Hewlett Packard
Enterprise

# UEFI Deployment Guide for HPE ProLiant Gen9 Servers

## Abstract

This guide details how to use various deployment methods to boot Unified Extensible Firmware Interface (UEFI)-based ProLiant Gen9 servers. This document is for the person who installs, administers, and troubleshoots servers and storage systems.

Part Number: 814283-004
Published: June 2016
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**

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

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

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

# Contents

# 1 Introduction

ProLiant Gen9 servers support an industry standard system firmware based on UEFI. By leveraging an enhanced built-in network stack and an Embedded UEFI Shell environment, ProLiant UEFI server deployment options offer improved reliability, security, scalability, and faster download speeds than existing network-based deployment solutions.

## UEFI deployment methods

The following methods are available for installing and deploying UEFI-based ProLiant Gen9 servers.

- **SD (Secure Digital)**—Uses an SD card as an alternative to hard drive or solid state drive storage as a target for deploying an OS on ProLiant Gen9 servers. You can use any capacity card that is on the market today.
- **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 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/FTP URL**—Downloads and boots files from a network location (URL) using HTTP and FTP 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 disk attached to a SATA AHCI or a Dynamic Smart Array RAID controller.

# 2 Configuring UEFI boot settings

## Boot settings

You can use the iLO 4 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.

More information

*HPE iLO 4 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

ProLiant Gen9 servers provide 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/FTP URL, the Embedded User Partition, and the Dynamic Smart Array B140i controller.
- **Legacy BIOS Mode**—This boot mode configures the server to boot traditional operating systems in legacy BIOS-compatibility mode. 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

### Prerequisite

When booting to **UEFI Mode**, leave **UEFI Optimized Boot** enabled so that the system to use native UEFI graphic drivers.

### To select the Boot Mode:

1. From the **System Utilities** screen, select **System Configuration→BIOS/Platform Configuration (RBSU)→Boot Options→Boot Mode** and press **Enter**.
2. Select a setting and press **Enter**.
   - **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. Press **F10**.
4. Reboot the server.

## UEFI Optimized Boot

Use this option to control whether the system BIOS boots using native UEFI graphic drivers.

Before changing this setting, consider the following:

- If you are running Microsoft Windows 2008 or Windows 2008 R2 operating systems, and the system is configured for UEFI Mode, set this option to disabled. Legacy BIOS Mode components are needed for video operations in Windows.
- When this option is enabled, **Boot Mode** must be set to **UEFI Mode**. See Selecting the boot mode.
- This option must be enabled to:
  - Enable and use **Secure Boot**. See “Secure Boot” in the *UEFI System Utilities User Guide for HPE ProLiant Gen9 Servers*.
  - Operate VMware ESXi.

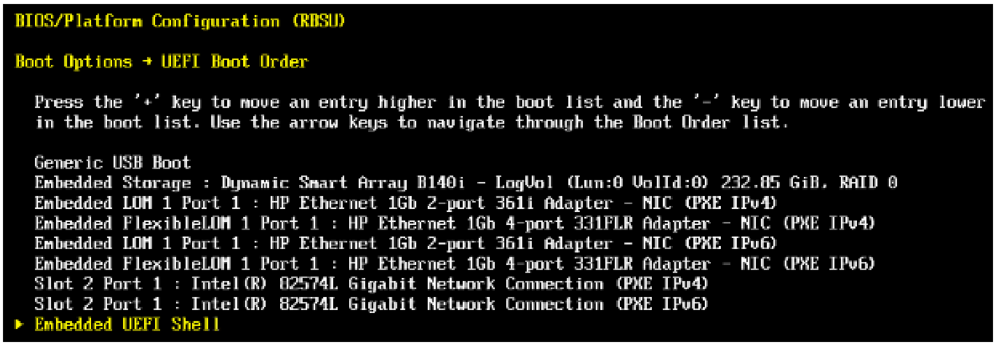
## Enabling or disabling UEFI Optimized Boot

1. From the **System Utilities** screen, select **System Configuration→BIOS/Platform Configuration (RBSU)→Boot Options→UEFI Optimized Boot** and press **Enter**.
2. Select an option and press **Enter**.
   - **Enabled** (default)—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.
3. Press **F10**.
4. Reboot the server.

## UEFI Boot Order

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

**Figure 1 UEFI Boot Order list example**



```
BIOS/Platform Configuration (RBSU)

Boot Options → UEFI Boot Order

  Press the '+' key to move an entry higher in the boot list and the '-' key to move an entry lower
  in the boot list. Use the arrow keys to navigate through the Boot Order list.

  Generic USB Boot
  Embedded Storage : Dynamic Smart Array B140i - LogVol (Lun:0 VolId:0) 232.85 GiB, RAID 0
  Embedded LOM 1 Port 1 : HP Ethernet 1Gb 2-port 361i Adapter - NIC (PXE IPv4)
  Embedded FlexibleLOM 1 Port 1 : HP Ethernet 1Gb 4-port 331FLR Adapter - NIC (PXE IPv4)
  Embedded LOM 1 Port 1 : HP Ethernet 1Gb 2-port 361i Adapter - NIC (PXE IPv6)
  Embedded FlexibleLOM 1 Port 1 : HP Ethernet 1Gb 4-port 331FLR Adapter - NIC (PXE IPv6)
  Slot 2 Port 1 : Intel(R) 82574L Gigabit Network Connection (PXE IPv4)
  Slot 2 Port 1 : Intel(R) 82574L Gigabit Network Connection (PXE IPv6)
▸ Embedded UEFI Shell
```

You can also configure the **UEFI Boot Order** list using the RESTful Interface Tool. See the RESTful Interface Tool documentation at: **http://www.hpe.com/info/restfulinterface/docs**.

## Changing the UEFI Boot Order list

1. From the **System Utilities** screen, select **System Configuration→BIOS/Platform Configuration (RBSU)→Boot Options→UEFI Boot Order** and press **Enter**.
2. Use the arrow keys to navigate within the boot order list.
3. Press the **+** key to move an entry higher in the boot list.
4. Press the **-** key to move an entry lower in the list.
5. Press **F10**.

## Legacy BIOS Boot Order

When your server is configured in Legacy BIOS Mode, you can use this setting to change the order in which the server looks for OS boot firmware within the **Legacy BIOS Boot Order** list.

## Changing the Legacy BIOS Boot Order list

### Prerequisite

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

### To change the Legacy BIOS boot order:

1. From the **System Utilities** screen, select **System Configuration→BIOS/Platform Configuration (RBSU)→Boot Options→Legacy BIOS Boot Order** and press **Enter**.
2. Use the arrow keys to navigate within the boot order list.
3. Press the **+** key to move an entry higher in the boot list.
4. Press the **-** key to move an entry lower in the list.
5. Press **F10**.
6. Reboot the server.

## Add Boot Option

Use this 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.

## Adding a boot option to the UEFI Boot Order list

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→Advanced UEFI Boot Maintenance→Add Boot Option** and press **Enter**.
3. Browse for an .EFI application from the list and 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 press **Enter**.

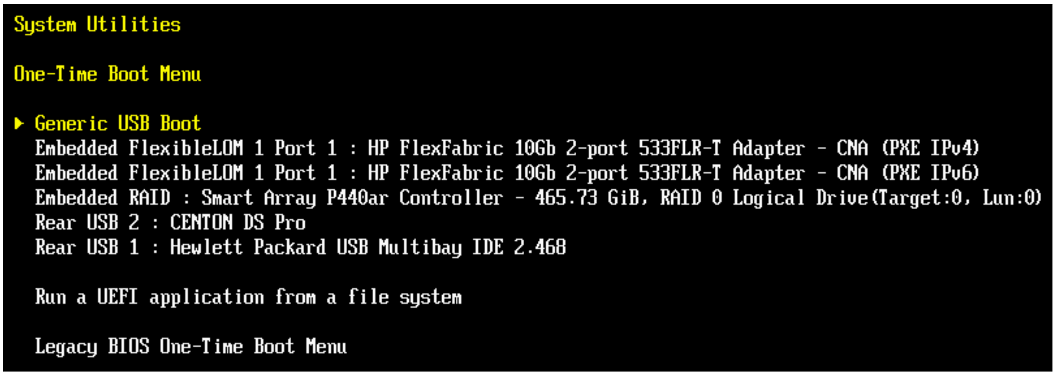
   The new boot option appears in the **UEFI Boot Order** list.
6. Select **Commit changes and exit** to save your selection.

## 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 4 Remote Console, exit and re-enter the System Utilities to refresh this menu so that the devices appear.

**Figure 2 One-Time Boot Menu**



```
System Utilities

One-Time Boot Menu

▶ Generic USB Boot
  Embedded FlexibleLOM 1 Port 1 : HP FlexFabric 10Gb 2-port 533FLR-T Adapter - CNA (PXE IPv4)
  Embedded FlexibleLOM 1 Port 1 : HP FlexFabric 10Gb 2-port 533FLR-T Adapter - CNA (PXE IPv6)
  Embedded RAID : Smart Array P440ar Controller - 465.73 GiB, RAID 0 Logical Drive(Target:0, Lun:0)
  Rear USB 2 : CENTON DS Pro
  Rear USB 1 : Hewlett Packard USB Multibay IDE 2.468

  Run a UEFI application from a file system

  Legacy BIOS One-Time Boot Menu
```

Boot options include:

- OS boot manager, such as **Windows Boot Manager**—Lists the boot manager for your installed OS.
- **Generic USB Boot**—Provides a place holder 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 Flexible LOMs
- Embedded UEFI Shell
- Embedded SATA Port
- **Run a UEFI Application from a file system**—Enables you 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 a .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.

## Selecting an option for a one-time boot

1. From the **System Utilities** screen, select **One-Time Boot Menu** and press **Enter**.
2. Select an option and press **Enter**.
3. Reboot the server.

# 3 SD boot

## OS to SD card deployment using Intelligent Provisioning

Intelligent Provisioning, a single-server deployment tool embedded in ProLiant Gen9 servers, supports deploying an OS to an SD card. The SD card then provides SD boot and OS drive space functionality similar to a hard disk drive or a solid state drive. When you deploy an OS through Intelligent Provisioning, a Hewlett Packard Enterprise-approved SD card (if installed) is listed as a target choice for OS deployment. This provides a consistent and simple method for deploying supported operating systems to SD media.

## Deploying an OS to an SD card

### Prerequisites

- To boot an OS in UEFI Mode, the SD card is GPT-formatted.
- To boot an OS in Legacy BIOS Mode, the SD card is MBR-formatted.
- To boot an OS installer or a pre-boot application in either boot mode, the SD card is formatted as FAT16 or FAT32.
- The **Internal SD Card Slot** option in the System Utilities is enabled. This is the default setting. See “Internal SD Card slot” in the *UEFI System Utilities User Guide for HPE ProLiant Gen9 Servers*.

### Deploying an OS:

1. Launch Intelligent Provisioning.
2. View and modify hardware settings, including system profiles, array configuration, fiber channel configuration, and SD card configuration.
3. View and modify the OS selection information, including OS family, install method, and source media type.
4. View and modify OS information, including version and keyboard type.
5. Review and save your settings.
6. Change the boot order so that the SD card is at the top of the boot order list.

More information

*HPE Intelligent Provisioning User Guide for HPE ProLiant Gen9 Servers*

# 4 USB boot

## USB boot options

- Creating a bootable UEFI USB flash drive for installing Windows
- Using the Embedded User Partition to install and boot VMware ESXi
- Booting from an HPE Dual MicroSD device
- Creating a dual boot USB key for both UEFI Mode and Legacy BIOS Mode

## 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:** 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.

More information

One-Time Boot Menu options

## Creating a bootable UEFI USB flash drive for installing Windows

### Prerequisites

- 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. This is the default configuration. For more information, see “USB Options” in the *UEFI System Utilities User Guide for HPE ProLiant Gen9 Servers*.

### To create a bootable USB:

1. Connect the USB flash drive to a Windows 7 or 8/8.1 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 2008 R2 or 2012 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 2008 R2 or Windows 2012/2012 R2 using UEFI with the bootable UEFI USB flash drive.

## Using the Embedded User Partition to install and boot VMware ESXi

**NOTE:** Hewlett Packard Enterprise recommends that you regularly back up data on the Embedded User Partition.

### Prerequisite

Boot mode is set to **UEFI Mode** (the default setting).

### To install and boot VMware ESXi:

1. Enable the **Embedded User Partition**:
   a. From the **System Utilities** screen, select **System Configuration→BIOS/Platform Configuration (RBSU)→System Options→USB Options→Embedded User Partition** and press **Enter**.
   b. Select **Enabled** and press **Enter**.
   c. If necessary, format the partition.
   d. Press **F10**.
2. Change the boot order so that the Embedded User Partition is at the top of the boot order list.
3. Reboot the server.

## Booting from an HPE Dual MicroSD device

An HPE Dual MicroSD is similar to a USB boot device. It can contain redundant images of an OS, such as VMware ESX. For information, see the *HPE Dual 8GB MicroSD Enterprise Midline USB User Guide*: **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:

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.

# 5 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** is the leading open source network boot firmware. It provides an enhanced PXE implementation that enables you to:
    - Boot from a web server or a WAN via HTTP.
    - “Chainload” into iPXE through an TFTP server to obtain the features of iPXE without having to reflash the network card of each server individually.
    - Unify the PXE boot process by using iPXE across all servers, including pre-UEFI HPE and non-HPE servers.
    - Control the boot process with a script.

      You can use iPXE to gather the server’s unique identifiers, such as serial number, product name, manufacturer, UUID, MAC addresses, and any SMBIOS value, and configure your scripts to execute specific actions based on the type or identity of the machine running an iPXE script.

  On ProLiant Gen9 servers, iPXE boots when the `filename` on the DHCP server is set to `ipxe.efi`. 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`.

  More information

  PXE configuration
  iPXE configuration

## 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

PXE operation requires DHCP and TFTP. Linux DHCP is usually provided by the ISC dhcpd service or by dnsmasq.

- Configuring PXE servers for Legacy BIOS clients in a Linux environment shows how most current DHCP servers are configured, including example configurations for both the ISC dhcpd service and dnsmasq options.
- Configuring PXE servers for UEFI-based clients in a Linux environment shows example configurations of what you must add to enable DHCP to operate UEFI PXE in addition to Legacy BIOS PXE.

More information

HTTP/FTP URL boot

## Configuring PXE servers for Legacy BIOS clients in a Linux environment

The standard Legacy BIOS boot loader is pxelinux. To configure it:

1. Configure DHCP using one of the following methods:
   - Configuring ISC dhcpd
   - Configuring dnsmasq
2. Configure TFTP. See Configuring TFTP.

### Configuring DHCP

#### Configuring ISC dhcpd

1. Install the DHCP server package using the command associated with your Linux distribution.
2. Add the `filename` option to the `dhcpd.conf` file as shown in the following example.

**Example 1 Legacy BIOS configuration `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;
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;
}
```

#### Configuring dnsmasq

1. Install the dnsmasq package using the command associated with your Linux distribution.
2. Add the `dhcp-boot` option to the `dnsmasq.conf` file as shown in the following example.

**Example 2 Legacy BIOS configuration `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
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
```

### Configuring TFTP

1. Install the `tftp` package using the command associated with your Linux distribution.
2. In the `/etc/xinetd.d/tftp` configuration file, change the `disable` parameter from `yes` to `no` as shown in the following example.

### Example 3 Legacy BIOS configuration `tftp` file

```
# default: off
# description: tftp service is provided primarily for booting or when a \
#       router need an upgrade. Most sites run this only on machines acting as
#       "boot servers".
service tftp
{
        socket_type = dgram
        protocol = udp
        wait = yes
        flags = IPv6 IPv4
        user = root
        server = /usr/sbin/in.tftpd
        server_args = -s /tftpboot
        disable = no
}
```

3. To obtain the `pxelinux.0` file, install the Syslinux package.
4. To start `xinetd` and make sure it is enabled at boot time, do one of the following.
   - For a SysVinit configuration:

     ```
     service xinetd start
     ```

     ```
     chkconfig xinted on
     ```

   - For a non-SysVinit configuration, such as systemd:

     ```
     systemctl xinetd enable
     ```

     ```
     systemctl xinetd start
     ```

5. Verify that the boot loader files are in the `/tftpboot` directory.
6. To verify that the boot loader file is readable, enter:

   ```
   # ls -l /tftpboot/pxelinux.0
   ```

7. To verify that the TFTP server is enabled, enter:

   ```
   chkconfig tftp
   ```

   **NOTE:** The command syntax you enter for `chkconfig tftp` might be different, depending on the Linux version of your server.

Your `tftp` directory structure should look like the following example.

### Example 4 Sample TFTP directory structure

```
/tftpboot
/tftpboot/pxelinux.0
/tftpboot/pxelinux.cfg/default
/tftpboot/RHEL6.6
/tftpboot/RHEL7.0
/tftpboot/SLES11
/tftpboot/SLES12
/tftpboot/Trusty
```

Your default menu file should look like the following example.

### Example 5 Sample default file for Legacy BIOS PXE

```
default menu.c32
prompt 0
timeout 250
ONTIMEOUT local
#CONSOLE 0
#SERIAL 0 9600
menu title Linux PXE services
menu autoboot Starting Local System in # seconds
label
menu default
label local
menu label ^boot from local disk
COM32 chain.c32
append hd0
label hdt
menu label ^Hardware Detection Tool
COM32 hdt.c32
MENU SEPARATOR
label RHEL-6.6Server-x86_64
kernel RHEL6.6/vmlinuz
append vga=normal initrd=RHEL6.6/initrd.img repo=http://192.168.100.10/repo/RHEL6.6/disc1
label RHEL-7.0Server-x86_64
kernel RHEL7.0/vmlinuz
append vga=normal initrd=RHEL7.0/initrd.img repo=http://192.168.100.10/repo/RHEL7.60/disc1
label SLES-11Server-x86_64_kISO
kernel SLES11/kISO/linux
append initrd=SLES11/kISO/initrd.img repo=http://192.168.100.10/repo/RHEL6.6/disc1
addon=http://192.168.100.10/repo/kISO_SLES11_Gen9/disc1
label SLES-12Server-x86_64
kernel SLES12/linux
append initrd=SLES12/initrd.img repo=http://192.168.100.10/repo/SLES12/disc1
label Trusty-x86_64
kernel Trusty/linux
append initrd=Trusty/initrd.gz preseed/url=http://192.168.100.10/answers/trusty.cfg
```

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

UEFI PXE requires a different boot loader than Legacy BIOS PXE. The most common are ELILO and GRUB2. With the release of Syslinux 6.x, Syslinux added PXE capability.

The following examples show how to modify the existing DHCP and TFTP configurations shown in Configuring PXE servers for Legacy BIOS clients in a Linux environment to include UEFI. Additions in the examples are indicated in bold.

### Example 6 Modified `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;
}
```

### Example 7 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.

**Example 8 Updated TFTP directory structure**

```
/tftpboot
/tftpboot/pxelinux.0
/tftpboot/pxelinux.cfg/default
/tftpboot/EFI/
/tftpboot/EFI/bootx64.efi
/tftpboot/EFI/textmenu-message.msg
/tftpboot/RHEL6.6
/tftpboot/RHEL7.0
/tftpboot/SLES11
/tftpboot/SLES12
/tftpboot/Trusty
```

## Configuring boot loaders

You can use a few different boot loaders for UEFI PXE. ELILO was one of the first boot loaders for EFI-based systems. This document shows detailed examples for configuring ELILO and a simple example using GRUB2. Syslinux is a fairly new option for UEFI PXE and is not covered here.

### Using ELILO

ELILO is an EFI Linux boot loader that can only be used for UEFI systems.

1. Download the source files at **http://sourceforge.net/projects/elilo/files/elilo/**.
2. In a temporary directory, extract `elilo-3.16-all.tar.gz`.
3. Rename `elilo-3.16-x86_64.efi` to `bootx64.efi`. Optionally, if you want a graphical boot menu, you can extract the source files `elilo-3.16-source.tar.gz` and use `/elilo-3.16-source/examples/textmenu_chooser/textmenu-message.msg` as the menu template.
4. Copy `bootx64.efi` to the `tftpboot/EFI/` directory. Optionally, copy `textmenu-message.msg` to `/tftpboot/EFI/`.

   **NOTE:** The ELILO configuration file must be in the same directory as the binary.

**Example 9 Sample `elilo.conf` file**

```
chooser=textmenu
message=textmenu-display.msg
prompt
delay=150
f1=general.msg
f2=params.msg
#timeout=300
#default=sles11kiso
image=/RHEL7.0/vmlinuz
label=rhel7.0
   description = "RHEL 7.0 x86_64"
   initrd=/RHEL7.0/initrd.img
   append="repo=http://192.168.100.10/repo/RHEL7.0/disc1"
   image=/RHEL6.6/vmlinuz
label=rhel6.6
   description = "RHEL 6.6 x86_64"
   initrd=/RHEL6.6/initrd.img
   append="repo=http://192.168.100.10/repo/RHEL6.6/disc1"
   image=/SLES11/kISO/linux
label=sles11kiso
   description = "SLES 11 SP3 kISO Gen9"
   initrd=/SLES11/kISO/initrd
   append="install=http://192.168.100.10/repo/SLES11/disc1\
   addon=http://192.168.100.10/
   repo/kISO_SLES11_Gen9/disc1"
   image=/SLES12/linux
label=sles12
   description = "SLES 12"
   initrd=/SLES12/initrd
   append="install=http://192.168.100.10/repo/SLES12/disc1"
   image=/Trusty/linux
label=trusty
   description = "Ubuntu 14.04 amd64"
   initrd=/Trusty/initrd.gz
   append="preseed/url=http://192.168.100.10/answers/trusty.cfg"
```

## Using GRUB2

GRUB2 is a boot loader that works with both UEFI and Legacy BIOS. It is also the only boot loader that currently supports Secure Boot. You can download the GRUB2 source files from **ftp://ftp.gnu.org/gnu/grub/**.

**NOTE:** GRUB2 should not be confused with GRUB Legacy.

**Example 10 Sample `grub2.conf` file**

```
menuentry 'RHEL 7.0 AUTO' {
insmod gzio
insmod part_gpt
insmod ext2
linuxefi /k/rhel-7.0 ip=dhcp inst.ks=http://192.168.100.10/ks/rhel-7.0 devfs=nomount
initrdefi /i/rhel-7.0.img
}
menuentry 'Ubuntu 14.04.2 AUTO' {
insmod gzio
insmod part_gpt
insmod ext2
insmod iso9660
# Cf: http://ubuntuforums.org/archive/index.php/t-1495706.html
loopback loop /ubuntu-14.04.2-server-amd64.iso
linux (loop)/install/netboot/ubuntu-installer/amd64/linux ip=dhcp ks=http://192.168.100.10/ks/ubuntu-14.04
initrd (loop)/install/netboot/ubuntu-installer/amd64/initrd.gz
}
```

Using Syslinux

Syslinux version 6.x supports UEFI. For more information about using Syslinux for PXE booting, see **http://www.syslinux.org/wiki/index.php/The_Syslinux_Project**.

## Configuring a PXE server for a 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.

More information

**https://technet.microsoft.com/en-us/library/jj648426.aspx**

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

### System requirements

The following versions of Windows can boot in UEFI mode:

- Windows 7
- Windows 2008 R2
- Windows Vista (64–bit)
- Windows 8
- Windows Server 2012
- Windows Server 2012 R2

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.

More information

How to use WDS: **https://technet.microsoft.com/en-us/library/jj648426.aspx**
How to set up the Configuration Manager on a Windows server:

**https://technet.microsoft.com/en-us/library/jj938037.aspx**

## 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 of ProLiant , 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.

To configure VLANs for UEFI network boot, select one of the following methods.

- Using the global VLAN Configuration menu provided by the System Utilities Network Options
- Using the configuration menu provided by specific NIC adapters

More information

Using the RESTful Interface Tool: **http://www.hpe.com/info/restfulinterface/docs**

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

1. From the **System Utilities** screen, select **System Configuration→BIOS/Platform Configuration (RBSU)→Network Options→VLAN Configuration** and press **Enter**.
2. Complete the following and press **Enter** after each selection or data entry.
   - **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. Press **F10**.

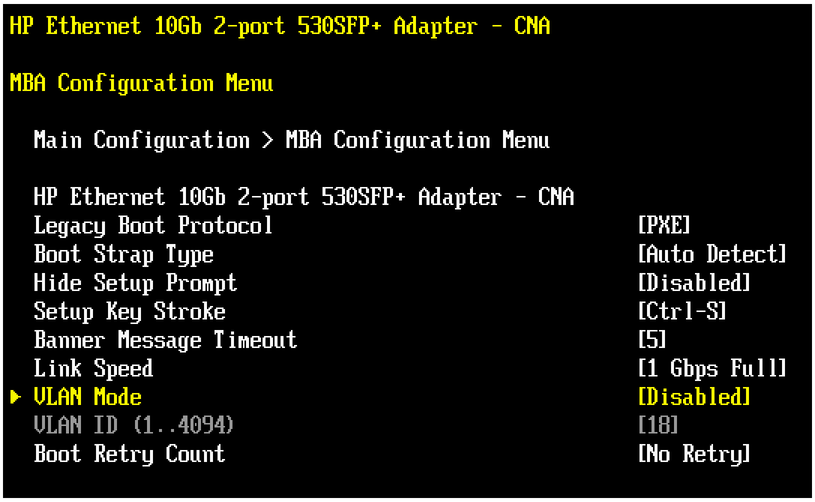
### Using the configuration menu provided by specific NIC adapters

1. From the **System Utilities** screen, select **System Configuration** and press **Enter**.

   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 and press **Enter**.

   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**) and press **Enter**.

   A menu appears similar to the one shown in the following example.



```
HP Ethernet 10Gb 2-port 530SFP+ Adapter - CNA

MBA Configuration Menu

  Main Configuration > MBA Configuration Menu

  HP Ethernet 10Gb 2-port 530SFP+ Adapter - CNA
  Legacy Boot Protocol                              [PXE]
  Boot Strap Type                                   [Auto Detect]
  Hide Setup Prompt                                 [Disabled]
  Setup Key Stroke                                  [Ctrl-S]
  Banner Message Timeout                            [5]
  Link Speed                                        [1 Gbps Full]
▸ VLAN Mode                                         [Disabled]
  VLAN ID (1..4094)                                 [18]
  Boot Retry Count                                  [No Retry]
```

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, 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.

More information

iPXE Open Source Boot Firmware website: **http://ipxe.org/**

# Configuring iPXE

## Process overview

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

## Downloading iPXE files

1. Go to the iPXE Open Source Boot Firmware website at **http://ipxe.org/download**, and select one of the following links.
   - For a pre-built binary ISO image: **http://boot.ipxe.org/ipxe.iso**.
   - For customizable iPXE source code: **http://git.ipxe.org/ipxe.git**.
2. Complete the download.

### Customizing iPXE

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

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

### Configuring HTTP

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

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.

More information

HTTP/FTP URL boot

## 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,

**Example 11 Sample `dhcpd.conf` 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**.

## To boot PXE or iPXE:

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

# 6 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.

The following procedure shows how to configure the iSCSI Software Initiator using the System Utilities. You can also configure iSCSI using the RESTful Interface Tool. See the RESTful Interface Tool documentation at: **http://www.hpe.com/info/restfulinterface/docs**.

## Configuring the iSCSI Software Initiator and booting iSCSI

### Prerequisites

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

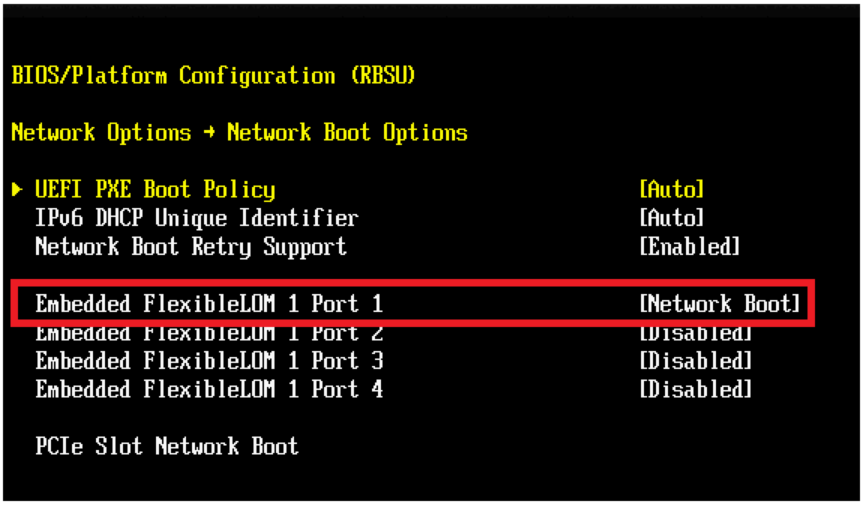
### Process overview

1. Select a device for iSCSI boot and make it network bootable.
2. Enter an iSCSI initiator name.
3. Add an iSCSI boot device.
4. Verify the iSCSI connection.
5. (If needed) Install and configure Red Hat Enterprise Linux 7.
6. Change the boot order of the iSCSI initiator.

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

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

   The **Network Boot Options** screen lists network devices and ports available for iSCSI boot as shown in the following example.



```
BIOS/Platform Configuration (RBSU)

Network Options → Network Boot Options

▶ UEFI PXE Boot Policy                                [Auto]
  IPv6 DHCP Unique Identifier                         [Auto]
  Network Boot Retry Support                          [Enabled]

  Embedded FlexibleLOM 1 Port 1                       [Network Boot]
  Embedded FlexibleLOM 1 Port 2                       [Disabled]
  Embedded FlexibleLOM 1 Port 3                       [Disabled]
  Embedded FlexibleLOM 1 Port 4                       [Disabled]

  PCIe Slot Network Boot
```

2. Select the device that is connected to the SAN network on which the iSCSI target is available and press **Enter**.

   A message appears stating that you must reboot for this boot option to appear in the boot order list.

3. Press **Enter**, select **Network Boot**, and press **Enter** again.
4. Press **F10**.

## Entering an iSCSI Initiator Name

If you want to set a new iSCSI initiator name and override the default name set for the initiator:

1. From the **System Utilities** screen, select **System Configuration→BIOS/Platform Configuration (RBSU)→Network Options→iSCSI Boot Configuration** and press **Enter**.
2. Select **iSCSI Initiator Name** and press **Enter**.
3. Enter a unique name for the iSCSI initiator using iSCSI Qualified Name (IQN) format.

   For example: `iqn.2001-04.com.example:uefi-13021088`.

## Adding an iSCSI boot device

1. From the **iSCSI Boot Configuration** screen, select **Add an iSCSI Boot Attempt** and press **Enter**.

   A message appears stating that you must reboot for this boot attempt to be in effect.

2. Press **Enter**.

   The **iSCSI Boot Network Interface Selection** lists network devices and ports available for iSCSI boot.

3. Select the NIC device for which you enabled network boot and press **Enter**.

   The **iSCSI Boot Attempt Configuration** screen appears.

4. Complete the following and press **Enter** to save each setting.

   - **iSCSI Attempt Name**—Enter a name.
   - **iSCSI Boot Control**—Select **Enabled**.

**NOTE:** You must change this setting from **Disabled** (the default setting) to **Enabled** to create a boot attempt.

- **IP Address Type**—Select **IPv4**.
- **Connection Retry Count**—Enter a value from 0 to 16. Default is 0 (no retries).
- **Connection Timeout**—Enter a value in milliseconds from 100 to 20000. Default is 1000.
- **Initiator DHCP**—Press **Enter** to enable configuring the iSCSI initiator address from DHCP.
- **Target DHCP Config**—Disable this option (clear the check box), and enter a target name, IP address, port and boot LUN. This option is enabled by default.
- **Target Boot LUN**—Enter the LUN number of the target for this boot attempt in hexadecimal format.

  **NOTE:** You must enter the LUN number in hexadecimal format

- Optional: **Authentication Type**—If Challenge-Handshake Authentication Protocol is required, select **CHAP** and complete the CHAP entries.

5. Press **PgDown** and select **Save Changes**.
6. Reboot the server.

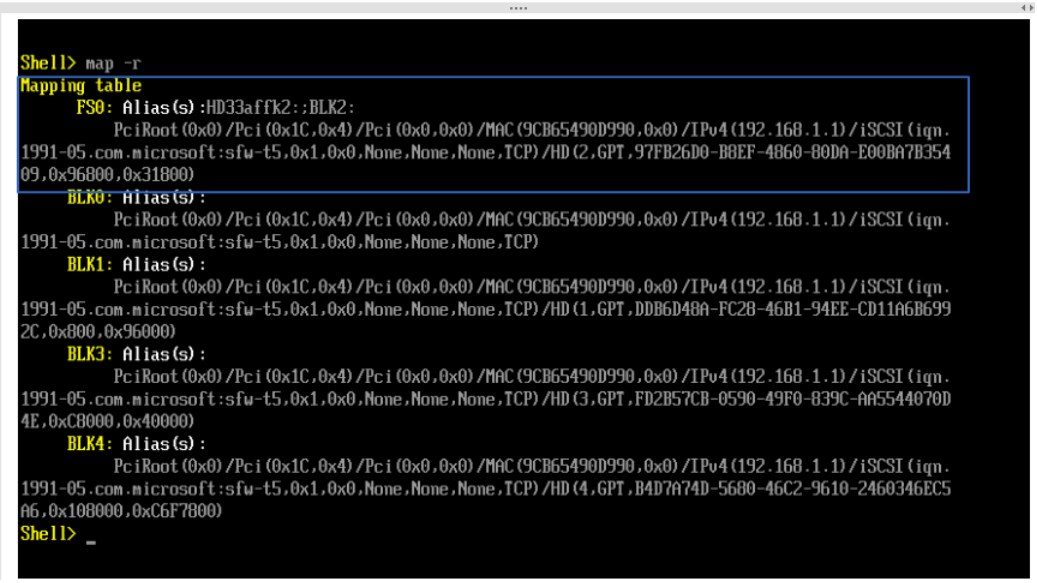
## Verifying the iSCSI connection

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

   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 device. 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** and press **Enter**.
4. At the Embedded UEFI Shell prompt, enter the following command:

   ```
   map -r
   ```

   A mapping table appears as shown in the following example.



```
Shell> map -r
Mapping table
      FS0: Alias(s):HD33affk2::BLK2:
          PciRoot(0x0)/Pci(0x1C,0x4)/Pci(0x0,0x0)/MAC(9CB65490D990,0x0)/IPv4(192.168.1.1)/iSCSI(iqn.
1991-05.com.microsoft:sfw-t5,0x1,0x0,None,None,None,TCP)/HD(2,GPT,97FB26D0-B8EF-4860-80DA-E00BA7B354
09,0x96800,0x31800)
     BLK0: Alias(s):
          PciRoot(0x0)/Pci(0x1C,0x4)/Pci(0x0,0x0)/MAC(9CB65490D990,0x0)/IPv4(192.168.1.1)/iSCSI(iqn.
1991-05.com.microsoft:sfw-t5,0x1,0x0,None,None,None,TCP)
     BLK1: Alias(s):
          PciRoot(0x0)/Pci(0x1C,0x4)/Pci(0x0,0x0)/MAC(9CB65490D990,0x0)/IPv4(192.168.1.1)/iSCSI(iqn.
1991-05.com.microsoft:sfw-t5,0x1,0x0,None,None,None,TCP)/HD(1,GPT,DDB6D48A-FC28-46B1-94EE-CD11A6B699
2C,0x800,0x96000)
     BLK3: Alias(s):
          PciRoot(0x0)/Pci(0x1C,0x4)/Pci(0x0,0x0)/MAC(9CB65490D990,0x0)/IPv4(192.168.1.1)/iSCSI(iqn.
1991-05.com.microsoft:sfw-t5,0x1,0x0,None,None,None,TCP)/HD(3,GPT,FD2B57CB-0590-49F0-839C-AA5544070D
4E,0xC8000,0x40000)
     BLK4: Alias(s):
          PciRoot(0x0)/Pci(0x1C,0x4)/Pci(0x0,0x0)/MAC(9CB65490D990,0x0)/IPv4(192.168.1.1)/iSCSI(iqn.
1991-05.com.microsoft:sfw-t5,0x1,0x0,None,None,None,TCP)/HD(4,GPT,B4D7A74D-5680-46C2-9610-2460346EC5
A6,0x108000,0xC6F7800)
Shell> _
```

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.
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.

# Installing and configuring Red Hat Enterprise Linux 7

## Installing Red Hat Enterprise Linux 7 using the kickstart menu

If necessary, you can use the UEFI kickstart menu to install RHEL.

1. Reboot the server.
2. When the POST screen appears, press **Esc** until a screen with the option to install Red Hat Enterprise Linux appears.
3. If necessary, use the up or down arrow keys to select the RHEL installation option, and then press **E** to edit the parameters.
4. Add your boot options to the `linuxefi` line as shown in the following example.

   ```
   linuxefi /images/pxeboot/vmlinuz
   inst.stage2=hd:LABEL=RHEL-7.1\x20Server.X86.64 quiet initrdefi/images/pxeboot/initrd.img
   ```

5. Press **Ctrl+X** to start the install boot process.

## Configuring Red Hat Enterprise Linux 7

If necessary, complete the following steps to ensure that the iSCSI Software Initiator works properly with RHEL 7.0 or 7.1.

1. Download and install the GRUB2 boot manager files from **ftp://ftp.gnu.org/gnu/grub/**.
2. From the GRUB2 menu, select `boot option`, and press **E**.
3. Add the iSCSI boot option `ip=ibft` to `vmlinuz inst.stage2` as shown in the following examples.
   - For RHEL 7.0:

     ```
     linuxefi /images/pxeboot/vmlinuz
     inst.stage2=hd:LABEL=RHEL-7.0\x20Server.X86.64 quiet ip=ibft
     ```

   - For RHEL 7.1:

```
linuxefi /images/pxeboot/vmlinuz
inst.stage2=hd:LABEL=RHEL-7.0\x20Server.X86.64 quiet rd.iscsi.ibft
```

4. Press **Ctrl+X** to start the install boot process.

**NOTE:** If the network connecting the host server with the iSCSI boot target loses connectivity, your original iSCSI boot option is lost and a new boot option is created. When this happens, you must restore the iSCSI boot option to the top of the Boot Order list. See Changing the UEFI Boot Order list.

## Changing the boot order of the iSCSI initiator

You can use the iLO 4 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.

# 7 HTTP/FTP URL boot

## Pre-Boot Network 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.

**IMPORTANT:** You do not need to use the Embedded UEFI Shell `ifconfig` command on a network interface if you plan to run `webclient` or `ftp` over the same interface because these interface and IP address settings are automatically selected by the **Pre-Boot Network Settings** configured in the System Utilities.

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

### System Utilities Pre-Boot Network Settings options

- **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.
  - **Embedded NIC**—The system uses the selected NIC. If the selected NIC has more than one port, the system only uses the first available port with a network connection.
- **DHCPv4**—Enables or disables obtaining the pre-boot network IPv4 configuration from a DHCP server.
  - **Enabled** (default)—Enables DHCPv4 IPv4 network address configuration, as shown in Figure 3 (page 32).

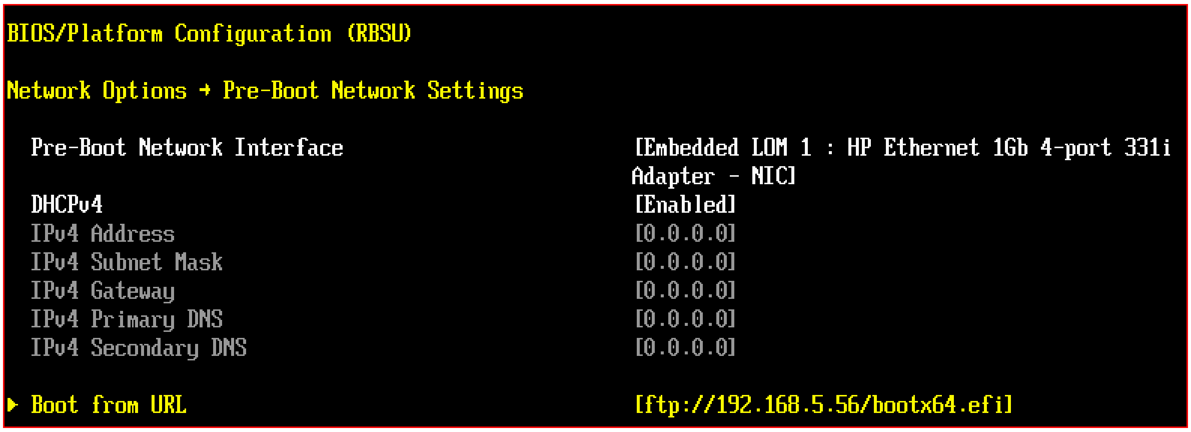
    **NOTE:** This setting makes IPv4 address, subnet mask, gateway, and DNS settings unavailable because values are supplied automatically.
  - **Disabled**—Disables DHCPv4 address configuration, requiring you to manually configure the following static IP address settings, as shown in Figure 4 (page 32).
    - **IPv4 Address**
    - **IPv4 Subnet Mask**
    - **IPv4 Gateway**
    - **IPv4 Primary DNS**
    - **IPv4 Secondary DNS**
- **Boot from URL**—Specifies a network URL to a bootable ISO or EFI file. Enter a URL in either HTTP or FTP format, using either an IPv4 server address or host name. IPv6 addresses are not supported. 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:** To use this setting:

- ○ Configure the pre-boot network settings to access the URL location.
- ○ Leave the boot mode set to UEFI Mode.

When booting from an ISO file, use a file that the system can use to boot a preliminary operating system environment, such as WinPE, mini-Linux, or VMware ESX installer. Doing so enables further installation to proceed over an OS network connection. ISO files that contain the full OS installation media are not supported.

**Figure 3 Setting the NIC IP address via DHCP**



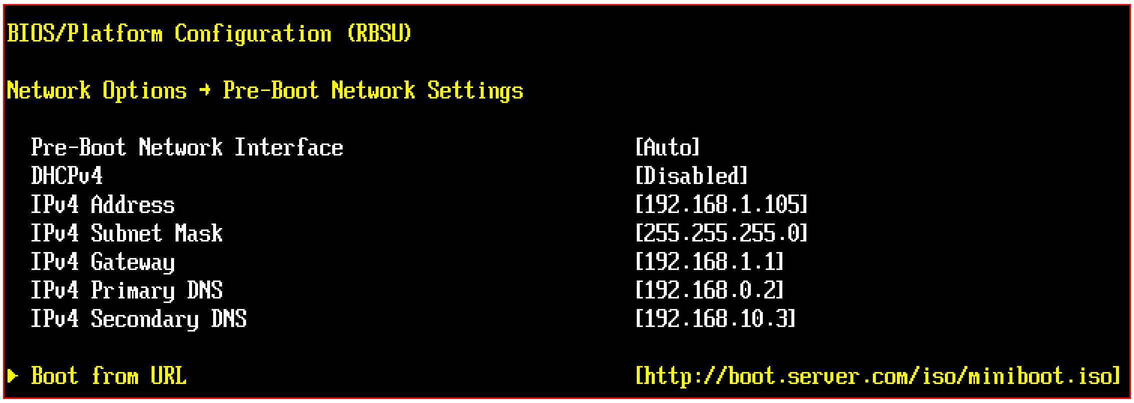
```
BIOS/Platform Configuration (RBSU)

Network Options → Pre-Boot Network Settings

  Pre-Boot Network Interface                            [Embedded LOM 1 : HP Ethernet 1Gb 4-port 331i
                                                        Adapter - NIC]
  DHCPv4                                                [Enabled]
  IPv4 Address                                          [0.0.0.0]
  IPv4 Subnet Mask                                      [0.0.0.0]
  IPv4 Gateway                                          [0.0.0.0]
  IPv4 Primary DNS                                      [0.0.0.0]
  IPv4 Secondary DNS                                    [0.0.0.0]

▶ Boot from URL                                         [ftp://192.168.5.56/bootx64.efi]
```

**Figure 4 Setting the NIC IP address statically**



```
BIOS/Platform Configuration (RBSU)

Network Options → Pre-Boot Network Settings

  Pre-Boot Network Interface                            [Auto]
  DHCPv4                                                [Disabled]
  IPv4 Address                                          [192.168.1.105]
  IPv4 Subnet Mask                                      [255.255.255.0]
  IPv4 Gateway                                          [192.168.1.1]
  IPv4 Primary DNS                                      [192.168.0.2]
  IPv4 Secondary DNS                                    [192.168.10.3]

▶ Boot from URL                                         [http://boot.server.com/iso/miniboot.iso]
```

**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

## Prerequisites

- Before configuring VLANs for HTTP/FTP URL boot in a Windows environment, complete the steps in Configuring VLANs for UEFI network boot.
- When using the **Boot from URL** setting:
  - Configure the pre-boot network settings to access the URL location.
  - Leave the boot mode set to **UEFI Mode**.
  - When booting from an ISO file, use a file that the system can use to boot a preliminary operating system environment, such as WinPE, mini-Linux, or VMware ESX installer.

## Using the System Utilities

1. From the **System Utilities** screen, select **System Configuration→BIOS/Platform Configuration (RBSU)→Network Options→Pre-Boot Network Settings** and press **Enter**.
2. Select any of the System Utilities Pre-Boot Network Settings options and press **Enter**, then select a setting or enter a value for that option and press **Enter** again.

## Using the sysconfig command in the Embedded UEFI Shell

1. From the **System Utilities** screen, select **Embedded Applications→Embedded UEFI Shell** and press **Enter**
2. At the Embedded UEFI Shell prompt, do one of the following:
   - Enable DHCPv4 to assign an IPv4 network address configuration, for example:

     ```
     Shell> sysconfig -s PreBootNetwork=EmbNic Dhcpv4=Enabled
     UrlBootfile=ftp://192.168.5.56/bootx64.efi
     ```

   - Specify a static NIC IP address configuration, for example:

     ```
     Shell> sysconfig -s PreBootNetwork=Auto Dhcpv4=Disabled
     Ipv4Address=192.168.1.105 Ipv4SubnetMask=255.255.255.0
     Ipv4Gateway=192.168.1.1 Ipv4PrimaryDNS=192.168.0.2
     Ipv4SecondaryDNS=192.168.10.3
     UrlBootfile=http://boot.server.com/iso/miniboot.iso
     ```

The system downloads the file and boots to it.

## Using the RESTful Interface Tool

1. Run the RESTful Interface Tool.
2. At the command line prompt, do one of the following:
   - Enable DHCPv4 to assign an IPv4 network address configuration, for example:

     ```
     hprest login proliantsystem1-ilo.domain.com -u username -p password
     hprest select HpBios.
     hprest set PreBootNetwork=EmbNic Dhcpv4=Enabled
     UrlBootFile=http://192.168.5.56/bootx64.efi
     hprest commit
     ```

   - Specify a static NIC IP address configuration, for example:

     ```
     hprest login proliantsystem1-ilo.domain.com -u username -p password
     hprest select HpBios.
     hprest set PreBootNetwork=-s PreBootNetwork=Auto
     Dhcpv4=Disabled Ipv4Address=192.168.1.105
     Ipv4SubnetMask=255.255.255.0 Ipv4Gateway=192.168.1.1
     Ipv4PrimaryDNS=192.168.0.2 Ipv4SecondaryDNS=192.168.10.3
     ```

```
UrlBootfile=http://boot.server.com/iso/miniboot.iso
hprest commit
```

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.
4. In the **System Utilities**, enter the **One-Time Boot Menu**. See "Selecting an option for a one-time boot" (page 10).
5. Select the **URL File** entry and press **Enter**.

# 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 for HPE ProLiant Gen9 Servers*.
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 admin privileges on 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 or FTP server is down or did not respond.

Action

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

## Cannot network boot from downloaded image file

Symptom

Booting from the image specified in the URL fails.

### Solution 1

Cause

The image is not signed and **Secure Boot** is enabled.

Action

1. Ensure that the image is signed and that its Secure Boot settings are correct. See “Secure Boot” in the *UEFI System Utilities User Guide for HPE ProLiant* .
2. Try to download the file in the URL again.

### Solution 2

Cause

The downloaded file is corrupt.

Action

1. Select a new file.
2. Repeat the URL configuration, specifying the new file.
3. Try to download the new file in the URL.

# 8 Embedded UEFI Shell boot

The system BIOS in all ProLiant 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 for HPE ProLiant Gen9 Servers*.

| **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. |

## 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. For more information, see “UEFI Shell Script Auto-start“ in the *UEFI System Utilities User Guide for HPE ProLiant Gen9 Servers*.

## Manually invoking a Shell script

1. Navigate to the location of the `.nsh` script file.
2. Double-click the file, or right-click it, and then select **Open**.

## 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

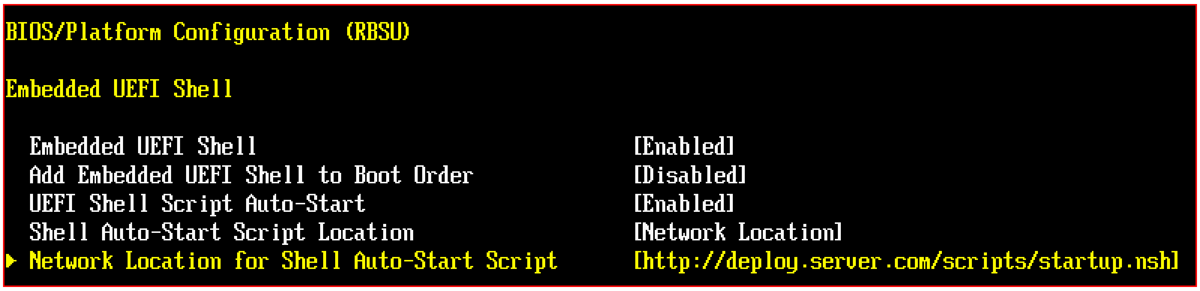
### Using the System Utilities

1. Complete the pre-boot network configuration as described for HTTP boot (Using the System Utilities), 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.
2. Enable UEFI Shell Script Auto Start. By default this is disabled. See “UEFI Shell Script Auto-start” in the *UEFI System Utilities guide for ProLiant Gen9 Servers*.
3. Set the auto-start location to Network. See “Shell Auto-start Script Location” in the *UEFI System Utilities Guide for HPE ProLiant Gen9 Servers*.

4. Set the location (URL) where the Embedded UEFI Shell downloads the script. See Figure 5 (page 38) and “Network Location for Shell Auto-start Script” in the *UEFI System Utilities guide for ProLiant Gen9 Servers*.

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

**Figure 5 Configuring the Embedded UEFI Shell startup script to run from a network location**



```
BIOS/Platform Configuration (RBSU)

Embedded UEFI Shell

  Embedded UEFI Shell                                 [Enabled]
  Add Embedded UEFI Shell to Boot Order               [Disabled]
  UEFI Shell Script Auto-Start                        [Enabled]
  Shell Auto-Start Script Location                    [Network Location]
▶ Network Location for Shell Auto-Start Script        [http://deploy.server.com/scripts/startup.nsh]
```

## Using the sysconfig command in the Embedded UEFI Shell

1. From the **System Utilities** screen, select **Embedded Applications→Embedded UEFI Shell** and press **Enter**.
2. At the Shell prompt, enter sysconfig parameters as shown in the following example.

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

3. Reboot the server.

## Using the RESTful Interface Tool

1. Run the RESTful Interface Tool.
2. At the command line prompt, do one of the following:
   - Enable DHCPv4 to assign an IPv4 network address configuration, for example:

     ```
     hprest login proliantsystem1-ilo.domain.com -u username -p password
     hprest set PreBootNetwork=Auto Dhcpv4=Enabled
     UefiShellStartup=Enabled UefiShellStartupLocation=NetworkLocation
     UefiShellStartupUrl=http://deploy.server.com/scripts/startup.nsh
      --selectorHpBios.
     hprest commit
     ```

   - Specify a static NIC IP address configuration, for example:

     ```
     hprest login proliantsystem1-ilo.domain.com -u username -p password
     hprest set PreBootNetwork=Auto Dhcpv4=Disabled
     Ipv4Address=192.168.1.105 Ipv4SubnetMask=255.255.255.0
     Ipv4Gateway=192.168.1.1 Ipv4PrimaryDNS=192.168.0.2
     Ipv4SecondaryDNS=192.168.10.3
     UefiShellStartup=Enabled UefiShellStartupLocation=NetworkLocation
     UefiShellStartupUrl=http://deploy.server.com/scripts/startup.nsh
      --selectorHpBios.
     hprest commit
     ```

3. Reboot the server.
4. Move the URL entry to the top of the boot order list as described in “Boot settings” (page 7).
5. Reboot the server.

# 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
```

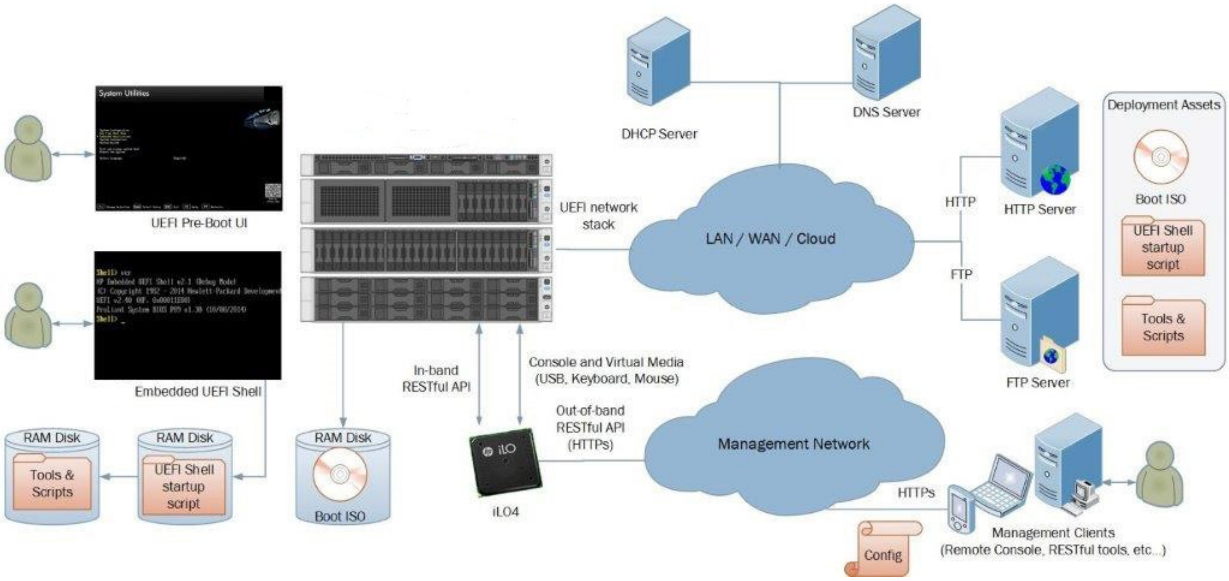
For more details, enter `help ramdisk, or ramdisk -?`

For more information about this command, see the *UEFI Shell User Guide for HPE ProLiant Gen9 Servers*.

## Deployment and scripting

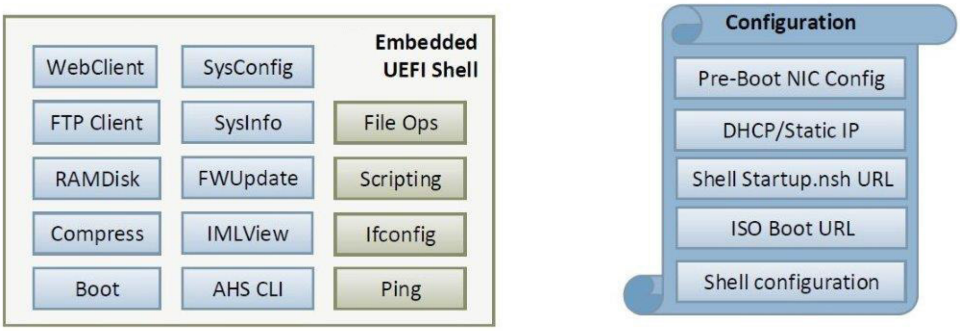
The following figure shows the Embedded UEFI Shell boot method.

**Figure 6 Deployment architecture**



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.

**Figure 7 Network stack**



The following sections detail how to configure the Embedded UEFI Shell to pick up and execute its startup script from the network.

## Configuring Embedded UEFI Shell boot: A sample deployment solution

You can use the System Utilities, Embedded UEFI Shell commands, or RESTful Interface Tool commands to configure or, where appropriate, leave the default values for booting to the Shell.

The following sections show how to use the System Utilities and the RESTful Interface Tool to complete the required configuration tasks.

## Configuring required settings using the System Utilities

1. Configure pre-boot network settings.
    - For the **Pre-Boot Network Interface**, select one of the following:
        - **Auto** (the default setting)
        - An available **Embedded NIC**
    - For **DHCPv4**, do one of the following:
        - Select **Enabled** (the default setting).
        - Select **Disabled** and manually configure the following static IP address settings, as shown in Figure 4.
            - **IPv4 Address**
            - **IPv4 Subnet Mask**
            - **IPv4 Gateway**
            - **IPv4 Primary DNS**
            - **IPv4 Secondary DNS**
2. Configure the Embedded UEFI Shell settings.
    - For **Embedded UEFI Shell**, select **Enabled** (the default setting).
    - For **UEFI Shell Script Auto-Start**, select **Enabled** (not the default setting).
    - For **Add Embedded UEFI Shell to Boot Order**, select **Enabled** (not the default setting).

      The Embedded UEFI Shell is added to the **UEFI Boot Order** list and the **One-Time Boot Menu**.
    - For **Shell Auto-Start Script Location**, select **Auto** (the default setting), **Network Location**, or **File Systems on Attached Media**.
3. Do one of the following:
    - Select the Embedded UEFI Shell from the **One-Time Boot** menu. See Selecting an option for a one-time boot.
    - 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.

## Configuring required settings using the RESTful Interface Tool

You can use the RESTful Interface Tool to complete the configuration described previously in Configuring required settings using the System Utilities. The following are sample RESTful Interface Tool commands:

```
#login to the target iLO and setup the session.
a) hprest login https://clientilo.domain.com -u username -p password

#Select the NIC. "Auto" is the default setting, uses the first available NIC port with cable connected.
b) hprest set PreBootNetwork=Auto --selector HpBios.
[OR]
#use the Network adapter in PCI slot 2's first port that has a cable connected.
   hprest set PreBootNetwork=PciSlot2 --selector HpBios.

#Set DHCP or static IP. DHCP is the default setting
c) hprest set Dhcpv4=Enabled
[OR]
#disable DHCP and set up manual IP address and DNS settings.
   hprest set Dhcpv4=Disabled Ipv4Address=192.168.1.121 Ipv4SubnetMask=255.255.255.0 Ipv4Gateway=192.168.1.1
#DNS is optional if you refer to the remote servers directly by their IP addresses.
```

```
   hprest set Ipv4PrimaryDNS=192.168.1.1 Ipv4SecondaryDNS=192.168.1.189

#ask the embedded Shell to run a startup script upon launch.
d) hprest set UefiShellStartup=Enabled

#Set the startup script location. "Auto" is the default - first network then local disk.
e) hprest set UefiShellStartupLocation=Auto
[OR]
#download and execute script from network location only
   hprest set UefiShellStartupLocation=NetworkLocation

#URL for the startup script. Supports host names or IP addresses of remote servers. Ensure DNS is configured
if using host names.
f) hprest set UefiShellStartupUrl=http://server.domain.com/deploy/startup.nsh

#Set one time boot to Embedded UEFI shell.
g) hprest set Boot/BootSourceOverrideEnabled=Once Boot/BootSourceOverrideTarget=UefiShell --selector
ComputerSystem.

#commit all changes and power on the system.
h) hprest commit --reboot=ON
```

## Sample start-up script

After the steps in Configuring Embedded UEFI Shell boot: A sample deployment solution are complete, and the target system is powered on for the first time, the system boots to a certain stage, configures itself with the requested settings, and reboots. The system then launches the Embedded UEFI Shell, which downloads and runs the startup script from the configured URL. The following is a sample configuration script that the Embedded UEFI Shell can run.

```
@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 http://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.
   - 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.
   - 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.

# Troubleshooting

## 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.
   - The Embedded UEFI Shell interface is added to the **UEFI Boot Order** list or **One-Time Boot Menu**.
   - 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.
   - UEFI Shell Script Auto-Start is enabled.
   - 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 either inside the `fs`*`X`*`:\` or the `fs`*`X`*`:\efi\boot\` directory.
   - The `.nsh` script only contains supported commands.
   - Your system has enough RAM memory to create RAM disks during automated script execution.
   - 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.

# 9 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

### 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**.
- The **Internal SD Card Slot** option in the System Utilities is enabled. This is the default setting. For more information, see “Internal SD Card Slot” in the *UEFI System Utilities User Guide for HPE ProLiant Gen9 Servers*.

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

1. If necessary, do one of the following to change your boot policy setting.
   - 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 Gen9 Servers*.
   - 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 Gen9 Servers*.
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.

# 10 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.

### To enable Embedded SATA support:

1. From the **System Utilities** screen, select **System Configuration**→**BIOS/Platform Configuration (RBSU)**→**System Options**→**SATA Controller Options**→**Embedded SATA Configuration** and press **Enter**.
2. Ensure that you are using the correct AHCI or RAID system drivers for your SATA option.
3. Select a setting and press **Enter**:
   - **Enable SATA AHCI Support**—Enables the embedded chipset SATA controller for AHCI.
   - **Enable HP Dynamic Smart Array RAID Support**—Enables the embedded chipset SATA controller for Dynamic Smart Array RAID.
4. Press **F10**.
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.

# 11 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/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, go to either of the following:
  - Hewlett Packard Enterprise Support Center **Get connected with updates** page:

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

  - Software Depot website:

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

- 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:

  **http://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.

## Related information

The latest documentation for the UEFI System Utilities and Embedded Shell is available at: **http://www.hpe.com/info/ProLiantUEFI/docs**. Available documents include:

- *UEFI Shell User Guide for HPE ProLiant Gen9 Servers*
- *UEFI System Utilities User Guide for HPE ProLiant Gen9 Servers*

- *UEFI Shell Quick Reference Card for HPE ProLiant Gen9 Servers*
- *UEFI System Utilities and Shell Release Notes for HPE ProLiant Gen9 Servers*

The *UEFI System Utilities and Shell Command Mobile Help for HPE ProLiant Gen9 Servers* is available by scanning the QR code located at the bottom of the System Utilities screen, or at **http://www.hpe.com/qref/ProLiantUEFI/Help**.

## Websites

| Website | Link |
|---|---|
| UEFI Specification | **http://www.uefi.org/specifications** |
| UEFI Learning Resources | **http://www.uefi.org/learning_center** |
| UEFI EDK2 project on SourceForge (download specifications and code) | **http://sourceforge.net/apps/mediawiki/tianocore** |
| Hewlett Packard Enterprise Information Library | **http://www.hpe.com/info/enterprise/docs** |
| Hewlett Packard Enterprise Support Center | **http://www.hpe.com/support/hpesc** |
| 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** |

## 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.

For more information and device support details, go to the following website:

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

## RESTful API

The RESTful API is the management interface that Hewlett Packard Enterprise’s server management tools use to perform configuration, inventory, and monitoring of a server. It provides a programmable interface and a lightweight data model specification that is simple, remote, secure, and extensible.

The RESTful API can be accessed via the following mechanisms:

- Install freely available REST plugins for browsers, like the Advanced REST plugin for Google Chrome browser
- Install freely available command line tools, such as CURL
- Install the REST Tool (available both for Windows and Linux) from: **www.hpe.com/info/redfish**

After the tool is installed, the RESTful API service on the target ProLiant system can be accessed over the system's iLO management network. For more details visit: **http://www.hpe.com/info/restfulinterface/docs**.

The RESTful Tool will eventually replace CONREP (Configuration Replication utility), a utility that operates with the BIOS/Platform Configuration (RBSU) to replicate hardware configuration on ProLiant servers. For more information, see the *Scripting Toolkit User Guide* for your OS environment on the Hewlett Packard Enterprise website (**http://www.hpe.com/info/stk/docs**).

## 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.

# Glossary

| | |
|---|---|
| **AHCI** | Advanced Host Controller Interface |
| **API** | Application Program Interface |
| **BIOS** | Basic Input/Output System |
| **Boot loader** | A program that loads an operating system on a server or computer when the system is on. |
| **CHAP** | Challenge-Handshake Authentication Protocol |
| **CONREP** | Configuration Replication, a utility that generates a system configuration XML file used to duplicate the hardware configuration of one ProLiant server onto another |
| **DHCP** | Dynamic Host Configuration Protocol |
| **DNS** | Domain Name System |
| **EFI** | Generic term that refers to one of the versions of the EFI specification: EFI 1.02, EFI 1.10, or UEFI 2.0 |
| **FAT** | File Allocation Table |
| **FTP** | File Transfer Protocol |
| **FW** | Firmware |
| **GRUB** | Grand Unified Boot loader |
| **iLO** | Integrated Lights-Out |
| **iSCSI** | Internet Small Computer System Interface |
| **ISO** | International Organization for Standardization |
| **LUN** | Logical Unit Number |
| **MBA** | Multi-Boot Agent |
| **MBR** | Master Boot Record |
| **MDT** | Microsoft Deployment Toolkit—A set of tools, processes, and guidance for automating desktop and server deployment |
| **Multicast** | Group communication where information is addressed to group destination systems at the same time |
| **PXE** | Preboot Execution Environment—A specification that defines how a client-server environment boots from network software on a client. It requires standard network protocols, such as DHCP and TFTP. |
| **RBSU** | ROM-Based Setup Utility |
| **REST** | Representational State Transfer |
| **SAS** | Serial Attached SCSI |
| **SATA** | Serial Advanced Technology Attachment |
| **SD** | Secure Digital |
| **Secure Boot** | A process added to new versions of UEFI that can improve the security of a system deployment by offering hardware-verified and malware free operating systems |
| **SFO** | Standard Format Output |
| **TFTP** | Trivial File Transfer Protocol—A file transfer protocol that lets clients to get from or copy to a remote host |
| **UDP** | User Datagram Protocol |
| **UEFI** | Unified Extensible Firmware Interface—The interface between the operating system and platform firmware during the boot process |
| **URI** | Uniform Resource Identifier |
| **USB** | Universal Serial Bus |
| **WDS** | Windows Deployment Services—A server technology made by Microsoft for network-based installation of Windows operating systems |

# Index