Hewlett Packard
Enterprise

UEFI Deployment Guide for HPE ProLiant Gen10 Servers and HPE Synergy

Abstract

This guide details how to use various deployment methods to boot Unified Extensible Firmware Interface (UEFI)-based ProLiant Gen10 servers and Synergy compute modules. This document is for the person who installs, administers, and troubleshoots servers and storage systems.

Part Number: 881329-003
Published: March 2019
Edition: 1

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise 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. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise 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.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Intel®, Itanium®, Pentium®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the United States and other countries.

UEFI® is a registered trademark of the UEFI Forum, Inc.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Contents

Introduction

HPE ProLiant servers and Synergy compute modules support an industry standard system firmware based on UEFI. By leveraging an enhanced built-in network stack, and an Embedded UEFI Shell environment, UEFI server deployment options offer improved reliability, security, scalability, and faster download speeds than existing network-based deployment solutions.

UEFI deployment methods

The following UEFI deployment methods are available:

- **USB**—Creates a bootable UEFI USB flash drive to launch any `.efi` type UEFI application, boots an OS launcher by storing it in `\efi\BOOT\BOOTX64.EFI`, or uses a USB flash drive partition as a storage for OS installation.
- **PXE (Preboot Execution Environment) and iPXE**—PXE boots by acquiring an IP address and downloading files from a server using the networking capabilities of the server. iPXE extends the boot methods of PXE by providing HTTP/ HTTPS boot and other options.
- **iSCSI (Internet Small Computer System Interface)**—Boots from an image stored on a LUN or in an iSCSI storage area network (SAN).
- **HTTP Boot and Boot from URL**—Downloads and boots files from a network location (URL) using HTTP/ HTTPS via TCP connections that are more reliable than the UDP connections used for PXE boot.
- **Embedded UEFI Shell**—Boots from the Embedded UEFI Shell, a method that provides an automated scripting solution for a faster network-based OS deployment.
- **SAN (FC and FCoE)**—Boots from disk operating system images located on a Fibre Channel storage area network (SAN).
- **Local HDD**—Boots from a local hard drive.

Configuring UEFI boot settings

Boot settings

You can use the iLO web interface, the UEFI System Utilities menus, or the RESTful Interface Tool to view and modify boot settings, including the order in which the server attempts to boot from configured boot options.

The following sections describe how to configure boot settings using the System Utilities menus.

For more information:

HPE iLO User Guide at **http://www.hpe.com/info/ilo/docs**

RESTful Interface Tool documentation at **http://www.hpe.com/info/restfulinterface/docs**

Boot mode comparison: UEFI and Legacy BIOS

This server provides two boot mode configurations: UEFI Mode (the default boot mode) and Legacy BIOS Mode. Certain boot options described in this guide require that you select a specific boot mode.

- **UEFI Mode**—This is the default boot mode. It configures the server to boot UEFI-compatible operating systems, and is required for you to use certain UEFI System Utilities options, including Secure Boot and booting to USB, IPv6 PXE, iSCSI, HTTP/HTTPS FTP URL, and the Embedded User Partition and the HPE Smart Array S100i Gen10 SW RAID device.
- **Legacy BIOS Mode**—This boot mode configures the server to emulate the traditional BIOS boot environment. Certain situations might require that you operate in Legacy BIOS Mode, such as booting custom OS images that were installed using legacy boot mode or created using a legacy BIOS system.

Selecting the boot mode

This server provides two **Boot Mode** configurations: UEFI Mode and Legacy BIOS Mode. Certain boot options require that you select a specific boot mode. By default, the boot mode is set to **UEFI Mode**. The system must boot in **UEFI Mode** to use certain options, including:

- Secure Boot, UEFI Optimized Boot, Generic USB Boot, IPv6 PXE Boot, iSCSI Boot, and Boot from URL
- Fibre Channel/FCoE Scan Policy

NOTE: The boot mode you use must match the operating system installation. If not, changing the boot mode can impact the ability of the server to boot to the installed operating system.

Prerequisite

When booting to **UEFI Mode**, leave **UEFI Optimized Boot** enabled.

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Boot Options** > **Boot Mode**.
2. Select a setting.

- **UEFI Mode** (default)—Configures the system to boot to a UEFI compatible operating system.
- **Legacy BIOS Mode**—Configures the system to boot to a traditional operating system in Legacy BIOS compatibility mode.

3. Save your setting.
4. Reboot the server.

Enabling or disabling UEFI Optimized Boot

Use **UEFI Optimized Boot** to control whether the system BIOS boots using native UEFI graphic drivers. **UEFI Optimized Boot** is enabled by default. You disable **UEFI Optimized Boot** only if you are using Windows Server 2008, Windows Server 2008 R2, or Windows 7.

Prerequisites

- When **UEFI Optimized Boot** is enabled, Boot Mode must be set to UEFI Mode. UEFI Mode is the default Boot Mode.
- **UEFI Optimized Boot** must be enabled to:
  - Enable and use Secure Boot.
  - Operate VMware ESXi.

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Boot Options** > **UEFI Optimized Boot**.
2. Select an option.
   - **Enabled**—When set to UEFI Mode, configures the system BIOS to boot using native UEFI graphic drivers.
   - **Disabled**—Configures the system BIOS to boot using INT10 legacy video expansion ROM. This setting is required if you are using Windows Server 2008, Windows Server 2008 R2, or Windows 7 as your operating system.
3. Save your setting.
4. Reboot the server.

Changing the UEFI Boot Order list

Use the **UEFI Boot Order** option to change the order in which entries in the UEFI Boot Order list boot.

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Boot Options** > **UEFI Boot Settings** > **UEFI Boot Settings** > **UEFI Boot Order**.
2. To navigate within the boot order list, use your pointing device or the arrow keys.
3. Select an entry and change its order in the list:

- To move an entry higher in the boot list, press the **+** key, or drag and drop the entry.
- To move an entry lower in the boot list, press the **-** key, or drag and drop the entry.

4. Save your changes.

Changing the Legacy BIOS Boot Order list

Prerequisite

Boot Mode is set to **Legacy BIOS Mode**.

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Boot Options** > **Legacy BIOS Boot Order**.
2. To navigate within the boot order list, use your pointing device or the arrow keys.
3. Select an entry and change its order in the list:
   - To move an entry higher in the boot list, press the **+** key, or drag and drop the entry.
   - To move an entry lower in the boot list, press the **-** key, or drag and drop the entry.
4. Save your changes.

Adding a boot option to the UEFI Boot Order list

Use **Add Boot Option** to select an x64 UEFI application with an .EFI extension, such as an OS boot loader or other UEFI application, to add as a new UEFI boot option.

The new boot option is appended to the UEFI Boot Order list. When you select a file, you are prompted to enter the boot option description (which is then displayed in the boot menu), as well as any optional data to be passed to an .EFI application.

Procedure

1. Attach media with a FAT16 or FAT32 partition on it.
2. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Boot Options** > **UEFI Boot Settings** > **Add Boot Option**.
3. Browse for an .EFI application from the list, and then press **Enter**.
4. If necessary, continue to press **Enter** to drill-down through the menu options.
5. Enter a boot option description and optional data, and then press **Enter**.

   The new boot option appears in the **UEFI Boot Order** list.
6. Select **Commit changes and exit**.

One-Time Boot Menu options

Use the **One-Time Boot Menu** to select a UEFI boot option for a one-time boot override.

NOTE: The option you select does not modify your predefined boot order settings. If you use a USB key or virtual media through the iLO 5 Remote Console, exit and re-enter the One-Time Boot Menu to refresh this menu so that the devices appear.

Boot options include:

- OS boot manager, such as **Windows Boot Manager**—Lists the boot manager for your installed OS.
- **Generic USB Boot**—Provides a placeholder for any USB device that is bootable in UEFI. You can set the boot priority of this option, and retain this priority for use with USB devices you might install in the future. Setting this priority does not affect priorities set for individual USB devices in the **UEFI Boot Order** list.

  NOTE: This option is only available in UEFI Mode. The system attempts to boot all UEFI bootable USB devices in the order you specify in the **Generic USB Boot** entry, even if installed individual USB devices are configured lower in the boot order.

- Embedded UEFI Shell — Depending on your configuration, this boot option may not be on the One-Time Boot Menu initially. You can add this boot option with the UEFI System Utilities. See "Adding the Embedded UEFI Shell to the UEFI Boot Order list" in the UEFI System Utilities Guide.
- **Run a UEFI Application from a file system** — Use this option to select a UEFI application to run from a file system. You can browse all FAT file systems that are available in the system. You can also select an x64 UEFI application (with an .EFI extension) to execute (can be an OS boot loader or any other UEFI application).
- **Legacy BIOS One-Time Boot Menu**—Exits and launches the **Legacy BIOS One-Time Boot Menu**, where you can select a specific override option for this boot only. This option does not modify your boot order mode settings.
- Device boot option — Use this option to boot with Embedded Flexible LOMs, Embedded SATA Port, Embedded RAID, and so on.
- Run a UEFI Application from a file system — Use this option to select a UEFI application to run from a file.

Selecting an option for a one-time boot

Procedure

1. From the **System Utilities** screen, select **One-Time Boot Menu**.
2. Select a **One-Time Boot Menu** option.

   If you select Legacy BIOS One-Time boot option, the system reboots.

USB boot

USB boot options

Generic USB Boot

The **Generic USB Boot** entry in the **One-Time Boot Menu** is a placeholder for any USB device that is bootable in UEFI. You can set the boot priority of this option, and retain this priority for use with USB devices you might install in the future. Setting this priority does not affect priorities set for individual USB devices in the **UEFI Boot Order** list. Newly-added USB devices appear at the bottom of the list by default, and you can move those entries in the list and boot from them as well.

NOTE: The system attempts to boot all UEFI bootable USB devices in the order you specify in the **Generic USB Boot** entry, even if installed individual USB devices are configured lower in the boot order.

Creating a bootable UEFI USB flash drive for installing Windows

Prerequisites

NOTE: See **Microsoft Docs** for information on Windows images and USB drives.

- The flash drive is formatted as FAT/FAT32.
- When booting an OS launcher, the boot loader is an `.efi` type and is stored at the default location `\efi\BOOT\BOOTX64.EFI`.
- The **USB Options** in the System Utilities are configured to support booting from a USB drive. Booting from a USB drive is the default configuration. For more information, see “USB Options” in the UEFI System Utilities user guide.

Procedure

1. Connect the USB flash drive to a Windows computer and open a command prompt.
2. Enter the following commands one at a time and press **Enter** after each:

   ```
   diskpart
   list disk
   ```

3. Record the disk number for the USB flash drive.
4. Right-click and format the USB key.
5. To extract image files from an `.iso` file for the OS installation:

   a. Open File Explorer, navigate to the `.iso` file, and right-click and select **Mount**.

The `.iso` file opens.

b. Select all files in the `.iso` file, right-click, and select **Send to** and the removable disk of your USB flash drive.

6. If you are creating a 64-bit Windows UEFI USB flash drive, copy the content from the `.iso` image to the USB.
7. On the USB flash drive, copy the `efi\microsoft\boot` folder up one level into the `efi` folder as `efi\boot`.
8. Open an archive utility and navigate to the `sources\install.wim\1\Windows\Boot\EFI\bootmgfw.efi` on the USB flash drive.

   NOTE: The `install.wim` file is not a directory. It is a Windows image formatted file.

9. Select the `bootmgfw.efi` file, and copy it to the `/efi/boot/` folder on the USB flash drive.
10. Click **OK**, and close the archive utility.
11. Rename the `bootmgfw.efi` file to `bootx64.efi`.
12. Perform a clean install of Windows using UEFI with the bootable UEFI USB flash drive.

Booting from an HPE Dual MicroSD device

Procedure

An HPE Dual MicroSD is similar to a USB boot device. It can contain redundant images of an OS, such as VMware ESXi. For information, see the *HPE Dual 8GB MicroSD Enterprise Midline USB User Guide* at **http://www.hpe.com/support/8GBDualMicrosd_ug_en**.

Creating a dual boot USB key for both UEFI Mode and Legacy BIOS Mode

You can create a dual boot USB device to boot both Legacy BIOS (for example DOS) and a UEFI Shell. This is useful, for example, for performing an offline ROM upgrade utility. To do so:

Procedure

1. Attach a USB device with a FAT/FAT32 partition that contains a Legacy BIOS Mode MBR.
2. Add the `\efi\BOOT\BOOTX64.EFI` boot loader to the legacy-bootable USB.

PXE and iPXE boot

PXE and iPXE operation

- **PXE** defines how a client system boots over a network connection using the networking capabilities of the server. It requires standard network protocols, such as DHCP and TFTP.
- **iPXE** provides an enhanced PXE implementation.

Standard PXE boots when the DHCP server `filename` is configured for your PXE environment. For example, to boot standard PXE in a Linux environment, you set the DHCP `filename` to `pxelinux.0`. iPXE boots when the `filename` on the DHCP server is set to `ipxe.efi`.

PXE multicast boot

PXE multicasting combines unicasting and broadcasting in group communications using one-to-many or many-to-many distribution. It is both a client-side and server-side feature. Operating systems, such as Microsoft Windows Server 2012, support deployment in PXE multicast mode. UEFI enables PXE multicast boot on the client side, a feature that Legacy BIOS or Option ROMs do not support. The PXE stack of UEFI-based systems “listens” on multicast addresses. This enables servers to install multiple PXE booting clients as part of a multicast group, which saves time and bandwidth, and reduces server load.

PXE configuration

Configuring PXE servers for UEFI-based clients in a Linux environment

The most common UEFI PXE boot loaders are GRUB and GRUB2.

The following examples show how to modify the existing BIOS-based PXE configuration to include the option to boot both BIOS and UEFI boot loaders. Additions in the examples are indicated in bold.

Modified ISC `dhcpd.conf` file

```
#/etc/dhcpd.conf
option domain-name "pxetest.org";
option domain-name-servers 192.168.100.10;
option routers 192.168.100.1;
ddns-update-style none;
subnet 192.168.100.0 netmask 255.255.255.0{
range 192.168.100.20 192.168.100.254;
default-lease-time 14400;
max-lease-time 172800;
next-server 192.168.100.10;
##filename "pxelinux.0"; #comment out for UEFI settings
##Added sections for UEFI
# In initial DHCP DISCOVER packet, PXE client sets option 93 to its arch.
# 0000 == IA x86 PC (BIOS boot)
# 0006 == x86 EFI boot
# 0007 == x64 EFI boot
option arch code 93 = unsigned integer 16; # RFC4578
class "pxe-clients" {
         match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";
         if option arch = 00:07 {
            filename "EFI/bootx64.efi";
         } else {
            filename "pxelinux.0";
        }
```

```
}
subnet 192.168.100.0 netmask 255.255.255.0{
range 192.168.100.20 192.168.100.254;
default-lease-time 14400;
max-lease-time 172800;
next-server 192.168.100.10;
}
```

Modified `dnsmasq.conf` file

```
#Configuration file for dnsmasq
#DHCP configuration
dhcp-option=option:domain-serch,foo.org
dhcp-boot=pxelinux.0,pxeserver,192.168.100.10
# UEFI IPv4 PXE
# currently using elilo boot file
dhcp-match=set:efi-x86_64,option:client-arch,7
dhcp-boot=tag:efi-x86_64,/EFI/bootx64.efi,pxeserver,192.168.100.10
dhcp-range=set:devnet,192.168.100.20,192.168.100.254,1h
dhcp-option=tag:devnet,121,0.0.0.0/0,192.168.100.1
```

NOTE: For more information on the DHCP option 93 arch codes, refer to RFC4578.

Updated TFTP directory structure

```
/tftpboot
/tftpboot/pxelinux.0
/tftpboot/pxelinux.cfg/default
/tftpboot/EFI/
/tftpboot/EFI/bootx64.efi
 /tftpboot/EFI/grub.cfg
/tftpboot/RHEL6.9
/tftpboot/RHEL7.4
/tftpboot/SLES11SP4
/tftpboot/SLES12SP3
/tftpboot/Ubuntu16.04.3
```

Configuring boot loaders

For UEFI-based systems, the common boot loaders are GRUB and GRUB2. Refer to the distribution documentation on how to obtain and configure the correct bootx64.efi file. GRUB2 has become the standard for UEFI PXE configurations. GRUB2 is the only boot loader that supports Secure Boot.

Using GRUB2

GRUB2 is the only boot loader that currently supports Secure Boot.

NOTE: GRUB2 should not be confused with GRUB Legacy.

Sample `grub2.conf` file

```
insmod gettext
insmod iso9660
insmod ntfs
insmod normal
insmod chain

menuentry 'RHEL 7.4'  --class gnu-linux --class gnu --class os {
        echo 'Loading Kernel ...'
        linuxefi /RHEL-7.4Server/vmlinuz repo=http://192.168.100.10/RHEL-7.4Server/disc1
```

```
        echo 'Loading initial Ramdisk ...'
        initrdefi /RHEL-7.4Server/initrd.img
        }
menuentry 'SLES 12 SP3 AHCI' --class gnu-linux --class gnu --class os {
        echo 'Loading Kernel ...'
        linuxefi /SLE12SP3Server/linux install=http://192.168.100.10/SLE12SP3Server/disc1
        echo 'Loading initial Ramdisk ...'
        initrdefi /SLE12SP3Server/initrd
        }
menuentry 'Xenial 16.04.3'  --class gnu-linux --class gnu --class os {
        echo 'Loading Kernel ...'
        linuxefi /Ubuntu-16.04.3/Linux
        echo 'Loading initial Ramdisk ...'
        initrdefi /Ubuntu-16.04.3/initrd.gz
        }
```

Configuring a PXE server for Legacy BIOS-based client in a windows environment

You can use software such as Windows Deployment Services (WDS) to configure PXE boot for Legacy BIOS in Windows, and to configure DHCP and TFTP servers together. Configurations steps might differ depending on the software you use.

Configuring a PXE server for a UEFI-based client in a Windows environment

System requirements

All versions of Windows supported by Gen10 servers and HPE Synergy compute modules can boot in UEFI mode.

Earlier versions of Windows, such as Windows XP and Server 2003, can only boot in Legacy BIOS mode.

Configuring a Windows server

You can use software, such as Windows Deployment Services (WDS), to configure PXE boot for UEFI in Windows. In addition, you can use WDS in combination with deployment solutions, such as Microsoft Deployment Toolkit (MDT) or Configuration Manager. Depending on the software you use, your configuration steps might vary.

Configuring VLANs for UEFI network boot

You can use the System Utilities **Network Options** > **VLAN Configuration** menu or the RESTful Interface Tool to set a global VLAN configuration on enabled network interfaces, including those used in PXE boot, iSCSI boot, and FTP/HTTP boot, and for all pre-boot network access from the Embedded UEFI Shell.

When supported by the NIC card, you can also use the NIC-specific configuration menu in the **System Utilities** > **System Configuration** options to set VLAN settings for that port.

IMPORTANT: When you are using a NIC card that supports an individual, card-specific VLAN configuration in a PCIe slot, you must only select one of the following methods: the global VLAN configuration method provided by the System Utilities Network Options, or the individual, card-specific VLAN configuration method. Both VLAN configurations must not be active under any circumstances.

Using the global VLAN Configuration menu provided by the System Utilities Network Options

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Network Options** > **VLAN Configuration**.
2. Complete the following.
   - **VLAN Control**—Select **Enabled** to enable VLAN tagging on all enabled network interfaces. This setting is disabled by default.
   - **VLAN ID**—When **VLAN Control** is enabled, enter a global VLAN ID of 0 to 4094 for all enabled network interfaces.
   - **VLAN Priority**—When **VLAN Control** is enabled, enter a priority value of 0 to 7 for VLAN tagged frames.
3. Save your changes.

Using the configuration menu provided by specific NIC adapters

Procedure

1. From the **System Utilities** screen, select **System Configuration**.

   The **System Configuration** screen lists the **BIOS/Platform Configuration (RBSU)** option and the other available device configurations, including the NICs.
2. Select the NIC port to be used for network boot.

   Configuration options for the NIC port appear. Option titles vary by NIC card.
3. Select the configuration menu option for your NIC (for example, **MBA Configuration Menu**).

   A configuration menu appears.
4. For **VLAN Mode**, select **Enabled**, and press **Enter**.
5. For **VLAN ID (1..4094)**, enter the VLAN ID that matches the VLAN setting on your network, and press **Enter**.
6. Press **Esc** twice, and then press **Y** to save and exit the configuration.
7. Reboot the server.

NOTE: You cannot use the RESTful Interface Tool to configure the NIC adapter provided VLAN settings.

iPXE configuration

While standard PXE clients use TFTP to transfer data, iPXE client firmware adds the ability to retrieve data through other protocols, including HTTP. When using UEFI iPXE boot, your TFTP and DHCP configuration settings are similar to those for a **PXE configuration** on page 13, but can also include steps to configure HTTP, including chainloading iPXE through your TFTP server. Chainloading enables you obtain the features of iPXE without having to reflash each server's network card individually.

Configuring iPXE

Process overview

Procedure

1. Download iPXE.
2. Optional: Customize iPXE.
3. Configure TFTP.
4. Configure HTTP.
5. Configure DHCP.

Downloading iPXE files

Procedure

1. Go to the iPXE Open Source Boot Firmware website at **http://ipxe.org/download**, and select the link for either of the following:
   - Pre-built binary ISO image
   - Customizable iPXE source code
2. Complete the download.

Customizing iPXE

Procedure

1. Compile the iPXE source code.
2. Enable the graphics console.
   - In the `config/console.h` file, uncomment the following.

     ```
     #define CONSOLE_FRAMEBUFFER /* Graphical framebuffer console */
     ```
   - In the `config/general.h` file, uncomment the following.

     ```
     #define CONSOLE_CMD /* Console command */
     ```
3. Enable the time command.
   In the `config/general.h` file, uncomment the following.

   ```
   #define TIME_CMD /* Time commands */
   ```

Configuring TFTP

Procedure

1. Install the TFTP server package prescribed by your OS.
2. Move the `pxelinux` file into the TFTP root directory.

Configuring HTTP

Procedure

1. Install the HTTP server package prescribed by your OS.
2. Move the `ipxe.efi` (or `snponly.efi`) file into the HTTP root directory.
3. Move the OS files into HTTP root directory.

Configuring DHCP

Procedure

1. Install the DHCP server package prescribed by your OS. For example, the ISC dhcpd service.
2. Modify the configuration file:
   - To enable PXE clients to load iPXE from the TFTP server, specify `ipxe.efi` for the filename option.
   - Add parameters to handle an iPXE DHCP client, and specify the iPXE script HTTP path for its `filename`.
   - Add parameters to send back the HTTP URL for ipxe.efi if HTTP boot is detected.

Chainloading configuration

Chainloading iPXE from an existing PXE ROM is useful if you have a large number of machines that you want to be able to boot using iPXE, but you do not want to reflash the network card on each individual machine. You can do this by enabling PXE clients to load iPXE from the TFTP server,

Sample `dhcpd.comf` file with chainloading enabled

```
# dhcpd.conf
#
option client-architecture code 93 = unsigned integer 16;
#
# Options common to all private networks
#
option domain-name "ipxedemo.net";
option domain-name-servers 16.110.135.52,16.110.135.51;
option ntp-servers 16.110.135.123;
default-lease-time 600;
max-lease-time 7200;
#
# Private network
#
subnet 192.168.0.0 netmask 255.255.0.0 {
  range 192.168.1.100 192.168.1.200;
  option routers 192.168.0.1;
  option broadcast-address 192.168.255.255;

  # iPXE is DHCP client
  if exists user-class and option user-class = "iPXE" {
    filename "http://192.168.0.1/ipxe-demo/menu.ipxe";

  # UEFI HTTP Boot
  # option client-architecture code 93 = unsigned integer 16;
  } elsif option client-architecture = encode-int ( 16, 16 ) {
```

```
    filename "http://192.168.0.1/ipxe.efi";
    option vendor-class-identifier "HTTPClient";

  # Chain-load iPXE from TFTP server
  } else {
    filename "ipxe.efi";
  }
}
```

Booting PXE or iPXE

Prerequisite

For iPXE and iPv6 PXE, set the boot mode to **UEFI Mode**.

Procedure

1. Boot the server.
2. When the POST screen appears, do one of the following:
   a. Press **F12** (Network Boot).
   b. Press **F11** and enter the **One-Time Boot Menu** and select the PXE or iPXE boot target. See **Selecting an option for a one-time boot** on page 10.

iSCSI boot

iSCSI software initiator configuration

iSCSI boot refers to a host system booting from a boot image stored on a LUN in an iSCSI Storage Area Network (SAN). You can use the iSCSI Software Initiator to implement iSCSI boot. The iSCSI software initiator provides access to storage devices over Ethernet network connections called iSCSI targets. These targets differ from other network-attached storage devices in that iSCSI provides block-level access to disks, rather than file-based access.

iSCSI boot enables you to boot Windows or Linux-based systems using a network adapter with the iSCSI stack implemented in software (the system ROM), or in an iSCSI Host Bus Adapter (HBA) as an alternative method to access block-level storage. By eliminating the need for local storage on each server or blade server, iSCSI boot can reduce power consumption and the amount of heat the system generates.

Configuring the iSCSI Software Initiator and booting iSCSI

Process overview

Prerequisites

- Before configuring VLANs for iSCSI boot in a Windows environment, complete the steps in **Configuring VLANs for UEFI network boot** on page 15.
- Leave the boot mode set to **UEFI Mode**.

Procedure

1. **Select a device for iSCSI boot and make it network bootable**.
2. **Add an iSCSI initiator name**.
3. **Adding an iSCSI boot attempt**.
4. **Verify the iSCSI connection**.
5. **Change the boot order of the iSCSI initiator**.

Selecting a device for iSCSI boot and making it network-bootable

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Network Options** > **Network Boot Options**.

   The **Network Boot Options** screen lists network devices and ports available for iSCSI.
2. Select the device that is connected to the SAN network on which the iSCSI target is available.
3. Select **Network Boot**.
4. Save the setting.

Adding an iSCSI initiator name

Use the iSCSI Initiator Name option to set a name for the iSCSI initiator using iSCSI Qualified Name (IQN) format. EUI format is not supported. This option replaces the default name set for the initiator.

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Network Options** > **iSCSI Boot Configuration** > **iSCSI Initiator Name**.
2. Enter a unique name for the iSCSI initiator using iSCSI Qualified Name (IQN) format. For example: `iqn.2001-04.com.example:uefi-13021088`

   This setting is saved automatically.

Adding an iSCSI boot attempt

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Network Options** > **iSCSI Boot Configuration** > **Add an iSCSI Boot Attempt**.

   A message appears stating that this boot attempt will not be in effect until the next server reboot.
2. Press **Enter**.
3. Select a port on which to attempt iSCSI connection.
4. Complete the configuration settings:

   - **iSCSI Attempt Name**—Enter a name.
   - **iSCSI Boot Control**—Select **Enabled**. (The default setting is **Disabled**).
   - **IP Address Type**—Select an address type.
   - **Connection Retry Count**—Enter a value from 0 to 16. Default is 3 retries.
   - **Connection Timeout**—Enter a value in ms from 100 to 20000. Default is 20000 (20 seconds).
   - **Initiator DHCP**—This is the default setting. If you must configure static IP addresses for the Initiator, clear this option. The target name, IP address, port, and boot LUN must also be configured manually (disable Target DHCP Config) if you configure static addresses for the Initiator.
   - **Target DHCP Config**—This is the default setting. If you must configure the target settings manually, clear this check box) and enter a target name, IP address, port, and boot LUN.
   - Optional: **Authentication Type**—Default is NONE. If required, select **CHAP**, and then complete the CHAP entries.
5. Select **Save Changes**.
6. Reboot the system.

Verifying the iSCSI connection

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Network Options** > **iSCSI Boot Configuration** > **iSCSI Attempts**.

   The **iSCSI Boot Attempt Configuration** screen appears.
2. Verify that correct information is listed for the boot attempt you configured in **Adding an iSCSI boot attempt**. If you selected **Initiator DHCP**, verify that the newly-assigned IP addresses are listed.

3. From the **System Utilities** screen, select **Embedded Applications** > **Embedded UEFI Shell**.
4. At the Embedded UEFI Shell prompt, enter the `map -r` command.
   A mapping table appears.
5. Verify that the iSCSI target device is identified as a file system. For example, `FS0`.
6. Enter the `exit` command to exit the Embedded UEFI Shell.
7. Verify that the iSCSI target device is listed in the **One-Time Boot Menu**. See **One-Time Boot Menu options** on page 9.
8. If you booted with a Windows or Linux boot disk or live image, verify that the mapped iSCSI target device is listed as an installation target.

Changing the boot order of the iSCSI initiator

You can use the iLO 5 web interface, the System Utilities menus, or the RESTful Interface Tool to change the boot order of the iSCSI initiator. For information about using the System Utilities, see **Changing the UEFI Boot Order list** on page 8.

HTTP boot

HTTP Boot is a new industry standard capability for network booting systems, and is meant as a faster and more secure replacement for TFTP-based PXE boot. It combines the DHCP, DNS, and HTTP protocols to provide the capability for system deployment and configuration over the network. HTTP Boot works similar to PXE boot, but uses HTTP or HTTPS to download images. HTTP Boot is only supported in UEFI boot mode.

Like PXE, HTTP Boot supports downloading and booting Network Bootstrap Programs (NBPs) like EFI loaders or applications. In addition, HTTP Boot also supports downloading ISO images and booting from them. This feature is new in HTTP Boot and not supported by PXE. The URL can be an IPv4 or IPv6 address, or fully qualified host names. DNS resolution will be used when host names are provided in the URL.

HTTP Boot is done in one of two ways:

1. Discover the boot file (EFI application/boot loader or ISO image) using DHCP, just like PXE boot. When the URL is provided by a DHCP server, the system downloads the file pointed to by the URL and boots from it.

   On the HPE ProLiant Gen10 Servers and HPE Synergy compute modules, when the **RBSU** > **Network Options** > **Network Boot Options** > **HTTP Support** option is set to **Auto** (default), **HTTP Only**, or **HTTPs Only**, an HTTP Boot Option is created and shown in the UEFI Boot Menu for every port enabled for network boot. The created HTTP Boot Option is selected in the **One-Time Boot Menu** and the system attempts to perform the HTTP Boot on that network device.

   For more information on setting up the server side configuration to perform HTTP Boot, see **Configuring the network services to provide the boot URL**.

2. Boot from a statically provided URL (pointing to an EFI application or ISO image). No special configuration is needed on the DHCP server to support booting from a statically provided URL. The DHCP server only provides the IP address configuration required for the system to be able to download this URL.

   HPE ProLiant Gen10 servers and HPE Synergy compute modules support configuring up to four user provided URLs. To configure booting from a specific URL or URLs, use the Boot from URL 1, 2, 3 or 4 setting under **RBSU** > **Network Options** > **Pre-Boot Network settings**. Each URL can be a bootable ISO or EFI file. Enter a URL in either HTTP or HTTPS format, using either an IPv4 or IPv6 server address or host name. For example, the URLs can be in any of the following formats: `http://192.168.0.1/file/image.iso`, `http://example.com/file/image.efi`, `https://example.com/file/image.efi`, `http://[1234::1000]/image.iso`. When a URL is configured, this URL is listed as a boot option in the UEFI Boot menu.

   These settings can also be done using the RESTful Interface Tool or the sysconfig command from the Embedded UEFI Shell. Then you can select this option from the boot menu to download the specified file to the system memory and enable the system to boot from the file. The network interface to download and boot this URL is chosen based on the values set in **RBSU** > **Network Options** > **Pre-Boot Network settings** > **Pre-Boot Network Interface**.

Configuring the network services to provide the boot URL

Use this set of tasks to setup the network server-side configurations. All examples shown are done on Ubuntu/Debian. Refer to the user and administration guides for your OS to perform the same configuration in your environment.

Procedure

1. **Installing the necessary packages**
2. **Setting up the network interface**

3. **Configuring the DHCP service**
4. **Configuring the Router Advertisement Daemon**
5. **Configuring the DNS service for IPv4 and IPv6**
6. **Configuring the HTTP service**
7. **Configuring HTTP Basic Authentication**
8. **Setting up HTTPs (TLS) on apache2**
9. **Installing certificates on the client machine**

Installing the necessary packages

This table lists the recommended services for the Debian/Ubuntu system:

DNS	bind9
DHCP IPv4	isc-dhcp-server
DHCP IPv6	isc-dhcp-server and radvd
Web Server	apache2

Procedure

Install the packages necessary for your environment. For information on installing packages for your OS, consult the documentation for your OS.

Setting up the network interface

Procedure

Set up the network interface on which you want the DHCP and other services to listen for client requests. Refer to the documentation for your OS for information on setting up network interfaces.

Configuring the DHCP service

You configure the DHCP service for:

- IPv4 and IPv6 address assignment for HttpBoot Client requests asking for the boot file or ISO image
- HPE Embedded UEFI shell client requests asking for a startup.nsh script

Procedure

1. To set up DHCP through IPv4, add these lines to `/etc/dhcp/dhcp.conf`:

```
option arch code 93 = unsigned integer 16;
option user-class code 77 = { integer 8, string };

subnet 192.168.111.0 netmask 255.255.255.0 {
```

```
  # Common response to all client requests on this subnet. IP, GW and DNS.
  range 192.168.111.100 192.168.111.120;
  option routers 192.168.111.1;
  option domain-name-servers 192.168.111.3;

  # This is to respond to requests from HttpBoot clients with what they require.
  class "http-clients" {
    match if substring (option vendor-class-identifier, 0, 10) = "HTTPClient";
    option vendor-class-identifier "HTTPClient";
    if option arch = 00:10 {  # x86-64 architecture systems.
      option bootfile-name "https://192.168.111.4/ISO/sles12sp2_x64.iso";
      #option bootfile-name "http://websrv.example.com/boot/grub2/x86_64-efi/grub.efi";
    }
  }

  # This is to respond to requests from Embedded UEFI Shell asking for a script URL.
  class "uefishell-clients" {
    match if substring (option user-class, 1, 9) = "UEFIShell";
    option user-class 9 "UEFIShell";
    if option arch = 00:10 {
      option bootfile-name "https://websrv.example.com/SCRIPT/start.nsh";
    }
  }
}
```

NOTE:

The ‘subnet’ declaration must match the prefix (address & netmask) of the network interface on which you want the DHCP service to listen on.

NOTE: For other architecture types, see "Processor architecture types" at **http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xhtml**.

2. To set up DHCP for IPv6:

   a. Create two new files: `/var/lib/dhcp/dhcpd6.leases` (empty file) and `/etc/dhcp/dhcpd6.conf`.

   b. Add these settings to the `/etc/dhcp/dhcpd6.conf` file:

```
ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
log-facility local7;

option dhcp6.bootfile-url code 59 = string;
option dhcp6.user-class code 15 = { integer 16, string };
option dhcp6.vendor-class code 16 = { integer 32, integer 16, string };

subnet6 1235::/64 {

  # Common response for all client requests on this subnet. Supply IP and DNS.
  range6 1235::2000 1235::2100;
  option dhcp6.name-servers 1235::1003;

  # This is to respond to requests from HttpBoot clients with what they require.
  class "http-clients" {
    match if substring (option dhcp6.vendor-class, 6, 10) = "HTTPClient";
    option dhcp6.vendor-class 0 10 "HTTPClient";
```

```
    if option dhcp6.client-arch-type = 00:10 {  # x86-64 architecture systems.
      option dhcp6.bootfile-url "https://[1235::1004]/ISO/sles12sp2_x64.iso";
      #option dhcp6.bootfile-url "http://websrv.example.com/boot/grub2/x86_64-efi/grub.efi";
    }
  }

  # This is to respond to requests from Emb UEFI Shell asking for a script URL.
  class "uefishell-clients" {
    match if substring (option dhcp6.user-class, 2, 9) = "UEFIShell";
    option dhcp6.user-class 9 "UEFIShell";
    if option dhcp6.client-arch-type = 00:10 {
      option dhcp6.bootfile-url "http://[1235::1004]/SCRIPT/start.nsh";
    }
  }
}
```

NOTE: The URL can be set to any .iso or .efi that you want to use to boot. The .iso or .efi must be placed under the root folder of the HTTP server. For more information, see **Configuring the HTTP service**.

3. Start or restart the DHCP service for both IPv4 and IPv6.

Configuring the Router Advertisement Daemon

You configure the Router Advertisement Daemon (radvd) for answering IPv6 ICMP Router Solicit messages from clients looking for router addresses and network prefixes for their automatic IPv6 address configuration.

Procedure

1. Create a file `/etc/radvd.conf`:

```
interface eth0 {
        AdvSendAdvert on;
        AdvManagedFlag on;
        prefix 1235::/64 {
        };
};
```

NOTE: This interface must match the one on which clients send requests for IPv6 router addresses.

2. Enable packet forwarding for IPv6 on the router (host) where the radvd service is running by editing the `/etc/sysctl.conf` file:

```
# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1
```

NOTE: If this information is not in the `/etc/sysctl.conf`, add it to the file.

3. Restart the radvd service.

Configuring the DNS service for IPv4 and IPv6

Configure the DNS service (named) to listen only on specific IP addresses for client requests. Otherwise, the service listens on all NIC interfaces by default, which may not be desired.

Procedure

1. Edit /`etc/bind/named.conf.options` and add these lines within the options { } section and remove or comment out any other listen commands:

```
listen-on { 192.168.1.3; };
listen-on-v6 { 1235::1003; };
```

NOTE: The address must match the IP address of the local NIC interface on which the DNS service listens for lookup requests.

2. Edit `/etc/bind/named.conf.local`:

```
zone “abc.com” {
type master;
file “/etc/bind/db.abc.com”;
};
```

3. Edit `/etc/bind/db.abc.com` and add these lines to configure forward lookups (Host to IP), setting `dnssrv.` and `root.dnssrv.` to the hostname and IP address of the server where the DNS service is running. The '.' at the end of the service name is intentional. The "Serial" field needs to be incremented every time you change the configuration and restart the DNS service.

```
;
; BIND data file for interface 192.168.111.3/1235::1003
;
$TTL    604800
@       IN      SOA     dnssrv.     root.dnssrv. (
                              3         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      dnssrv.
@       IN      A       192.168.111.3
@       IN      AAAA    1235::1003

; other hosts in this network. websrv.abc.com is the only one for now.
websrv     IN      A       192.168.111.4
websrv     IN      AAAA    1235::1004
```

Optionally, you may configure for reverse lookup (IP to Host) as well, but is not necessary for our purpose.

4. Restart the DNS service.

Configuring the HTTP service (apache2)

This task enables HTTP GET and PUT on apache2.

When you complete this task, you can verify if apache2 is listening on both IPv4 and IPv6 addresses on the configured ports with the `netstat -antep | grep apache2` command:

```
# netstat -antep | grep apache2
```

```
tcp    0    0 192.168.111.4:80   0.0.0.0:*  LISTEN    0     36264       1897/apache2
tcp6   0    0 1235::1004:80    :::*       LISTEN    0     36262       1897/apache2
```

Prerequisites

The files (.iso/.efi) that you use to boot from a URL must be copied to the root directory: `/var/www/html/`.

Procedure

1. Edit `/etc/apache2/ports.conf` and setup the ports and IP addresses the server will use.

   Specify all the addresses on which you want apache2 to listen for client requests. In the following example, the network interface on the system where apache2 is running is configured with the IPv4 address 192.168.1.4 and an IPv6 address of 1235::1004.

   ```
   Listen [1235::1004]:80
   Listen 192.168.111.4:80

   <IfModule ssl_module>
           Listen 443
   </IfModule>

   <IfModule mod_gnutls.c>
           Listen 443
   </IfModule>
   ```

   NOTE: Some systems may not have the file `ports.conf`. If you find the line “Include ports.conf” in `/etc/apache2/apache2.conf`, the `ports.conf` file will exist and the example settings can be added to `ports.conf`. Otherwise, you must add the lines from the example into apache2.conf itself.

2. Edit `/etc/apache2/apache2.conf` and:

   a. Add these lines

   ```
   LoadModule dav_module /usr/lib/apache2/modules/mod_dav.so
   LoadModule dav_fs_module /usr/lib/apache2/modules/mod_dav_fs.so
   LoadModule dav_lock_module /usr/lib/apache2/modules/mod_dav_lock.so
   DavLockDB /var/www/DavLock
   <Directory /var/www/>
           Options Indexes FollowSymLinks
           AllowOverride None
           AuthType None
           Require all granted
           Dav On
   </Directory>
   ```

   b. Comment out all other `<Directory> </Directory>` blocks in the file.

3. Set `KeepAlive` to On, if it is not already set.

   ```
   KeepAlive On
   ```

4. Restart apache2.

Configuring HTTP Basic Authentication

Do this task only if you want to enable HTTP Basic Authentication on apache2.

Procedure

1. Add the following lines to `/etc/apache2/sites-enabled/000-default.conf`:

```
<Directory "/var/www/html">
  AuthType Basic
  AuthName "Restricted Content"
  AuthUserFile /etc/apache2/.htpasswd
  Require valid-user
</Directory>
```

2. To generate a password and store it in `/etc/apache2/.htpasswd`, run the `htpasswd` command. Set the user name to whatever appropriate. This username is the username and password that clients will need to specify when they send HTTP requests to this server. In this example, `httpuser` is the user name.

```
htpasswd -c /etc/apache2/.htpasswd httpuser
```

3. Restart apache2.

Setting up HTTPs (TLS) on apache2

Procedure

1. Create a self-signed certificate for the apache2 server with the **`mkdir`** and **`openssl`** commands. For example:

```
mkdir /etc/apache2/ssl
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out
/etc/apache2/ssl/apache.crt
```

   When you press Enter after the `openssl` command, you are asked several questions:

2. Answer the questions when prompted:

```
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Your State
Locality Name (eg, city) []:Your City
Organization Name (eg, company) [Internet Widgits Pty Ltd]:YourCompany
Organizational Unit Name (eg, section) []:Department of Kittens
Common Name (e.g. server FQDN or YOUR name) []:abc.com
Email Address []:your_email@abc.com
```

   The key and certificate will be created and placed in the `/etc/apache2/ssl` directory.

3. Enable ssl:

```
a2enmod ssl
```

4. Edit `/etc/apache2/sites-available/default-ssl.conf`, make sure the Virtual Host setting is configured:

```
<VirtualHost _default_:443>
```

```
ServerAdmin admin@abc.com
ServerName abc.com
ServerAlias www.abc.com
DocumentRoot /var/www/html
SSLCertificateFile /etc/apache2/ssl/apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/apache.key
```

5. Enable the virtual host:

   ```
   a2ensite default-ssl.conf
   ```

6. Restart apache2.

Installing certificates on the client machine

Procedure

1. Do one of the following:

   - Get the certificate (`/etc/apache2/ssl/apache.crt`) from the server to the HPE ProLiant Gen10 system over HTTP and place it in the local file system. To do this, copy the certificate to the HTTP server's root directory /var/www/html/ and on the SUT Embedded UEFI Shell, run:

     ```
     FS0:\> webcli –g http://websrv.abc.com/apache.crt
     ```

     NOTE: A RAM Disk may be created if there is no local storage option available on the Gen10 system before downloading the certificate:

     ```
     Shell:> ramdisk –c –s 128
     ```

   - Use a USB device. Copy the certificate on the server `/etc/apache2/ssl/apache.crt` to a USB key and plug it on the Gen10 system.

2. Use the `tlsconfig` command to install the certificate:

   ```
   tlsconfig –e –f apache.crt
   ```

   NOTE: You can also use the iLO RESTful Interface Tool. For more information, see **https://hewlettpackard.github.io/ilo-rest-api-docs/ilo5/#https-boot-tls-configuration**.

Performing an HTTP(S) Boot using a supported boot loader

HTTP(S) Boot or Boot from URL can be performed in two ways:

1. Download and boot an ISO image (could be a mini/net installer image or full media image). By completing all the tasks in **Configuring the network services to provide the boot URL**, you have configured the system to boot this way.
2. By downloading a Network Bootstrap Program such as an EFI boot loader (for example, grub.efi). This is similar to the PXE boot.

For the GEN10 release, SLES12 SP2, SP3, SLES 15, and RHEL 7.6 support both HTTP(S) boot methods.

	SUSE Linux Enterprise Server 12 SP2, SP3	SUSE Linux Enterprise Server 15	Red Hat Enterprise Linux 7.6
Method 1 with HTTP URL	Yes	Yes	Yes
Method 2 with HTTP URL	Yes	Yes	Yes
Method 1 with HTTPS URL	Yes	Yes	Yes
Method 2 with HTTPS URL	X	Yes	Yes

By using the bootloader from a supported OS installation image, you can HTTP(S) boot another supported OS.

This task lists the steps used to set up the supported boot loader.

Procedure

1. Copy the install media contents to the server. For example, create a folder (`/sles12sp3`) in HTTP root directory (`/var/www/html/`). Mount the SLES12 SP3 installation media (ISO image or CD/DVD drive) and copy the files (or make a symbolic link) to `/var/www/html/sles12sp3/`.
2. Create `/boot/grub2/x86_64-efi/` folder in HTTP root directory and then copy `grub.efi` from installation media to the newly created folder. For example, copy `grub.efi` from `/var/www/html/sles12sp3/EFI/BOOT/` to `/var/www/html/boot/grub2/x86_64-efi/`.
3. Create `grub.cfg` in `/var/www/html/boot/grub2/` and configure it like the example following:

   NOTE: You can put `grub.cfg` in `/var/www/html/` and `grub.efi` in `/var/www/html/x86_64-efi/`.

   ```
   set timeout=60
   terminal_output console
   menuentry ' SLES 12 SP3' --class os {
   echo 'Loading Linux ...'
   linuxefi /sles12sp3/boot/x86_64/loader/linux install=http://192.168.111.4/sles12sp3/
   echo 'Loading initial ramdisk ...'
   initrdefi /sles12sp3/boot/x86_64/loader/initrd
   }
   menuentry ' SLES 15 (HTTPS)' --class os {
   echo 'Loading Linux ...'
   linuxefi (https,192.168.111.4)/sles15/boot/x86_64/loader/linux install=http://192.168.111.4/sles15/
   echo 'Loading initial ramdisk ...'
   initrdefi (https,192.168.111.4)/sles15/boot/x86_64/loader/initrd
   }
   menuentry 'RHEL 7.6' --class os {
   echo 'Loading Linux ...'
   linuxefi linuxefi /images/pxeboot/vmlinuz inst.stage2=http:// 192.168.111.4/EFI/BOOT/grubx64.efi inst.noverifyssl quiet
   echo 'Loading initial ramdisk ...'
   initrdefi /images/pxeboot/initrd.img
   }
   menuentry 'RHEL 7.6 (HTTPS)' --class os {
   echo 'Loading Linux ...'
   linuxefi linuxefi /images/pxeboot/vmlinuz inst.stage2=http:// 192.168.111.4/EFI/BOOT/grubx64.efi inst.noverifyssl quiet
   echo 'Loading initial ramdisk ...'
   initrdefi /images/pxeboot/initrd.img
   }
   ```

   NOTE: If the boot loader is not using the network port which is connected to HTTP server as its default network port, you may see that boot loader stops at the installation source selection menu. For this case, you can manually enter the URL (http://192.168.111.4/sles12sp3/) to let boot loader try through all network ports to continue the installation process.

4. Configure `dhcpd.conf` on the DHCP server to set the boot URL to point to `grub.efi`:

a. Enter one of the following:

- HTTP boot:

```
option bootfile-name "http://192.168.111.4/boot/grub2/x86_64-efi/
grub.efi";
```

- HTTPS boot:

```
option bootfile-name "https://192.168.111.4/boot/grub2/x86_64-efi/
grub.efi";
```

b. Restart the `isc-dhcp-server` service.

NOTE: If the client system (HPE ProLiant Gen10 or HPE Synergy compute module) uses Boot from URL instead of HTTP Boot, then the example URL may be entered in the "Boot from URL x" options directly.

Configuring Boot from URL

Prerequisites

Configuring Boot from URL with the System Utilities

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Network Options** > **Pre-Boot Network Settings**.
2. To change the Pre-boot Network interface to use a different NIC port, select and configure the **Pre-Boot Network Interface** option. Default is "Auto", which automatically uses the first network interface port with a network connection.
3. To set the URL, select and configure the **Boot from URL 1** option with a valid HTTP or HTTPs URL pointing to a bootable ISO or EFI file.

Configuring Boot from URL with the sysconfig command in the Embedded UEFI Shell

Procedure

1. From the **System Utilities** screen, select **Embedded Applications** > **Embedded UEFI Shell**.
2. At the Embedded UEFI Shell prompt, do ONE of the following:

- Specify a pre-boot network interface and configure Boot from URL 1 with an IPv4 address. For example:

```
Shell> sysconfig -s PreBootNetwork=Slot1NicPort1
UrlBootFile=http://192.168.1.1/bootx64.efi
```

- Set a URL to boot from the first network interface with a network connection (Pre-boot Network Interface=Auto). For example:

```
Shell> sysconfig -s UrlBootFile=http://server.domain.com/boot.iso
```

- Set an IPv6 URL to boot from a FlexibleLOM. For example:

```
Shell> sysconfig -s PreBootNetwork=FlexLom1Port1
UrlBootFile=http://[1235::1000]/bootx64.efi
```

3. From the **System Utilities** screen, select **One-Time Boot Menu**.
4. Select the **URL File** option.

Configuring Boot from URL with the RESTful Interface Tool

Locate the RESTful Interface Tool at **http://www.hpe.com/info/redfish**.

Procedure

1. Run the RESTful Interface Tool.
2. At the command-line prompt, do one of the following:
   - Specify a pre-boot network interface and configure Boot from URL 1 with an IPv4 address:

     `ilorest set PreBootNetwork=Slot1NicPort1 UrlBootFile=http://192.168.1.1/bootx64.efi`
   - Set a URL to boot from the default interface (Pre-boot Network Interface=Auto):

     `ilorest set PreBootNetwork=Auto UrlBootFile=https://server.domain.com/boot.iso`
   - Set an IPv6 URL to boot from a FlexibleLOM:

     `ilorest set PreBootNetwork=FlexLom1Port1 UrlBootFile=http://[1235::1000]/bootx64.efi`
3. Do one of the following for your changes to take effect:
   - If the system is running, reboot it.
   - If the system is powered off, power it on.

Pre-Boot network settings configuration

You can use the System Utilities, the Embedded UEFI Shell, or the RESTful Interface Tool to configure a pre-boot network interface and related settings.

NOTE: After you complete configuration changes using the RESTful API when the system is running, you must reboot the system for the changes to take effect. If the system was powered off at the time of configuration using the RESTful API, then the settings appear after you power on the system. Changes you make using the System Utilities or the UEFI Embedded Shell `sysconfig` command take effect immediately.

Configuring Pre-Boot Network Settings with the System Utilities

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **Network Options** > **Pre-Boot Network Settings**.
2. Select and configure the **Pre-Boot Network Settings**.

Pre-Boot Network Settings

- **Pre-Boot Network Interface**—Specifies the network interface used for pre-boot network connections.
  - **Auto** (default)—The system uses the first available port with a network connection.
  - **Select Specific Port** — The system uses the selected NIC port.
- **DHCPv4** — Enables or disables obtaining the preboot network IPv4 configuration from a DHCP server for Network operations from the Embedded UEFI Shell.
  - **Enabled** — Enables DHCPv4 network address configuration. Individual settings are not available.
  - **Disabled** — Disables DHCPv4 address configuration, requiring you to configure the following static IP address settings manually.
    - **IPv4 Address**
    - **IPv4 Subnet Mask**
    - **IPv4 Gateway**
    - **IPv4 Primary DNS**
    - **IPv4 Secondary DNS**
- **Preboot Network Proxy**—Specifies a preboot network proxy. When set, network operations for the Pre-Boot Network Interface are attempted through the configured proxy. The proxy must be in an HTTP URL format, and can be specified as `http://IPv4_address:port`, `http://[IPv6_address]:port` or `http://FQDN:port`.
- **IPv6 Config Policy**

- **Automatic**—Enables preboot network IPv6 configuration to be automatically obtained for Network operations from the Embedded UEFI Shell. Individual settings are not available.
- **Manual**—Enables you to configure static IP address settings individually.

- **Boot from URL 1, 2, 3 or 4**—Specifies a network URL to a bootable ISO or EFI file. Enter a URL in either HTTP or HTTPS format, using either an IPv4 or IPv6 server address or host name. For example, the URLs can be in any of the following formats: `http://192.168.0.1/file/image.iso`, `http://example.com/file/image.efi`, `https://example.com/file/image.efi`, `http://[1234::1000]/image.iso`. When configured, this URL is listed as a boot option in the UEFI Boot menu. Then you can select this option from the boot menu to download the specified file to the system memory and enable the system to boot from the file.

NOTE: Boot from URL does not depend on the "DHCPv4" and "IPv6 Config Policy" settings.

Booting from an ISO file can involve only booting a preliminary OS environment image, such as WinPE or a mini Linux, or a complete OS install image if the OS supports the HTTP Boot feature (Old OS versions may not support booting from an ISO file or OS install image). Please check your OS documentation for the HTTP Boot feature support.

IMPORTANT: If you plan to run `webclient` or `ftp` over the same interface, you do not need to use the Embedded UEFI Shell `ifconfig` command on a network interface. The **Pre-Boot Network Settings** configured in the System Utilities automatically selects these interface.

If the interface used by `ftp` and `webclient` are configured by `ifconfig`, that setting is erased. Instead, the System Utilities **Pre-Boot Network Settings** menu is applied on the interface when the commands are run.

Configuring Pre-Boot Network Settings with the sysconfig command in the Embedded UEFI Shell

Procedure

1. From the **System Utilities** screen, select **Embedded Applications** > **Embedded UEFI Shell**.
2. At the Embedded UEFI Shell prompt, do ONE of the following:

   - Use DHCP to get an IPv4 address on a specific NIC port. For example:

     ```
     Shell> sysconfig -s PreBootNetwork=EmbNicPort1 Dhcpv4=Enabled
     ```

   - Use IPv6 auto configuration on a specific NIC port. For example:

     ```
     Shell> sysconfig -s PreBootNetwork=FlexLom1Port1 Ipv6ConfigPolicy=Automatic
     ```

   - Let the system choose the interface with a network connection, but use a static IPv4 address. For example:

     ```
     Shell>sysconfig -s PreBootNetwork=Auto Dhcpv4=Disabled Ipv4Address=192.168.1.66
     Ipv4SubnetMask=255.255.255.0 Ipv4Gateway=192.168.1.1 Ipv4PrimaryDNS=192.168.1.1
     ```

```
Ipv4SecondaryDNS=192.168.1.2
```

- Let the system choose the interface with a network connection, but use a static IPv6 address. For example:

```
Shell>sysconfig -s PreBootNetwork=Auto Ipv6ConfigPolicy=Manual
Ipv6Address=1235::1022 Ipv6Gateway=1235::1000 Ipv6PrimaryDNS=1235::1000
```

3. Use the **webcli** or **ftp** command to download or upload files.

Configuring Pre-boot Network Settings with the RESTful Interface Tool

Locate the RESTful Interface Tool at **http://www.hpe.com/info/redfish**.

Procedure

1. Run the RESTful Interface Tool.

```
ilorest login proliantsystem1.hpe.com –u username –p password
ilorest select Bios.v1_0_0
```

2. At the command-line prompt, do one of the following:

   - Use DHCP to get an automatic IPv4 address configuration on a specific NIC port:

     ```
     ilorest set PreBootNetwork=EmbNicPort1 Dhcpv4=Enabled
     ```

   - Use IPv6 auto configuration on a specific NIC port. For example:

     ```
     ilorest set PreBootNetwork=FlexLom1Port1 Ipv6ConfigPolicy=Automatic
     ```

   - Let the system choose the Dhcpv4 interface with a network connection, but use a static IPv4 address:

     ```
     ilorest set PreBootNetwork=Auto Dhcpv4=Disabled Ipv4Address=192.168.1.66
     Ipv4SubnetMask=255.255.255.0 Ipv4Gateway=192.168.1.1 Ipv4PrimaryDNS=192.168.1.1
     Ipv4SecondaryDNS=192.168.1.2
     ```

   - Let the system choose the IPv6 interface with a network connection, but use a static address:

     ```
     ilorest set PreBootNetwork=Auto Ipv6ConfigPolicy=Manual Ipv6Address=1235::1022
     Ipv6Gateway=1235::1000 Ipv6PrimaryDNS=1235::1000 Ipv6SecondaryDNS=1235::1001
     ```

3. Do one of the following for your changes to take effect:

   - If the system is running, reboot it.
   - If the system is powered off, power it on.

Embedded UEFI Shell boot

The system BIOS includes an Embedded UEFI Shell in the ROM. Based on the **UEFI Shell Specification**, the Embedded UEFI Shell environment provides an API and a CLI you can use for scripting, file manipulation, and obtaining system information. The Embedded UEFI Shell also runs other UEFI applications. These features enhance the capabilities of the UEFI System Utilities. Access to the Embedded UEFI Shell is enabled by default.

The following information describes how to use the Embedded UEFI Shell CLI to configure the system BIOS and automate OS deployment and installation.

Embedded UEFI Shell commands for network deployment

The following table lists Embedded UEFI Shell commands that you can use for network deployment. For more information about each command, see the UEFI Shell user guide.

Command	Description
`ramdisk`	Creates and deletes RAM disks.
`restclient`	Interacts with the local RESTful API service.
`sysconfig`	Displays or configures system BIOS settings.
`webclient`	Downloads files from HTTP or FTP, and mounts an ISO file system.
`tlsconfig`	Displays and modifies TLS connection settings and certificates.

Standard Format Output (SFO)

The general, table-based standard output format (SFO) command option enables you to easily process Embedded UEFI Shell command output. Shell commands that use the standard formatted output display the same information that would normally be displayed, except that the commands use rows and columns of comma-delimited data which can be parsed using a parse command. The first column always contains a C-style identifier that describes the type of data on the row. This identifier is known as the `table name`. Table names that begin with the '_' character are implementation-specific.

The second and subsequent columns are quoted C-style strings containing the actual Embedded UEFI Shell command data. For each Embedded UEFI Shell command, the format and meaning of each column depends on the column number and the `table name`.

Embedded UEFI Shell commands that support the `-sfo` option always produce the table name `ShellCommand`. The second column contains the name of the Embedded UEFI Shell command without any extension. For example: `ShellCommand,"ls"`.

Example

```
FileInfo,"fs0:/efi/boot/winloader.efi","45670","arsh"
FileInfo,"fs0:/efi/boot/timsfile.txt","1250","a"
FileInfo,"fs0:/efi/boot/readme.txt","795","a"
```

Extended syntax

In the syntax below, an `identifier` is a C-style identifier that starts with an alphabetic character or underscore. A quoted string starts with a double-quotation mark (") character, followed by zero or more

characters, and concluding with a double-quotation mark (") character. Quotation marks in the string must be escaped by using a ^ character (such as ^"). The ^ character can be inserted using ^^.

```
sfo-format := sfo-row
sfo-row <EOL> <sfo-row>
sfo-row := sfo-table-name, sfo-columns
sfo-table-name := identifier
sfo-columns := sfo-column |
sfo-columns, | sfo-column
sfo-column := quoted-string |
<empty>
```

Example

```
VolumeInfo, "TimsVolume","400000000","32000000","16000000"
FileInfo, "fs0:/efi/boot/winloader.efi","45670","arsh"
FileInfo, "fs0:/efi/boot/timsfile.txt","1250","a"
FileInfo, "fs0:/efi/boot/readme.txt","795","a"
```

Invoking Shell scripts

You can invoke UEFI Shell scripts using either of the following two methods.

UEFI Shell Script Auto-Start in the System Utilities

The startup script enables you to create a RAM disk, download files from the network, collect data, upload results back to network, and then boot to the OS without rebooting the system. You can store the script file on local media, or access it from a network location.

By default, **UEFI Shell Script Auto-Start** is disabled in the System Utilities and can be configured so that the Shell looks for the `startup.nsh` file in any FAT16 or FAT32 file systems available. You can modify these settings so that the Shell looks for the startup script in a specific file system on attached media, or in a specific network location. When configured for a network location, you can specify the URL in HTTP or FTP format of the `startup.nsh` file location or have the system use DHCP to get the location (URL) of the script. For more information, see “UEFI Shell Script Auto-start“ in the UEFI System Utilities user guide.

Manually invoking a Shell script

Procedure

- Do one of the following:
  - Navigate to the location of the `.nsh` script file and enter the name of the script.
  - Enter the full pathname of the script.

Editing Shell scripts

You can edit script files offline or in the Shell using the `edit` command. You can also use the `type` command to output the script to the screen.

Configuring the Embedded UEFI Shell to read the startup script from the network

You can configure the Embedded UEFI Shell to read the startup script from the network. You can do the configuration with either the UEFI System Utilities, the `sysconfig` command, or the RESTful Interface Tool.

Configuring the Embedded UEFI Shell to read the startup script (System Utilities)

Procedure

1. Complete the preboot network configuration, but do not enter data in the **Boot from URL** field. This field is only needed for booting from an ISO image or EFI application. See **Configuring Pre-Boot Network Settings with the System Utilities**.
2. Enable UEFI Shell Script Auto Start. By default this setting is disabled. See “UEFI Shell Script Auto-start” in the UEFI System Utilities user guide.
3. Set the auto-start location to Network. See “Shell Auto-start Script Location” in the UEFI System Utilities user guide.
4. Set the location (URL) that contains the path to the Embedded UEFI Shell Script. See “Network Location for Shell Auto-start Script” in the UEFI System Utilities user guide. Alternatively, you can let the Shell discover the script URL using DHCP by enabling "Discover Shell Auto-Start Script using DHCP". For additional information, see "Enabling or disabling discovery of the Shell auto-start script using DHCP" in the UEFI System Utilities Guide for HPE ProLiant Gen10 Servers and HPE Synergy.

NOTE: You set the URL statically to point to an Embedded UEFI Shell script (`.nsh`). No other files are supported.

NOTE: When using an https URL, ensure the server certificate is enrolled in the system. Use the `tlsconfig` shell command or **RBSU** > **Server Security** > **TLS (HTTPS) Options**.

NOTE: If the URL is accessible through an HTTP proxy, then set the **Pre-Boot Network Proxy** option under **RBSU** > **Network Options** > **Pre-Boot Network Settings**.

Configuring the Embedded UEFI Shell to read the startup script (sysconfig command)

Procedure

1. From the **System Utilities** screen, select **Embedded Applications** > **Embedded UEFI Shell**. See **Configuring Pre-Boot Network Settings with the sysconfig command**.
2. At the Shell prompt, enter one of the `sysconfig` commands as shown in the following example.

   - To set a static URL, enter:

     ```
     Shell> sysconfig -s UefiShellStartup=Enabled
     UefiShellStartupLocation=NetworkLocation
     UefiShellStartupUrl=https://deploy.server.com/scripts/startup.nsh
     ```

   - To use DHCP to discover the URL, enter:

     ```
     Shell> sysconfig -s UefiShellStartup=Enabled
     UefiShellStartupLocation=NetworkLocation
     UefiShellStartupUrlFromDhcp=Enabled
     ```

Configuring the Embedded UEFI Shell to read the startup script (RESTful Interface Tool)

Procedure

1. Run the RESTful Interface Tool.

   ```
   ilorest login proliantsystem1.hpe.com –u username –p password
   ilorest select Bios.v1_0_0
   ```

2. Set up the Pre-Boot Network Settings. See **Configuring Pre-Boot Network Settings with the RESTful Interface Tool**.
3. At the command-line prompt, enter one of the following:

   - `ilorest UefiShellStartupLocation=NetworkLocation UefiShellStartupUrl=https://deploy.server.com/scripts/startup.nsh`
   - `ilorest UefiShellStartupLocation=NetworkLocation UefiShellStartupUrlFromDhcp=Enabled`

4. Do one of the following for your changes to take effect:

   - If the system is running, reboot it.
   - If the system is powered off, power it on.

Creating RAM disks

RAM disks are “in memory” file systems (block/storage devices emulated in RAM) that can be accessed like any storage device.

To create a 512 MB FAT32 RAM disk with the volume label `MYRAMDISK`, enter the following command at the Embedded UEFI Shell prompt:

```
Shell> ramdisk -c -s 512 -v MYRAMDISK -t F32
```

For more details, enter `help ramdisk, or ramdisk -?`

For more information about this command, see the UEFI Shell user guide.

Deployment and scripting

After you complete the **configuration steps** and boot to the Embedded UEFI Shell, the Shell automatically locates the `startup.nsh` file at the configured attached media or network location. It provisions RAM disks to download the script from network. The startup script can then automatically perform system configuration and start the OS deployment and installation process.

Configuring Embedded UEFI Shell boot: A sample deployment solution

Procedure

1. Configure preboot network settings using either the System Utilities, the sysconfig command, or the RESTful Interface Tool.

See **Configuring Pre-Boot Network Settings with the System Utilities**, **Configuring Pre-Boot Network Settings with the sysconfig command**, or **Configuring Pre-Boot Network Settings with the RESTful Interface Tool**.

2. Configure the Embedded UEFI Shell settings to read the startup script using either the System Utilities, the `sysconfig` command, or the RESTful Interface Tool.

   See **Configuring the Embedded Shell to read the startup script (System Utilities)**, **Configuring the Embedded UEFI Shell to read the start up script (sysconfig command)**, or **Configuring the Embedded UEFI Shell to read the startup script (RESTful Interface Tool)**.

3. Do one of the following:

   a. Select the Embedded UEFI Shell from the **One-Time Boot** menu. See **Selecting an option for a one-time boot** on page 10.

   b. If you want the Embedded UEFI Shell to always boot first, move it to the top of the **UEFI Boot Order** list. See **Changing the UEFI Boot Order list** on page 8.

Sample startup script

If you configured the Embedded UEFI Shell boot using the RESTful Interface tool, then the system boots up, configures itself with the requested settings, and reboots again. The system then launches the Embedded UEFI Shell, which downloads and runs the startup script from the configured URL.

If you configured the Embedded UEFI Shell boot using the System Utilities, you can launch the Embedded Shell immediately. The Embedded Shell downloads and runs the startup script from the URL.

```
@echo -off

#
# Setup the environment variables. All of them are created as volatile.
#

#
# The volume label for the RAMDISK.
#
set -v VolumeLabel MYRAMDISK

#
# Variable to store the file system index that will be looped
# to determine the FS<x> number for the RAMDISK that is created.
#
set -v FsIndex 0

#
# Variable to store the output string of the ramdisk -c command.
# Successful creation of RAMDISK will give the following output:
# "RAM disk 'FSx:' created successfully." where x=0,1,2,...
#
set -v RamDiskStr 0

#
# Size of the RAMDISK in MegaBytes (MB).
#
set -v RamDiskSize 512

#
# Server URL hosting the OS loader and images.
# Can be HTTP or FTP. Names or IP addresses are allowed.
# Ensure DNS service is available and configured (see pre-requisites)
# when server names are used.
#
set -v Url https://192.168.1.1
```

```
#
# Files to be downloaded
#
set -v DownloadFile1 efilinux.efi
set -v DownloadFile2 deploy.kernel
set -v DownloadFile3 deploy.ramdisk

#
# Step 1. Create RAMDISK to store the downloaded OS programs.
#
echo "Creating a RAM Disk to save downloaded files..."
ramdisk -c -s %RamDiskSize% -v %VolumeLabel% -t F32 >v RamDiskStr
if %lasterror% ne 0x0 then
  echo "Cannot create a RAMDISK of size %RamDiskSize%."
  goto EXITSCRIPT
endif
echo "RAM Disk with Volume Label %VolumeLabel% created successfully."

#
# Step 2. Check each word in the output (RamDiskStr) and see if it matches
# the FSx: pattern. The newly created RAMDISK will be FS1: or higher.
# Here the check goes up to FS3: (the inner for loop), but a larger limit
# may be used in case many other file systems already exist before
# the creation of this RAMDISK. The FS for the RAMDISK is found when the
# FsIndex matches the FS<x> in RamDiskStr. Change the working directory
# to FS<FsIndex>:, so all downloads get saved there.
#
# FS0: is ignored. In the worst case, when no other usable
# file system is present, FS0: will map to the file system
# that this script is executing from.
#
#
for %a in %RamDiskStr%
  for %b run (1 10)
    set -v FsIndex %b
    if 'FS%FsIndex%:' == %a then
      FS%FsIndex%:
      goto RDFOUND
    endif
  endfor
endfor

#
# The following message appears if the newly created RAMDISK cannot be found.
#
echo "RAMDISK with Volume Label %VolumeLabel% not found!"
goto EXITSCRIPT

#
# The following message appears if the RAMDISK FS<x> has been found and you are in the
# RAMDISK's root folder.
#
:RDFOUND
echo "RAMDISK with Volume Label %VolumeLabel% found at FS%FsIndex%:."

#
# Step 3: Download the required files into the RAMDISK.
#
echo "Downloading %Url%/deploy/%DownloadFile1% (File 1 of 3...)"
webclient -g %Url%/deploy/%DownloadFile1% -o %DownloadFile1%
if %lasterror% ne 0x0 then
  goto EXITSCRIPT
endif

echo "Downloading %Url%/deploy/%DownloadFile2% (File 2 of 3...)"
webclient -g %Url%/deploy/%DownloadFile2% -o %DownloadFile2%
if %lasterror% ne 0x0 then
```

```
  goto EXITSCRIPT
endif

echo "Downloading %Url%/deploy/%DownloadFile3% (File 3 of 3...)"
webclient -g %Url%/deploy/%DownloadFile3% -o %DownloadFile3%
if %lasterror% ne 0x0 then
  goto EXITSCRIPT
endif

#
# Step4: Launch the boot loader.
#
echo "Starting the OS..."
%DownloadFile1% -f %DownloadFile2% initrd=%DownloadFile3%

#
# You reach here only if the downloads and booting failed.
#
:EXITSCRIPT
echo "Exiting Script."
```

The sample script does the following:

1. Creates a temporary RAM disk for saving the downloaded boot loader, the OS kernel, file system, and any configuration files required for the boot loader and kernel to initialize themselves and proceed with the installation over the network.
2. Determines the `FS<x>` ID for the just-created RAM disk.
3. Sets the working directory to the root of the RAM disk. For example: `FS1:\`.
4. Downloads the required files to launch the OS: the boot loader, the OS kernel, and an in-memory file system for the OS kernel.
5. Does one of the following.
   a. If the download succeeds, launches the boot loader. The path to the OS kernel file, its in-memory file system, and any arguments to the OS kernel (that the boot loader passes to the kernel upon launching it), are passed to the boot loader as command line arguments.
   b. If the download fails, performs cleanup and exits the startup script.

The role of the Embedded UEFI Shell and the pre-boot script ends here, and the OS can proceed with the deployment on its own, with the help of OS-specific deployment scripts embedded in its in-memory file system.

FC/FCoE SAN boot in UEFI Mode

FC/FCoE SAN boot configuration

By default, the System Utilities enables installed FC/FCoE adapters to only scan boot targets that are preconfigured in the device settings for the target. You can change the boot policy settings. You can then boot from FC/FCoE SAN in either UEFI Mode or Legacy BIOS Mode. The following describes doing so in UEFI Mode.

NOTE: To avoid long boot times, do not set your FC scan policy to scan all available targets. You can disable this setting in the UEFI System Utilities options for your specific adapter or by setting a global scan policy in the System Utilities.

Booting from FC/FCoE SAN

To boot from FC/FCoE SAN using the System Utilities:

Prerequisites

- EFI System Partition (ESP) and boot loader files are stored at the default location `\efi\BOOT\BOOTX64.EFI`.
- The SAN boot target is GPT (GUID Partition Table) configured.
- Boot mode is set to **UEFI Mode**.

Procedure

1. If necessary, do one of the following to change your boot policy setting.
   a. For a Fibre Channel controller that is not in a PCIe slot, modify your setting as described in “Fibre Channel/FCoE Scan Policy” in the *UEFI System Utilities User Guide for HPE ProLiant Gen10 Servers and HPE Synergy*.
   b. For a Fibre Channel controller that is in a PCIe slot, modify your setting as described in “PCIe Storage Boot Policy” in the *UEFI System Utilities User Guide for HPE ProLiant Gen10 Servers and HPE Synergy*.
2. Move this boot option to the top of the boot order list as described in **Changing the UEFI Boot Order list**.
3. Reboot the server.

Local HDD boot in UEFI Mode

HDD boot configuration

You can boot from a local hard disc drive, including one that is attached to an embedded SATA or embedded (SAS-based) Smart Array controller, in either UEFI Mode or Legacy BIOS Mode. The following describes local HDD boot in UEFI Mode.

Embedded SATA support

To boot from a local SATA hard disk drive that is attached to an embedded SATA, you must enable the appropriate SATA controller for your installed OS. Depending on your server model, either SATA AHCI or Dynamic Smart Array RAID support are enabled by default.

CAUTION: Dynamic Smart Array is not supported when your boot mode is configured to Legacy BIOS Mode. You must use UEFI Mode when enabling Dynamic Smart Array RAID to avoid data loss or data corruption on existing SATA drives. Back up all drives before enabling this option.

See your operating system documentation before enabling SATA AHCI support to ensure your base media drivers support this feature.

Enabling Embedded SATA support

Prerequisite

For partitions larger than 2.2 TB, boot mode is set to UEFI Mode.

Procedure

1. From the **System Utilities** screen, select **System Configuration** > **BIOS/Platform Configuration (RBSU)** > **System Options** > **SATA Controller Options** > **Embedded SATA Configuration**.
2. Ensure that you are using the correct AHCI or RAID system drivers for your SATA option.
3. Select a setting:
   a. **Enable SATA AHCI Support**—Enables the embedded chipset SATA controller for AHCI.
   b. **Enable HP Dynamic Smart Array RAID Support**—Enables the embedded chipset SATA controller for Dynamic Smart Array RAID.
4. Save your setting.
5. In UEFI mode, move this UEFI boot option to the top of the UEFI boot order list as described in **Changing the UEFI Boot Order list**.
6. Reboot the server.

Troubleshooting

Cannot download the file in the network boot URL

Symptom

You see an error message when you try to download the file in the URL you specified for a network boot.

Solution 1

Cause

The network connection settings you specified during static configuration are incorrect.

Action

1. Use the Embedded UEFI Shell `ping` command to check the network connection. See “Ping” in the UEFI Shell user guide.
2. Change your static networking connection settings and try to download the file in URL again.

Solution 2

Cause

The DHCP server did not respond.

Action

1. Ensure there is a DHCP server available and it is operational.
2. Try to download the file in the URL again.

Solution 3

Cause

No cable is connected to the selected NIC port.

Action

1. Ensure there is a cable connection.
2. Try to download the URL again.

Solution 4

Cause

The file is incorrect or not present on the server, or it cannot be downloaded due to insufficient privileges. Check the file name and that it exists on the server. Make sure you have the privilege to download from the server.

Action

1. Ensure the file is present, that you are using the correct file name and have sufficient privileges to download it.
2. Try to download the file in the URL again.

Solution 5

Cause

The HTTP server is down or did not respond.

Action

1. Ensure that the HTTP server you specified is available and that it is operational.
2. Try to download the file in the URL again.

Cannot deploy from the UEFI Shell script

Symptom

You attempted to deploy an OS using the UEFI Shell script and you see an error message that the deployment failed.

Cause

Configuration settings are not correct.

Action

1. Verify the following.
   a. The Embedded UEFI Shell interface is added to the **UEFI Boot Order** list or **One-Time Boot Menu**.
   b. When added to the **UEFI Boot Order** list, the Embedded UEFI Shell interface is the first boot option in **UEFI Boot Order** list so that it overrides other boot options to load.
   c. UEFI Shell Script Auto-Start is enabled.
   d. The correct `startup.nsh` script file location in attached media or a network location is specified. If it is in attached media, the `startup.nsh` script must be inside the `fs`*`X`*`:\` directory.
   e. The `.nsh` script only contains supported commands.
   f. Your system has enough RAM memory to create RAM disks during automated script execution.
   g. Any OS boot loader or diagnostics application launched using the `.nsh script`is supported to run in UEFI the environment.
2. Try the deployment again.

Websites

General websites

Hewlett Packard Enterprise Information Library

www.hpe.com/info/EIL

Single Point of Connectivity Knowledge (SPOCK) Storage compatibility matrix

www.hpe.com/storage/spock

Storage white papers and analyst reports

www.hpe.com/storage/whitepapers

UEFI Specification

www.uefi.org/specifications

UEFI Learning Resources

www.uefi.org/learning_center

RESTful API Tool

http://www.hpe.com/info/redfish

Contact Hewlett Packard Enterprise Worldwide

http://www.hpe.com/assistance

Subscription Service/Support Alerts

http://www.hpe.com/support/e-updates

Software Depot

http://www.hpe.com/support/softwaredepot

Customer Self Repair

http://www.hpe.com/support/selfrepair

Insight Remote Support

http://www.hpe.com/info/insightremotesupport/docs

For additional websites, see **Support and other resources**.

Support and other resources

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:

  http://www.hpe.com/info/assistance

- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:

  http://www.hpe.com/support/hpesc

Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates:

  Hewlett Packard Enterprise Support Center

  www.hpe.com/support/hpesc

  Hewlett Packard Enterprise Support Center: Software downloads

  www.hpe.com/support/downloads

  Software Depot

  www.hpe.com/support/softwaredepot

- To subscribe to eNewsletters and alerts:

  www.hpe.com/support/e-updates

- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:

  www.hpe.com/support/AccessToSupportMaterials

IMPORTANT: Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HPE Passport set up with relevant entitlements.

Customer self repair

Hewlett Packard Enterprise customer self repair (CSR) programs allow you to repair your product. If a CSR part needs to be replaced, it will be shipped directly to you so that you can install it at your convenience. Some parts do not qualify for CSR. Your Hewlett Packard Enterprise authorized service provider will determine whether a repair can be accomplished by CSR.

For more information about CSR, contact your local service provider or go to the CSR website:

http://www.hpe.com/support/selfrepair

Remote support

Remote support is available with supported devices as part of your warranty or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which will initiate a fast and accurate resolution based on your product's service level. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

If your product includes additional remote support details, use search to locate that information.

Remote support and Proactive Care information

HPE Get Connected

www.hpe.com/services/getconnected

HPE Proactive Care services

www.hpe.com/services/proactivecare

HPE Proactive Care service: Supported products list

www.hpe.com/services/proactivecaresupportedproducts

HPE Proactive Care advanced service: Supported products list

www.hpe.com/services/proactivecareadvancedsupportedproducts

Proactive Care customer information

Proactive Care central

www.hpe.com/services/proactivecarecentral

Proactive Care service activation

www.hpe.com/services/proactivecarecentralgetstarted

Warranty information

To view the warranty information for your product, see the links provided below:

HPE ProLiant and IA-32 Servers and Options

www.hpe.com/support/ProLiantServers-Warranties

HPE Enterprise and Cloudline Servers

www.hpe.com/support/EnterpriseServers-Warranties

HPE Storage Products

www.hpe.com/support/Storage-Warranties

HPE Networking Products

www.hpe.com/support/Networking-Warranties

Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at the Hewlett Packard Enterprise Support Center:

www.hpe.com/support/Safety-Compliance-EnterpriseProducts

Additional regulatory information

Hewlett Packard Enterprise is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (Regulation EC No 1907/2006 of the European Parliament and the Council). A chemical information report for this product can be found at:

www.hpe.com/info/reach

For Hewlett Packard Enterprise product environmental and safety information and compliance data, including RoHS and REACH, see:

www.hpe.com/info/ecodata

For Hewlett Packard Enterprise environmental information, including company programs, product recycling, and energy efficiency, see:

www.hpe.com/info/environment

Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (**docsfeedback@hpe.com**). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.