HP SmartStart Scripting Toolkit Linux Edition
Best Practices

April 2005 (Third Edition)
Part Number 365443-003

April 2005 (Third Edition)

Part Number 365443-003

Contents

Introduction

In this section

Linux Toolkit overview

This document describes how to best utilize the Linux edition of the SmartStart Scripting Toolkit to configure HP ProLiant servers. It also contains information about using the Toolkit utilities in an unattended environment. This document does **not** include information about installing the operating system.

The best practices also include suggestions about how to organize utilities and data files by following a set of operational procedures that standardize configuration procedures and help reduce errors.

CAUTION: Because of the potential risk of data loss, be sure that all necessary precautions are taken so that mission-critical systems are not disrupted if a failure occurs.

Toolkit changes

Previous versions of the SmartStart Scripting Toolkit utilities were designed for the MS-DOS environment. However, limitations in MS-DOS have become a significant problem as hardware and software have evolved. In response to this issue, HP has migrated the SmartStart Scripting Toolkit to a Linux environment to provide better scripting and better hardware support.

The Linux edition of the Toolkit provides the same functionality as the earlier MS-DOS version. However, tools, arguments, and data files are different in the Linux edition of the Toolkit, so you must update your customized scripts to work in a Linux environment. Data files have been migrated to industry-standard XML format to provide improved extensibility. For more information about the Toolkit utilities, refer to the *HP SmartStart Scripting Toolkit Linux and Win32 Editions User Guide* on the Toolkit website (http://www.hp.com/servers/sstoolkit).

Minimum requirements

Before beginning the deployment process, be sure to have the following items available:

- *HP SmartStart Scripting Toolkit Linux and Win32 Editions User Guide*
- SmartStart Scripting Toolkit Linux Edition
- A Linux workstation (any Linux distribution)
- SYSLINUX package, downloaded from the Web

The Linux Toolkit environment

The Linux Toolkit environment is a small, multi-purposed, Linux environment that can be booted from various media, such as CD, network or PXE, or bootable USB device. After the Linux environment is booted, it has virtually the same capabilities as Linux, provided that the tools are available. This document discusses the following topics:

- Booting the environment
- Using the environment

The first section describes how to set up the pieces needed to boot the environment from CD or using PXE. The second section describes how to use the environment to perform certain functions, such as capturing configurations, flashing configurations, and setting up partitions for operating system installation.

Because the Toolkit is Linux-based, this document assumes that the operations described will be performed under Linux. Unless otherwise stated, a full Linux installation of any distribution of Linux released in the last two years enables you to perform all the operations described in this document.

Booting the Linux Toolkit environment

In this section

Overview of booting the Linux Toolkit

The three main components needed to boot any Linux environment are the bootloader, the Linux kernel, and the Linux file system. For the Linux edition of the SmartStart Scripting Toolkit, these components are:

- Bootloader: SYSLINUX
- Kernel: vmlinuz, a kernel based on Red Hat 7.3
- File system: initrd.img, a reduced Red Hat 7.3 installation

In general, you must only modify the bootloader to boot the Toolkit in your environment.

Whether you boot from a CD or through a network, the following general steps must occur for the Toolkit environment to boot:

1. The system boots and performs a POST.
2. The BIOS determines how the system boots, the BIOS settings, and the available media.
3. The BIOS retrieves the bootloader using the available media.
4. The bootloader (SYSLINUX) loads the Linux file system into memory.
5. The bootloader boots the Linux kernel.
6. A startup script is executed to finish the environment setup.

SYSLINUX

SYSLINUX is a free third-party bootloader available at http://syslinux.zytor.com/index.php (http://syslinux.zytor.com/index.php).

SYSLINUX consists of a suite of programs that perform various boot functions. The Toolkit uses the following two bootloader programs:

- isolinux.bin—This program enables you to boot from ISO media.
- pxelinux.0—This program enables you to boot using PXE protocol.

Both bootloaders require a configuration file to run:

- isolinux.cfg—This configuration file is used for booting from ISO media.
- default.cfg—This configuration file is used for booting using PXE.

The following is a sample configuration file:

```
default normal_boot
prompt 1
timeout 100
label normal_boot
   kernel vmlinuzA
   append initrd=initrd.img rw root=/dev/ram0
   ramdisk_size=39567 init=/first_script
label bash_prompt
   kernel vmlinuzB
   append initrd=initrd.img rw root=/dev/ram0
   ramdisk_size=39567 init=/bin/bash
```

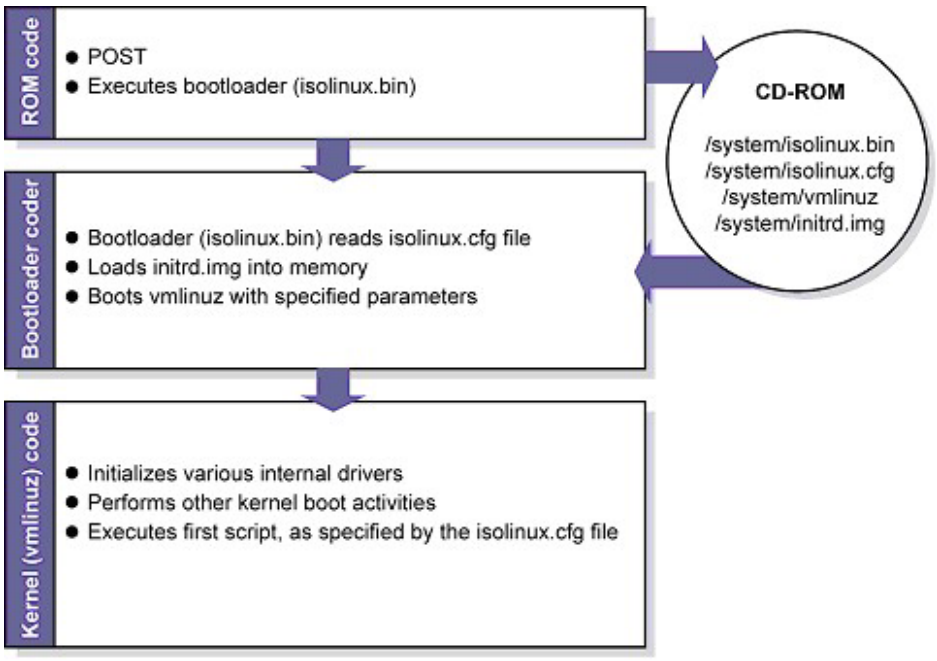
In this example, there are two distinct boot directives: normal_boot and bash_prompt. The configuration file instructs the bootloader to prompt the user to pick a boot directive, wait 10 seconds, and then boot the default directive, normal_boot, if no user input is recorded.

The normal_boot directive instructs the bootloader to use vmlinuzA as the kernel. The append line specifies which parameters the bootloader passes to the kernel. The parameter `init=/first_script` instructs the kernel to pass control to the file first_script.

For more information about SYSLINUX usage, refer to the SYSLINUX documentation.

Booting from the CD

The following figure illustrates a Toolkit CD boot.



Creating a Toolkit bootable CD requires the following general steps:

1. Create a CD build directory.
2. Create an ISO image to be written to CD.

Creating a CD build directory

1. Create a directory on the Linux workstation; for example, ./linuxbootCD.
2. Create a subdirectory on which to store boot files; for example, ./linuxbootCD/system.

3. Copy the necessary boot files to the ./linuxbootCD/system directory:
   - isolinux.bin (the SYSLINUX binary used for ISO media)
   - isolinux.cfg (the boot directive used by isolinux.bin)
   - initrd.img (the Linux file system)
   - vmlinuz (the Linux kernel)

If needed, copy additional files to ./linuxbootCD. These files might include Toolkit files, configuration files, or third-party tools. Alternately, refer to "Using the Linux Toolkit environment (on page 23)" for information about downloading the files from a network share.

Creating an ISO image

The `mkisofs` command is used to create an ISO image. The following table describes the arguments used with this command.

Argument	Description
`-o linuxbootCD.iso`	This argument is the output of the mkisofs command, the ISO file.
`-b system/isolinux.bin`	This argument sets isolinux.bin as the bootloader.
`-v LinuxBootCD`	This argument sets the volume label of the CD.
`./linuxbootCD`	This argument specifies the target directory that will be the root of the CD.

To create the ISO image, execute the following command at the shell prompt:

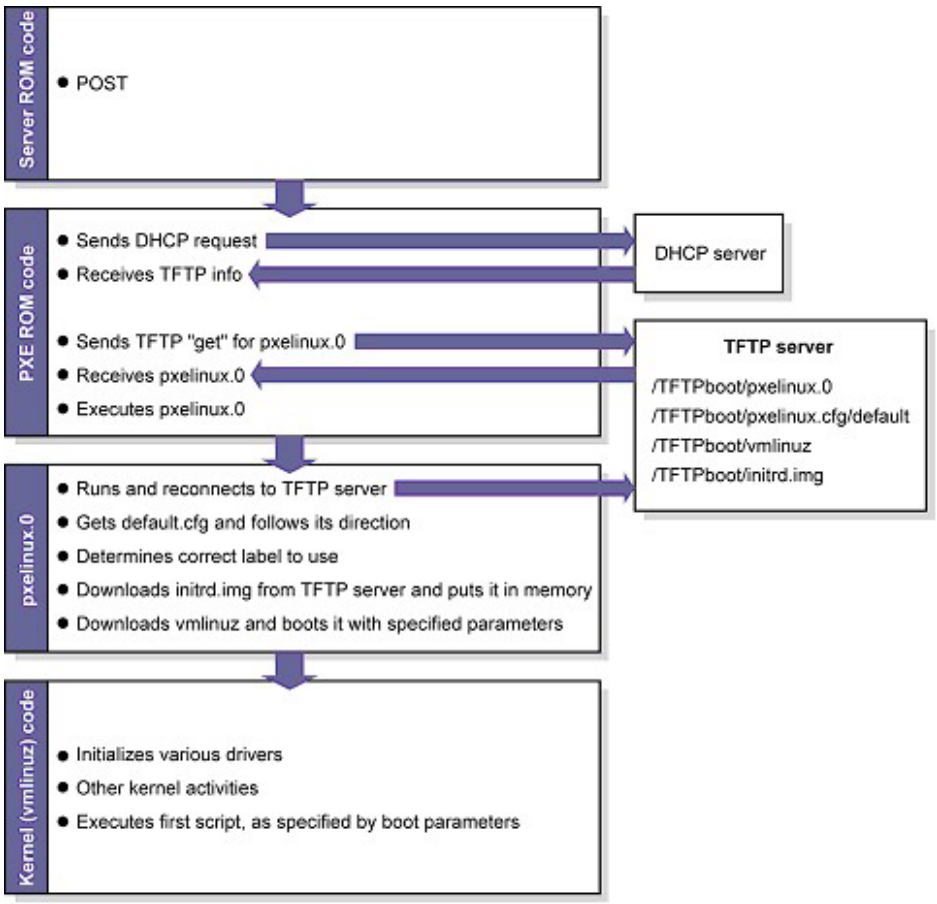
```
mkisofs -J -iso-level 3 -R -L -o linuxbootCD.iso \
   -b system/isolinux.bin -c system/boot.cat \
   -V LinuxBootCD \
   -no-emul-boot -boot-load-size 4 \
   -boot-info-table \
   ./linuxbootCD
```

The ISO file can now be written to a CD.

Booting using PXE

IMPORTANT: A basic understanding of DHCP, PXE, and TFTP is required to perform the procedure described in this section.

The following figure illustrates a simplified Toolkit PXE boot.



Setting up a PXE boot environment requires the following general steps:

1. Set up a DHCP server with the appropriate options.

2. Set up a TFTP server with the appropriate options.
3. Populate the TFTP directory share with the Linux Toolkit boot components.

These steps assume that a Linux workstation is used as the DHCP/TFTP server. You might need to download additional components and adapt the following instructions to suit your environment.

Setting up a DHCP server

To set up a DHCP server, create and edit a dhcpd.conf file in the /etc directory of your server. The following is a sample dhcpd.conf file:

```
allow booting;
allow bootp;
ddns-update-style ad-hoc;
```

[Insert the usual DHCP directives, IP ranges, subnet masks, and so on here.]

```
next-server IP_ADDRESS_OF_TFTP_SERVER;
filename "pxelinux.0";
```

The next-server command tells the DHCP client where to send the TFTP get request.

The filename command tells the DHCP client which file to get. In this example, the file is pxelinux.0.

When you have finished creating the dhcpd.conf file, restart the dhcpd process:

```
/etc/init.d/dhcpd restart
```

Setting up a TFTP server

Most Linux installations include a TFTP server and an automated method of launching the server upon receiving a TFTP request. The parent process for detecting a TFTP request and launching the TFTP server is called xinetd. However, you might have to enable the TFTP service. The TFTP file is located in the /etc/xinetd.d/ directory. The following is a sample TFTP file:

```
# default: off
# description: The tftp server serves files using the \
# trivial file transfer protocol. The tftp protocol is \
# often used to boot diskless workstations, download \
```

```
# configuration files to network-aware printers and to \
# start the installation process for some operating
systems.
service tftp
{
   socket_type = dgram
   protocol = udp
   wait = yes
   user = root
   server = /usr/sbin/in.tftpd
   server_args = -s /tftpboot
   disable = yes
   per_source = 11
   cps = 100 2
}
```

In this example, "disabled" is the default setting, and /tftpboot is the root directory for all client access.

To enable the TFTP service on your server, edit the disable line to read:

```
disable = no
```

Populating the TFTP directory share

Use the /tftpboot directory from the "Setting up a TFTP server (on page 14)" section as the directory from which TFTP clients will get their files. To populate the TFTP directory share:

1. Create a /tftpboot directory, if needed.
2. Copy the necessary boot files to the /tftpboot directory:
   - pxelinux.0 (the SYSLINUX binary used for PXE boot)
   - initrd.img (the Linux file system)
   - vmlinuz (the Linux kernel)
3. Create a pxelinux configuration file subdirectory called /tftpboot/pxelinux.cfg.
4. Copy the default.cfg file (the boot directive used by pxelinux.0) into the /tftpboot/pxelinux.cfg/ directory.

The tftpboot directory should now contain the following items:

```
/tftpboot/pxelinux.0
/tftpboot/initrd.img
/tftpboot/vmlinuz
/tftpboot/pxelinux.cfg/default.cfg
```

Booting from a USB flash device

Some applications, such as the firmware update components, require the use of a writable medium. A writable file system allows for the expansion of the contents of the components and provides a scratch area for the backup firmware image copied from the device under flash. While using booting from CD is not suitable for this purpose, a USB flash device provides the ideal medium for this type of activity.

> **NOTE:** Booting from a USB flash device is supported only on the HP ProLiant BL20p G3 Server and ProLiant G4 and later servers.

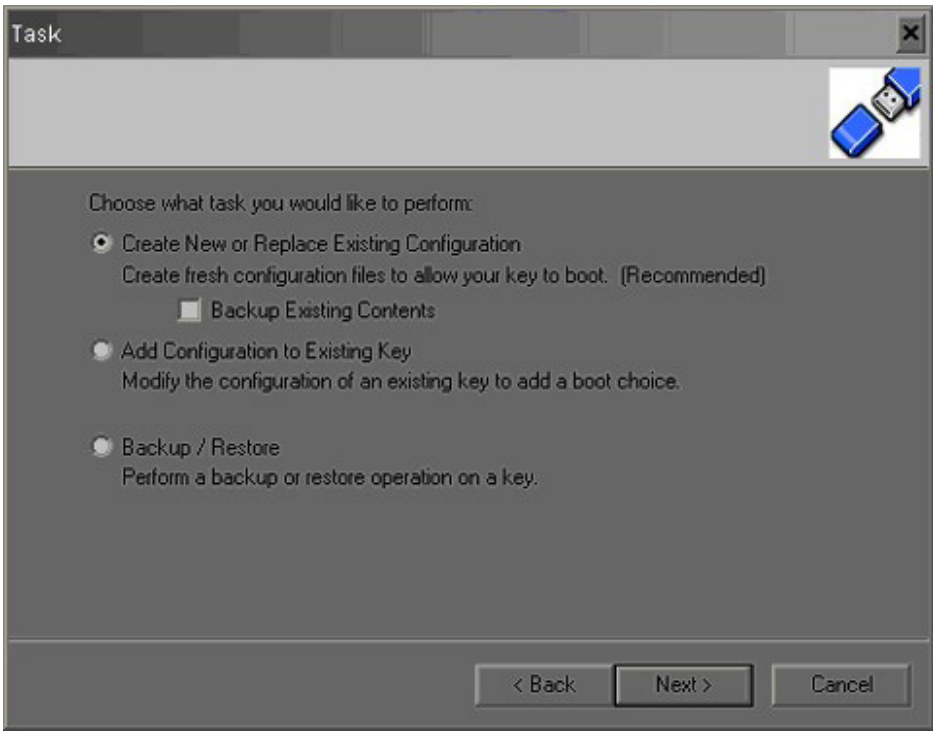
To boot from a USB flash device:

1. Download the HP Drive Key Boot Utility from the HP support website (http://h18023.www1.hp.com/support/files/server/us/download/21621.html), and place it in a temporary directory.
2. Execute the component, and select the install option.
3. Start the HP Drive Key Boot Utility. A welcome window appears.

   > **NOTE:** You must use Microsoft® Windows® 2000 or later to use the HP Drive Key Boot Utility.

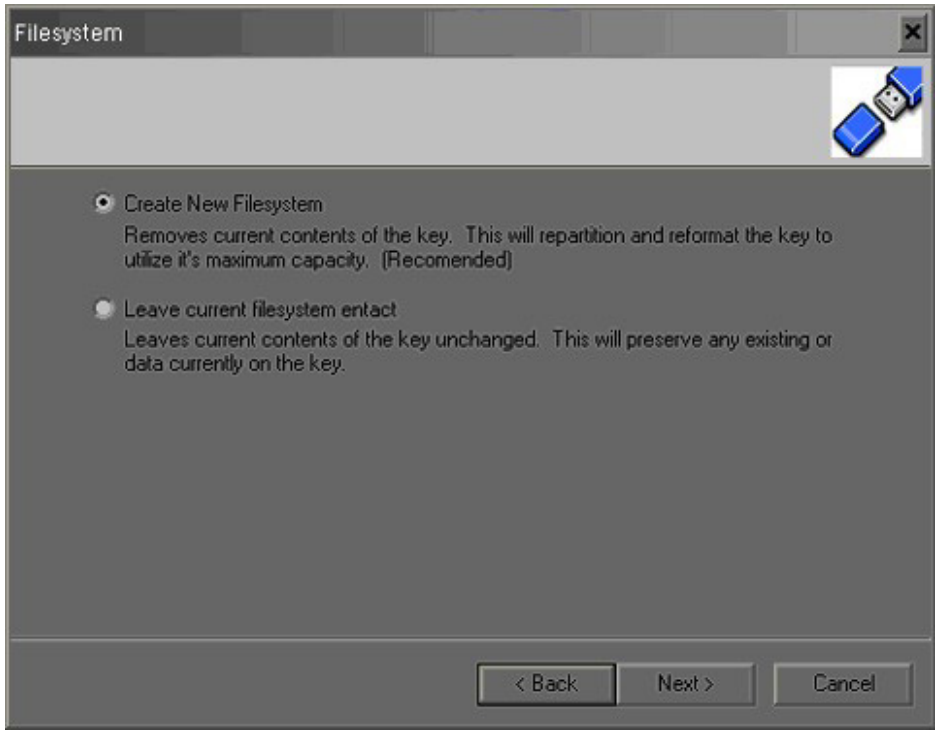
4. Insert the USB flash device, and click **Next.**

5. Select the drive letter for the USB flash device you want to use. The following screen appears.



6. Select **Create New or Replace Existing Configuration.**

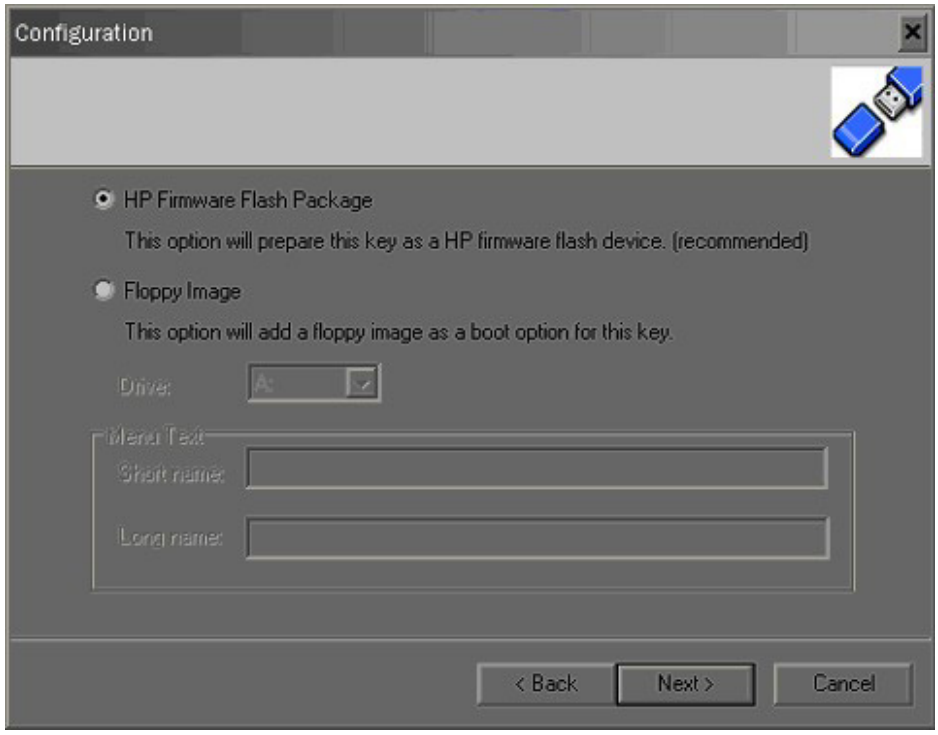
7. Click **Next.** The following screen appears.



8. Select **Create New Filesystem.** The utility creates an MBR, creates a partition table, and formats the flash device.

CAUTION: Selecting the "Create New Filesystem" option will erase the USB flash device.

9. Click **Next.** The following screen appears.



10. Select **HP Firmware Flash Package.**
11. Click **Next.**

The flash device has been successfully prepared. The Toolkit can now be installed on the USB flash device.

To install the Toolkit on the USB flash device:

1. Prepare the initrd.img file, and replace the existing bootstrap script with a prepared installation script. For more information, refer to "Editing the file system file (on page 29)."
2. Copy the modified initrd.img file and the vmlinuz file from the Toolkit onto the USB flash device, overwriting the existing files.
3. Copy the isolinux.cfg file onto the USB flash device.

4. Delete the existing syslinux.cfg file.
5. Rename the isolinux.cfg file to "syslinux.cfg."
6. Use a text editor to modify the new syslinux.cfg file:
   a. Change the init=/bootstrap parameter to refer to your custom script.
   b. Remove the ssstk options because these options are used only by the original bootstrap script.

   Sample modified syslinux.cfg file:

```
say HP Bootable Media
say Copyright 2001, 2004 Hewlett-Packard Development
Company,L.P.
say press <enter> or wait 30 seconds for normal
toolkit boot
say type 'bash' and press <enter> for a bash prompt
say type 'local' and press <enter> to boot from local
drive.
default toolkit
prompt 1
timeout 300
label toolkit
kernel vmlinuz
append initrd=initrd.img rw noht noapic root=/dev/ram0
   ramdisk_size=39996 init=/custom.sh ide=nodma
label bash
kernel vmlinuz
append initrd=initrd.img rw noht noapic root=/dev/ram0
   ramdisk_size=39996 init=/bin/bash ide=nodma
label local
localboot 0x80
```

   NOTE: HP recommends that the /custom.sh script placed in the initrd.img file contains commands to load the USB drivers, mount the USB flash device file system, and pass execution to a script stored on the USB flash device, which is easier to update than the initrd.img file.

7. Test the USB flash boot process:
   a. Insert the USB flash device in the server. If the server already has a C drive, you must change the IPL order to ensure the USB boot device boots before the C drive of the primary controller. After the flash device boots, the syslinux information and a prompt appear.

b. Press **Enter** at the "Boot:" prompt. A progress bar and the Toolkit boot messages appear. After processing is complete, the /custom.sh script from the initrd is executed.

8. Customize the Linux Toolkit environment on the USB flash device for your use.

Using the Linux Toolkit environment

In this section

Overview of using the Linux Toolkit environment

The general tasks for setting up a target server for operating system installation include:

- Basic Toolkit strategy
- Toolkit environment setup
- Operating system-independent configuration
- Microsoft® Windows® pre-installation setup
- Linux pre-installation setup

You might need to adapt these tasks to suit your specific needs.

Basic Toolkit strategy

The three primary components of a Toolkit Linux boot are the bootloader, the kernel, and the file system (initrd.img):

- The choice of bootloader depends on whether the boot process occurs using the CD media or over a network using PXE.

- The kernel is generally static and cannot be modified easily. HP recommends that you use the kernel that is shipped with the Toolkit because it has been tested on all servers supported by the Toolkit.
- The initrd.img must be created during the Toolkit setup process and is easily modified. For details on creating an initrd.img file, refer to "Editing the file system file (on page 29)."

The initrd.img file enables a target machine to boot a Linux environment with network connectivity. You can edit the network settings in the packaged initrd.img file to be specific to your deployment environment. For more information, refer to "Editing the file system file (on page 29)."

The bootstrap script in the initrd.img file copies the next script in the execution path and executes it, beginning the Toolkit process. Because this script, toolkit_redirect.sh, resides on a network share and not inside initrd.img, you can modify it as often as needed without needing to rebuild the initrd.img file.

The primary purpose of the toolkit_redirect.sh script is to connect to the main network share and copy the main_toolkit_launch.sh script. This redirection is necessary because the deployment environment can change over time. If the main network share crashes and a secondary network share takes its place, the toolkit_redirect.sh script can be changed to point to the new network share. The initrd.img file does not need to be revised.

The main_toolkit_launch.sh script copies the remaining toolkit utilities to the RAM drive created by the bootstrap script. It then determines the server ID of the target server and uses that information to copy the server-specific configuration scripts and data files.

The main_toolkit_launch.sh script then executes the server-specific script. The remaining steps are specific to each deployment situation.

Toolkit environment setup

To set up the Toolkit environment, the bootstrap script at the root of the initrd.img file system:

1. Mounts the /usr directory. This step makes the /usr directory available to the rest of the Linux environment.

2. Creates an expandable RAM drive to serve as the Toolkit work area.
3. Loads all common network drivers needed by HP ProLiant servers. You can edit this script to add or remove network drivers.
4. Configures the network interface device (eth0). You must edit the bootstrap script to match your deployment environment. To assist you in configuring the eth0 device, the ifconfig and dhclient programs are provided in the /bin directory.
5. Mounts a network share on the server.
6. Copies the initial toolkit script (toolkit_redirect.sh) from the network share and executes it.

The Toolkit boot is complete.

The commands are located in the toolkit_bootstrap.sh script in the /root/compaq/ directory. Steps 1 and 2 are required and should not be modified. Steps 3, 4, and 5 can be modified for your specific deployment environment.

Operating system-independent configuration

After the main network share is mounted, the main_toolkit_launch.sh script initiates Toolkit operations.

The sample main_toolkit_launch.sh script:

1. Copies all toolkit utilities from the network share to the target server.
2. Runs hardware discovery to determine server type.
3. Obtains server ID information from the hardware discovery file.
4. Copies server-specific configuration script and data files from the network share.
5. Runs the server-specific configuration script:
   a. Loads drivers for storage controllers and any other devices that must be configured.
   b. Runs the CONREP utility.
   c. Runs the CPQACUXE utility, if needed. This action is server-specific.

d. Runs any other configuration utilities.

e. Runs hardware discovery to determine the boot controller.

f. Obtains the device node of the boot controller. This information is required for the disk carving portion of the operating system setup, as described in the following two sections.

g. Runs the operating system-specific setup script.

These steps can be adapted to your server deployment process.

Windows pre-installation setup

You can use the win_diskcarve.sh script to set up a DOS partition for Microsoft® Windows® Server 2000 or Windows® 2003 Server setup. Run win_diskcarve.sh in its own directory to create a 4-GB bootable DOS partition on the system boot controller.

After the bootable DOS partition is created, you can populate the partition with the Windows® installation files (i386 directory and unattend.txt), HP drivers, and ProLiant Support Packs.

This step is not specific to the Linux Toolkit environment. For more information on setting up a Windows® unattended install, refer to Microsoft® documentation.

Linux pre-installation setup

To set up a Red Hat kickstart install using the Toolkit Linux Edition:

1. Run the linux_diskcarve.sh script in its own directory. This script:
   - Creates a FAT16 partition.
   - Copies ldlinux.sys, which is the syslinux binary for booting off FAT partitions, to the partition.
2. Copy the Red Hat vmlinuz kernel to the partition.
3. Copy an initrd.img file that contains the necessary drivers for the install to the partition:

a. Obtain a Red Hat bootnet.img diskette image from the Red Hat media or from the HP website (http://www.hp.com), where there is a downloadable bootnet.img file for each ProLiant server that supports Red Hat Linux.

b. Mount the image on your Linux system.

c. Copy the vmlinuz and initrd.img files from the diskette image to your network share.

NOTE: The initrd.img from the bootnet.img file is not equivalent to the initrd.img delivered with this Toolkit.

4. Copy a syslinux.cfg file that specifies a kickstart install to the partition.
5. Reboot to the FAT16 partition.

The following is a sample syslinux.cfg file:

```
default ks
prompt 1
timeout 6
display boot.msg
F1 boot.msg
label linux
kernel vmlinuz
   append initrd=initrd.img lang= devfs=nomount
   ramdisk_size=9216
label ks
   kernel vmlinuz
   append
   ks=nfs:10.7.90.120:/STORAGE/examples/ks_rh80.cfg
   initrd=initrd.img lang= devfs=nomount
   ramdisk_size=9216 network
```

The following line points the Red Hat install to an NFS share to obtain the kickstart file ks_rh80.cfg:

```
append ks=nfs:10.7.90.120:/STORAGE/examples/ks_rh80.cfg
```

You must modify this line to match your deployment environment.

Most recent Red Hat installations provide a kickstart file, called anaconda-ks.cfg, in the /root directory. Modify this file, and put it on your network share for subsequent installations.

To perform a Red Hat unattended install, modify the following lines:

```
network --device eth0 --bootproto dhcp
```

The preceding line specifies which device to configure and what protocol to use.

```
nfs --server 10.7.90.120 --dir /STORAGE/rh80
```

The preceding line specifies that NFS be used as the file sharing protocol. You must set up the NFS share with the Red Hat installation tree. For more information about how to set up the installation tree, refer to the README file on the Red Hat CD media.

```
clearpart --all --drives=cciss/c0d0
```

The preceding line erases everything from the install drive, including the FAT16 partition that was created in step 1 of this section. HP recommends erasing the FAT16 partition.

Modify the rest of the ks.cfg file as needed.

Editing the file system file

In this section

Overview of editing the file system file

The ability to edit the file system file (initrd.img) enables you to adapt the Linux Toolkit to various deployment environments.

Most users are familiar with the concept of a file system located on a disk partition. The initrd.img file is a file system located in a file. In this case, a large file (initrd.img) is created and formatted as a file system. The file acts as a complete disk partition.

In Linux and UNIX®-like systems, partitions on a hard drive must be mounted at a mount point before they can be accessed. The initrd.img must also be mounted, but because it is a file, it must be mounted as a loopback device. When mounted, the initrd.img can be manipulated like any other mounted partition. Files can be deleted from and added to it.

NOTE: After the initrd.img file is created, its size cannot be changed.

Editing the initrd.img file

CAUTION: Improper editing of the initrd.img file can result in an unusable file. Be sure to back up your work before proceeding.

IMPORTANT: All steps described must be performed on a Linux system as the root user.

The following basic steps are required to edit the initrd.img file:

1. Uncompress the initrd.img file.

2. Mount the initrd.img.uncompressed file.
3. Edit the initrd.img.uncompressed file.
4. Unmount the initrd.img.uncompressed file.
5. Compress the initrd.img file.

These steps are described in detail in the following sections.

Uncompressing the initrd.img file

The initrd.img file was compressed with the gzip utility. To uncompress initrd.img, use the gunzip utility.

1. Rename the packaged initrd.img file with a file name that has the extension .gz:
   ```
   mv (or cp) initrd.img initrd.img.uncompressed.gz
   ```
2. Uncompress the renamed file:
   ```
   gunzip (or gzip -u) initrd.img.uncompressed.gz
   ```

This procedure produces a new file called initrd.img.uncompressed. Notice that the size of this file is different from the original initrd.img file.

Mounting the uncompressed initrd.img file

Linux enables you to mount file systems that exist in a file (as opposed to file systems on a disk partition) by using a loopback device. Discussion of loopback devices is outside the scope of this document.

Mount the initrd.img.uncompressed file:

```
mount -o loop initrd.img.uncompressed mnt_point/
```

After mounting the file system, you can navigate to it using the mount point.

Editing the uncompressed initrd.img file

When the initrd.img.uncompressed file is mounted, you can make any desired changes to it by using the mount point, just as you would edit files on a mounted partition.

The initrd.img file shipped with the Toolkit has approximately 5 MB of free space. Linux requires a small amount of writable free space on the file system to operate properly, so be sure not to fill the entire free space of the initrd.img file.

The bootstrap script that begins the Toolkit execution process is located at the root of the file system. You might need to make changes to this script to suit your specific needs.

Unmounting the uncompressed initrd.img file

After editing the initrd.img.uncompressed file, you must unmount it:

```
umount mnt_point/
```

Compressing the uncompressed initrd.img file

1. After unmounting the file, compress it:
   ```
   gzip -v -9 initrd.img.uncompressed
   ```
   The resulting file is called initrd.img.uncompressed.gz.
2. Rename initrd.img.uncompressed.gz with the original name:
   ```
   mv (or cp) initrd.img.uncompressed.gz initrd.img
   ```

The new initrd.img is ready to be used.

Technical support

In this section

Reference documentation

For issues or problems not addressed by this guide, refer to the following resources for more information:

- The SmartStart Scripting Toolkit website (http://www.hp.com/servers/sstoolkit)
- The Red Hat Linux website (http://www.redhat.com)

Toolkit support

E-mail support for the SmartStart Scripting Toolkit is available from the HP support website (http://atwnt947.external.hp.com/fd2/email_form.cfm?countrycode=US&langcode=en&sni=437764&lang=en&cc=us).

HP contact information

For the name of the nearest HP authorized reseller:

- In the United States, call 1-800-345-1518.
- In Canada, call 1-800-263-5868.
- In other locations, refer to the HP website (http://www.hp.com).

For HP technical support:

- In North America:
  - Call 1-800-HP-INVENT (1-800-474-6836). This service is available 24 hours a day, 7 days a week. For continuous quality improvement, calls may be recorded or monitored.
  - If you have purchased a Care Pack (service upgrade), call 1-800-633-3600. For more information about Care Packs, refer to the HP website (http://www.hp.com).
- Outside North America, call the nearest HP Technical Support Phone Center. For telephone numbers for worldwide Technical Support Centers, refer to the HP website (http://www.hp.com).

Acronyms and abbreviations

BIOS

Basic Input/Output System

ConRep

Configuration Replication utility

DHCP

Dynamic Host Configuration Protocol

IP

Internet Protocol

NFS

network file system

POST

Power-On Self Test

PXE

preboot eXecution environment

RAM

random access memory

TFTP

Trivial File Transfer Protocol

USB

universal serial bus

XML

extensible markup language

Index

P

R

S

T

U

V

W