

Automating the deployment of FreeBSD & PC-BSD® systems

by

Kris Moore
kris@pcbsd.org

In PC-BSD 9.x every installation is fully-scripted, due to the the **pc-sysinstall** backend. This backend can also be used to quickly automate the deployment of FreeBSD servers and PC-BSD desktops using a PXE boot environment. In PC-BSD & TrueOS™ 9.1 and higher, this functionality is easy to setup and deploy using the “**pc-thinclient**” utility. PXE booting allows you to boot systems via the LAN interface, as opposed to using traditional media, such as DVD or USB. In order for clients to boot via PXE they will need a PXE capable network adapter.

The initial PXE setup

To get started, you will need to have a system with two network interfaces running PC-BSD or TrueOS 9.1 and a complete ports tree in /usr/ports. If you do not have the ports tree installed, you can download it by running the command “**portsnap fetch extract update**” as root. With these pieces in place, open a root prompt and run the “**pc-thinclient**” command. The first screen you see will look something like this:

```
root@pxehost:/root # pc-thinclient
/usr/local/bin/pc-thinclient will install the components to convert this system
into a thin-client server.
Continue? (Y/N) █
```

Enter “y” to continue, and the following screen will be shown:

```
Do you wish to make this a remote X desktop server or install server?
(r/i) █
```

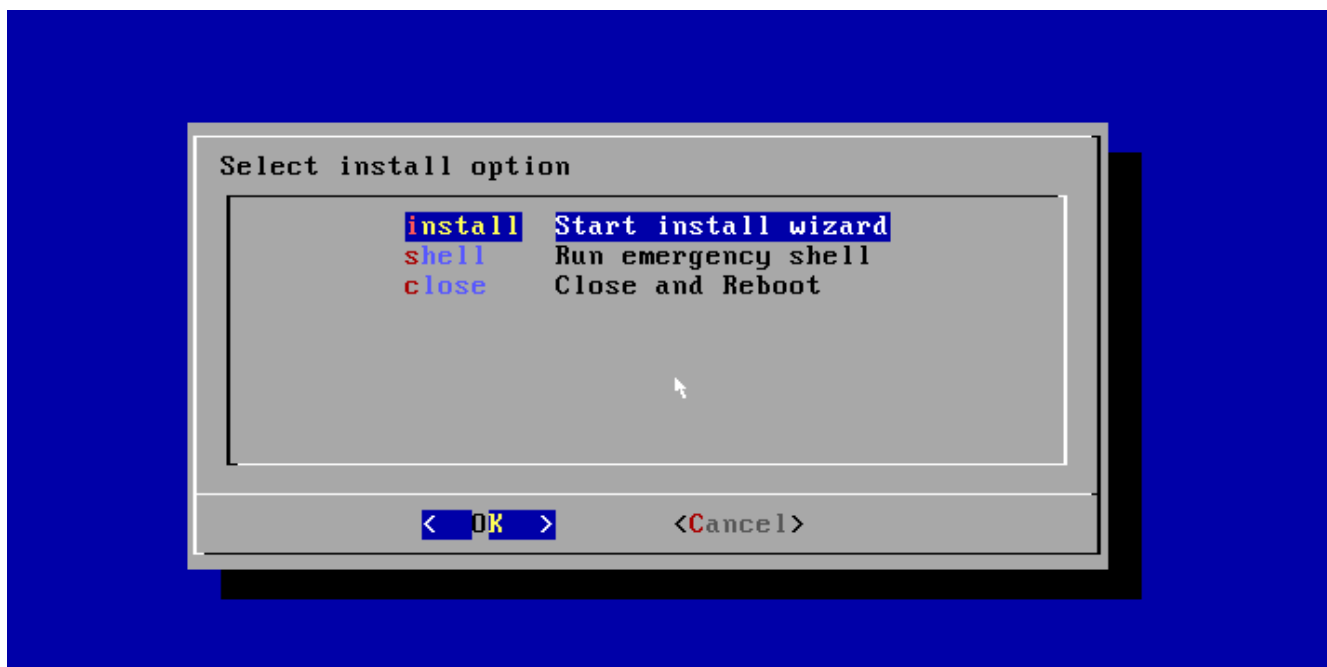
In this case you are going to be setting up a PXE installation server, so enter “i” to continue. (The “r” option can be used to make your system a X thin-client server. More information about this can be found on the wiki page at the end of the article.) After selecting your type of PXE system, the thin-client wizard will then begin to build the “**net/isc-dhcp42-server**” port. You will most likely only need the default port options, and can hit enter at any confirmation screens that appear. Once the port has finished installing, the thin-client setup will prompt you again for the PXE network interface to use:

```
Setting up system for PXE booting...
What NIC do you wish DHCPD to listen on? (I.E. re0)
nic) █
```

Enter the interface name of the network card you wish to use as the PXE interface. This interface will be running the DHCP server, and should not be connected to an existing network with another DHCP server running. The wizard will then finish up the configuration for PXE booting, and display a message similar to this:

```
To perform system installations, place your custom pc-sysinstall scripts in:  
/usr/home/thinclient/installscripts  
  
An example script is provided in the above directory  
  
For unattended installations, save your pc-sysinstall script as:  
/usr/home/thinclient/installscripts/unattended.cfg  
  
Your system is now setup to do PXE booting!
```

Your initial PXE setup is now complete! You may now try to PXE boot a client connected on the network interface you specified. If the client boots successfully, you should be presented with an installation screen like below:



By selecting the “install” option you will be presented with the example configuration option, which can be used to install a basic FreeBSD system. (Below we will look at adding your own) By selecting it, you will be asked to confirm one last time that you wish to perform the installation using this configuration, and then the installation will proceed. In addition to performing installs, you can also access an emergency shell prompt. This can be very useful if you have a system which can no longer boot, and you wish to access the disk or attempt repairs of some kind.

Customizing the installation scripts

With your initial PXE configuration finished, you will now most likely want to create your own installation scripts. The thin-client wizard creates an example installation script in the

`/usr/home/thinclient/installscripts/` directory, which is where you will want to place your custom scripts as well. Any scripts placed in this directory will be selectable as an installation option on the PXE client.

With the location of the install scripts in hand, lets now take a look at the provided `pc-sysinstall.example` file in that directory.

```
# Sample configuration file for an installation using pc-sysinstall

installMode=fresh
installInteractive=no
hostname=examplesystem

# Set the disk parameters
disk0=ada0
partition=all
bootManager=none
commitDiskPart

# Setup the disk label
# All sizes are expressed in MB
# Avail FS Types, UFS, UFS+S, UFS+J, ZFS, SWAP
# Size 0 means use the rest of the slice size
disk0-part=UFS+SUJ 1000 /
disk0-part=SWAP 2000 none
disk0-part=UFS+SUJ 0 /usr
commitDiskLabel

# Set if we are installing via optical, USB, or FTP
installType=FreeBSD
installMedium=local
localPath=/installarchive
packageType=tar
installFile=fbsd-release.txz

# Set the root pass
rootPass=root

# Setup our users
# Setup our users
userName=kris
userComment=Kris Moore
userPass=kris
userShell=/bin/csh
userHome=/home/kris
userGroups=wheel,operator
commitUser
```

The included comments in the example make most of the functionality fairly self-explanatory. However there are a few sections we will take a closer look at customizing. The first place to note is the “`installMedium=local`” and “`localPath=/installarchive`” options. The `pc-sysinstall` backend supports a

number of methods of fetching the archive files for installation. In this case we are using the local file “**fbbsd-release.txz**”, stored in **/usr/home/thinclient/installarchive**. This directory will appear to PXE clients as **/installarchive**, allowing you on the host OS to supply your own tar archives of any FreeBSD / PC-BSD release or configuration you wish.

PRO Tip!

By using the **Warden** utility in PC-BSD / TrueOS, it is possible to setup your install environment inside a jail on the host system. Then when you are satisfied with the configuration, you can stop the jail and create the install archive on the host system with the “**tar**” command as shown below:

```
# tar cvJf /usr/home/thinclient/installarchive/myarchive.txz -C /usr/jails/<jailip> .
```

After selecting your installation archive options you will most likely want to customize the disk layout of the new system. In the example provided we are performing a very simple full-disk installation to the disk **ada0**, using UFS with Soft Updates + Journaling. The resulting installation will contain a 1000MB root partition, a 2000MB swap, and use the rest of the disk space for **/usr** as seen below.

```
# Set the disk parameters
disk0=ada0
partition=all
bootManager=none
commitDiskPart

# Setup the disk label
# All sizes are expressed in MB
# Avail FS Types, UFS, UFS+S, UFS+J, ZFS, SWAP
# Size 0 means use the rest of the slice size
disk0-part=UFS+S+UJ 1000 /
disk0-part=SWAP 2000 none
disk0-part=UFS+S+UJ 0 /usr
commitDiskLabel
```

In addition to UFS support, the **pc-sysinstall** backend can also handle more advanced ZFS disk layouts in a similar manner:

```
# Setup the disk label
# All sizes are expressed in MB
# Avail FS Types, UFS, UFS+S, UFS+J, ZFS, SWAP
# Size 0 means use the rest of the slice size
disk0-part=ZFS 0 /,usr,var,root (mirror: ada1)
commitDiskLabel
```

In the example above we have changed our disk layout from UFS to a single ZFS pool using the entire disk, and added the disk drive **ada1** to the resulting **zpool** as a mirror device. While this example is fairly simplistic much more complex layouts can be created using ZFS, including **raidz**, **dataset**

options and more.

PRO Tip!

Since every single install with pc-sysinstall is fully scripted, a good way to experiment with alternative ZFS layouts is using a virtualization tool such as VirtualBox and your PC-BSD installation DVD. By running the install GUI, you can customize your ZFS disk layout in an easy-to-use manner inside a virtual machine. Then when the installation is finished, a copy of the saved pc-sysinstall configuration file will be saved onto the installed system at `/root/pc-sysinstall.cfg`.

One last feature to look at with the thinclient utility is the ability to perform completely unattended installations. If you plan on only doing fully-automated installations via this PXE interface, you may do so simply by creating a configuration script named “**unattended.cfg**” and placing it in the `/usr/home/thinclient/installscripts/` directory alongside the example. When a client first boots it will check for this **unattended.cfg** file, and if found it will automatically use it for installation. Some caution should be taken when using this method, since simply plugging a PXE booting system into the wrong LAN cable could cause it to be re-installed.

Conclusion

We have briefly looked at some of the ways PC-BSD and TrueOS 9.1 make it easy to setup PXE booting and perform rapid deployment of FreeBSD based systems. We have also just begun to scratch the surface of the type of installations that can be performed with the pc-sysinstall backend. If you want to do further research into more complex installations please take a look at the included wiki references on pcbsd.org. We also will be happy to continue the discussion with you on the PC-BSD mailing lists or forums.

References

Wiki Articles:

http://wiki.pcbsd.org/index.php/Thin_Client

http://wiki.pcbsd.org/index.php/Creating_an_Automated_Installation_with_pc-sysinstall

Mailing Lists & Forums

<http://lists.pcbsd.org>

<http://forums.pcbsd.org/>