



Table of Contents

Abstract.....	2
Executive summary.....	2
Section I: Installation and recovery booting processes	3
HP9000 systems:	3
Booting from a CD or DVD	3
Booting from a CD or DVD with HP-UX 11i v2 and greater	4
Booting from a tape.....	4
Booting from a LAN.....	4
Booting a HP9000 virtual partition	7
HP Integrity systems:	9
Booting from a CD/DVD	9
Booting a vPars system.....	10
Booting from a tape.....	10
Booting from a LAN.....	15
Section II: Troubleshooting the secondary loader	17
Debugging HP9000 booting problems:	17
Errors booting HP9000 systems without virtual partitions	17
Debug level logging	18
Setting debug level logging on HP Integrity systems.....	19
Additional debug level logging for both HP9000 and HP Integrity systems	19
Troubleshooting early install kernel boot problems on HP Integrity systems	21
The “pre_reboot_cmd” command hook.....	22
Network recovery/installation without network boot helpers:.....	23
Booting from tape (HP9000 only).....	24
Booting from CD/DVD (HP9000 and/or HP Integrity systems)	25
The <i>bootsys(1m)</i> command	25
Customizing the media	25
Section III: Network boot helpers	26
Glossary	27
For more Information.....	28

Abstract

Ignite-UX for HP-UX addresses the need for system administrators to perform operating system installations, deployment, and recovery, often on a large scale. It provides the means for creating and reusing standard operating system configurations. Additionally, Ignite-UX delivers the ability to archive operating system configurations, and to use these archives to replicate systems, with the added benefit of speeding up the process. Ignite-UX also permits various customizations, and is capable of both interactive and unattended operating modes.

Executive summary

The necessity to debug problems encountered by the specific secondary loader (SSL) or bootstrap program may occur. This white paper is intended to provide additional information to augment the *hpux(1M)* and *hpux.efi(1M)* documentation and aid you in advanced troubleshooting of secondary loader problems.

It is important to understand how Ignite-UX boots a system in preparation for installing HP-UX, which differs from the normal system boot process. The first section of this paper discusses the installation booting process for HP9000 and HP Integrity systems. The second section presents troubleshooting techniques to assist you in resolving secondary loader issues.

Section I: Installation and recovery booting processes

This section provides a high-level outline of the system booting process that occurs prior to installation or recovery of the HP-UX operating system using Ignite-UX.

HP9000 systems:

Booting from a CD or DVD

System booting from a CD or DVD is as follows:

1. The command to boot from the CD or DVD is entered at the boot console.

Please consult the documentation appropriate to the system for information on how to boot from CD/DVD.

2. The firmware reads the LIF from the boot device and copies the ISL from the location on the device designated by the ISL LIF entry.
3. ISL is invoked.

It is assumed that you did not answer **y** when asked, "Do you want to interact with IPL?" The booting process is different if you answer **y**, since ISL does not read and automatically execute the `AUTO` file.

4. ISL reads the `AUTO` file from the LIF on the boot device.

For installation media, the typical contents of the `AUTO` file are `hpux (; 0) : INSTALL`, while for a normal system boot the contents are `hpux (; 0) /stand/vmunix`. ISL completes the `(; 0)` information read from the `AUTO` file to include the full device information.

5. ISL reads the secondary loader and invokes it with the options described in the previous step.

The secondary loader, `hpux`, reads the file name part of the boot command. The `:` (colon) in the boot command means the kernel is to be read from the LIF on the boot device, whereas in a normal boot command the file name starts with `/` (forward slash) to indicate it is to be read from a file system.

Note:

ISL might also alter the name of the installation kernel to prepend a `w` if the PDC indicates that the system can only run a 64-bit kernel. The translation to add a "W" was new at HP-UX 11.0.

6. The secondary loader loads the installation kernel and invokes it.

By its name the kernel knows that it has been loaded as `INSTALL` (or one of the `[W|V|I]INSTALL` kernels), so it must act as an installation kernel. Early in the boot process it has to load the corresponding `[W|V|I]INSTALLFS` file system into a RAM disk.

The installation file system is very limited in size, approximately 8MB for HP9000 systems.

7. The kernel loads the installation file system from the LIF into the RAM disk.

The kernel loads the installation file system by executing calls to the secondary loader to load the file system. Since the kernel has not initialized the input/output (I/O) subsystem, it has no ability to access a disk device to read from the installation media.

8. The kernel starts `init` from the installation file system that was just loaded.
9. Lastly, `init` starts the installation or recovery process.

Booting from a CD or DVD with HP-UX 11i v2 and greater

The Ignite-UX C.6.0.x version, which supports HP-UX 11i v2 (B.11.23) and greater, contains installation kernels specific to each supported release of the operating system. In order to select the correct kernel for your client at network installation time, the bootloader was enhanced to understand additional `AUTO` file syntax.

This additional syntax causes `hpux` to prompt you for the target HP-UX release for the client, for example B.11.23, and then it retrieves the installation kernel appropriate for that release. The default prompts and other options of the additional `AUTO` file syntax can be modified using the `auto_adm` command introduced in the Ignite-UX C.6.0.x version. For more information, refer to `auto_adm(4)`.

Booting from a tape

The only difference between booting from media and tape is that the tape is a serial device so it requires rewinding and forwarding to access different parts of the tape to locate the files the LIF points to on the tape.

Booting from a LAN

System booting from a LAN is as follows:

1. The command to boot from the LAN is entered at the BCH prompt.
The IP Address of the server to boot from is optional, and if given, only responses from that server are accepted.
2. The client performs a `bootp` broadcast in an attempt to find an Ignite-UX server. The `bootp` broadcasts do *not* go outside of the local LAN segment. If the Ignite-UX server is on a different LAN segment and there is no local boot helper, the system fails to boot
3. The Ignite-UX server provides a `bootp` response to the client.

If the `install` option was given to the BCH to boot from the LAN, the `bootp` protocol is still used. However, the User Datagram Protocol (UDP) ports used are 1067/1068 instead of the standard 67/68. These custom ports for installation allow you to install HP9000 HP-UX systems without any interference from other systems. Only the Ignite-UX daemon `instl_bootd` is set to listen on these ports for `bootp` traffic, so only `instl_bootd` responds. All of the necessary client boot information is provided in the `bootp` response.

The response from `instl_bootd` returns `/opt/ignite/boot/boot_lif` as the boot file.

4. The client retrieves the boot file from the Ignite-UX server using firmware implemented `tftp`, and then extracts the ISL.
5. The ISL on the client reads the `AUTO` file from the boot LIF.
6. The secondary loader reads from the boot LIF and is started with the information read from the `AUTO` file.

At this point, both ISL and the secondary loader are set to read from the LAN device. Using a `/` (forward slash) in the file name when designating a LAN device causes `tftp` to access a file, or if a disk is designated to read from a file system.

In a default Ignite-UX installation, the `AUTO` file read by a system booting via the network should contain `hpux (;0)/boot/INSTALL`. Due to the way NFS diskless was originally designed, the secondary loader translates the kernel path by taking the path of the boot file, `/opt/ignite/boot/boot_lif`, and removing the last two path elements, leaving `/opt/ignite`, and then by appending the name from the `AUTO` file, resulting in `/opt/ignite/boot/[W|V|I]INSTALL`¹. This design allows the kernels to be in client-specific directories without needing each LIF volume to have a different `AUTO` file.

7. The secondary loader loads the `INSTALL` kernel and executes it.
8. The `INSTALL` kernel loads the RAM file system, `INSTALLFS`.

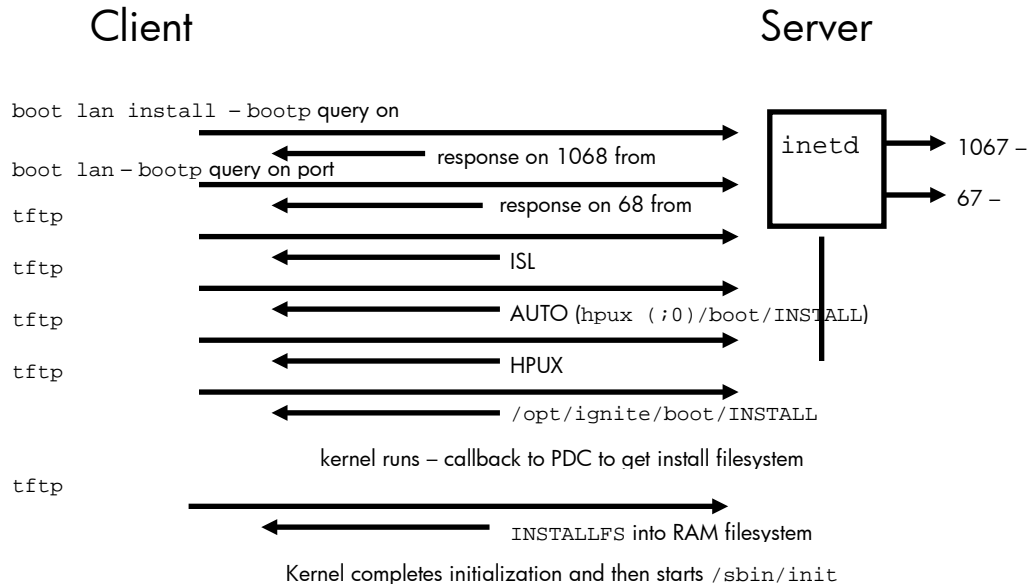
The kernel loads the installation file system by making calls back into secondary loader to load the file system. At this stage, the kernel cannot read anything from the Ignite-UX server. The kernel does not have any initialized LAN interfaces from which the installation file system can be read.

The kernel starts `init` from the installation file system just loaded. The `init` process then starts the installation or recovery process.

¹ As of Ignite-UX version C.6.0.x, all installation kernels are located in release-specific directories so the boot kernel for an 11.11 PA-RISC system is `/boot/Rel_B.11.11/INSTALL` not `/boot/INSTALL`.

Figure 1 illustrates the process of booting from the network by HP9000 systems:

Figure 1

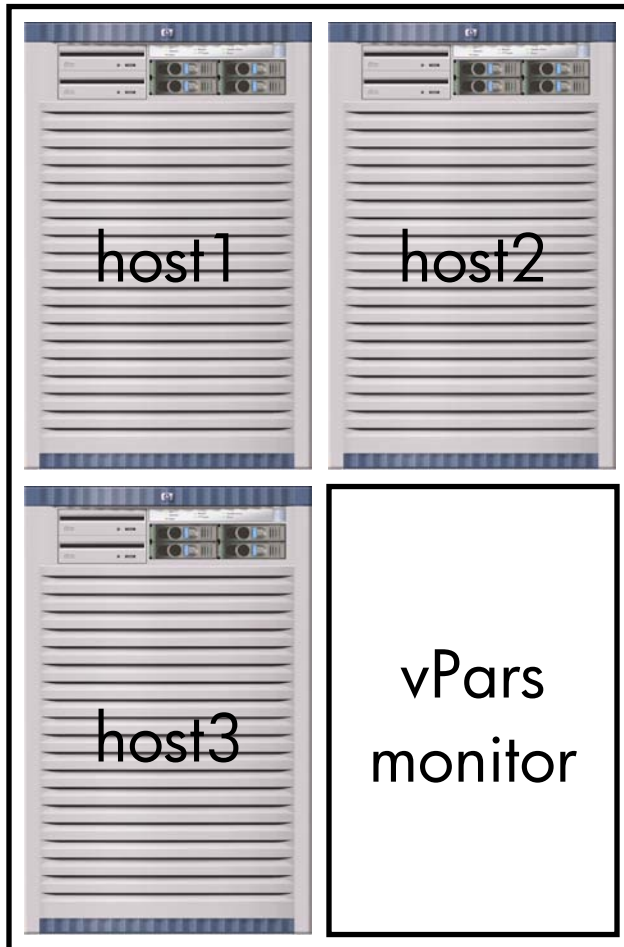


Note:

The method that very old HP9000 systems used to boot over the network, Remote Maintenance Protocol (RMP) via `rbootd(1m)`, is not discussed in this white paper.

Booting an HP9000 virtual partition

This example is illustrated with an rp8400 system that has been split into three Virtual Partitions (vPars). The virtual partition, *host1*, has already been installed with the vPars software and the vPars monitor is running. The remaining two virtual partitions have been configured but not installed.



The virtual partition *host1* has Ignite-UX installed previously and has depots that contain the HP-UX B.11.11 OE, the vPars software, and enablement patches. The required Ignite-UX configuration is complete.

You execute the `vparboot` command to start one of the other virtual partitions so that you can begin the operating system installation using Ignite-UX with the following command:

```
vparboot -p host2 -I WINSTALL2
```

The `-I` option of `vparboot(1M)` by default obtains files from `/opt/ignite/boot` if no path is specified.

In addition, the `-I` option allows a host name or IP address with the kernel. When used in this way, the `vparboot` command will `tftp` the installation kernel and file system from the remote Ignite-UX server to the local vPars partition, and then the installation kernel is passed to the vPars monitor.

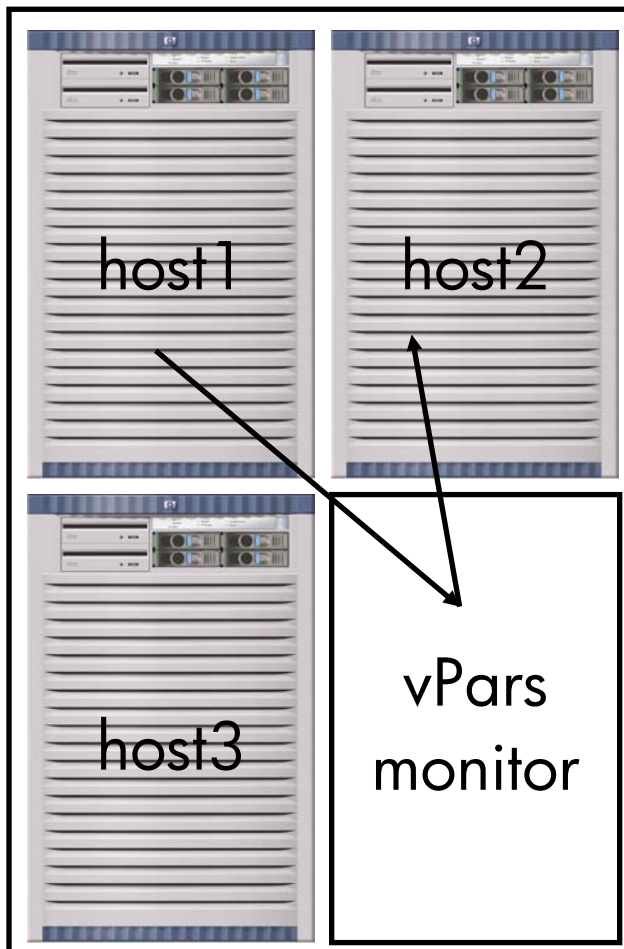
```
vparboot -p host2 -I \  
10.0.0.5,WINSTALL
```

The monitor uses the installation kernel and file system to boot the vPar³. The monitor puts the `WINSTALL` and `WINSTALLFS` into the memory owned by the partition that is booting and then starts the partition running.

This boot process is completely different from the way that a HP9000 system would normally boot.

² As of Ignite-UX version C.6.0.x, the `vparboot` command will not know about the new locations of installation kernels within the Ignite-UX product. You must give the full path to the installation kernel if you are using vPars and Ignite-UX version C.6.0.x or later. For example, HP-UX 11i v1 (B.11.11) requires `/opt/ignite/boot/Rel_B.11.11/WINSTALL` and HP-UX 11i v2 (B.11.23) requires `/opt/ignite/boot/Rel_B.11.23/WINSTALL`.

³ This description indicates that if there are no partitions running (just the monitor), it is not possible to use a network-based Ignite-UX server from just the monitor prompt since the monitor is not capable of communicating over the network.



The vPars boot process is as follows:

1. The `vparboot` command locates the installation kernel and file system.
2. The `vparboot` command transfers the installation kernel and file system to the vPars monitor.
3. The monitor places them into the memory that is used by the host being started.
4. The kernel in the virtual partition is started.

Once the vPar host2 has started and is running the kernel in the installation file system, it must be able to contact the Ignite-UX server.

This is accomplished either by the client contacting a DHCP server to obtain an IP address, or by entering the information on the client console designating the IP address and Ignite-UX server to use (and possibly a default route).

To advance the booting process, the client must be connected via the network to the Ignite-UX server.

If the booting partition cannot contact the Ignite-UX server because there is no network connectivity, the installation cannot succeed.

Remember that being able to boot a virtual partition from an Ignite-UX server does not mean there is network connectivity between the two systems, so consider the following:

- When you choose a network interface on the booting vPars system, it ***must*** have network connectivity to the Ignite-UX server.
- If you experience problems, troubleshoot the connectivity issue as a network problem.

HP Integrity systems:

Booting from a CD/DVD

System booting from a CD/DVD is as follows:

1. Enter the commands to boot from an appropriate DVD or use the boot manager menu (if configured).

Please consult the documentation appropriate to the system for information on how to use the EFI or how to configure the boot manager menu.

2. The EFI reads the CD or DVD media to determine if it is an El Torito format CD or DVD (HP Integrity systems can only boot from El Torito format CD or DVD media).
3. Invoke the bootloader from command line or use the boot manager menu to start the installation process.

The bootloader on the CD or DVD is invoked from the EFI shell by changing partitions to the EFI partition for the CD or DVD, and then entering `install`. If no other boot devices are found before the CD or DVD is found, the EFI automatically boots via the `/STARTUP.NSH` EFI script on the CD or DVD. If you change EFI partitions to the CD or DVD, the `STARTUP.NSH` EFI script should automatically start the boot process.

4. The bootloader, `install.efi`, reads the `AUTO` file from the LIF. It is possible to interact with the bootloader.

The HP Integrity installation media used to install HP-UX contains a LIF, just as with HP9000 media. However, the EFI does nothing with this LIF because its purpose is just to allow common code to be used to install HP-UX on HP9000 and HP Integrity systems. The `AUTO` file contains `boot : IINSTALL`; the `:` (colon) designates that the installation kernel `IINSTALL` is read from the LIF on the boot device. Only the installation bootloader reads LIF files; the EFI does not.

5. The bootloader executes the contents of the `AUTO` file; however, both the installation kernel (`IINSTALL`) and installation file system (`IINSTALLFS`) are loaded by the bootloader at the same time before the execution of the kernel starts.

Additional tasks are not available with the HP Integrity systems bootloader as with the secondary bootloader on HP9000 because the state of EFI is undefined once the operating system starts running. The bootloader must load the installation kernel and file system before passing control to the kernel. The bootloader passes the location of the loaded installation file system to the kernel and then the kernel completes its initialization.

6. The installation continues at this stage with the kernel running `init` from the installation file system loaded by the bootloader.

Booting a vPars system

When you boot HP Integrity servers running vPars you are using EFI to perform the boot. This is different from the way vPars works on HP9000 systems. Please consult the other areas in this white paper for information on how HP Integrity systems boot. Please consult the vPars release notes for the version you are running to see if there are any limitations on what boot methods are allowed.

Booting from a tape

Some HP Integrity servers released after June, 2006 have support for native tape boot. Some systems available before that time received firmware updates that enable tape boot. However, not all systems will receive firmware updates to enable tape boot. Also, only some SCSI HBAs will be supported for tape boot. Fibre Channel and SAS HBAs do not support native tape boot on all systems. Refer to table "Tape Drive Capabilities by Protocol" to see where they are supported.

For HP Integrity systems that do not support tape boot, or those that do with tape drives connected to HBAs that do not support tape boot, Ignite-UX will still continue to support tape recovery support via a method called Tape Recovery with No Tape Boot Support, Dual-Media Recovery, or Two-Step Media Recovery. This method requires that a tape drive be connected to the client, that you boot the client from the HP-UX installation media, and that you choose options to continue installation or recovery from tape. For more information, refer to the "Tape Recovery with No Tape Boot Support - Two-Step Media Recovery" section of the *Ignite-UX Administration Guide* at:

<http://www.docs.hp.com/en/IUX/infolib.html>

The information presented in this section is different from other sections in this white paper. Because native tape boot is new functionality on HP Integrity servers and there are combinations that allow and prohibit native tape boot, you need to review the following four tables:

- The first table will show if the system will support tape boot.
- The second will show what add-on SCSI HBAs a tape drive must be connected to in order to allow tape boot.
- The third shows tape drive capability according to protocol.
- The fourth will show the version of Ignite-UX required to support bootable recovery tapes.

On systems supporting native tape boot, the parallel SCSI interfaces on the core I/O are also supported for tape boot.

Important:

HP does not support native tape boot on interfaces shared with internal or external disks. This restriction applies to both built-in parallel SCSI interfaces⁴ and add-on cards.

⁴ If the parallel SCSI interfaces on the core I/O of your system are used for the internal disks and also have an external connector, you cannot use these interfaces for native tape boot, as no disks can be present on a parallel SCSI interface used for native tape boot.

The following table shows the firmware requirements for various HP Integrity systems in order to support tape boot. All systems support two-step media recovery.

Native Tape Boot by System Model		
Model⁵	Native Tape Boot	Minimum Firmware Revision
rx1600, rx1620	Yes	04.05 or later
rx2600, rx5670	No	NA
rx2660	Yes	01.03 or later
rx2620, rx4640	Yes	04.05 or later
rx3600, rx6600	Yes ⁶	Any – supports tape boot from first release
rx7620, rx8620	Yes	5.0 or later
rx7640, rx8640	Yes	Version 2 or later
Superdome (sx1000 based)	Yes	7.0 or later
Superdome (sx2000 based)	Yes	5.5 or later
cx2620	Yes	Any – supports tape boot from first release
HP Integrity blades	No ⁷	NA

⁵ Future HP Integrity systems should support tape boot when they are released and will not be documented in this table, future systems that will not support tape boot will be listed in this table.

⁶ These systems have no parallel SCSI HBAs on the core I/O. An add-on parallel SCSI HBA is required for native tape boot.

⁷ Itanium(R)-based Blades have no built-in parallel SCSI interfaces and it is not possible to have an add-on parallel SCSI HBA.

The following table shows the HBAs that are supported for native tape boot along with the minimum EFI firmware driver revision.

Native Tape Boot by SCSI Model	
Parallel SCSI Model	Minimum Required EFI firmware Revision
A6828A (Single port U160)	1.03.01.00
A6829A (Dual port U160)	1.03.01.00
A7173A (Dual port U320)	1.04.02.00
AB290A (U320 Gigabit Ethernet combo)	1.04.02.00

The following table illustrates the tape drive capabilities of HP systems according to protocol.

Tape Drive Capabilities by Protocol		
	Native Tape Boot	Two-Step Media Recovery
Parallel SCSI	Yes	Yes
SAS	Yes (see note 2)	Yes (see note 1)
FC	Yes (see note 2)	Yes

Note 1: Requires HP-UX 11i v3 HWE0803 or later.

Note 2: Is available on HP Integrity 2 systems and requires HP-UX 11iv3 HWE1103 or later.

Ignite-UX as of version C.6.8 creates tapes that conform to the UEFI 2.0 specification for bootable tapes; only tapes created by this version of Ignite-UX onwards are natively bootable. The following table shows the format of the tapes created by different versions of Ignite-UX and if they are bootable on HP Integrity servers:

Native Tape Boot by Ignite-UX Version and Tape Format		
Ignite-UX Version	Format	Native Tape Boot
Up to and including C.6.2	HP9000 format, LIF in first file followed by archive.	No
C.6.3 to C.6.7	New format, but not conformant to UEFI 2.0 specifications	No
C.6.8 onwards	New format, conformant to UEFI 2.0 specifications	Yes

Without going into detail about the new format for bootable tapes on HP Integrity servers, the tape contains the following files (this assumes the recovery tape does not span multiple tapes):

Recovery Tape Contents			
File Number	Contents	File Number	Contents
1	ANSI Label containing VOL1 header	13	ANSI Label
2	EFIBOOTHPUX	14	IINSTALL
3	ANSI Label	15	ANSI Label
4	ANSI Label	16	ANSI Label
5	BOOTLOADER	17	IINSTALLFS
6	ANSI Label	18	ANSI Label
7	ANSI Label	19	ANSI Label
8	FPWSAEFI	20	HPUXIUXLIF
9	ANSI Label	21	ANSI Label
10	ANSI Label	22	ANSI Label describing archive
11	AUTO	23	Archive
12	ANSI Label		

When reading the above table, please note the numbering of the files going down the columns - the files appear on the tape in the order of the file numbers given in the table.

Starting with Ignite-UX version C.7.7, the `ansitape` command is recommended for file access instead of counting files.

To boot the tape, the EFI loadfile interface is used to load the file called "BOOTLOADER" from the tape (it is found via the ANSI label introducing the file). This file contains an executable EFI program; once loaded it is set running.

This program is responsible for loading everything else required from the tape to start the kernel (along with its install file system) and hence set the recovery going.

The kernel completes its initialization. The installation continues with the kernel running `init` from the installation file system loaded by the bootloader.

The Ignite-UX Administration Guide contains information on how to configure boot manager menu options for tape boot on HP Integrity servers that support it.

Booting from a LAN

System booting from a LAN is as follows:

1. At the EFI boot manager menu, select a LAN device to boot from, or use the `lanboot` command at the EFI shell prompt.

Please consult the documentation appropriate to the system for information on how to use EFI or how to configure the boot manager menu.

2. The firmware sends a DHCP request with an Itanium®-based `class-id` set.

The DHCP server on HP-UX 11i v2 (B.11.23) and later replies to Itanium®-based PXE boot requests, if configured appropriately.

For more information regarding how to configure DHCP on an HP-UX B.11.23 or later system to respond correctly to Itanium®-based PXE DHCP requests, refer to the *Ignite-UX Administration Guide*.

If a DHCP PXE response to a PXE boot request includes a `class-id`, it will attempt to contact a PXE proxy server on the responding host. HP-UX does not currently provide a PXE proxy server, so a DHCP PXE response including a `class-id` will cause a network boot to fail. HP-UX releases prior to HP-UX 11.11 do not have the appropriate option to allow you to configure DHCP to exclude a `class-id` in the DHCP PXE response. On HP-UX 11.11 (with PHNE_28762 or later installed) and on HP-UX 11.23 and later from first release, the `ncid` keyword prevents the `class-id` from being included in the PXE response. Other keywords were also added at HP-UX 11.23 to make configuring DHCP for booting HP Integrity systems easier. One of these keywords is `re` – without this keyword it is significantly harder to create a `dhcp_device_group` in `/etc/dhcptab` that supports only anonymous booting of Itanium clients, as you cannot use basic regular expressions to match the Itanium PXE DHCP class id.

3. The DHCP response directs the system to download `/opt/ignite/boot/nbp.efi` (“nbp” means “network bootstrap program”) to continue booting.
4. After the `nbp.efi` process is downloaded and executed, it downloads and executes the bootloader, `hpux.efi`.
5. The bootloader downloads the `AUTO` file, `/opt/ignite/boot/AUTO`.

The contents of the `AUTO` file control the operation of the bootloader and usually contain the `boot IINSTALL` command.

6. The bootloader downloads the installation kernel, `IINSTALL`, and the installation file system, `IINSTALLFS`, and executes the kernel.

HP Integrity systems cannot ask their bootloader to perform additional tasks (like the secondary bootloader does on HP9000 systems). Because of this, the bootloader must load the installation

kernel and the installation file system before passing control to the kernel. The bootloader tells the kernel where it loaded the installation file system. This is different from how booting works on HP9000 systems.

7. The kernel completes its initialization. The installation continues with the kernel running `init` from the installation file system loaded by the bootloader.

Section II: Troubleshooting the secondary loader

Debugging HP9000 booting problems:

Errors booting HP9000 systems without virtual partitions

It is likely that at some point errors will appear when booting a HP9000 system (without Virtual Partitions) that are similar to the following errors, though the actual errors vary across systems (the amount of information provided varies and you might or might not see a hex dump).

```
ENTRY_INIT
```

```
Status = -7
```

```
Failed to initialize
```

The status value helps you isolate and correct the problem.

Table 1 defines some of the errors you might encounter:

Table 1: Errors

Status Value	Description
-3	Cannot complete call without error – An error of unknown type prevented the call from completing correctly.
-4	Unrecoverable hardware error – An unrecoverable hardware error has occurred between the processor PDC and IODC on the target controller or device.
-5	Unrecoverable data error - ENTRY_INIT encountered an error while transferring data to/from the target controller or device.
-6	Illegal device address -- The device address supplied to ENTRY_INIT is invalid; it could never be a valid device address.
-7	Nonexistent device - The device address specified is a valid device address. However, it points to either a device that is not installed and/or not responding.
-8	Module/device not ready - The module or device is not ready to be initialized or tested. A module may not be ready because it is still performing its self-test after reset or power-on.
-9	Cannot locate a console device or boot device - The search could not locate a console device or boot device on the module.
-10	Invalid argument.
-13	Protocol error - A protocol violation was encountered on the module-device connection while transferring data to or from the device, for example, a SCSI protocol error.
-14	Invalid or uninitialized SCSI parms in NVRAM – When initializing a SCSI module, either the SCSI parameters in the system NVRAM or module NVRAM were not initialized, or they did not agree.

Examples:

- A value of -7 for the `ENTRY_INIT` error can mean different things for different devices. For a LAN device it could mean that nothing responded to its boot request. For a disk device, it could mean that the device does not exist (for example, the device path is incorrect).
- A value of -8 for the `ENTRY_INIT` error could mean that the disk you are attempting to boot from has not spun up yet, the tape is still loading, or the CD-ROM may have only just been inserted into a drive and is in the process of becoming ready.

Debug level logging

There are times when you encounter errors when using Ignite-UX when you may be asked by HP Support to enable debug level logging within Ignite-UX to gather more detailed information to help with troubleshooting.

The following steps enable debug level logging during an installation or recovery session:

1. Boot the system from the media and answer **y** when queried if you wish to interact with the IPL.
2. At the ISL prompt, enter the appropriate command from Table 2 for the media you used to boot:

Table 2: Command to Enable Debugging

Boot Media	Command
CD or DVD and tape	<code>hpux -i3 (;0):INSTALL</code>
Network before Ignite-UX C.6.0.x	<code>hpux -i3 (;0)/boot/INSTALL</code>
Network after Ignite-UX C.6.0.x	<code>hpux -i3 (;0)/boot/Rel_B.xx.yy/INSTALL</code>

Other older HP9000 systems may require different syntax (for example the V series) so consult the operational documentation for the system for specifics regarding how to boot and provide flags to the bootloader.

There are times when you might experience bootloader problems such as the loading of a kernel or installation file system. At these times, HP Support might ask you to enter a loader option from Table 3 for the media you used to boot from, on systems running HP-UX 11i v1 (B.11.11) and greater:

Table 3: Loader Option Commands for HP-UX 11i v1 (B.11.11) and above

Boot Media	Command
CD or DVD and tape	<code>hpux -D0xffffffff -B0xffffffff (;0):INSTALL</code>
Network before Ignite-UX C.6.0.x	<code>hpux -D0xffffffff -B0xffffffff (;0)/boot/INSTALL</code>
Network after Ignite-UX C.6.0.x	<code>hpux -D0xffffffff -B0xffffffff (;0)/boot/Rel_B.<xx.yy>/INSTALL</code>

Setting debug level logging causes the system to print information about the instructions that the secondary loader is executing. Typically this information is useful only to HP Support. Enable these loader debug options only when requested to do so by HP Support.

Setting debug level logging on HP Integrity systems

There are times when you encounter errors when using Ignite-UX when you may be asked by HP Support to enable debug level logging within Ignite-UX to gather more detailed information to help with troubleshooting.

The following steps enable debug level logging during an installation or recovery session:

1. At the EFI shell, choose one of the following options:
 - If you have configured an option to boot from the CD or DVD drive, select that option. The system starts to boot automatically.
 - Otherwise, quit to the EFI shell, and then change your EFI partition to the CD drive containing the installation media. If the system does not automatically boot, type **startup** and then press **Enter**.

2. Interrupt the boot sequence at the message `Press Any Key to interrupt Autoboot` by pressing a key.

```
fs0:\> startup
...
HP-UX Bootloader for IA64 Revision 1.713
Press Any Key to interrupt Autoboot
AUTO ==> boot :IINSTALL
Seconds left till autoboot - 7
type 'help' for help
HPUX>
```

3. At the HPUX> prompt, enter **boot :IINSTALL -i3** so that the bootloader loads the kernel and installation file system into memory, and then passes control to the kernel.

Debug messages start to appear once the kernel has started the `init` process and the `init` process sets Ignite-UX running.

Additional debug level logging for both HP9000 and HP Integrity systems

The information in this section is applicable to HP9000 and HP Integrity systems.

In addition to booting with the `-i3` flag (how you do this depends on the type of system you are using), you can enable debug level logging during an installation or recovery before the Ignite-UX Graphical User Interface (GUI)⁸ program starts.

On the screen that provides the option to **Install HP-UX**, select **Advanced Options**. In the screen that appears, scroll down to debug level and press **Enter** to select **level 3**. The same amount of detail is provided for levels 3, 4 and 5. This action starts debug level logging exactly as it would from boot except that debug messages will appear once you have continued with the installation or recovery (that is, selected **Install HP-UX** from the main menu) instead of when `init` starts.

Tip:

Unless told to use a different debug level by HP Customer Support or you need to enable debug level logging, always use debug level 3.

If you are experiencing problems during initial device discovery (for example, some hardware is not found), you must enable debug level logging from boot time. For more information, see Debugging HP9000 booting problems: or Setting debug level logging on HP Integrity systems.

You may choose to enable debug level logging during an installation or recovery to avoid filling the RAM file system to capacity when debug level logging is enabled from boot (it is more likely to happen the more hardware you have attached to a system). A full RAM file system can cause an installation or recovery session to fail if the RAM file system fills before the Ignite-UX GUI starts and causes cosmetic issues with the GUI. However, it should not cause recovery problems after the GUI is invoked.

Ignite-UX version C.7.0 and later also allows you to define the configuration variable `_hp_debug_level` in the installation file system. If you set the debug level this way, it should only be set to an integer value between 1 and 5. You will not see all the debug messages you would see if you set the debug level via the boot loader, however this setting does take effect a very short time after boot. Setting this configuration variable in any place other than the installation file system has no effect on the debug level.

Important:

When using Ignite-UX versions before version C.7.0 you can set the debug level to a number between 1 and 9. With Ignite-UX version C.7.0 and above you must only use debug levels between 1 and 4 when providing a debug level to the boot loader⁹.

⁸ The Ignite-UX Graphical User Interface helps you manage multiple, simultaneous installation or recovery sessions using tabs and dialog boxes for task navigation.

⁹ Do not boot using debug level 5 or above unless you have been requested to do so by HP Customer Support as part of a problem investigation, you will be provided explicit instructions on what to do if you are asked to run in debug levels 5 or 6. You will not be asked to boot with debug levels 7 to 9.

Troubleshooting early install kernel boot problems on HP Integrity systems

When you encounter the rare occurrence of a kernel hang or delay during boot on HP Integrity systems, you can localize where the problem might be occurring. There is a boot loader option `-v` that directs the kernel to print out what is happening in the kernel during its initialization.

The output looks like the following:

```
Init[ 0]:0xe0000000015ealc0: - DONE (0x000000000a8b iticks)
Init[ 1]:0xe0000000017c4f90: - DONE (0x000000000dd2d iticks)
Init[ 2]:0xe000000001856240: - DONE (0x0000000000b4 iticks)
Init[ 3]:0xe000000000557520: - DONE (0x0000000004c62 iticks)
Init[ 4]:0xe000000000b1c8e0: - DONE (0x00000000006cf iticks)
Init[ 5]:0xe00000000133d0c0: - DONE (0x000001faa603 iticks)
Init[ 6]:0xe00000000085a750: unwind_init - DONE (0x0000000001e93 iticks)
Init[ 7]:0xe0000000016093c0:kwdb_wait_for_consol - DONE (0x000000000181 iticks)
Init[ 8]:0xe000000000b14c80: fpswa_init - DONE (0x0000000001915 iticks)
Init[ 9]:0xe000000000b0be90: ex_init - DONE (0x00000000000e9 iticks)
Init[10]:0xe000000000af9720: fw_psm_init - DONE (0x000000000d0a8 iticks)
Init[11]:0xe0000000010599d0: io_memory_config - DONE (0x00001cbf9196 iticks)
Init[12]:0xe0000000015ea9d0:kwdb_boot_init_lan_s - DONE (0x0000000000f6 iticks)
Init[13]:0xe000000000edfaa0: GetPainfo - DONE (0x000000010ae4 iticks)
...
Init[105]:0xe0000000005bb6b0: vasinit - DONE (0x0000000adddc iticks)
Init[106]:0xe0000000005cd160: preginit - DONE (0x0000000025c9 iticks)
Init[107]:0xe0000000005c1480: reginit - DONE (0x0000000006ald iticks)
Init[108]:0xe0000000004ed200: pgcopy_util_init - DONE (0x00000000019c iticks)
Init[109]:0xe000000000ee1780: pdk_malloc_nomore - DONE (0x000001b5cd82 iticks)
Init[110]:0xe000000000efc620: vm_pre_main_init - DONE (0x00003e2929ba iticks)
Init[111]:0xe000000000f18ee0: miscmeminit - DONE (0x000000000232 iticks)
Init[112]:0xe000000000f2b220: vhpt_init2 - DONE (0x0000003f55a8 iticks)
Init[113]:0xe0000000017a45c0: treg_copy_trs - DONE (0x00000000cb9a iticks)
Init[114]:0xe0000000008b8d40: p0init - DONE (0x00000001782d iticks)
Init[115]:0xe00000000081c190: swmain - Invoked 0xe000000100105040
```

Important:

The format and functions listed might change with patch and HP-UX revisions, but the information provided will remain generally the same.

The output is dynamic. On the console, you will see:

```
Init[ 11]:0xe0000000010599d0: io_memory_config - In Progress ...
```

When the function completes, the “In Progress ...” gets erased and replaced with:

```
Init[ 11]:0xe0000000010599d0:    io_memory_config - DONE (0x00001cbf9196 iticks)
```

To enable this output, you should perform the following actions when booting the IINSTALL install kernel (these instructions assume that you are booting from a CD or DVD):

1. At the EFI shell, choose one of the following options:
 - If you have configured an option to boot from the CD or DVD drive, select that option. The system starts to boot automatically.
 - Otherwise, quit to the EFI shell, and then change your EFI partition to the CD drive containing the installation media. If the system does not automatically boot, type **startup** then press **Enter**.
2. Interrupt the boot sequence at the message Press Any Key to interrupt Autoboot by pressing a key.

```
fs0:\> startup
...
HP-UX Bootloader for IA64 Revision 1.713
Press Any Key to interrupt Autoboot
AUTO ==> boot :IINSTALL
Seconds left till autoboot - 7
type 'help' for help
HPUX>
```

3. At the HPUX> prompt, enter **boot :IINSTALL -v** and then press **Enter** so the bootloader loads the kernel and installation file system into memory, and then passes control to the kernel.

Note that the information about what routines are being called during initialization will appear only on the console of the system.

The “pre_reboot_cmd” command hook

The following information about the command hook “pre_reboot_cmd” is not intended for general use and should only be used when requested by HP support personnel. This new keyword is only available from Ignite-UX version C.7.2 onwards and it provides no functionality that is useful for installing systems.

The purpose of the pre_reboot_cmd command hook is to execute the given command early in an installation or recovery session prior to a reboot. The reason this command hook is documented in this white paper is to provide convenient instructions on how to use it when you need it.

The most common reason HP might ask you to use this command hook is to provide a shell prompt after an error that would have led to a reboot and failed installation or recovery. The shell can then be used to gather information that will aid in the investigation and resolution of the problem.

The following is an example of how to use the pre_reboot_cmd command hook to run a shell after an error that would have caused a reboot early during an installation or recovery session.

To enable this feature, the `pre_reboot_cmd` command hook must be specified in the install file system. The following is an example showing how to set it:

1. Save the default contents of the install file system to a temporary file

```
# instl_adm -d > /var/tmp/myconfig
```

2. Edit the file:

```
# vi /var/tmp/myconfig
```

At the end of the file, add a line that looks like (note that the location of spaces and other information in the command is important and it must be preserved exactly "as is"):

```
pre_reboot_cmd += "/sbin/sh -c /sbin/sh</dev/console>/dev/console\ 2>&1 ;"
```

3. Apply the changes to the install file systems

```
# instl_adm -f /var/tmp/myconfig
```

4. Verify the contents of the install file system to ensure it contains the correct information:

```
# instl_adm -d
```

```
...
```

```
pre_reboot_cmd += "/sbin/sh -c /sbin/sh</dev/console>/dev/console\ 2>&1 ;"
```

Important:

The above commands apply this change to all the install file systems present on the system. This change applies to network booted systems; you might need to make additional customizations to have this apply to CD/DVD or tape media you create.

When using the shell provided by this command hook, it is important to note that very few commands will be available - you can rely on the presence of `/sbin/cat` but not `/sbin/ls` or `/sbin/ll`. To view the contents of a directory run the following command:

```
# echo /tmp/*
```

Where `/tmp` is the directory you want a list of files from.

Network recovery/installation without network boot helpers:

There are several methods for doing this. Note that some methods work only on HP9000 systems or HP Integrity systems.

As of Ignite-UX version C.6.2, an example script is provided that can create many types of custom media:

```
/opt/ignite/data/scripts/examples/make_media_install
```

Booting from tape (HP9000 only)

The `make_boot_tape(1m)` command can create a bootable tape containing just enough information to boot a HP9000 system. However, you must use a combination of an interface card and a tape drive that is bootable. Booting the tape will get the system to the point where it can contact an Ignite-UX server.

Note the `-f` option to `make_boot_tape` which allows you to specify the configuration that will be placed into the LIF on the tape. In this configuration file (which is limited to 8 KB¹⁰), you need to give information such as the IP address of the Ignite-UX server.

¹⁰ Ignite-UX takes advantage of the fact that the first 8Kb of filesystems are not used (true for HFS and VxFS filesystems currently) to allow for up to 8Kb of configuration information to be kept in the install filesystem.

Bootimg from CD/DVD (HP9000 and/or HP Integrity systems)

This section no longer discusses how to create custom CD/DVD boot media for HP9000 or HP Integrity servers. Instead, please refer to the Ignite-UX Custom Configuration Files document in the section titled, "How do I create the CD equivalent of a tape created by `make_boot_tape?`"

The `bootsys(1m)` command

The `bootsys` command allows you to reinstall or recover an existing HP-UX system on its next reboot. The `bootsys` command does this by copying an install kernel and filesystem to `/stand` on the client from an Ignite-UX server. (This means you must have enough free space on the client to hold a copy of the install kernel and filesystem.) The `bootsys` command then modifies how the system will next boot so it points at the install kernel in `/stand`.

Note that you must have a functional HP-UX system before you can use `bootsys`. If the system is down and is not currently bootable, you will not be able to take advantage of `bootsys` as an installation or recovery method.

Important:

The `bootsys` command does not support Auto-Port Aggregation (APA). If you need to contact an Ignite-UX server through an aggregate link, the aggregate must be broken before running `bootsys`, and a LAN interface must be set up with the IP address information through which you wish to contact the Ignite-UX server.

Please consult the `bootsys(1m)` man page for more information on the `bootsys` command.

Customizing the media

The following are customization ideas for environments where it is not feasible or desirable to boot systems over the network.

If you provide enough information in the install filesystem, you can have individually customized media that can set the IP address and disable DHCP so you do not need to gain an IP address over the network. This would require sets of CDs/DVDs unique to subnets to allow you to boot the CDs/DVDs on systems and not have conflicting IP addresses.

An extension to this would be to add tests to the configuration in the install filesystem to test on MAC addresses of systems on a subnet. This would allow a system to use the IP address it normally uses (with a default route defined) so the system could be booted for install or recovery using its own IP address without having to worry about network booting, `bootp`, DHCP, boot helper setup, and so forth.

Section III: Network boot helpers

The `bootp` protocol query UDP broadcast is not routable (cannot relay or forward). In view of this, Ignite-UX provides you the ability to configure a boot helper. A boot helper is the supported method for accessing an Ignite-UX server when it is located on a different subnet than the client.

By default, an Ignite-UX boot helper does not have enough information to point to the correct server so you must modify the server line in the `INSTALLFS` that gets downloaded from the boot helper using `instl_adm` as follows:

```
instl_adm -d
```

```
# instl_adm defaults:
# NOTE: Manual additions between the lines containing "instl_adm
defaults"
#           and "end instl_adm defaults" will not be preserved.
server="10.0.0.2"
netmask[ ]="255.255.255.0"
route_gateway[0]="10.0.1.1"
route_destination[0]="default"
disable_dhcp=true
# end instl_adm defaults.
```

This `INSTALLFS` indicates that the Ignite-UX client was somewhere on the class C subnet based upon the netmask 10.0.1.1, and must contact the Ignite-UX server at 10.0.0.2.

If the `server` line in the `INSTALLFS` does not match one of the IP addresses of the system that the Ignite-UX server booted from, you are using a boot helper. The remainder of the network boot process occurs transparently as the IP address the firmware used is immaterial to the kernel.

Once the client systems boots from the boot helper, it sends a DHCP request to attempt to obtain an IP address. If the request does not succeed, Ignite-UX prompts for the IP address information to be entered on the client console.

For more information regarding boot helpers, refer to the *Ignite-UX Administration Guide*.

Glossary

BCH	Boot Console Handler is the firmware prompt that is available when the system first boots. At the BCH prompt, you can start a system boot and execute other BCH commands.
DHCP	Distributed Host Configuration Protocol allows systems to gain IP addresses and other configuration information dynamically.
EFI	Extensible Firmware Interface is the firmware interface for HP Integrity systems.
El Torito	A bootable CD or DVD ISO9660 filesystem variant that allows the booting of systems from CD or DVD disks.
HPUX	The <code>hpux</code> utility is the specific secondary loader program used for bootstrap of the kernel. For more information, refer to <code>hpux(1M)</code> or <code>hpux.efi(1M)</code> .
ISL	Initial System Loader is the program that starts a system boot and loads the secondary loader, <code>hpux</code> . This term is used interchangeably with IPL on HP9000 systems.
IPL	Initial Program Load means that a system has begun to boot. This term is used interchangeably with ISL on HP9000 systems.
LIF	Logical Interchange Format is the directory format that contains the information that is needed to boot HP9000 systems. For more information, refer to <code>lif(4)</code> and <code>lifls(1M)</code> .
PA-RISC	Precision Architecture-Reduced Instruction Set Computer is the name of the CPU architecture used in HP9000 servers and workstations.
PDC	The processor-dependent code (firmware) that implements all processor-dependent functionality, including initialization and self-test of the processor. Upon completion, it loads and transfers control to the ISL. Firmware behavior varies somewhat, depending on the hardware architecture.
PXE	Pre-eXecution Boot Environment is part of the Wired For Management specifications from the Intel® Corporation and allows a PXE compliant system to boot over the network. All HP Integrity systems support PXE.

For more Information

The Ignite-UX Administration Guide, along with other documents and white papers, are available at <http://www.docs.hp.com/en/IUX/infolib.html>

Product information regarding Ignite-UX is available at <http://www.docs.hp.com/en/IUX/>

The following relevant documents are available online at the HP Technical Documentation Web site at <http://www.docs.hp.com/>:

HP-UX Virtual Partitions Ordering and Configuration Guide

HP-UX 11i v1 Installation and Update Guide

HP-UX 11i v2 Installation and Update Guide

HP-UX 11i v3 Installation and Update Guide

HP-UX 11i v1 Release Notes

HP-UX 11i v2 Release Notes

HP-UX 11i v3 Release Notes

Managing Systems and Workgroups: A Guide for HP-UX System Administrators

HP-UX System Administrator's Guide

Some or all of these documents are available on the Instant Information media and in printed form.

© Copyright 2004-2008 Hewlett-Packard Development Company, L.P.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Intel Itanium® Logo, Intel, Intel Inside and Itanium are trademarks or registered trademarks of Intel Corporation in the US and other countries and are used under license.

Intel® Itanium® Processor Family is a trademark of Intel Corporation in the US and other countries and is used under license

MPN 5992-5274, 09/2008