JCATS Diskless Terminal Setup

Setting up diskless terminal workstations using JCATS 7.1.x and Red Hat Enterprise Linux 4.x.

22.08.2008, version 1.5

Table of Contents

0. Introduction

This manual documents setting up a Red Hat Enterprise Linux (RHEL) 4.4 server and diskless terminal workstations to run JCATS version 7.1.3. Although not tested, the procedures will most likely work for subsequent RHEL 4.x and JCATS 7.1.x versions. For other versions and setups, this guide might still be of use as reference and source of information.

While effort is made to explain the theory behind all configurations and commands, it is assumed that the reader is proficient with Linux in general and RHEL in particular. The setup is unsupported by the author, and does not constitute or imply endorsement by JCATS or Red Hat proponents. During utlization of this manual, you will have to resolve problems on your own. You are encouraged to send in their observations and suggestions. Suggestions and noted problems encountered while using this manual will assist to enhance the document.

The manual is divided into four main sections:

0. Background information on terminals.
1. Instructions and explanations for setting up a terminal solution.
2. List of commands.
3. Questions and Answers (Q&A).

1. About the author

Lemmit Kaplinski is currently assigned as a Systems Administrator for the Estonian Simulation Center, Estonia. The author can be contacted through email: lemmit.kaplinski@ksk.edu.ee.

2. Acknowledgements

The author would like to thank the following persons for their contributions:

- Mr. Elmo Kuslap for the cover photograph and for testing.
- Mr. Rain Sirendi for testing.
- Mr. Steven Edwards for proofreading.

3. Manual conventions

The manual consists of:

- Explanations. Explanations use the following style:

  This is an explanation

- Code and commands. Code and commands must be typed exactly as they appear. Commands use the following style:

  ```
  echo ”Hello World“
  ```

  NOTE: ALL COMMANDS HAVE TO BE EXECUTED AS THE ROOT USER.

  Sometimes long lines will not fit horizontally in the manual. In this case they have been split with the backslash character \. When entering these lines into

configuration files do not enter the backslash and do not split the line. If your terminal is not wide enough, they might be automatically wrapped, but must not contain a newline character. Here is an example of a long line:

```
This is a long line that will not fit horizontally in the manual, \
but must be entered on one line in a configuration file.
```

- Warnings. Warnings are indicated by the ”binary hazard“ sign:


This is a warning

4. About terminals

Terminal is a rather ambiguous term, which can refer to anything from an office workstation to a device or system for processing credit card payments. For clarity, throughout this manual the term “terminal“ refers to a client workstation that:

1. Is in most aspects a standard IBM-PC compatible device. Usually terminals have a smaller form-factor and less powerful components than standalone workstation computers. Nothing prevents one from using a standard PC as a terminal though.
2. Has no storage devices, or the storage devices are not used, and relies on the server to provide the operating system, applications and data files. The terminal must still have a way to initialize the hardware and boot the operating system. This is done with the help of BIOS or similar firmware.
3. Executes applications locally, using its own processor, RAM and other resources. In many instances, terminals only provide a graphical environment and applications are executed in a central server. JCATS does not support this setup however, and thus this benefit of terminals is not available.

Terminals have, depending on the setup, several advantages over standalone workstations. In the specified configuration the advantages are as follows:

- Hard drives have moving parts and they will fail sooner or later. They also generate noise and heat. Using resources over a network connection makes hardware errors less likely to occur and results in a better environment for the operator.
- All software and data files needed by terminals are stored on the server. This makes central management possible without any custom or third-party add-ons. Updating software, modifying configurations and other tasks most likely involves updating or changing just one set of files.
- If terminal devices used are physically identical or very similar, faulty devices can be swapped for backup devices with minimal or no changes in a very short time. No additional installation is necessary. This gives the administrator breathing space to deal with the problem while the operator can continue her activities unhindered.

  Same applies to adding new terminals.
- Cost benefits. Devices capable of operating as terminals can be considerably cheaper than standalone workstations. A great source for terminals can be previous generation PC-s that have hard drives removed and some components (most likely RAM and video card) updated.

Terminals also have weaknesses or considerations. These include:

- Complicated and unsupported setup.

  While properly configured terminals are easier to administer from day to day, doing the initial setup requires the proper configuration of several services, is error-prone and sometimes hard to debug. When problems arise, you have to be familiar with a wide array of concepts, from BIOS procedures to oddities of the Linux operating system.

- Higher demands on network and server infrastructure.

  Loading the entire operating system over the network requires more bandwidth. It is recommended that you should consider (as a minimum) 100Mbps properly installed CAT 5e cabling, workstations connected to a switch (no hubs) and server to a 1000Mbps uplink port. This setup will definitely be sufficient for 20+ terminals.

  In this setup, server becomes a single point of failure and extra care should be taken to minimize data loss and network connectivity problems. Using at least two hard drives in RAID1 configuration is highly recommended. Also having two network connections to the switch would provide better redundancy. Purchasing a backup server and configuring it with identical settings as the first, would guarantee quick restoration of services in case of errors and again provide a secondary resolution for the administrator. Ideally the servers are identical and software setup, including the operating system, can be cloned.

  Overall, the requirements are obtainable and once the terminals have loaded the operating system, load on the network will decrease considerably.

- Limited upgrade options.

  When using dedicated terminal hardware, your options with regard to upgrades may be limited. Small form-factors might mean proprietary or little-used connections and expansion cards. Processor speeds and available RAM might be limited and so on. Before investing in terminals, research your options.

5. Terminal start-up sequence

To properly administer and troubleshoot a terminal setup, you must understand the sequence of events that boot up the terminal and the services involved. The following gives a broad overview of these sequential events. Although the next chapter provides some details and step-by-step instructions for configuring the process, you should familiarize yourself with all steps and required software involved to accommodate your specific needs.

1. BIOS initialization.

   After turning on the terminals, integrated BIOS performs a quick Power On Self Test (POST) and initializes the hardware.

2. PXE (Pre Boot Execution Environment).

   You will have to enable the PXE capabilities of your network adapters. Once BIOS finishes initialization, it executes the PXE firmware embedded in the network adapter. PXE does a standard DHCP query and the server replies with an IP address and location of boot loader software.

3. Fetching the bootloader.

   PXE uses the TFTP protocol to retrieve the boot loader pxelinux from the location

specified in the DHCP response from step 2.. The software is similar to GRUB or LILO bootloaders but is optimized for network booting. After fetching pxelinux, the software is executed.

4. Fetching the kernel.

   pxelinux consults a server-side configuration file and fetches the kernel (and initial ramdisk with critical device drivers and startup scripts) specified. After successfully retrieving the files, pxelinux executes the kernel with several options from the configuration file.

5. Kernel execution.

   Once pxelinux fetches the kernel, it is executed. Hardware is detected and, if drivers are available at this stage, configured for operation. The root directory (shared from the server) and initial ramdisk are mounted and system startup script /disklessrc is executed. Note that the root directory is mounted read-only and shared between all terminals.

6. /disklessrc script.

   The /disklessrc script does several things, but its most important function is to mount a snapshot directory from the server in read-write mode and map several important system files and directories to locations on this mount. This is done using bind mounts and explained more thoroughly in Chapter 1. The result is that the system can use these files as if they were local and applications do not need to be modified to work in a diskless, server-mounted setup. While the root directory is read-only and shared between terminals, this snapshot directory is unique to each terminal and allows per-terminal differences in configuration.

7. Normal start-up.

   Once the system has finished all terminal-specific start-up procedures it continues with the execution of /sbin/init and from there on the standard initscripts. Please refer to RHEL documentation for procedural descriptions.

   The start-up concludes with launching of the graphical environment X, and from there on the terminal is ready to launch a JCATS client.

1. Setting up a server and terminals

The manual refers to RHEL 4.4 and JCATS 7.1.3, but is probably similar for other RHEL 4.x and JCATS 7.1.x versions.

When installing, follow the steps as outlined in the “JCATS System Administrator's Guide“, Chapter 3-1. There are a few changes and considerationoons to the aforementioned installation procedure though

1. Install Red Hat Linux

1.1. Network configuration.

This server will assign IP addresses to terminal workstations. To minimize confusion (and provide better bandwidth), you should use a separate subnet connected to a dedicated switch. This documentation assumes your server has an IP address of 10.0.0.1/24 and terminals will be assigned addresses from the range 10.0.0.101-200/24.

Additionally, if you are using a separate and isolated subnet with no access to outside, you must still insert a gateway address. JCATS will lag and hang if this address is not defined. If you do not have a gateway address, use the servers IP address.

1.2. Package selection.

When selecting package groups, open the “Network Servers“ group and select dhcp.

Open the “Legacy Network Server“ group and select tftp. Deselect other packages in this group.

1.3. After finishing the installation, log in as root user, insert the 4th installation CD, mount it and install the package busybox-anaconda. You can do this by executing the following commands:

```
mount /media/cdrom
rpm -Uvh /media/cdrom/RedHat/RPMS/busybox-anaconda-1.00.rc1-5.i386.rpm
eject cdrom
```

NOTE: This package was not found in any package groups. (If you know how to select this package during installation, please notify the author for inclusion in updates)

2. Install JCATS

Follow the instructions in the “JCATS System Administrator's Guide“ Chapter 4-1. You can use the quick start script.

3. Disable NIS (optional)

Once JCATS is installed you might want to disable NIS (Yellow Pages). While this is not strictly necessary, NIS adds complexity to user accounts management and is a possible point of failure. If and only if the server only services terminals (that is clients that boot and mount system directories from the server), all account, host and other relevant data is

shared as it is and NIS is not needed. Based on this information, it is recommended that you disable it.

If the server has standalone clients, they need to access shared data and you should use NIS. Disabling NIS client services for terminals if still recommended though.

This manual assumes you turn off NIS.

3.1. Disable the NIS services by executing the following commands:

```
chkconfig --level 35 ypbind off
chkconfig --level 35 ypserv off
chkconfig --level 35 yppasswdd off
```

3.2. Disable NIS lookups by editing the file /etc/nsswitch.conf. Find the part that looks like:

```
passwd: files nis
shadow: files nis
group: files nis
hosts: files nis dns
services: files nis
```

Delete the phrase nis from every line (do not touch other lines that have nisplus in them). The lines should look like:

```
passwd: files
shadow: files
group: files
hosts: files dns
services: files
```

4. Configure hostnames, DHCP, TFTP

For the terminals to operate properly, they must get an IP address from the server. The IP address should be tied to the MAC address of the terminal, so that it gets the same IP address every time. A more elegant solution is to set up DNS with dynamic updates, but that is currently left as an exercise to the reader.

Additionally the DHCP server will give terminals information they need to boot - TFTP server address and location of the network bootloader (pxelinux).

Also all terminals must have an entry in system hosts file.

Last the TFTP service must be running, so the terminals can fetch both the bootloader and kernel from server.

4.1. Configure the DHCP server

Open the file /etc/dhcpd.conf, delete the existing few lines and enter the following:

```
authoritative;
ddns-update-style none;
subnet 10.0.0.0 netmask 255.255.255.0 {
  option            routers 10.0.0.1;
  option            subnet-mask 255.255.255.0;
  option            domain-name „simctrest“;
  next-server       10.0.0.1;
  filename          „linux-install/pxelinux.0“;
}
```

Next enter a host section for every terminal you have. You will have to know the MAC addresses of all terminals. The host entries must be within the subnet declaration (that is inside the angular brackets). For every terminal enter the following lines, where you replace “ws101“ with the terminals hostname, “00:60:72:10:61:A2“ with its MAC address and “10.0.0.101“ with its IP address:

```
host ws101 {
  hardware ethernet       00:60:72:10:61:A2;
  fixed-address           10.0.0.101;
}
```

4.2. Launch the DHCP server

To have the DHCP server start automatically at next boot, enter the following command:

```
chkconfig --level 35 dhcpd on
```

To launch the server immediately, run the following script:

```
/etc/init.d/dhcpd start
```

4.3. Configure system hosts file

Do this step only if you have not entered the hostnames during JCATS installation.

If you do not have a DNS server, all hostname lookups are done from the file /etc/hosts. This file must have an entry for every terminal. Open the file and for every terminal you configured in the dhcpd.conf file, enter the following line:

```
10.0.0.101   ws101.simctrest     ws101
```

4.4. Activate the TFTP service by editing the file /etc/xinetd.d/tftp and replacing the line

```
disable      = yes
```

with the line

```
disable      = no
```

Now you need to reload the service xinetd for the change to take effect. Execute the script

```
/etc/init.d/xinetd reload
```

5. Set up an environment for terminals

The following procedures are based on the official RHEL procedures for setting up diskless workstations. You can read them online at http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/en-US/System_Administration_Guide/Diskless_Environments.html

As a forewarning, it is recommended that you not try to follow those instructions to the letter. They are rather vague, contain some errors and result in a setup that is hard to update and manage. You should, however, browse through the cited document to get a general understanding for the whole process.

The theory behind terminals is that they will use NFS to mount all system directories from the server in read-only mode. These system directories are shared between all terminals. They will also mount a separate network share in read-write mode and this share is specific to each terminal. They use this share to keep files they need write access to.

Note that the /tmp directory can be located on the read-write share, but to minimize

network traffic, it is described how to set up a ramdisk based /tmp instead. If your your terminals have limited memory, you can place /tmp on NFS share to save some memory.

After mounting system shares from the server, the terminals function as a standard workstation. They will mount /home with JCATS data from the server and launch jcatsd from xinetd as needed.

5.1. Create the terminal system directory

This is the shared, read-only directory that contains most Red Hat system files. Create the directory with the following command:

```
mkdir -p /diskless/root
```

Populate the directory with a copy of RHEL, essentially all the files that JCATS needs to run properly. Because there are literally thousands of them, it is easy to just copy the whole RHEL directory structure, which is the official recommendation of Red Hat. This will of course work, but will give you a setup very difficult to update. This might be desirable when you want to support legacy JCATS on a new RHEL installation, in which case you should follow the Red Hat procedures (don't forget to skip /dev, /proc, /sys and other virtual filesystems, see below). This manual assumes you want to keep both Red Hat and JCATS up-to-date and will document a procedure that results in a system that is far easier to manage, because most system files are shared not only between terminals themselves, but between server and terminals as well. Updates to the server will automatically manifest themselves for terminals.

To do this, use bind mounting - which makes one directory or file in the filesystem available in another location as well. To understand this, think of email aliases – there are several addresses that are all delivered to same mailbox. In the case of bind mounts, a file might be accessible from both /etc/passwd and /diskless/root/etc/fstab. But it is the same file and modifications done using one path (address) are there when when opening the file using the other path. This example also explains why NIS if not needed in this setup.

5.2. Create mountpoints for the system directories

Before doing the bind mounts, you need to create mountpoints for them. You also need to create mountpoints for virtual filesystems like /proc. Execute the following commands:

```
cd /diskless/root
mkdir bin boot dev etc home lib proc root sbin sys tmp usr var
```

5.3. Copy etc directory to /diskless/root/etc

Because /etc contains many system files that need to be modified for terminals to function properly, you need to make a copy of the /etc directory under /diskless/root/etc. This way terminal-specific changes will not affect the server. You can use the following command to create a copy of the /etc directory:

```
rsync -av /etc/ /diskless/root/etc
```

The first /etc/ must have a trailing slash

5.4. Create the bind mount entries

Open the file /etc/fstab and add the following lines at the end:

```
# These mounts create the environment for terminal workstations
/bin               /diskless/root/bin        none   bind   0 0
/boot              /diskless/root/boot       none   bind   0 0
/lib               /diskless/root/lib        none   bind   0 0
/sbin              /diskless/root/sbin       none   bind   0 0
/usr               /diskless/root/usr        none   bind   0 0
/var               /diskless/root/var        none   bind   0 0
/etc/hosts         /diskless/root/etc/hosts  none   bind   0 0
/etc/group         /diskless/root/etc/group  none   bind   0 0
/etc/shadow        /diskless/root/etc/shadow none   bind   0 0
/etc/passwd        /diskless/root/etc/passwd none   bind   0 0
```

Now mount the directories using the command

```
mount -a
```

You should see no errors and when you check the contents of directories under /diskless/root, they should have identical contents as the corresponding directories in /.

5.5. Create the snapshot directory

Next, create the snapshot directory where every terminal keeps its own writeable system files. We will then use a Red Hat tool to create the necessary template files in this directory. To create the snapshot directory, execute the following command:

```
mkdir /diskless/snapshot
```

5.6. Export system and snapshot directories

The system directory and snapshot directory must be made accessible to clients over NFS protocol. Open the file /etc/exports and add the following lines:

```
/diskless/root      *(ro,async,no_root_squash,no_subtree_check)
/diskless/root/bin  *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/boot *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/lib  *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/sbin *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/usr  *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/var  *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/etc/hosts \
      *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/etc/passwd \
      *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/etc/group  \
      *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/etc/shadow \
      *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/snapshot  *(rw,async,no_root_squash)
```

Note that all entries added are exported in read-only mode, but the /diskless/snapshot directory is exported in read-write mode.

All the bind mounted directories under /diskless/root need to have a separate entry in /etc/exports and be mounted with the nohide option. Otherwise clients will not see the contents of the directories because NFS works per mountpoint, not per directory. (NOTE: this is the authors understanding).

After creating the entries, reload the NFS service by executing the following script:

```
/etc/init.d/nfs restart
```

5.7. Assemble the boot files and populate the snapshot directory

Next, gather and assemble certain files needed for booting. Among the files are the bootloader, kernel, initial ramdisk, terminal startup script and a list of files that terminals must have read-write access to. Red Hat provides a graphical tool that automates most of this process.

Execute the following graphical tool (accessible also from Applications > System Settings > Server Settings > Network Booting Service)

```
system-config-netboot
```

- From the “First Time Druid“ window select “Diskless”
- Click “Forward“
- In the „Name“ field enter “JCATS terminal“ or other suitable name for this setup
- In the “Description“ field enter a description for this setup, for example “Diskless JCATS 7.1.3 terminals in classroom 205“
- Click “Forward“
- For “Server IP Address“ enter 10.0.0.1 (or the address this server is using)
- For “Directory“ enter /diskless/root
- Click “Forward”
- Next select the kernel that diskless clients will be using. By default the following kernels are available: 2.6.9-42.EL and 2.6.9-42.ELsmp. As my terminals have single core processors, I choose the 2.6.9-42.EL kernel. Making other kernels available is something to investigate further.
- Click “Forward“
- Click Apply
- Exit the utility

The utility will now do several things, among them a kernel and initrd in /tftpboot/linux-install/JCATS_terminal, a custom initscript and fstab file in /diskless/root/etc/. Also a list of files that will be accessible to the terminal in read-write mode is written to /diskless/snapshot/files.

5.8. Create a boot configuration for terminals

Edit the file /tftpboot/linux-install/pxelinux.cfg/default to look like the following:

```
default JCATS_terminal/vmlinuz initrd=JCATS_terminal/initrd.img \
append root=/dev/ram0 init=/disklessrc NFSROOT=10.0.0.1:/diskless \
ETHERNET=eth0
```

Every time you open and close the system-config-netboot tool from step 5.7, it will overwrite the /tftpboot/linux-install/pxelinux.cfg/default file. After entering the above line, create a backup copy of the file for easy restoration.

6. Configure the terminals

Before the terminals can boot up, there are a few modifications that must be done. First they need read-write access to several locations and files. They also need access to the

system hosts file, account data and the /home partition. You must also turn off system services terminals not needed – that is most of them.

6.1. Create missing /dev/entries

Terminals need some entries in the /dev directory to boot properly. To create these entries, execute the following commands:

```
cp -pr /dev/console /dev/null /dev/tty1 /diskless/root/dev
```

6.2. Allow read-write access to necessary system files

Open the file /diskless/snapshot/files and add the following lines:

```
/etc/pam.d/
/etc/mtab
```

6.3. Edit the fstab file

Because the whole /diskless/root tree was exported as read-only and JCATS needs write address, you need to mount /home with JCATS files in another way. You need to edit the fstab file for the clients and add an entry to mount /home from server using NFS. To do this open the file /diskless/root/etc/fstab and add the following line:

```
10.0.0.1:/home/     /home  nfs    defaults     0 0
```

Unless you want to give terminal users access to local floppy and cd-rom drives, you will most likely also want to delete the lines starting with /dev/cdrom and /dev/floppy.

6.4. Disable services

To disable unneeded services, chroot into the /diskless/root directory and use the standard chkconfig tool to turn off unneeded services. First change the root directory:

```
chroot /diskless/root
```

Next use the chkconfig tool. The general syntax is:

```
chkconfig --level 12345 service off
```

Replace the string „service“ with the actual name of the service you are going to disable. It is recommended that you disable the following services:

```
mdmonitor syslog rhnsd readahead_early crond cups-config-daemon nfs
haldaemon readahead openibd isdn lm_sensors iptables irqbalance atd
anacron rpcidmapd messagebus cpuspeed cups pcmcia acpid dhcpd
arptables_jf microcode_ctl rpcgssd kudzu rawdevices gpm tftp
```

Last, be sure to exit the chroot environment:

```
exit
```

6.5. Configure X

Terminals most likely have different X configuration than the server. Find out the correct parameters for terminals and insert them into the file /diskless/root/etc/X11/xorg.conf. Configuring X is outside the scope of this document, but most likely you need to change the video device and screen resolution.

6.6. Boot the terminals

And hope for the best...

2. List of commands

This section is for those eager to get terminals operational or for quick reference. If a particular step is not understood, you can find explanations in Chapter 1.

1. chkconfig –level 35 ypbind off
   chkconfig --level 35 ypserv off
   chkconfig --level 35 yppasswdd off
2. edit /etc/nsswitch.conf, delete the string "nis" from passwd, shadow, group, hosts and services lines.
3. edit /etc/dhcpd.conf file to look like:

```
authoritative;
ddns-update-style none;
subnet 10.0.0.0 netmask 255.255.255.0 {
  option            routers 10.0.0.1;
  option            subnet-mask 255.255.255.0;
  option            domain-name „simctrest“;
  next-server       10.0.0.1;
  filename          „linux-install/pxelinux.0“;
}
```

4. for every terminal, enter a host section into /etc/dhcpd.conf, inside the subnet declaration:

```
host ws101 {
  hardware ethernet       00:60:72:10:61:A2;
  fixed-address           10.0.0.101;
}
```

5. chkconfig --level 35 dhcpd on
6. /etc/init.d/dhcpd start
7. if not done previously, enter a line for every terminal into /etc/hosts:

```
10.0.0.101          ws101.simctrest     ws101
```

8. change the line disable = yes to disable = no in /etc/xinetd.d/tftp
9. /etc/init.d/xinetd reload
10. mkdir -p /diskless/root
11. cd /diskless/root
12. mkdir bin boot dev etc home lib proc root sbin sys tmp usr var
13. rsync -av /etc/ /diskless/root/etc
14. insert the following lines to /etc/fstab:

```
# These mounts create the environment for terminal workstations
/bin                /diskless/root/bin        none   bind   0 0
/boot               /diskless/root/boot       none   bind   0 0
/lib                /diskless/root/lib        none   bind   0 0
/sbin               /diskless/root/sbin       none   bind   0 0
/usr                /diskless/root/usr        none   bind   0 0
/var                /diskless/root/var        none   bind   0 0
/etc/hosts          /diskless/root/etc/hosts  none   bind   0 0
/etc/group          /diskless/root/etc/group  none   bind   0 0
/etc/shadow         /diskless/root/etc/shadow none   bind   0 0
/etc/passwd         /diskless/root/etc/passwd none   bind   0 0
```

15. mount -a
16. mkdir /diskless/snapshot
17. insert the following lines to /etc/exports:

```
/diskless/root            *(ro,async,no_root_squash,no_subtree_check)
```

```
/diskless/root/bin  *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/boot *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/lib  *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/sbin *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/usr  *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/var  *(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/etc/hosts \
*(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/etc/passwd \
*(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/etc/group \
*(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/root/etc/shadow \
*(ro,async,no_root_squash,no_subtree_check,nohide)
/diskless/snapshot  *(rw,async,no_root_squash)
```

18. /etc/init.d/nfs restart
19. system-config-netboot
    - 19.1. select „Diskless“
    - 19.2. click „Forward“
    - 19.3. enter „JCATS terminal“ for „Name“
    - 19.4. enter „Diskless JCATS 7.1.3 terminals“ for „Description“
    - 19.5. click „Forward“
    - 19.6. enter 10.0.0.1 for „Server IP Address“
    - 19.7. enter „/diskless/root“ for „Directory“
    - 19.8. click "Forward"
    - 19.9. select kernel 2.6.9-42.EL
    - 19.10. click „Forward“
    - 19.11. click „Apply“
    - 19.12. exit utility
20. insert the following lines into the file /tftpboot/linux-install/pxelinux.cfg/default:

```
default JCATS_terminal/vmlinuz initrd=JCATS_terminal/initrd.img \
append root=/dev/ram0 init=/disklessrc NFSROOT=10.0.0.1:/diskless \
ETHERNET=eth0
```

21. cp -pr /dev/console /dev/null /dev/tty1 /diskless/root/dev
22. insert the following lines to /diskless/root/snapshot/files:

```
/etc/pam.d/
/etc/mtab
```

23. insert the following line to /diskless/root/etc/fstab:

```
10.0.0.1:/home      /home        nfs    defaults     0 0
```

24. chroot /diskless/root
25. use the command "chkconfig --level 12345 service off" to disable the following services:

```
mdmonitor syslog rhnsd readahead_early crond cups-config-daemon nfs
haldaemon readahead openibd isdn lm_sensors iptables irqbalance atd
anacron rpcidmapd messagebus cpuspeed cups pcmcia acpid dhcpd
arptables_jf microcode_ctl rpcgssd kudzu rawdevices gpm tftp
```

26. exit
27. configure the /diskless/root/etc/X11/xorg.conf file

3. Q&A

Based on user feedback and operational usage, this section will hopefully have more suitable recommendations in future release editions

1. I modified /etc/hosts file on the server or added a user, but clients don't see the changes

Bind mounts over NFS don't update their contents immediately, for some reason. To change a file in /etc, you must do the following:

```
umount /diskless/root/etc/hosts
/etc/init.d/nfs stop
edit the file
mount /diskless/root/etc/hosts
/etc/init.d/nfs start
```

It is probably wise to do this when terminals are not running.

An alternative is to not bind mount hosts, passwd and other files from the server /etc directory, but use local copies in /diskless/root/etc. You will have to keep them in sync by hand (if they need to be in sync at all), but this seems less error-prone than running NIS.

2. How do I perform additional configuration of the terminal environment – install applications, configure their behaviour?

There are two thing to consider:

- Everything you install on the main server is automatically visible to clients as well. This is because the system directories like /usr are shared between server and terminals. Most often you can install additional software using standard Red Hat tools (rpm) and procedures. See Red Hat Linux Documentation for more details.

  Keep in mind the explanations given for question 1 above as well.

- The configuration files are not shared between the server and terminals. To change the behaviour of terminals, you will have to edit the files in /diskless/root/etc. It is advisable to use the chroot method introduced in 6.4. to minimize the risk of accidentally editing server configuration files.

  Keep in mind that not all configuration files are shared between terminals. The files listed in /diskless/snapshot/files are specific to every terminal. If you need to customize the configuration of individual terminals, add the necessary files to this list and edit the copies in /diskless/snapshot/<terminal IP address>/etc.

3. The NFS server does not seem to be able to handle this many clients. What do I do?

By default Red Hat Linux starts 8 NFS server processes. Depending on the number of terminals, this may not be sufficient to service all of them properly. To increase the number of NFS server processes, create the file (it does not exist by default) /etc/sysconfig/nfs and insert the following line:

```
RPCNFSDCOUNT=32
```

You can experiment with different number of processes to find out your requirements.

After you have edited the file, restart the NFS service by executing the command:

```
/etc/init.d/nfs restart
```