Hewlett Packard Enterprise

# Scripting Tools for Windows PowerShell User Guide

## BIOS cmdlets v1.1

### Abstract

This document contains instructions for using Scripting Tools for Windows PowerShell to manage the BIOS on HPE ProLiant servers. It is intended for system administrators who use the Scripting Tools for Windows PowerShell to manage their IT infrastructure. The BIOS cmdlets support both Legacy and UEFI (Unified Extensible Firmware Interface) boot modes.

Part Number: 823063-003
Published: September 2016
Edition: 1

**Acknowledgments**

# Contents

# 1 Introduction to Scripting Tools for Windows PowerShell

The Scripting Tools for Windows PowerShell provides a simplified and consistent infrastructure management experience. These sets of PowerShell utilities provide comprehensive Hewlett Packard Enterprise integration tools. These tools are designed for IT experts with experience in PowerShell scripting and configuring HPE ProLiant server hardware.

The Scripting Tools for Windows PowerShell includes sets of PowerShell cmdlets for configuring Hewlett Packard Enterprise hardware using familiar PowerShell syntax. Documentation describing how to apply these new tools to configure Hewlett Packard Enterprise hardware is also included.

This guide is intended for system administrators who use the Scripting Tools for Windows PowerShell to manage their IT infrastructure. Users should be familiar with Windows PowerShell and the system ROM of ProLiant servers. For more information about configuring the system ROM options, see the guide applicable to your ProLiant server:

- For Gen8: *HP ROM-Based Setup Utility User Guide* and other related RBSU documents on the RBSU information library (**http://www.hpe.com/info/rbsu/docs**)
- For Gen9: *HPE UEFI System Utilities User Guide for HPE ProLiant Gen9 Servers* and other related UEFI documents on the UEFI System Utilities information library (**http://www.hpe.com/info/ProLiantUEFI/docs**)

## Windows PowerShell

Windows PowerShell is Microsoft's task automation framework, consisting of a command-line shell and associated scripting language built on a .NET Framework. As businesses face the need to configure large numbers of servers in a quick and reliable fashion, the Scripting Tools for Windows PowerShell is a powerful set of utilities that can be used to perform various configuration tasks on Hewlett Packard Enterprise hardware. These cmdlets follow the standard PowerShell syntax and scripting model, making it easy for you to incorporate these functions into your administrative scripts.

## Features

Scripting Tools for Windows PowerShell provides the following features:

- Uses proven PowerShell technology to provide consistent and reliable server configuration.
- Supports the standard PowerShell architecture and scripting model.
- Object oriented—output from one command can be piped to another command.
- Built-in help for all HPE PowerShell cmdlets, documenting syntax and usage examples.

# 2 Installation

## System prerequisites

Windows Management Framework 3.0 or later (which includes PowerShell 3.0 or later) must be installed on your system before installing the Scripting Tools for Windows PowerShell. Microsoft .NET Framework 4.5 or later must be installed before installing Windows Management Framework 3.0 or later. The Windows Assessment and Deployment Kit (Windows ADK) version 5.0 or later is needed when using BIOS in Microsoft WinPE environment.

The following links provide access to the Microsoft download sites for these applications. Make sure you read and understand the system requirements and other information provided.

- **Microsoft .NET Framework 4.5**
- **Windows Management Framework 3.0**
- **Windows Management Framework 4.0**
- **Windows Management Framework 5.0**
- **Windows Assessment and Deployment Kit** (Applies only to Gen8 servers for creating a customized WinPE image.)

## Supported operating systems

The BIOS cmdlets are supported on the following operating system versions:

- Microsoft Windows 7 SP1
- Microsoft Windows 8
- Microsoft Windows 8.1
- Microsoft Windows 10
- Microsoft Windows Server 2008 R2 SP1
- Microsoft Windows Server 2012
- Microsoft Windows Server 2012 R2

## Installing the BIOS cmdlets

The BIOS cmdlets can be downloaded from the following website:

**http://www.hpe.com/servers/powershell**

Use the following guidelines when installing the BIOS cmdlets:

- Close any PowerShell windows before the installation and open new ones after the installation is complete.
- Run the installer from an account with administrative privileges using any standard method of execution (command line or double click). If you run from an account without administrative privileges, use one of the following options:

### Procedure 1

1. Click **Start** and select **All Programs→Accessories→Windows PowerShell**.
2. Right-click one of the PowerShell options and select **Run as administrator**.
3. Change (CD) to the directory where you unzipped the installer.
4. On the PowerShell command line run either the 64-bit (`HPBIOSCmdlets-x64.msi`) or the 32-bit (`HPBIOSCmdlets-x86.msi`) installer as appropriate for your system.

### Procedure 2

1. Click **Start** and select **Run...**
2. In the Run dialog enter the path and filename of the correct installer for your system, either the 64-bit (`HPBIOSCmdlets-x64.msi`) or the 32-bit (`HPBIOSCmdlets-x86.msi`).

- It might be necessary to change the execution policy for PowerShell. Use the following help command to get more information to help you to decide what to select:

  ```
  help about_Execution_Policies
  ```

  Use the following command to see your current execution policy settings:

  ```
  Get-ExecutionPolicy -list
  ```

  You can use the following PowerShell command until you determine if it meets your needs:

  ```
  Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned
  ```

- Upgrading from a previous release is supported.
- The installation will halt and not complete successfully if any of the following conditions are detected:
  - Attempting to install the X86 package on a 64-bit operating system
  - Attempting to install without .NET 4.5 or above
  - Attempting to install without PowerShell 3.0 or above

## Uninstalling the BIOS cmdlets

To uninstall the BIOS cmdlets:

1. Open Windows Control Panel.
2. Select **Programs and Features**.
3. Select **Hewlett Packard Enterprise BIOS Cmdlets**.
4. Click **Uninstall**.

# 3 Scripting Tools for Windows PowerShell cmdlets

Table 1 provides a list and brief description of all the BIOS cmdlets.

## Cmdlet help

Help is available with the BIOS cmdlets and is supported in the same way as other PowerShell cmdlets. To display a complete list of the BIOS cmdlets in PowerShell, type:

```
help *hpbios*
```

**NOTE:** You can also use the following command to display the BIOS cmdlets:

```
Get-Command -Module HPBIOSCmdlets
```

To display complete help for a specific cmdlet, type:

```
help <cmdlet> -Full
```

where `<cmdlet>` is the name of the BIOS cmdlet.

The BIOS cmdlets support the PowerShell `Update-Help` feature. When you execute this command, it accesses a Hewlett Packard Enterprise website, gets the most current help file(s), and puts them in the correct location on your system.

**Table 1 BIOS cmdlets**

| Cmdlet | Description |
|---|---|
| Clear-HPBIOSUserDefault | Clears the user default configuration. |
| Connect-HPBIOS | Creates connections to one or multiple BIOS targets. |
| Disable-HPBIOSEmbeddedLOMPort | Disables the network boot for the installed network interface card (NIC). |
| Disable-HPBIOSPCIDeviceOption | Sets the status of the embedded and add-in BIOS PCI devices for the target server. |
| Disable-HPBIOSPCIeSlotNetworkBootOption | Disables the UEFI PXE boot status for the installed NIC in PCIe slots. |
| Disconnect-HPBIOS | Closes the connection. |
| Disconnect-HPBIOSAllConnection | Closes all the active BIOS connections in the current PowerShell session. |
| Enable-HPBIOSEmbeddedLOMPort | Enables the network boot for the installed NIC. |
| Enable-HPBIOSPCIDeviceOption | Enables the embedded and add-in BIOS PCI devices for the target server. |
| Enable-HPBIOSPCIeSlotNetworkBootOption | Enables the UEFI PXE boot status for the installed NIC in PCIe slots on the target server. |
| Get-HPBIOSACPI_SLIT | Gets BIOS ACPI SLIT preferences information. |
| Get-HPBIOSAdminInfo | Gets the reference information for the server administrator. |
| Get-HPBIOSAdvancedMemoryProtection | Gets BIOS advanced memory protection options information. |
| Get-HPBIOSAdvancedPerformanceTuningOption | Obtains the BIOS advanced performance tuning options. |
| Get-HPBIOSAdvancedSystemROMOption | Gets BIOS advanced system ROM options. |
| Get-HPBIOSBootControllerOrder | Gets the current Boot Controller Order. |
| Get-HPBIOSBootMode | Gets the current Boot Mode for the systems that support UEFI. |
| Get-HPBIOSBootOrderPolicy | Gets the current Boot Order Policy in UEFI systems. |

**Table 1 BIOS cmdlets** *(continued)*

| Cmdlet | Description |
|---|---|
| Get-HPBIOSBootTimeMemoryOptimization | Gets the BIOS boot time optimization settings. |
| Get-HPBIOSCustomPostMessage | Gets BIOS custom post message. |
| Get-HPBIOSDataDirectIO | Gets BIOS Data Direct I/O information. |
| Get-HPBIOSDateTimeOption | Obtains the BIOS daylight-saving time, time format, and time zone settings. |
| Get-HPBIOSEmbeddedDiagnostics | Obtains the embedded diagnostics settings for the BIOS. |
| Get-HPBIOSEmbeddedLOMPort | Obtains the network boot status for the installed NIC. |
| Get-HPBIOSEmbeddedUEFIShell | Gets BIOS Embedded UEFI Shell information. |
| Get-HPBIOSEmbeddedUserPartition | Gets the Embedded User Partition settings. |
| Get-HPBIOSEMSConsole | Gets the EMS console configuration. |
| Get-HPBIOSFanOption | Gets the BIOS system fan installation and policy configurations. |
| Get-HPBIOSHyperTransport | Gets the current HyperTransport Frequency. |
| Get-HPBIOSIntelNICDMAChannel | Gets BIOS Intel DMA Channels information. |
| Get-HPBIOSIntelTurboBoost | Gets BIOS Intel Turbo Boost information. |
| Get-HPBIOSInterfaceMode | Gets the interface mode displayed for ROM-based utilities. |
| Get-HPBIOSInternalSDCardSlot | Obtains the BIOS internal Secure Digital (SD) card slot configuration. |
| Get-HPBIOSIPLOrder | Gets the current Standard Boot Order (IPL) configuration. |
| Get-HPBIOSMemoryChannel | Gets BIOS memory channel mode information. |
| Get-HPBIOSMemoryPower | Gets the current settings of memory related power management. |
| Get-HPBIOSMemoryProximityReportingForIO | Gets BIOS Memory Proximity Reporting for I/O information. |
| Get-HPBIOSModuleVersion | Gets the module details for the BIOS cmdlets. |
| Get-HPBIOSNetworkBootOption | Obtains the BIOS network boot and pre-boot network options. |
| Get-HPBIOSNodeInterleaving | Gets BIOS Node Interleaving information. |
| Get-HPBIOSNUMLOCK | Obtains the BIOS NUMLOCK power-on state. |
| Get-HPBIOSOneTerabyteMemoryLimit | Gets BIOS One Terabyte Memory Limit information. |
| Get-HPBIOSPCIDeviceOption | Gets the status of the embedded and add-in PCI devices. |
| Get-HPBIOSPCIePower | Gets the current PCIe related configuration, which may impact system power usage. |
| Get-HPBIOSPCIeSlotNetworkBootOption | Obtains the UEFI PXE boot status for NIC installed in PCIe slots. |
| Get-HPBIOSPowerCapping | Gets the current status of Memory Power Capping and Dynamic Power Capping Functionality. |
| Get-HPBIOSPowerProfile | Gets the level of power versus performance for the system. |
| Get-HPBIOSPowerRegulator | Gets the current HPE Power Regulator. |
| Get-HPBIOSPrefetcher | Gets BIOS HW prefetcher, Adjacent Sector prefetch, DCU prefetcher and DCUIP prefetcher information. |
| Get-HPBIOSProcessorOption | Gets the current settings of processor options. |

**Table 1 BIOS cmdlets** *(continued)*

| Cmdlet | Description |
|---|---|
| Get-HPBIOSProcessorPower | Gets the current settings of processor related power management. |
| Get-HPBIOSQPI | Gets BIOS QPI snoop configuration, QPI bandwidth optimization information. |
| Get-HPBIOSQPILinkPower | Gets the current Intel QPI Link Power Management and Frequency. |
| Get-HPBIOSRedundantPowerSupplyMode | Gets the current redundant power supply configuration. |
| Get-HPBIOSRemovableFlashMediaBootSequence | Obtains the BIOS removable flash media boot sequence. |
| Get-HPBIOSSATAControllerOption | Obtains the BIOS SATA controller configuration. |
| Get-HPBIOSSecureBootState | Obtains the Secure Boot option settings. |
| Get-HPBIOSSerialConsole | Gets the serial console configuration. |
| Get-HPBIOSSerialPort | Gets the serial port configuration. |
| Get-HPBIOSServerAvailability | Gets BIOS server availability information. |
| Get-HPBIOSServerInfo | Gets the reference information for the server administrator. |
| Get-HPBIOSServerSecurity | Gets BIOS server security information. |
| Get-HPBIOSServiceContact | Gets the reference information for the server service. |
| Get-HPBIOSStorageOption | Gets the available storage options of the target server. |
| Get-HPBIOSSystemInfo | Obtains the BIOS system information. |
| Get-HPBIOSThermalOption | Obtains the fan cooling solution for the system. |
| Get-HPBIOSTPMConfiguration | Obtains the TPM (Trusted Platform Module) configurations. |
| Get-HPBIOSUEFIBootOrder | Gets the current advanced UEFI boot order list. |
| Get-HPBIOSUEFIDevicePriority | Obtains the BIOS UEFI device priority. |
| Get-HPBIOSUEFIOptimizedBoot | Gets the current UEFI Optimized Boot configuration. |
| Get-HPBIOSUSBOption | Obtains BIOS USB options. |
| Get-HPBIOSUtilityLanguage | Gets BIOS system utility language. |
| Get-HPBIOSVideoOption | Gets BIOS video options information. |
| Get-HPBIOSVirtualization | Gets the hardware virtualization configuration. |
| Get-HPBIOSVLANConfiguration | Obtains the details of global VLAN configuration for all enabled network interfaces. |
| New-HPBIOSCustomWinPEImage | Creates HPE WinPE image. |
| Reset-HPBIOSAdminPassword | Resets the BIOS Administrator Password |
| Reset-HPBIOSDefaultManufacturingSetting | Resets all BIOS configuration settings to default manufacturing values. |
| Reset-HPBIOSPowerOnPassword | Resets the Power On Password. |
| Reset-HPBIOSUserDefault | Resets all BIOS configuration settings to saved default user values. |
| Set-HPBIOSACPI_SLIT | Sets BIOS ACPI SLIT preferences. |
| Set-HPBIOSAdminInfo | Sets the reference information for the server administrator. |
| Set-HPBIOSAdminPassword | Sets the BIOS Administrator Password. |

**Table 1 BIOS cmdlets** *(continued)*

| Cmdlet | Description |
|---|---|
| Set-HPBIOSAdvancedMemoryProtection | Sets the BIOS advanced memory protection method. |
| Set-HPBIOSAdvancedPerformanceTuningOption | Sets the BIOS advanced performance tuning options. |
| Set-HPBIOSAdvancedSystemROMOption | Sets BIOS system ROM options. |
| Set-HPBIOSBootControllerOrder | Sets the Boot Controller Order. |
| Set-HPBIOSBootMode | Sets the Boot Mode for the systems that support UEFI. |
| Set-HPBIOSBootOrderPolicy | Sets the Boot Order Policy in UEFI systems. |
| Set-HPBIOSBootTimeMemoryOptimization | Sets the boot time optimization options. |
| Set-HPBIOSCustomPostMessage | Sets a custom post message. |
| Set-HPBIOSDataDirectIO | Sets BIOS Data Direct I/O. |
| Set-HPBIOSDateTimeOption | Sets the BIOS daylight-saving time, time format, and time zone settings. |
| Set-HPBIOSEmbeddedDiagnostics | Sets the BIOS Embedded Diagnostic settings. |
| Set-HPBIOSEmbeddedUEFIShell | Sets BIOS embedded UEFI shell. |
| Set-HPBIOSEmbeddedUserPartition | Sets the Embedded User Partition information. |
| Set-HPBIOSEMSConsole | Sets the EMS console configuration. |
| Set-HPBIOSFanOption | Sets the BIOS system fan installation and policy configurations. |
| Set-HPBIOSHyperTransport | Sets the HyperTransport Frequency. |
| Set-HPBIOSIntelNICDMAChannel | Sets BIOS Intel DMA Channels. |
| Set-HPBIOSIntelTurboBoost | Sets Intel Turbo Boost. |
| Set-HPBIOSInterfaceMode | Sets the interface mode displayed for ROM-based utilities. |
| Set-HPBIOSInternalSDCardSlot | Sets the BIOS internal SD card slot configuration. |
| Set-HPBIOSIPLOrder | Sets the Standard Boot Order (IPL). |
| Set-HPBIOSMemoryChannel | Sets BIOS memory channel mode. |
| Set-HPBIOSMemoryPower | Sets the memory related power management. |
| Set-HPBIOSMemoryProximityReportingForIO | Sets BIOS Memory Proximity Reporting for I/O information. |
| Set-HPBIOSNetworkBootOption | Sets BIOS network boot and pre-boot network options. |
| Set-HPBIOSNodeInterleaving | Sets BIOS Node Interleaving. |
| Set-HPBIOSNUMLOCK | Sets the BIOS NUMLOCK power-on state. |
| Set-HPBIOSOneTerabyteMemoryLimit | Configures the BIOS One Terabyte Memory Limit. |
| Set-HPBIOSPCIePower | Sets the current PCIe related configuration, which may impact system power usage. |
| Set-HPBIOSPowerCapping | Sets the status of Memory Power Capping and Dynamic Power Capping Functionality. |
| Set-HPBIOSPowerOnPassword | Sets the BIOS Power On Password. |
| Set-HPBIOSPowerProfile | Sets the level of power versus performance for the system. |
| Set-HPBIOSPowerRegulator | Sets the Power Regulator. |

**Table 1 BIOS cmdlets** *(continued)*

| Cmdlet | Description |
|---|---|
| Set-HPBIOSPrefetcher | Sets BIOS HW, Adjacent Sector, DCU and DCUIP prefetcher. |
| Set-HPBIOSProcessorOption | Sets the current processor options settings. |
| Set-HPBIOSProcessorPower | Sets the processor related power management. |
| Set-HPBIOSQPI | Sets BIOS QPI snoop configuration and QPI bandwidth optimization. |
| Set-HPBIOSQPILinkPower | Sets the Intel QPI Link Power Management and Frequency. |
| Set-HPBIOSRedundantPowerSupplyMode | Sets the Redundant Power Supply Mode. |
| Set-HPBIOSRemovableFlashMediaBootSequence | Sets the BIOS removable flash media boot sequence. |
| Set-HPBIOSSATAControllerOption | Sets the BIOS SATA controller configuration. |
| Set-HPBIOSSecureBootState | Sets the Secure Boot option settings. |
| Set-HPBIOSSerialConsole | Sets the serial console configuration. |
| Set-HPBIOSSerialPort | Sets the serial port configuration. |
| Set-HPBIOSServerAvailability | Sets BIOS server availability. |
| Set-HPBIOSServerInfo | Sets the reference information for the server administrator. |
| Set-HPBIOSServerSecurity | Sets BIOS server security options. |
| Set-HPBIOSServiceContact | Sets the reference information for the server service. |
| Set-HPBIOSStorageOption | Sets the storage options of the target server. |
| Set-HPBIOSSystemInfo | Sets the BIOS system information. |
| Set-HPBIOSThermalOption | Sets the fan cooling solution for the system. |
| Set-HPBIOSTPMConfiguration | Sets the TPM (Trusted Platform Module) configurations. |
| Set-HPBIOSUEFIBootOrder | Sets the UEFI Boot Order configuration. |
| Set-HPBIOSUEFIOptimizedBoot | Sets the UEFI Optimized Boot state. |
| Set-HPBIOSUSBOption | Sets the BIOS USB options. |
| Set-HPBIOSUserDefault | Sets the BIOS user default configuration. |
| Set-HPBIOSUtilityLanguage | Selects the language for the system. |
| Set-HPBIOSVideoOption | Sets BIOS video options. |
| Set-HPBIOSVirtualization | Sets the hardware virtualization configuration. |
| Set-HPBIOSVLANConfiguration | Sets the details of global VLAN configuration for all enabled network interfaces. |
| Test-HPBIOSConnection | Checks if the connection to the target is still valid. |
| Update-HPBIOSModuleVersion | Determines if an updated version of the BIOS cmdlets is available and returns the link to download this new version. |

# User scenarios for managing BIOS

There are two ways to use the BIOS cmdlets to manage server BIOS: either by using `Connect-HPBIOS` and the iLO IP address or by using `Connect-HPBIOS` with Window Server IP address.

## Using `Connect-HPBIOS` and the iLO IP address for Gen8 servers

In this scenario, `Connect-HPBIOS` connects to Hewlett Packard Enterprise servers remotely using an iLO IP address if the target server is in the WinPE environment.

### Prerequisites

- Install BIOS cmdlets on the management client.
- Make sure the target server iLO IP address is pingable from the management client where the BIOS cmdlets are installed.

### Procedure

1. Create a Hewlett Packard Enterprise customized WinPE image by running `New-HPBIOSCustomWinPEImage` on the management client. For example, the following command creates a new WinPE image with .iso file output.

   ```
   PS C:\> New-HPBIOSCustomWinPEImage -OutputImageType ISO_File -OutputImage "C:\WinPE.iso" -Log | fl

   StatusType       : Ok
   StatusMessage    : WinPE image path: C:\WINPE.ISO
   NewWinPELogPath  : C:\New_HPBIOSCustomWinPEImage_2015021209451002.log
   ```

   The command in this example creates a new WinPE image with USB drive output.

   ```
   PS C:\>New-HPBIOSCustomWinPEImage -OutputImageType USB_Drive -OutputImage "D:" -Log | fl

   StatusType       : Ok
   StatusMessage    : WinPE image path: D:
   NewWinPELogPath  : C:\New_HPBIOSCustomWinPEImage_2015021209451002.log
   ```

2. If the customized WinPE image is created using the .iso file option, modify the standard boot order in the server’s BIOS to boot from CD-ROM.

   If the customized WinPE image is created using the USB drive option, insert the created USB media into the target server. Modify the standard boot order in server’s BIOS to boot from USB drive.

   The standard boot order can also be modified from the **iLO web→Virtual Media→Boot Order** page. If the iLO cmdlets are installed, the `Set-HPiLOOneTimeBootOrder` cmdlet or the `Set-HPiLOPersistentBootOrder` cmdlet can be used to modify the boot order.

3. Boot up the target server and enter customized WinPE environment. Booting up the server can be done using the **iLO web→Virtual Media→Boot Order** page. If the iLO cmdlets are installed, the `Reset-HPiLOServer` cmdlet can also be used to boot up the server.

   **NOTE:** When the server boots up into the customized WinPE environment for the first time, a one-time automatic reboot might occur and then it will automatically boot up into WinPE again.

4. The target server boots the WinPE image. Verify that the WinPE console appears on the target server. You can check this from the iLO Integrated Remote Console or using an SSH client like PuTTY.
5. Before using `Connect-HPBIOS`, ping the management client and target server networks to ensure they are both reachable.
6. Ensure that both client and server firewalls are disabled.
7. If a firewall is needed, add a firewall exception for remote WMI connections.
   a. At the command prompt, enter:

      ```
      netsh advfirewall firewall set rule group=”Windows Remote Management” new enable=yes
      ```

   b. Enter `Exit`.

8. At the WinPE prompt, ping the management client IP address to ensure it is reachable. Contact your network administrator to resolve any issues.
9. Execute `Connect-HPBIOS` with an iLO IP address. A successful connection will return the session object.
10. Use the session object from the previous step to run BIOS cmdlets.

## Using `Connect-HPBIOS` with Windows server IP address for Gen8 systems

In this scenario, the BIOS cmdlets modify the BIOS settings of the target server remotely if there is a Windows system on the target server.

### Prerequisites

Prepare a management client on which BIOS cmdlets are installed.

### Procedure

1. Ensure that both client and server firewalls are disabled.
2. If a firewall is needed, add a firewall exception for remote WMI connections.
    a. At the command prompt, enter:

    ```
    netsh advfirewall firewall set rule group=”Windows Remote Management” new enable=yes
    ```

    b. Enter `Exit`.
3. At the Windows Server, ping the management client IP address to ensure it is reachable. Contact your network administrator to resolve any issues.
4. Verify that the Windows Management Instrumentation service is running on the Windows server.
5. Verify that the File Share service is running on the Windows server.
6. If the File Share service is not running, enter `net start “Server”` at the command prompt.
7. Verify that the default net share is available using the following procedure.
    a. Open a command line or PowerShell window.
    b. Run the `net share` command and make sure the Windows default shares of C$, IPC$, ADMIN$ are listed as shown.

    ```
    c:\ >net share
    Share name   Resource                                Remark
    ------------------------------------------------------------------
    C$           C:\                                     Default share
    IPC$                                                 Remote IPC
    ADMIN$       C:\Windows                              Remote Admin
    The command completed successfully.
    ```

    If these shares are not listed, use the `net share` command to create them. For detailed help for `net share`, enter `net share /?`.
8. Execute `Connect-HPBIOS` with the Windows server IP address. A successful connection will return the session object.
9. Use the session object to run the BIOS cmdlets. For more information, see “Using cmdlets to manage the BIOS” (page 17).

## Using `Connect-HPBIOS` and the iLO IP address for Gen9 systems

### Prerequisites

- Install BIOS cmdlets on the management client.
- Make sure the target server iLO IP address is pingable from the management client where the BIOS cmdlets are installed.

### Procedure

1. Execute `Connect-HPBIOS` with the iLO IP address of the Gen9 target server. A successful connection will return the session object.
2. Use the session object to run the BIOS cmdlets. For more information, see “Using cmdlets to manage the BIOS” (page 17).

# IPv6 support

Consider the following when using IPv6.

- IPv6 is supported and enabled on the system.
- IPv6 is supported in addition to IPv4 for network addresses on all cmdlets that have an IP address parameter. The double colon zero subnet format for IPv6 addresses is supported. For example, `1a00::1fe8` equates to `1a00:0000:0000:0000:0000:0000:0000:1fe8`.
- Address ranges are supported with the dash character. For example, `1a00::1fe8-1fef` resolves to eight addresses from `1a00::1fe8` through `1a00::1fef`.
- Sets of addresses are supported with the comma character. For example, `1a00,1b00::1fe8` resolves to two addresses, `1a00::1fe8` and `1b00::1fe8`.
- Examples in this document use IPv4 but could use IPv6 instead if supported in the network. Both IPv4 and IPv6 addresses can be used within one cmdlet.

For more information on IPv6, see the following website or the references it links to:

**http://en.wikipedia.org/wiki/IPv6**

# Multithreaded operation of cmdlets

For better performance, multithreading is used when one cmdlet sends data to multiple targets. Most cmdlets that support multiple targets use the multithreading feature in the cmdlets.

Up to 256 threads are used. This number was chosen after measuring response times and observing greatly diminishing returns by using more. Performance of the cmdlets depends on factors such as current system load, available memory, number of processors, network configuration, other systems in the network, and other network traffic.

To take advantage of multithreading, a single cmdlet is used but it is directed it to multiple targets in a single invocation by passing parameter sets as an array. Multiple threads are used automatically when you do this.

For example, the following executes a single invocation of the cmdlet, passing one parameter set at a time. This does not take advantage of multithreading.

```
foreach ($parameterset in $arrayofparametersets) {
 $parameterset | Get-HPBIOSMemoryPower
}
```

To take advantage of multithreading, send a cmdlet an array of parameter sets in a single invocation. The following uses multithreading, sending commands to up to 256 targets in parallel.

```
$arrayofparametersets | Get-HPBIOSMemoryPower
```

## Piping output from one command to another

A useful feature of PowerShell is the ability to pipe output from one command to another. The following example shows piping output from `Connect-HPBIOS` to `Get-HPBIOSPowerProfile` to produce the power profile information of those connected servers. The `-Verbose` parameter is used to view more information.

**PowerShell script:**

```
$PowerProfile= Connect-HPBIOS -IP 192.168.243.100-102 -Username "username" -Password
"password" -DisableCertitificateAuthentication | Get-HPBIOSPowerProfile -Verbose
VERBOSE: Using 3 threads.
```

**Script output:**

```
$PowerProfile | fl

IP             : 192.168.243.100
Hostname       : server1.Company.net
StatusType     : OK
HPPowerProfile : Maximum Performance

IP             : 192.168.243.101
Hostname       : server2.Company.net
StatusType     : OK
HPPowerProfile : Balanced

IP             : 192.168.243.102
Hostname       : server3.Company.net
StatusType     : OK
HPPowerProfile : Custom
```

The verbose output indicates that three threads are being used for `Get-HPBIOSPowerProfile`. This threading enables multiple commands to multiple servers to be sent at the same time. `Connect-HPBIOS` makes the connection object array of the three servers. Those are in turn passed through to `Get-HPBIOSPowerProfile`, which uses those connections and requests the power profile information from each server. The final results are the power profile information for the three servers connected.

## Using the `Get-HPBIOSModuleVersion` and `Update-HPBIOSModuleVersion` cmdlets

These cmdlets are used to determine the current version of the BIOS cmdlets module installed and update the BIOS cmdlets module if necessary.

The `Get-HPBIOSModuleVersion` cmdlet has no parameters. It accesses the installed module file and help files and displays information about them including version numbers. The following is typical `Get-HPBIOSModuleVersion` cmdlet output.

```
PS C:\> Get-HPBIOSModuleVersion

Name                      : HPBIOSCmdlets
Path                      : C:\Program Files\Hewlett-Packard\PowerShell\Modules\HPBIOSCmdlets\HPBIOSCmdlets.dll
Description               : Cmdlets to interface with HP BIOS
GUID                      : 05545ade-5f25-4696-bfcc-e1d67fe32519
Version                   : 1.1.0.0
CurrentUICultureName      : en-US
CurrentUICultureVersion : 1.1.0.0
AvailableUICulture        : {@{UICultureName=en-US; UICultureVersion=1.0.0.1}, @{UICultureName=zh-CN;
UICultureVersion=1.0.0.1}, @{UICultureName=ja-JP;
            UICultureVersion=1.0.0.1}}"
```

The `Update-HPBIOSModuleVersion` cmdlet has no parameters. This cmdlet checks the version number of the installed cmdlets against the version number available for download. If the local version is the most recent, the output will indicate so.

```
PS C:\Users\Username> Update-HPBIOSModuleVersion
The currently installed version 1.0.0.0 is the most current.
```

If there is a more recent version available than the one currently installed locally, the output indicates so and provides the option to download the latest version.

```
PS C:\Users\Username> Update-HPBIOSModuleVersion
There is a newer version of HPBIOSCmdlets available at
http://www.hpe.com/servers/powershell.
Do you want to go there to download the new version?(Y/N): Y
```

If you respond Yes to the download prompt, a browser window opens and you can download and install the newer version.

## Using the `Connect-HPBIOS` cmdlet

The `Connect-HPBIOS` cmdlet is used to connect to a BIOS target represented by its iLO (if running under WinPE) or OS (if running Windows) IP address. The target can be a single IP address/hostname, or multiple IP addresses/hostnames. The target can also be the management client IP address/hostname.

Whenever the BIOS administrator password is set on a Gen9 target server, any further modifications to the BIOS properties on the same server will require the administrator password during connection or all Set cmdlets will fail. By default, all Get cmdlets will work without providing the administrator password during connection even though the BIOS administrator password is already set on those systems.

```
PS C:\> $conObj = Connect-HPBIOS 192.168.242.61 -Username "username" -Password "password" -AdminPassword
```

The following is an example of using the `Connect-HPBIOS` cmdlet. In this example, the connection is successful and the value of `$conObj.IsConnected` is True. Make sure to pass the `Connect-HPBIOS` output to a variable (`$conObj` in the example). This is necessary because managing the BIOS with other BIOS cmdlets needs this connection object.

```
PS C:\> $conObj = Connect-HPBIOS 192.168.242.61 -Username "username" -Password "password"
-DisableCertitificateAuthentication

PS C:\> $conObj

TargetType    : Windows
IP                  : 192.168.242.61
Hostname       : server.company.com
IsConnected    : True
ConnectionInfo : HPBIOSCmdlets.HPBIOSConnection+HPBIOSConnectionInfo
```

If the connection fails, an error message similar to the following is displayed:

```
Connect-HPBIOS : Failed for 192.168.242.61:A connection attempt failed because
                 the connected party did not properly respond after a period of time,
                 or established connection failed because connected host has failed to
                 respond 192.168.242.61:22
At line:1 char:1
+ Connect-HPBIOS 192.168.242.61 -Username Administrator -Password Admin
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidData: (:) [Connect-HPBIOS], HPBIOSErrorMsg
    + FullyQualifiedErrorId : HPBIOSCmdlets.ConnectHPBIOS
```

You can use either Username and Password parameters or a Credential parameter to connect to the target. The following is an example of using the Credential parameter:

```
PS C:\> $credential = Get-Credential -Message "Please input username and password"
PS C:\> $conObj = Connect-HPBIOS 192.168.242.61 -Username "username" -Password "password"
-DisableCertitificateAuthentication
PS C:\> $conObj

TargetType    : Windows
IP                  : 192.168.242.61
Hostname       : server.company.com
IsConnected    : True
ConnectionInfo : HPBIOSCmdlets.HPBIOSConnection+HPBIOSConnectionInfo
```

## Using cmdlets to manage the BIOS

After a connection to the BIOS is established, other cmdlets can be used to manage the BIOS. For example, the `Get-HPBIOSPowerRegulator` cmdlet can be used to display power regulator information as shown in the following example.

```
PS C:\>Get-HPBIOSPowerRegulator $conObj | fl

IP                : 192.168.1.2
Hostname          : server.company.net
StatusType        : OK
HPPowerRegulator : HP Dynamic Power Savings Mode
```

Other cmdlets can be used to set or update BIOS settings. For example, the `Set-HPBIOSPowerRegulator` cmdlet can be used to change the BIOS power regulator as shown in the following example.

```
PS C:\> $conObj | Set-HPBIOSPowerRegulator -Regulator OS_Control
```

If the cmdlet is successful, no other message is displayed. If an error occurs, an output message similar to the following is displayed.

```
IP            : 192.168.1.2
Hostname      : server.company.net
StatusType    : Error
StatusMessage : The target does not support Regulator. Check more details about how to use
                this cmdlet from help.
```

You can use output type `RawText` to display detailed information returned from the BIOS cmdlet as shown in the following example with the `Set-HPBIOSPowerRegulator` cmdlet.

```
PS C:\> $conObj | Set-HPBIOSPowerRegulator -Regulator Dynamic_Power_Savings -OutputType RawText

<Section name="HP_Power_Regulator" helptext="Allows tuning of the system power usage">
               HP_Dynamic_Power_Savings_Mode</Section>
```

## Using the `Disconnect-HPBIOS` and `Disconnect-HPBIOSAllConnection` cmdlet

Use the `Disconnect-HPBIOS` cmdlet to disconnect the connection object when you are finished using the BIOS settings. This is necessary when the connection is for WinPE. If there is a connection to WinPE that has not disconnected, another connection cannot be made to the WinPE because only one VSP session is allowed at a time.

```
PS C:\Windows\system32> Disconnect-HPBIOS $conObj

PS C:\Windows\system32>
```

If the cmdlet is successful, no other message is displayed. If an error occurs, an appropriate output message is displayed.

Use the `Disconnect-HPBIOSAllConnection` to close all active BIOS connections on the current PowerShell session and clean up resource files.

## Using parameters from a file

In some situations it may be more convenient to manage scripts by having the parameters for calling them contained in an external file or database. This is especially true if either the list of systems to communicate with or the number of parameters to enter is unwieldy. PowerShell provides cmdlets that support many different types of input data and can convert them to internal

PowerShell data objects. The following example illustrates the method of using a Comma Separated Value (CSV) file.

## CSV file input

CSV files are easy to create and maintain with Microsoft Excel or a text editor like Notepad. For this example script, the following CSV file is used.

```
ServerList.csv:

IP,Username,Password
192.168.1.2,Administrator,Admin
192.168.1.3,Administrator,Admin
192.168.1.4,Administrator,Admin
```

By using the Windows PowerShell embedded cmdlet `import-csv ServerList.csv`, all CSV data will be converted to one object collection by row. For information on the `import-csv` cmdlet, see **https://technet.microsoft.com/en-us/library/hh849891.aspx**.

The `import-csv` results are passed to `Connect-HPBIOS` in the pipeline. All data fields supported by `Connect-HPBIOS` will be used as input parameters for each object. In the following example there are three objects in the collection. All three objects will be used by `Connect-HPBIOS`. Then it is available to pass the `Connect-HPBIOS` result to `Get-HPBIOSPowerProfile` in the pipeline.

### CSV file input script

```
Import-csv ServerList.csv | Connect-HPBIOS -DisableCertitificateAuthentication | Get-HPBIOSPowerProfile |
Format-List
```

### Script output

```
IP             : 192.168.1.2
Hostname       : test1.company.com
StatusType     : OK
HPPowerProfile : Custom

IP             : 192.168.1.3
Hostname       : test2.company.com
StatusType     : OK
HPPowerProfile : Maximum Performance

IP             : 192.168.1.4
Hostname       : test3.company.com
StatusType     : OK
HPPowerProfile : Maximum Performance
```

# Script writing methodology

When deciding to write a script, you generally know what you want to accomplish. One of the powerful features of PowerShell ISE is that you can build a script piece-by-piece, testing code and viewing objects to get a better understanding how to accomplish what you want to do.

Here is a typical process you might want to use for creating PowerShell scripts.

1. Determine what type of data you want to get.
2. Execute the appropriate cmdlet interactively to retrieve the data.
3. After viewing the command results, decide what part of the object you are interested in.
4. Create the main processing to manage BIOS by different cmdlets.
5. Summarize or output the data in the desired format.

If there are many steps, repeat the process until all of the requirements of the data collection or setting have been completed.

When using data sources such as .csv, .xml or databases to store and retrieve data to use for targets it may be necessary to provide their usernames and passwords. These may need to be encrypted for security purposes. Encrypted storage and data use is beyond the scope of this document. It is not a recommended practice to embed passwords in scripts; instead they can be prompted for by omitting them as a parameter. You must be cognizant of your organizations security policies and code accordingly.

### Script examples

BIOS script examples are packaged along with the .msi installer and Readme First installation document. Beginning with BIOS 1.1, comprehensive PowerShell script examples are available on the Hewlett Packard GitHub repository at **https://github.com/HewlettPackard/PowerShell-ProLiant-SDK**.

## Understanding the operating process

Each BIOS cmdlet is designed to follow a particular operating sequence when executed. The functions performed often depend upon certain prerequisites. The following cmdlet examples detail the flow of this process.

### `Connect-HPBIOS` execution

1. When executed, the `Connect-HPBIOS` cmdlet first detects whether the target is an iLO IP address or a ProLiant host with a Windows-loaded IP address.
2. If the target is an iLO address, `Connect-HPBIOS` identifies whether the server is Gen8 or Gen9.
3. If the target iLO address is a Gen8 server, `Connect-HPBIOS` determines whether the WinPE EMS console is available.
4. If the WinPE EMS console is available, `Connect-HPBIOS` performs the following:
   a. Creates a local share folder in the management client and grants read/write access permissions to all users.
   b. Runs `net use` in WinPE to map to the share.
   c. Executes `ForWINpECheckingEnv.bat`, and populates the connection object by parsing previously generated files.
5. If the target iLO address is a Gen9 server, `Connect-HPBIOS` performs the following:
   - Verifies that the target iLO RESTful interfaces are accessible from the management client and returns the session object back to the user.
6. If the target is a Window server IP address, `Connect-HPBIOS` performs the following:
   a. Verifies that the remote drive is accessible. If it is not accessible, `Connect-HPBIOS` grants access to it.
   b. Copies the necessary RCU files and scripts to remote drive under the HPBIOSCmdlets_Tools folder.
   c. Executes `ForCheckEnv.bat` and populates the connection object.
7. If the cmdlet fails to connect, it returns an error record with the reason for failure.

```
Connect-HPBIOS : Failed for 192.168.243.56 eso-ironside1-os.chn.hpe.com:Access is denied. (Exception from
 HRESULT: 0x80070005 (E_ACCESSDENIED))
At line:1 char:6
+ $c = Connect-HPBIOS -IP "192.168.243.56" -Username "Administrator" -Password "Abc ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (:) [Connect-HPBIOS], HPBIOSException
    + FullyQualifiedErrorId : HPBIOSCmdlets.ConnectHPBIOS
```

8. If the cmdlet successfully connects to the target, it returns a connection object with information about the target system. The information will be used in later Get or Set cmdlets.

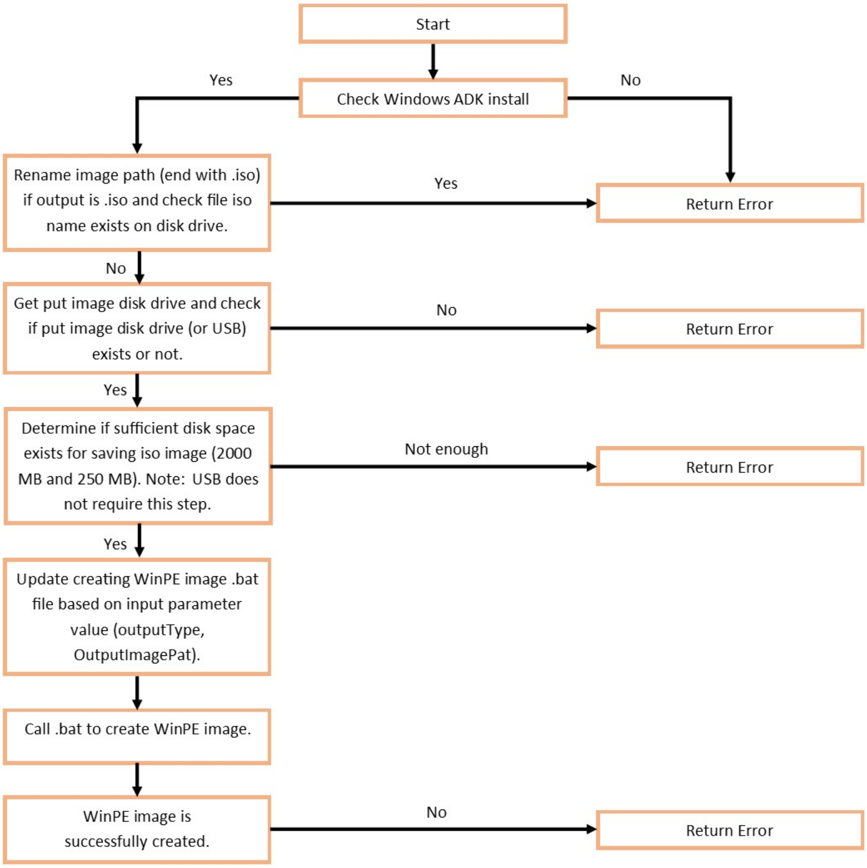
**NOTE:** Most of the properties are added into a structure titled `ConnectionInfo`.

```
TargetType     : Windows
IP             : 192.168.243.56
Hostname       : eso-ironside1-os.chn.hpe.com
IsConnected    : True
ConnectionInfo : HPBIOSCmdlets.HPBIOSConnection+HPBIOSConnectionInfo
```

## `Disconnect-HPBIOS` execution

1. When closing a connection using `Disconnect-HPBIOS`, properties in `ConnectionInfo` are removed and files generated by this connection are deleted.
2. If the connection successfully closed or was previously disconnected, no message is displayed.
3. If `Disconnect-HPBIOS` is used during a Get/Set operation, the error message `already disconnected` appears.

## `New-HPBIOSCustomWinPEImage` execution


Start
Yes
Check Windows ADK install
No
Rename image path (end with .iso) if output is .iso and check file iso name exists on disk drive.
Yes
Return Error
No
Get put image disk drive and check if put image disk drive (or USB) exists or not.
No
Return Error
Yes
Determine if sufficient disk space exists for saving iso image (2000 MB and 250 MB). Note: USB does not require this step.
Not enough
Return Error
Yes
Update creating WinPE image .bat file based on input parameter value (outputType, OutputImagePat).
Call .bat to create WinPE image.
WinPE image is successfully created.
No
Return Error

# 4 Troubleshooting

## General issues

### A disk error occurs when managing BIOS in WinPE on Gen8 servers

When managing the BIOS in a WinPE environment on some Gen9 servers, a disk error might occur indicating that the server cannot find the disks attached to the Dynamic Smart Array. This occurs when the ISO image created by the `New-HPBIOSCustomWinPEImage` cmdlet is loaded to WinPE and the boot mode is changed to Legacy BIOS Boot Mode. The Dynamic Smart Array is not supported in Legacy BIOS Boot Mode.

**Solution**: Backup all drives attached to the Dynamic Smart Array before using BIOS cmdlets to manage the BIOS on a Gen9 server. Set the boot mode to UEFI Boot Mode using the `Set-HPBIOSBootMode` cmdlet as required. Reconfiguring the Dynamic Smart Array may be necessary using the HPE Smart Storage Administrator (HPE SSA) in Intelligent Provisioning.

### IP address details not listed in WinPE on Gen8 servers

The most likely reason IP address details are not appearing in WinPE on Gen8 servers is because NIC drivers were not previously installed.

**Solution**: Check the NIC details and download the appropriate NIC drivers from the Hewlett Packard Enterprise website. Downloaded driver files can be made available either through virtual media such as iLO web or using the iLO cmdlet `Mount-HPiLOVirtualMedia`. Run `drvload <driver_path>` on the WinPE console to install the drivers. After the drivers are loaded, verify IP address details are available.

### `Connect-HPBIOS` returns `RPC server is unavailable` error message

When executing `Connect-HPBIOS` cmdlet, the following error message is displayed:

```
RPC server is unavailable
```

**Solution**: Ping the management client and target server to verify the connection. Also, check that the target server's firewall is disabled. If the firewall cannot be disabled, add the Windows Management Instrumentation service to the firewall's exception list.

Another possible reason for the error message is that the default SSH and its port were disabled for the target server.

**Solution**: Enable the SSH and set the port to `22` in iLO settings. Post this update, and the `Connect-HPBIOS` cmdlet should work properly.

### Verifying BIOS cmdlet version

**Solution**: If a problem occurs, verify that the most current version of the BIOS cmdlets is installed. Updating to the most current version might solve the problem.

To determine if there is a newer version of the BIOS cmdlets is available, see "Using the `Get-HPBIOSModuleVersion` and `Update-HPBIOSModuleVersion` cmdlets" (page 15).

### Error executing `New-HPBIOSCustomWinPEImage`

If executing `New-HPBIOSCustomWinPEImage` cmdlet fails to create an ISO image, the following error message is displayed:

```
WINPE failed to create the image
```

If you check the cmdlet's log files in `New_HPBIOSCustomWinPEImage_XXXXXX.log`, you will find this error:

```
Processing 1 of 1 - Adding package WinPE-HTA-Package~31bf3856ad364e35~x86~~6.2.9200.16384, An error occurred -
 WinPE-HTA-Package Error: 0x80070005
```

**Solution**: Because a certain version of McAfee antivirus is installed, the Microsoft ADK is unable to add the package. To resolve this issue, download and install McAfee VirusScan Enterprise (VSE) 8.8 Patch 5 or the latest available patch from McAfee.

### User closes PowerShell without disconnecting the session

If you close a PowerShell session without using `Disconnect-HPBIOS` to exit, the following error message will appear when attempting to establish new session using `Connect-HPBIOS`:

```
Connect-HPBIOS : Failed for 1.12.11.19:
Virtual Serial Port is currently in use by another session
```

**Solution**: Reset the iLO from either iLO web or using the iLO PowerShell cmdlet `Reset-HPiLORIB`.

### Remote server returns `(401) Unauthorized`

You may encounter an `(401) Unauthorized` error when attempting to Get/Set BIOS configurations over a stale target server connection. Stale connections result on established connections where no activity takes place over a prolonged period of time. Under these conditions, a `(401) Unauthorized` error will occur even when attempting `Test-HPBIOSConnection` cmdlet.

**Solution**: Use the `Disconnect-HPBIOS` cmdlet to disconnect the existing stale connection object, and then reconnect with the `Connect-HPBIOS` cmdlet. Try executing any Get/Set cmdlet again.

## Usage tips

### Input parameter matching

For input parameters, arrays of objects are supported. If an array of objects is provided to both parameter A and parameter B of a BIOS cmdlet, it matches input values for parameter A with values for parameter B.

In the following example, `Get-HPBIOSXXXX` has a mandatory parameter (named “`MandatoryParam`”), the execution result of each case is added as comments before each case.

```
#”value1” is used with $con1 and ”value2” is used with $con2
Get-HPBIOSXXXX –Connection @($con1, $con2) –MandatoryParam @(“value1”, “value2”)

#”value1” is used with $con1, ”value2” is used with $con2, and “value3” is discarded
Get-HPBIOSXXXX –Connection @($con1, $con2) –MandatoryParam @(“value1”, “value2”, “value3”)

#”value” is used with both $con1 and $con2
Get-HPBIOSXXXX –Connection @($con1, $con2) –MandatoryParam “value”

#”value1” is used with $con1 and it will ask the user to input the value of MandatoryParam for $con2
Get-HPBIOSXXXX –Connection @($con1, $con2) –MandatoryParam @(“value1”)

#”value1” is with for $con1 and an error record is written for $con2 because the Force parameter is used.
#(With the Force parameter, users will not be asked to input data but will get an error.)
Get-HPBIOSXXXX –Connection @($con1, $con2) –MandatoryParam @(“value1”) -Force
```

In the following example, `Get-HPBIOSYYYY` has an optional parameter (named “`OptionalParam`”).

```
#”value1” is used with $con1, ”value2” is used with $con2
Get-HPBIOSYYYY –Connection @($con1, $con2) –OptionalParam @(“value1”, “value2”)

#”value1” is used with $con1, ”value2” is used with $con2, and “value3” is discarded
Get-HPBIOSYYYY –Connection @($con1, $con2) –OptionalParam @(“value1”, “value2”, “value3”)

#”value” is used with both $con1 and $con2
Get-HPBIOSYYYY –Connection @($con1, $con2) –OptionalParam “value”

#”value1” is used with $con1. No OptionalParam value for $con2 is given so a default value (if any) is used
Get-HPBIOSYYYY –Connection @($con1, $con2) –OptionalParam @(“value1”)

#Value3 is used with $con1, value4 is used with $con2. value1 and value2 are not used.
```

```
#Priority is given to values from the commandline if both pipeline and commandline have the values.
$p1 = New-Object -TypeName PSObject -Property @{ "Connection"=$con1;"Parameter"=value1} ;
$p2 = New-Object -TypeName PSObject -Property @{ "Connection"=$con2;"Parameter"=value2} ;
@($p1,$p2)| Get-HPBIOSXXX -Parameter @(value3, value4)
```

# 5 Support and other resources

## Accessing Hewlett Packard Enterprise Support

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

  **www.hpe.com/assistance**

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

  **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:

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

  - Software Depot website:

    **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:

  **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 HP Passport set up with relevant entitlements.

## Support Information

Hewlett Packard Enterprise offers a number of additional software support services, many of which are provided to our customers at no additional charge.

## Hewlett Packard Enterprise Support Center

**Join the discussion** at the Support Center, a community-based, user-supported tool for Hewlett Packard Enterprise customers to participate in discussions amongst the customer community about Hewlett Packard Enterprise products.

## Reporting errors to Hewlett Packard Enterprise

If you get a PowerShell error that indicates that it is reporting something within the HPBIOSCmdlet module code, please contact Hewlett Packard Enterprise. Provide as much information as possible, including screen captures if appropriate. Also include the output of the following command:

```
PS C:\Users\yourname> Get-HPBIOSModuleVersion
```

# Related information

The following documents and websites provide related information:

### Documents

- *Scripting Tools for Windows PowerShell Release Notes: BIOS Cmdlets*
- *HP ROM-Based Setup Utility User Guide*
- *HPE UEFI System Utilities User Guide for HPE ProLiant Gen9 Servers*

### Websites

- Scripting Tools for Windows PowerShell: **http://www.hpe.com/servers/powershell**
- Scripting Tools for Windows PowerShell Information Library: **http://www.hpe.com/info/powershell/docs**
- RBSU Information Library: **http://www.hpe.com/info/rbsu/docs**
- UEFI System Utilities Information Library: **http://www.hpe.com/info/ProLiantUEFI/docs**
- PowerShell-ProLiant-SDK (for infrastructure script examples): **https://github.com/HewlettPackard/PowerShell-ProLiant-SDK**

# Windows PowerShell resources

The following websites provide useful information for using PowerShell.

- **Scripting Tools for PowerShell Documentation**
- **Microsoft Script Center**
- **Windows PowerShell Blog**
- **PowerShell.com**
- **PowerShell Community Groups**
- **PowerShell.org**
- **PowerShell Magazine**

## Websites

| Website | Link |
|---|---|
| Hewlett Packard Enterprise Information Library | **www.hpe.com/info/enterprise/docs** |
| Hewlett Packard Enterprise Support Center | **www.hpe.com/support/hpesc** |
| Contact Hewlett Packard Enterprise Worldwide | **www.hpe.com/assistance** |
| Subscription Service/Support Alerts | **www.hpe.com/support/e-updates** |
| Software Depot | **www.hpe.com/support/softwaredepot** |
| Customer Self Repair | **www.hpe.com/support/selfrepair** |
| Insight Remote Support | **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:

**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:

**www.hpe.com/info/insightremotesupport/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.

# Index