extremely fast upgrade and come back on-line in terms of minutes.

Previous to building our unattended installation system, we need to check if the downloadable ISO images are up to being used for installing servers in an unattended way. I'm saying this, basically, because FreeBSD's generated CDs and its content (packages, sources, doc...) are done per release; after that happens and until the next release ISO images are out, they provide you patches for your installation (in binary or source format) for fixing bugs, but IMHO you can get better, faster and more customizable results if you proceed the way described here.

So first of all, we're going to create our own patched CD images (never with unnecessary changes but yes with the code properly patched and bugs fixed at the time of iso creation) and later will build our unattended installation server.

This short introduction has tried to allow people who have started reading this article to know whether they are interested in this topic or not. I assume people with big computing parks will be interested in continuing the read.

Step 1: Building our own Release

First of all, I'm going to clarify one aspect – the main reason, because I'm going to generate a release in this article. This is, that it is extremely important for you to be able to generate your own upgraded or customized ISO images of the svn branch/tag you need to work with. You could need to customize them for using Sysinstall instead of BSDinstaller or just to have an updated medium for deploying unattended installation services, like the following one (or for upgrading it). You can also need it just for using this ISOs with your managed servers, instead of using a FreeBSD image with some bugs discovered previous to a new release launch (the new upgraded ISOs are not able to be downloaded from ftp.freebsd.org).

So, as I advanced before, at present, the new default FreeBSD installer (BSDinstaller), is undergoing an improvement process and does not support performing an installation in a similar way to Sysinstall with install.cfg config file, so I'm going to generate a custom release for obtaining ISO images with Sysinstall for now.

Let's grab an ISO image from ftp.freebsd.org for setting up our release generation machine. For example, fetch or wget:

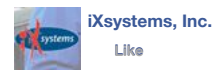


(/truenas/)



(/server-buying-guide/)

Find us on Facebook



Like

1,312 people like iXsystems, Inc.



Facebook social plugin

Reviews

```
ftp://ftp.freebsd.org/pub/FreeBSD/releases/ISO-IMAGES/9.1/FreeBSD-9.1-RELEASE-amd64-disc1.iso.
```

We will proceed with a normal installation, but without installing sources (just do install lib32 and ports). After having booted our installed system, we will continue on to create our source directories:

```
mkdir -p /usr/src
```

```
mkdir -p /datamountpoint/
```

Let's move into the directory in which we're going to 'svn export' the RELENG_9_1 needed data as can be seen in listing 1.

Listing 1. SVN exporting content

```
cd /datamountpoint/
```

```
svn export svn://svn.FreeBSD.org/base/releng/9.1 src_releng91
```

```
svn export svn://svn.freebsd.org/ports/head ports_releng91
```

```
svn export svn://svn.freebsd.org/doc/release/9.1.0/ doc_releng91
```

Now, we should have an up to date collection of sources, ports and documentation, for building our release and ISO images. I'm assuming we're creating a release for a 64 bit capable machine. So, for our purpose, move into the amd64 src directory:

```
cd /datamountpoint/src_releng91/release/amd64
```

Let's continue by slightly modifying mkisoimages.sh as I'm going to describe in listing 2.

Listing 2. Diff of changes applied in mkisoimages.sh

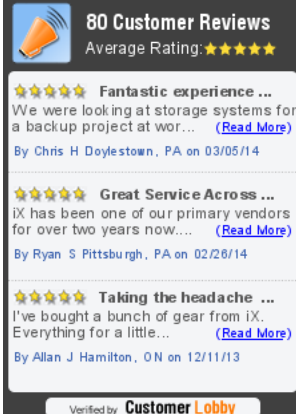
```
diff -u mkisoimages.sh-unmodified mkisoimages.sh
--- mkisoimages.sh-unmodified 2013-01-12 20:06:42.000000000 +0100
+++ mkisoimages.sh 2012-12-30 10:36:09.000000000 +0100
@@ -39,6 +39,6 @@
 LABEL=$1; shift
 NAME=$1; shift

-echo "/dev/iso9660/"`echo $LABEL | tr '[:lower:]' '[:upper:]'` / cd9660 ro
0 0" > $1/etc/fstab
+## echo "/dev/iso9660/"`echo $LABEL | tr '[:lower:]' '[:upper:]'` / cd9660
ro 0 0" > $1/etc/fstab
 makefs -t cd9660 $bootable -o rockridge -o label=$LABEL $NAME $*
-rm $1/etc/fstab
+## rm $1/etc/fstab
```

So now, let's move backwards one level and let's modify Makefile.sysinstall in order to end up like in listing 3.

Listing 3. Diff of Makefile.sysinstall

```
diff -u Makefile.sysinstall Makefile.sysinstall-modified
--- Makefile.sysinstall 2012-10-24 04:20:07.000000000 +0200
+++ Makefile.sysinstall-modified 2013-01-12 20:12:19.000000000 +0100
@@ -1,4 +1,4 @@
-# $FreeBSD$
+# $FreeBSD$
+# $FreeBSD: release/Makefile.sysinstall 241979 2012-10-24 02:20:07Z kensmith $
#
# make release [BUILDNAME=somename] CHROOTDIR=/some/dir CVSR00T=/cvs/dir \
# [RELEASETAG=tag] [SVNR00T=svn://svn.freebsd.org/base] \
@@ -1149,21 +1149,11 @@
```



80 Customer Reviews
Average Rating: ★★★★★

★★★★★ **Fantastic experience ...**
We were looking at storage systems for a backup project at wor... [\(Read More\)](#)
By Chris H Doylestown, PA on 03/05/14

★★★★★ **Great Service Across ...**
iX has been one of our primary vendors for over two years now... [\(Read More\)](#)
By Ryan S Pittsburgh, PA on 02/26/14

★★★★★ **Taking the headache ...**
I've bought a bunch of gear from iX. Everything for a little... [\(Read More\)](#)
By Allan J Hamilton, ON on 12/11/13

Verified by **Customer Lobby**

(<https://www.customerlobby.com/reviews/204inc/>)

Archive

2014

November (/whats-new/archive/2014/11/) (2)
October (/whats-new/archive/2014/10/) (5)
September (/whats-new/archive/2014/9/) (3)
August (/whats-new/archive/2014/8/) (3)
July (/whats-new/archive/2014/7/) (2)
June (/whats-new/archive/2014/6/) (4)
May (/whats-new/archive/2014/5/) (8)
April (/whats-new/archive/2014/4/) (4)
March (/whats-new/archive/2014/3/) (7)
February (/whats-new/archive/2014/2/) (6)
January (/whats-new/archive/2014/1/) (1)

2013

December (/whats-new/archive/2013/12/) (3)
November (/whats-new/archive/2013/11/) (8)
October (/whats-new/archive/2013/10/) (8)
September (/whats-new/archive/2013/9/) (12)
August (/whats-new/archive/2013/8/) (7)
July (/whats-new/archive/2013/7/) (4)
June (/whats-new/archive/2013/6/) (9)
May (/whats-new/archive/2013/5/) (5)
April (/whats-

```

FreeBSD_Install \
  ${CD}/FreeBSD-${BUILDNAME}-${TARGET}-disc1.iso      {CD_DISC1}
\
  ${CD_DISC1_PKGS}
-   @sh ${CURDIR}/${TARGET}/mkisoimages.sh \
-   FreeBSD_Packages \
-   ${CD}/FreeBSD-${BUILDNAME}-${TARGET}-disc2.iso ${CD_DISC2} \
-   ${CD_DISC2_PKGS}
.   if defined(MAKE_DVD)
-   @sh ${CURDIR}/${TARGET}/mkisoimages.sh ${BOOTABLE} \
-   FreeBSD_Install \
-   ${CD}/FreeBSD-${BUILDNAME}-${TARGET}-dvd1.iso ${CD_DVD1} \
-   ${CD_DVD1_PKGS}
+   @echo "No DVD1 needed..."
.   endif
.   if !defined(NODOC)
-   @sh ${CURDIR}/${TARGET}/mkisoimages.sh \
-   FreeBSD_Documentation \
-   ${CD}/FreeBSD-${BUILDNAME}-${TARGET}-disc3.iso ${CD_DOCS} \
-   ${CD_DOCS_PKGS}
+   @echo "No doc iso image needed..."
.   endif
.   if defined(SEPARATE_LIVEFS)
-   @sh ${CURDIR}/${TARGET}/mkisoimages.sh ${BOOTABLE} \

```

So, at this point we have the fresh code recently downloaded from FreeBSD's subversion properly patched for generating our customized release that uses Sysinstall as the installer.

Now, we should copy all the content to /usr/src. So, we could do:

```
rsync -av /datamountpoint/src_releeng91/ /usr/src/
```

And the following step is to buildworld in order to be able to generate release later.

```
cd /usr/src
make buildworld
```

Now let's launch the release generation make as shown in listing 4.

Listing 4. Making release
cd release

```
make -f Makefile.sysinstall release CHROOTDIR=/datamountpoint/release_generation
EXTSRCDIR=/datamountpoint/src_releeng91 EXTPORTSDIR=/datamountpoint/ports_releeng91
EXTDOCDIR=/datamountpoint/doc_releeng91 MAKE_ISOS=1
```

NOTE

If instead of generating the release specifying the makefile Makefile.sysinstall, and without having applied previous changes, we would obtain a fresh release with the new installer (BSDinstaller).

If all went OK, We will have the ISO images can be seen in listing 5.

Listing 5. Recently generated ISO images of the new release
ls -la /datamountpoint/release_generation/R/cdrom/

```

-rw-r--r--  1 root  wheel   67524608 Dec 30 17:47 FreeBSD-9.1-20121230-SN
AP-amd64-bootonly.iso
-rw-r--r--  1 root  wheel   508581888 Dec 30 17:48 FreeBSD-9.1-20121230-SN
AP-amd64-disc1.iso
-rw-r--r--  1 root  wheel          253 Dec 30 17:49 FreeBSD-9.1-20121230-SN
AP-amd64-iso.CHECKSUM.MD5
-rw-r--r--  1 root  wheel          358 Dec 30 17:49 FreeBSD-9.1-20121230-SN

```

new/archive/2013/4/) (4)
March (/whats-new/archive/2013/3/) (5)
February (/whats-new/archive/2013/2/) (7)
January (/whats-new/archive/2013/1/) (5)

2012

December (/whats-new/archive/2012/12/) (4)
October (/whats-new/archive/2012/10/) (1)
September (/whats-new/archive/2012/9/) (3)
August (/whats-new/archive/2012/8/) (3)
July (/whats-new/archive/2012/7/) (1)
May (/whats-new/archive/2012/5/) (2)
April (/whats-new/archive/2012/4/) (2)
March (/whats-new/archive/2012/3/) (5)
February (/whats-new/archive/2012/2/) (2)
January (/whats-new/archive/2012/1/) (2)

2011

December (/whats-new/archive/2011/12/) (1)

Feeds

RSS (/whats-new/feeds/rss/) /
Atom (/whats-new/feeds/atom/)

```

AP-amd64-iso.CHECKSUM.SHA256
-rw-r--r-- 1 root wheel 422400000 Dec 30 17:49 FreeBSD-9.1-20121230-SN
AP-amd64-livefs.iso
drwxr-xr-x 3 root wheel 512 Dec 30 17:47 bootonly
drwxr-xr-x 4 root wheel 512 Dec 30 17:47 disc1
drwxr-xr-x 2 root wheel 512 Dec 30 17:47 disc2
drwxr-xr-x 2 root wheel 512 Dec 30 17:47 docs
drwxr-xr-x 18 root wheel 1024 Dec 30 17:47 dvd1
drwxr-xr-x 17 root wheel 512 Dec 30 17:47 livefs

```

Step 2. Installation of the FreeBSD Unattended Installation Server

Let's continue by performing a new server installation (our unattended installing server) with our recently created ISO named FreeBSD*****-SNAP-amd61-disc1.iso. We will select a Standard installation, with a custom distribution set (please, select Custom) which contains the following parts: base, kernels (all), info, lib32, man, src, ports, and local. Ensure you say to use cd as media (as source).

After the installation process, let's install a dhcpd server which will be the ip allocator in our PXE server. We will use isc-dhcp41-server. For building:

```
cd /usr/ports/net/isc-dhcp41-server
```

```
(let's unselect all options)
```

```
make install clean
```

Now, let's configure our dhcpd server by setting /usr/local/etc/dhcpd.conf as in listing 6.

Listing 6. Configuracion of /usr/local/etc/dhcpd.conf

```

allow booting;
allow bootp;
authoritative;
option domain-name "freebsdpxe.sarenet.es";
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;
ddns-update-style none;
log-facility local7;
local-address 10.0.0.1;
subnet 10.0.0.0 netmask 255.255.255.0 {
    range 10.0.0.70 10.0.0.80;
    next-server 10.0.0.1;
    filename "boot/pxeboot";
    option root-path "/datamountpoint/netboot/freebsd91";
}

```

At this point, we don't need to launch the dhcpd server. Now, let's configure /etc/inetd.conf in order to enable tftpd. We need to make the tftp line appear like this:

```

(in a single line)
tftp dgram udp wait root /usr/libexec/tftpd tftpd -l -s /datamountpoint/netboot/freebsd91

```

We will use NFS as our install media (data source) for Sysinstall on the unattended installations. So /etc/exports in our pxe server should look like this:

```
/datamountpoint -alldirs,ro -network 10.0.0 -mask 255.255.255.0
```

Yes, read-only... nothing should be written to our NFS exported content.

Now let's populate /datamountpoint/netboot/freebsd91 with our recently generated release's disc1:

```
tar -C /datamountpoint/netboot/freebsd91 -pxvf FreeBSD-9.1-20121230-SNAP-amd64-disc1.iso
```

We should now set the loader.conf for our unattended installations service properly in /datamountpoint/netboot/freebsd91/boot/loader.conf:

```
mfsroot_load="YES"
mfsroot_type="mfs_root"
mfsroot_name="/boot/mfsroot"
vfs.root.mountfrom="ufs:/dev/md0"
```

Let's follow by decompressing what will become the mfsroot of our netbooted installation OS and let's copy to it's root (optionally of course) the install.cfg in order to have Sysinstall perform all automated tasks that belongs to it.

NOTE

Install.cfg generation and syntax is beyond the scope of this article and will not be covered.

```
cd /datamountpoint/netboot/freebsd91/boot
```

```
gzip -d mfsroot.gz
```

Now, we'll mount the memory disk in order to be able to copy the install.cfg to it's root :

```
mkdir /onerandommountpoint
```

```
mdconfig -a -t vnode -f
/datamountpoint/netboot/freebsd91/boot/mfsroot -u 200
```

I have used the number 200 but it's really optional, so you could certainly not specify -u ___ and it will be attached to the first free kernel memory disk in numerical order.

```
mount /dev/md200 /onerandommountpoint
```

```
cp /placewherewehaveourcustominstallcfg/install.cfg /onerandommountpoint
```

Now let's unmount the attached mfsroot.

```
cd
```

```
umount /onerandommountpoint
```

After ensuring it's properly unmounted, let's remove our attached kernel memory disk.

```
mdconfig -d -u 200
```

We ensure it's removed, by typing the following command and seeing no output:

```
mdconfig -l
```

Now finally, let's configure our pxe server's /etc/rc.conf to launch all services automatically as in listing 7.

Listing 7. /etc/rc.conf.

```
## fxp0 is the PXE-Boot interface
ifconfig_fxp0="inet 10.0.0.1 netmask 255.255.255.0"
```

```
## PXE services
dhcpd_enable="YES"
dhcpd_ifaces="fxp0" (serve dhcp leases just using this interface)
inetd_enable="YES"
rpcbind_enable="yes"
mountd_enable="yes"
nfs_server_enable="yes"
```

Now important, without doing this the tftp server won't be able to serve pxeboot under /datamountpoint/netboot/freebsd91/boot/ and the nfsd would also run into troubles:

```
chmod -R 755 /datamountpoint/netboot/freebsd91
```

Conclusion

That's all :). Now, to install or upgrade a new server, you can connect one PXE capable server's Ethernet port with a crossover cable to our recently built PXE server (to it's PXE interface) in order to boot from net and load a FreeBSD installation. I've just used this system for upgrading or installing one server at a time, but it should work properly as well with some more.

As this is my first article in BSD Magazine many thanks to all for reading it!

Bio

I'm a sysadmin and system's programmer at Sarenet (www.sarenet.es) and am very proud to be able to contribute with this article, because the Open Source community, documentation and software are basically the most powerful strength in the computing world. I wanted to dedicate this work to all my family, but specially to my grandmother who has very recently left us. I wanted also to give special thanks to all Sarenet's people, because working with them, is a really nice experience. You could send me you're questions or comments to egoitz@sarenet.es and I'll be happy and proud to answer or clarify whatever is needed.

Share on Twitter (<http://twitter.com/home?status=http%3A//www.ixsystems.com/whats-new/freebsd-unattended-installation-of-servers/%20FreeBSD%20Unattended%20Installation%20of%20Servers%20>)

Share on Facebook (<http://facebook.com/sharer.php?u=http://www.ixsystems.com/whats-new/freebsd-unattended-installation-of-servers/&t=FreeBSD%20Unattended%20Installation%20of%20Servers%20>)

← [What's Around the Corner: A Look at Upcoming PC-BSD Changes \(/whats-new/whats-around-the-corner-a-look-at-upcoming-pc-bsd-changes/\)](#)

[iXsystems Debuts Exclusive ZFS Encryption Feature in FreeNAS 8.3.1](#) → ([/whats-new/ixsystems-debuts-exclusive-zfs-encryption-feature-in-freenas-831/](#))

[Storage \(/storage/\)](#) · [Servers \(/servers/\)](#) · [Support \(/solutions/support/\)](#) · [What's New \(/whats-new/\)](#) · [Downloads \(/about-ix/downloads/\)](#) · [About iX \(/about-ix/\)](#) · [Partners \(/about-ix/partners/\)](#) · [Our Clients \(/clients/\)](#) · [Privacy Policy \(/privacy-policy/\)](#)

Copyright © iXsystems, Inc. (<http://www.iXsystems.com>)



(<http://www.customerlobby.com/reviews/20407/ixsystems/>)

VMware and VMware Ready is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions.

