Hewlett Packard
Enterprise

Scripting Tools for Windows PowerShell User Guide

iLO Cmdlets v3.0.0.0

Abstract

This document contains instructions for using Scripting Tools for Windows PowerShell to manage iLO. It is intended for system administrators who use the Scripting Tools for Windows PowerShell to manage their IT infrastructure.

Part Number: P06780-005
Published: March 2020
Edition: 1

Notices

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

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

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

Acknowledgments

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

Contents

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 management 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 ProLiant servers using familiar PowerShell syntax. Documentation describing how to apply these new tools to configure HPE ProLiant servers is also included.

This guide is intended for system administrators who use the Scripting Tools for Windows PowerShell to manage their IT infrastructure. Users must be familiar with Windows PowerShell and iLO. For more information about iLO, see HPE iLO 4 and iLO 5 user guide and other related iLO documents on the iLO information library (**http://www.hpe.com/info/ilo/docs**).

Major changes included in the iLO 3.x cmdlets

The following changes have been made from version 2.x to 3.x:

- The connection object created using HPEiLOCmdlets module can be used across other modules such as HPEBIOSCmdlets. This allows a single session to be established on a particular iLO and the same is used to configure iLO/BIOS settings instead of creating multiple sessions to single iLO. (Version 3.0.0.0 onwards)
- The output of the connection object is modified to support interoperability.
- Introduced two new cmdlets for modifying and getting the log configuration settings.

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, Scripting Tools for Windows PowerShell is a powerful set of utilities that can be used to perform various configuration tasks on hardware. These cmdlets follow the standard PowerShell syntax and scripting model, making it easy for you to incorporate these functions into your administrative scripts.

Installation

Before installation, ensure that your system meets all requirements for supported operating systems, environments, and hardware. For more information, see *Scripting Tools for Windows PowerShell Release Notes: iLO Cmdlets v3.0.0.0*.

System prerequisites

Install the following before installing Scripting Tools for Windows PowerShell: iLO Cmdlets. The following links provide access to the Microsoft download sites for these applications. Make sure that you read and understand the system requirements and other information provided.

1. Install Microsoft .NET Framework 4.7.1 or later.

   Microsoft .NET Framework 4.7.1

   NOTE: Microsoft .NET Framework must be installed **before** installing Windows Management Framework.

2. Install Windows Management Framework 3.0 or later (which includes PowerShell 3.0 or later).
   - **Windows Management Framework 3.0**
   - **Windows Management Framework 4.0**
   - **Windows Management Framework 5.0**
   - **Windows Management Framework 5.1**

Supported operating systems

Scripting Tools for Windows PowerShell: iLO Cmdlets are supported on the following operating system versions:

- Microsoft Windows 7 SP1
- Microsoft Windows 8.1
- Microsoft Windows 10
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016
- Microsoft Windows Server 2019

IMPORTANT: Using multiple targets in a single cmdlet on a 32-bit operating system can produce unsatisfactory cmdlet output if there are more than 100 target IP addresses. HPE recommends using a 64-bit operating system to run the cmdlets with more than 100 targets.

Installing Scripting Tools for Windows PowerShell - iLO Cmdlets from HPE Website

Procedure

1. Download the Scripting Tools for Windows PowerShell: iLO Cmdlets installer from the following website: **http://www.hpe.com/servers/powershell**
2. Close all PowerShell windows before the installation.
3. Run the installer from an account with administrative privileges, by using any standard method of execution (command line or double-click).

It might be necessary to change the execution policy for PowerShell. In PowerShell, enter the following command to get more information and to help you to decide what to select:

```
PS C:/> help about_Execution_Policies
```

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

```
PS C:/> Get-ExecutionPolicy -list
```

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

```
PS C:/> Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned
```

The installation will halt and not complete successfully in case any of the following conditions are detected:

- Attempting to install without .NET 4.7.1 or above.
- Attempting to install without PowerShell 3.0 or above.

Installing iLO cmdlets from PowerShell Gallery

PowerShell Gallery is a market place where PowerShell module or scripts from different vendors, users and individuals will be stored in a cloud environment. It's a central repository for PowerShell content.

You can choose to install the online version of iLO Cmdlets from the Microsoft PowerShell Gallery.

```
PS C:\> Install-Module -Name HPEiLOCmdlets -Verbose
```

Visit the Microsoft PowerShell Gallery at **https://www.powershellgallery.com** and search for "HPEiLOCmdlets" for more details.

Repairing the iLO cmdlets

Use the installer repair option for the following scenarios:

- The iLO cmdlets module is installed, but PowerShell is not able to import the iLO cmdlets module.
- iLO cmdlets module files, dependent files, or registry entries are corrupted.

Procedure

1. Open Windows Control Panel.
2. Select **Programs and Features**.
3. Select **Scripting Tools for Windows PowerShell: iLO cmdlets**.
4. Click **Repair**.

Uninstalling the iLO cmdlets using the `Uninstall-Module` cmdlet

Use the `Uninstall-Module` cmdlet to remove the module from your system.

```
PS C:/> Uninstall-Module -Name HPEiLOCmdlets -Verbose
```

Overview

The following topics provide information about how to use the iLO cmdlets:

Description of iLO Cmdlets

The following table provides a list and brief description of all the iLO Cmdlets.

Cmdlet help

The iLO cmdlets include help support similar to other PowerShell cmdlet help. To display a list of the iLO cmdlets, type:

```
help *hpeilo*
```

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

```
Get-Command -Module HPEiLOCmdlets
```

To display complete help for a specific cmdlet, type:

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

where `<cmdlet>` is the name of the iLO cmdlet

The iLO cmdlets support the PowerShell `Update-Help` feature. This command retrieves the most current help files from an HPE website and puts them in the correct location on your system.

Sr. No	Cmdlet Name	Description	iLO Supportability
1	Find-HPEiLO	Finds an array of iLO in a specified subnet.	iLO 4, iLO 5
Connection cmdlets			
2	Connect-HPEiLO	Creates connection to one or multiple iLO targets.	iLO 4, iLO 5
3	Disconnect-HPEiLO	Closes the connection.	iLO 4, iLO 5
4	Test-HPEiLOConnection	Checks if the connection to the target is valid.	iLO 4, iLO 5
User Administration cmdlets			
5	Add-HPEiLOUser	Creates a local user account to the iLO.	iLO 4, iLO 5
6	Get-HPEiLOUser	Gets the information of all local iLO users.	iLO 4, iLO 5
7	Set-HPEiLOUser	Modifies the existing local user account present in iLO.	iLO 4, iLO 5
8	Remove-HPEiLOUser	Removes an existing local user account.	iLO 4, iLO 5
9	Set-HPEiLOAdministratorPassword	Sets the administrator password to the specified value.	iLO 4, iLO 5
Federation cmdlets			
10	Add-HPEiLOFederationGroup	Creates an iLO Federation group or includes an iLO in an existing group membership.	iLO 4, iLO 5
11	Get-HPEiLOFederationGroup	Gets a list of all iLO Federation group names.	iLO 4, iLO 5
12	Set-HPEiLOFederationGroup	Modifies an existing iLO Federation group.	iLO 4, iLO 5
13	Remove-HPEiLOFederationGroup	Removes the specified iLO from an iLO Federation group membership.	iLO 4, iLO 5

Table Continued

14	Get-HPEiLOFederationMulticast	Gets the iLO Federation status and also the iLO federation multicast options.	iLO 4, iLO 5
15	Set-HPEiLOFederationMulticast	Enables iLO Federation and sets iLO Federation multicast operations.	iLO 4, iLO 5
16	Get-HPEiLOFederationPeer	Gets the iLO Federation peers.	iLO 5
Language cmdlets			
17	Add-HPEiLOLanguagePack	Adds the language pack to the iLO.	iLO 4, iLO 5
18	Remove-HPEiLOLanguagePack	Removes the language pack from iLO.	iLO 5
19	Get-HPEiLOLanguage	Gets all languages installed on the iLO.	iLO 4, iLO 5
20	Set-HPEiLOLanguage	Sets the default language on iLO.	iLO 4, iLO 5
SSO setting cmdlets			
21	Add-HPEiLOSSORecord	Adds a new HPE SIM Single Sign-On (SSO) Server Record.	iLO 4, iLO 5
22	Remove-HPEiLOSSORecord	Removes an HPE SIM Trusted SSO Server record.	iLO 4, iLO 5
23	Clear-HPEiLOSSORecord	Clears all the SSO Record.	iLO 5
24	Get-HPEiLOSSOSetting	Gets the SSO setting for the iLO.	iLO 4, iLO 5
25	Set-HPEiLOSSOSetting	Modifies the SSO settings for the iLO.	iLO 4, iLO 5
EventLog, IML, and Security Log cmdlets			
26	Get-HPEiLOEventLog	Gets the iLO event logs.	iLO 4, iLO 5
27	Clear-HPEiLOEventLog	Clears the iLO event logs.	iLO 4, iLO 5
28	Get-HPEiLOIML	Gets the iLO integrated management logs.	iLO 4, iLO 5
29	Clear-HPEiLOIML	Clears the Integrated Management Logs.	iLO 4, iLO 5
30	Get-HPEiLOSecurityLog	Gets Security Log information.	iLO 5

Table Continued

31	Clear-HPEiLOSecurityLog	Clears the iLO Security logs.	iLO5
Power Setting cmdlets			
32	Get-HPEiLOSNMPPowerAlertThreshold	Gets the power alert threshold for the iLO devices.	iLO 4, iLO 5
33	Set-HPEiLOSNMPPowerAlertThreshold	Sets the power alert threshold value for the iLO.	iLO 4, iLO 5
34	Get-HPEiLOPowerCapSetting	Gets the power capping information of the server.	iLO 4, iLO 5
35	Set-HPEiLOPowerCapSetting	Sets the power cap feature on the host server.	iLO 4, iLO 5
36	Get-HPEiLOPowerReading	Gets the power readings from the server power supply.	iLO 4, iLO 5
37	Get-HPEiLOPowerSupply	Gets the power supply details for the host server where the iLO is located.	iLO 4, iLO 5
38	Get-HPEiLOPowerOnTime	Gets the virtual clock value, in minutes, since the server was last powered on.	iLO 4, iLO 5
39	Get-HPEiLOServerPowerRestoreSetting	Gets the current automatic power- on and power-on-delay settings.	iLO 4, iLO 5
40	Set-HPEiLOServerPowerRestoreSetting	Sets the automatic power-on and power-on-delay settings.	iLO 4, iLO 5
41	Get-HPEiLOServerPower	Gets the power state of the server.	iLO 4, iLO 5
42	Set-HPEiLOServerPower	Toggles the power on the host server.	iLO 4, iLO 5
43	Get-HPEiLOPowerRegulatorSetting	Gets the state of the processor power regulator feature of the server.	iLO 4, iLO 5
44	Set-HPEiLOPowerRegulatorSetting	Sets the state of the processor power regulator feature of the server.	iLO 4, iLO 5
45	Clear-HPEiLOPowerOntime	Clears the virtual clock counter without power-cycling the server.	iLO 4
Access Setting cmdlets			
46	Get-HPEiLOAccessSetting	Gets the iLO access settings.	iLO 4, iLO 5

Table Continued

47	Set-HPEiLOAccessSetting	Sets the iLO access settings.	iLO 4, iLO 5
Remote Syslog Setting cmdlets			
48	Get-HPEiLORemoteSysLog	Gets the iLO Remote Syslog settings.	iLO 4, iLO 5
49	Set-HPEiLORemoteSysLog	Sets the iLO Remote Syslog settings.	iLO 4, iLO 5
50	Send-HPEiLORemoteSysLog	Sends TestSyslog to configured Remote Syslog email address.	iLO 5
AlertMail Setting cmdlets			
51	Get-HPEiLOAlertMailSetting	Gets the AlertMail setting of the server.	iLO 4, iLO 5
52	Set-HPEiLOAlertMailSetting	Modifies AlertMail settings of the host server.	iLO 4, iLO 5
53	Send-HPEiLOTestAlertMail	Sends the test alert mail to the configured mail address.	iLO 5
Virtual Media-related cmdlets			
54	Mount-HPEiLOVirtualMedia	Mounts the specified media image.	iLO 4, iLO 5
55	Dismount-HPEiLOVirtualMedia	Dismounts the Virtual Media image if one is mounted.	iLO 4, iLO 5
56	Get-HPEiLOVirtualMediaStatus	Gets the Virtual Media drive status.	iLO 4, iLO 5
57	Set-HPEiLOVirtualMediaStatus	Sets the virtual media drive status.	iLO 4, iLO 5
Remote Support Setting cmdlets			
58	Register-HPEiLORemoteSupportSetting	Registers a supported server for Insight Online direct connect remote support.	iLO 4, iLO 5
59	Unregister-HPEiLORemoteSupportSetting	Disables Insight Remote Support functionality and unregisters the server.	iLO 4, iLO 5
60	Get-HPEiLORemoteSupportSetting	Gets the Insight Remote Support setting.	iLO 4, iLO 5

Table Continued

61	Set-HPEiLORemoteSupportSetting	Sets the web proxy settings for servers that use Insight Online direct connect remote support.	iLO 4, iLO 5
62	Send-HPEiLORemoteSupportAHSSubmission	Sends Active Health System data submission to the Insight Remote Support server.	iLO 4, iLO 5
63	Send-HPEiLORemoteSupportDataCollectionInfo	Sends a Data collection submission to the Insight Remote Support server.	iLO 4, iLO 5
64	Send-HPEiLORemoteSupportTestEvent	Sends a test service event submission to the Insight Remote Support server.	iLO 4, iLO 5
65	Get-HPEiLORemoteSupportServiceEventLog	Gets the service event logs.	iLO 5
66	Clear-HPEiLORemoteSupportServiceEventLog	Clears all the service event logs.	iLO 5
Login Security Banner cmdlets			
67	Get-HPEiLOLoginSecurityBanner	Gets the iLO Login banner information.	iLO 4, iLO 5
68	Set-HPEiLOLoginSecurityBanner	Sets the security message for the iLO login screen.	iLO 4, iLO 5
Security Encryption cmdlets			
69	Get-HPEiLOEncryptionSetting	Gets the current security encryption settings.	iLO 4, iLO 5
70	Set-HPEiLOEncryptionSetting	Sets the security encryption settings.	iLO 4, iLO 5
AHS Setting cmdlets			
71	Get-HPEiLOAHSStatus	Gets the Active Health System (AHS) status.	iLO 4, iLO 5
72	Set-HPEiLOAHSStatus	Enables or disables AHS logging.	iLO 4, iLO 5
73	Clear-HPEiLOAHSData	Clears the AHS Log.	iLO 4, iLO 5
74	Save-HPEiLOAHSLog	Saves the AHS Log.	iLO 5
Asset Tag Setting cmdlets			
75	Get-HPEiLOAssetTag	Gets the Asset tag information.	iLO 4, iLO 5

Table Continued

76	Set-HPEiLOAssetTag	Sets or clears the asset tag.	iLO 4, iLO 5
Temperature Setting cmdlets			
77	Get-HPEiLOTemperature	Gets the temperature health details of the server	iLO 4, iLO 5
78	Get-HPEiLOCriticalTemperatureAction	Gets the configured critical temperature shutdown behavior of the server.	iLO 4
79	Set-HPEiLOCriticalTemperatureAction	Sets the server to Power On or Power Off after shutting down due to a critical temperature condition.	iLO 4
Boot Mode and Boot Order cmdlets			
80	Get-HPEiLOBootMode	Gets the supported boot mode, current boot mode, and pending boot mode of the server.	iLO 4, iLO 5
81	Set-HPEiLOBootMode	Sets the pending boot mode that will take effect when the server is rebooted.	iLO 4, iLO 5
82	Get-HPEiLOOneTimeBootOption	Gets the current state of the one-time boot.	iLO 4, iLO 5
83	Set-HPEiLOOneTimeBootOption	Sets one time boot order.	iLO 4, iLO 5
84	Get-HPEiLOPersistentBootOrder	Gets the current boot order.	iLO 4, iLO 5
85	Set-HPEiLOPersistentBootOrder	Sets the persistent boot order.	iLO 4, iLO 5
86	Get-HPEiLOBootOption	Gets the boot options available in the server.	iLO5
Directory Setting cmdlet			
87	Add-HPEiLODirectoryGroup	Adds a new Directory Group.	iLO 4, iLO 5
88	Set-HPEiLODirectoryGroup	Modifies the Directory Group user.	iLO 4, iLO 5

Table Continued

89	Get-HPEiLODirectoryGroup	Gets the Directory Group user information.	iLO 4, iLO 5
90	Get-HPEiLODirectorySetting	Gets the Directory Setting detail.	iLO 4, iLO 5
91	Set-HPEiLODirectorySetting	Modifies the directory setting.	iLO 4, iLO 5
92	Start-HPEiLODirectorySettingTest	Starts the Directory Setting Test.	iLO 4, iLO 5
93	Get-HPEiLODirectorySettingTestResult	Gets the Directory Setting Test result.	iLO 4, iLO 5
94	Stop-HPEiLODirectorySettingTest	Stops the Directory Setting Test.	iLO 4, iLO 5
95	Remove-HPEiLODirectoryGroup	Removes the directory group.	iLO 4, iLO 5
96	Get-HPEiLOLDAPCACertificate	Gets LDAP CA certificate from iLO server.	iLO 5
97	Import-HPEiLOLDAPCACertificate	Imports LDAP CA certificate into iLO server.	iLO 5
SSL Certificate cmdlet			
98	Start-HPEiLOCertificateSigningRequest	Starts the certificate signing request.	iLO 4, iLO 5
99	Get-HPEiLOCertificateSigningRequest	Gets the certificate signing request status.	iLO 4, iLO 5
100	Import-HPEiLOCertificate	Imports a signed certificate into iLO.	iLO 4, iLO 5
101	Get-HPEiLOSSLCertificateInfo	Gets the SSL certificate information.	iLO 4, iLO 5
Hot Key Config cmdlets			
102	Get-HPEiLOHotKeyConfig	Gets hot keys available for use in remote console sessions.	iLO 4, iLO 5
103	Set-HPEiLOHotKeyConfig	Configures the remote console hot key settings in iLO.	iLO 4, iLO 5
104	Clear-HPEiLOHotKeyConfig	Clears the remote console hot key settings.	iLO 5
Computer Lock Configuration cmdlets			
105	Get-HPEiLOComputerLockConfiguration	Gets the Computer lock configuration information.	iLO 5

Table Continued

106	Set-HPEiLOComputerLockConfiguration	Configures the computer lock settings.	iLO 4, iLO 5
Persistent Mouse Key Board cmdlets			
107	Get-HPEiLOPersistentMouseKeyboard	Gets the current persistent mouse and keyboard status.	iLO 4, iLO 5
108	Set-HPEiLOPersistentMouseKeyboard	Sets the persistent mouse and keyboard setting.	iLO 4, iLO 5
EncryptKeyManagerSetting cmdlets			
109	Get-HPEiLOESKM	Gets the Enterprise Secure Key Manager (ESKM) setting information.	iLO 4, iLO 5
110	Set-HPEiLOESKM	Sets the communication setting for the ESKM in iLO.	iLO 4, iLO 5
111	Clear-HPEiLOESKMLog	Clears the ESKM log.	iLO 4, iLO 5
112	Test-HPEiLOESKMConnection	Validates the ESKM connection of the iLO.	iLO 4, iLO 5
BrownOut cmdlets			
113	Get-HPEiLOBrownout	Gets the brownout recovery setting of the server.	iLO 5
114	Set-HPEiLOBrownout	Sets the brownout recovery feature on or off.	iLO 4, iLO 5
Module version cmdlets			
115	Get-HPEiLOModuleVersion	Gets the module details for the iLO cmdlets.	iLO 4, iLO 5
Network Setting cmdlets			
118	Get-HPEiLOSNTPSetting	Gets the SNTP settings of the server.	iLO 4, iLO 5
119	Set-HPEiLOSNTPSetting	Sets the SNTP setting.	iLO 4, iLO 5
120	Get-HPEiLOIPv4NetworkSetting	Gets the current iLO IPv4 network setting.	iLO 4, iLO 5
121	Set-HPEiLOIPv4NetworkSetting	Modifies the IPv4 network setting of the host server corresponding to iLO.	iLO 4, iLO 5

Table Continued

122	Get-HPEiLOIPv6NetworkSetting	Gets the current iLO IPv6 network setting.	iLO 4, iLO 5
123	Set-HPEiLOIPv6NetworkSetting	Modifies the IPv6 network setting of the host server corresponding to iLO.	iLO 4, iLO 5
Secure Shell Key cmdlets			
124	Import-HPEiLOUserSSHKey	Imports an SSH Key and associated username into iLO.	iLO 4
125	Remove-HPEiLOUserSSHKey	Removes any SSH keys associated with a particular UserLogin.	iLO 4
License cmdlets			
126	Get-HPEiLOLicense	Gets the license information.	iLO 4, iLO 5
127	Set-HPEiLOLicense	Applies a license key for the Integrated Lights-Out Advanced Pack.	iLO 4, iLO 5
UID Status Setting cmdlets			
128	Get-HPEiLOUIDStatus	Gets the UID Status of the server.	iLO 4, iLO 5
129	Set-HPEiLOUIDStatus	Toggles the UID on the host server.	iLO 4, iLO 5
SMHFQDN cmdlets			
130	Get-HPEiLOSMHFQDN	Gets the System Management Homepage (SMH) fully qualified domain name.	iLO 4
131	Set-HPEiLOSMHFQDN	Sets the fully qualified domain name for SMH and allows it to be placed at a different FQDN or IP address.	iLO 4
Server Info cmdlets			
132	Get-HPEiLOHealthSummary	Gets the health information summary of the host server.	iLO 4, iLO 5
133	Get-HPEiLOServerInfo	Gets the server information detail.	iLO 4, iLO 5
134	Get-HPEiLOMemoryInfo	Gets the memory details for the host server where the iLO is located.	iLO 4, iLO 5

Table Continued

135	Get-HPEiLOProcessor	Gets the processor details for the host server.	iLO 4, iLO 5
136	Get-HPEiLOOAInfo	Gets the Onboard Administrator information from the enclosure where iLO is located.	iLO 4, iLO 5
137	Get-HPEiLONICInfo	Gets the NIC details for the system NIC and the iLO NIC.	iLO 4, iLO 5
138	Get-HPEiLORackSetting	Gets the rack setting information.	iLO 4, iLO 5
139	Get-HPEiLOFan	Gets the fan details from the server.	iLO 4, iLO 5
SNMP cmdlets			
140	Get-HPEiLOSNMPAlertDestination	Gets the SNMP alert destination.	iLO 4, iLO 5
141	Set-HPEiLOSNMPAlertDestination	Modifies the SNMP alert destination.	iLO 4, iLO 5
142	Get-HPEiLOSNMPAlertSetting	Gets the SNMP Alert setting.	iLO 4, iLO 5
143	Set-HPEiLOSNMPAlertSetting	Sets the SNMP alert setting.	iLO 4, iLO 5
144	Get-HPEiLOSNMPSetting	Gets the SNMP setting.	iLO 4, iLO 5
145	Set-HPEiLOSNMPSetting	Sets the SNMP setting.	iLO 4, iLO 5
146	Get-HPEiLOSNMPv3Setting	Gets the SNMPv3 setting.	iLO 4, iLO 5
147	Set-HPEiLOSNMPv3Setting	Sets the SNMPv3 setting.	iLO 4, iLO 5
148	Get-HPEiLOSNMPv3User	Gets the SNMPv3User detail.	iLO 4, iLO 5
149	Set-HPEiLOSNMPv3User	Sets the SNMPv3User detail.	iLO 4, iLO 5
150	Add-HPEiLOSNMPAlertDestination	Adds the respective iLO SNMP Alert Destination.	iLO 5
151	Add-HPEiLOSNMPv3User	Adds the SNMPv3User in the iLO.	iLO 5
152	Remove-HPEiLOSNMPAlertDestination	Removes the SNMP Alert Destination.	iLO 5
153	Remove-HPEiLOSNMPv3User	Removes the SNMPv3 user.	iLO 5
154	Invoke-HPEiLOSNMPTestTrap	Sends a test SNMP trap to the configured alert destinations.	iLO 4, iLO 5

Table Continued

Kerberos Config Setting cmdlets			
155	Get-HPEiLOKerberosConfig	Gets the Kerberos configuration detail.	iLO 4, iLO5
156	Set-HPEiLOKerberosConfig	Configures the Kerberos authentication.	iLO 4, iLO5
Profile cmdlets			
157	Get-HPEiLOProfile	Gets all the Profile Descriptors and the data stored in them in the permanent (perm) directory of the blob store.	iLO 4
158	Invoke-HPEiLOProfileApply	Applies a deployment setting profile in iLO.	iLO 4
159	Invoke-HPEiLOProfileDownload	Modifies a Profile Description, downloads a specific blob, and writes the blob in permanent (perm) and temporary(tmp) directory of the blob store.	iLO 4
160	Remove-HPEiLOProfile	Removes a deployment profile.	iLO 4
161	Get-HPEiLOProfileApplyResult	Gets the result of the Invoke-HPEiLOProfileApply cmdlet.	iLO 4
162	Get-HPEiLOBatteryBackUpUnit	Gets the battery back up unit detail.	iLO 5
163	Set-HPEiLOBatteryBackUpUnit	Sets the battery back up unit.	iLO 5
Post Setting cmdlets			
164	Get-HPEiLOPostSetting	Gets the post setting detail.	iLO 5
165	Set-HPEiLOPostSetting	Sets the Post setting detail.	iLO 5
Processor Jitter Control cmdlets			
166	Get-HPEiLOProcessorJitterControl	Gets the Processor Jitter Control detail.	iLO 5
167	Set-HPEiLOProcessorJitterControl	Sets the Processor Jitter control.	iLO 5
Secure boot cmdlets			

Table Continued

168	Get-HPEiLOSecureBoot	Gets the secure boot detail.	iLO 5
169	Set-HPEiLOSecureBoot	Enables the secure boot mode.	iLO 5
170	Reset-HPEiLOSecureBootKey	Resets the SecureBootKey value.	iLO 5
Service Port cmdlets			
171	Get-HPEiLOServicePort	Gets the service port detail.	iLO 5
172	Set-HPEiLOServicePort	Sets the service port detail.	iLO 5
Rest API State cmdlets			
173	Get-HPEiLORestApiState	Gets the Rest API State.	iLO 5
174	Clear-HPEiLORestApiState	Clears the persistent state of the REST API.	iLO 5
Logging cmdlets			
175	Enable-HPEiLOLog	Enables iLO cmdlets logging for the current PowerShell session.	iLO 4, iLO 5
176	Disable-HPEiLOLog	Disables iLO cmdlets logging for the current PowerShell session.	iLO 4, iLO 5
177	Clear-HPEiLOLog	Clears any logs created by the iLO cmdlets module.	iLO 4, iLO 5
178	Get-HPEiLOLogConfig	Gets the existing log configuration settings.	iLO 4, iLO 5
179	Set-HPEiLOLogConfig	Modifies the existing log configuration settings.	iLO 4, iLO 5
Device Inventory cmdlets			
180	Enable-HPEiLOMCTP	Enables MCTP recovery.	iLO 5
181	Disable-HPEiLOMCTP	Disables MCTP recovery.	iLO 5
182	Get-HPEiLODeviceInventory	Gets the Device inventory detail.	iLO 5
Install Set Setting cmdlets			
183	Get-HPEiLOInstallSet	Gets the install set detail.	iLO 5

Table Continued

184	Add-HPEiLOInstallSet	Creates an install set.	iLO 5
185	Remove-HPEiLOInstallSet	Removes the install set.	iLO 5
186	Clear-HPEiLOInstallSet	Clears the Install set.	iLO 5
187	Invoke-HPEiLOInstallSet	Invokes the install set to add on to the installation queue.	iLO 5
Installation queue cmdlets			
188	Get-HPEiLOInstallationTaskQueue	Gets the Installation Task Queue detail.	iLO 5
189	Remove-HPEiLOInstallationTaskQueue	Removes the specified task from Installation Task queue.	iLO 5
190	Clear-HPEiLOInstallationTaskQueue	Clears the Installation queue.	iLO 5
191	Add-HPEiLOInstallationQueueWaitTask	Adds a wait task in the installation queue.	iLO 5
Repository Component Setting cmdlets			
192	Get-HPEiLORepositoryComponent	Gets the repository component detail.	iLO 5
193	Add-HPEiLORepositoryComponent	Adds the repository component to the iLO repository.	iLO 5
194	Invoke-HPEiLORepositoryComponent	Invokes the repository component and adds it to the installation queue.	iLO 5
195	Remove-HPEiLORepositoryComponent	Removes the repository component.	iLO 5
196	Set-HPEiLORepositoryComponent	Sets the existing repository component information settings.	iLO 5
197	Clear-HPEiLOUnlockedRepositoryComponent	Clears all the unlocked repository components.	iLO 5
Firmware cmdlets			
198	Update-HPEiLOFirmware	Updates the firmware in iLO.	iLO 4, iLO 5
199	Get-HPEiLOFirmwareInventory	Gets the firmware detail.	iLO 4, iLO 5
200	Get-HPEiLOFirmwareVersion	Gets the iLO firmware information.	iLO 4, iLO 5

Table Continued

201	Get-HPEiLOFirmwarePolicy	Gets the firmware downgrade policy information.	iLO 5
202	Set-HPEiLOFirmwarePolicy	Modifies the firmware downgrade policy settings.	iLO 5
Smart Storage cmdlets			
203	Get-HPEiLOSmartStorageBattery	Gets the smart storage battery detail.	iLO 4, iLO 5
204	Get-HPEiLOSmartArrayStorageController	Gets the Smart Array Storage Controller detail.	iLO 4, iLO 5
Reset cmdlet			
205	Reset-HPEiLO	Resets the iLO or server.	iLO 4, iLO 5
Disable iLO cmdlet			
206	Disable-HPEiLOFunctionality	Disables the iLO functionality.	iLO 4, iLO 5
Factory default cmdlet			
207	Set-HPEiLOFactoryDefault	Sets the Integrated Lights-Out device to factory default settings.	iLO 4, iLO 5
Miscellaneous cmdlets			
208	Get-HPEiLOTPMStatus	Gets the HPE Trusted Platform Module Information.	iLO 4, iLO 5
209	Get-HPEiLOIntelligentProvisioningInfo	Gets the Intelligent Provisioning info	iLO 5
210	Get-HPEiLOPCIDeviceInventory	Gets the PCI device Inventory information.	iLO 5
211	Get-HPEiLOPCISlot	Gets the PCI slot information.	iLO 5
212	Get-HPEiLOPhysicalSecurity	Gets the Physical security detail.	iLO 5
213	Get-HPEiLODeviceDiscoveryStatus	Gets the device discovery status of the iLO.	iLO 5
214	Get-HPEiLOSelfTestResult	Gets the Self-Test Result detail.	iLO 5
215	Get-HPEiLOServerSoftwareInventory	Gets the server HPE software inventory detail.	iLO 5

Table Continued

216	Get-HPEiLOSessionInfo	Gets the Session detail.	iLO 5
217	Get-HPEiLOSystemMaintenanceSwitch	Gets the system maintenance switch information.	iLO 5
218	Get-HPEiLOUSBDevice	Gets the USB device detail.	iLO 5
219	Get-HPEiLOUserCertificateMapping	Gets the User Certificate Mapping detail.	iLO 5
220	Get-HPEiLOCACSmartCardAuthentication	Gets the CAC smart card detail.	iLO 5
221	Get-HPEiLORedfishMessageInfo	Gets the Redfish message detail.	iLO 5
222	Get-HPEiLOEmbeddedMedia	Gets the embedded media information.	iLO 4, iLO 5
223	Get-HPEiLOSpatial	Gets the location information and system data with Asset Manager to obtain more precise and complete asset data.	iLO 4
224	Get-HPEiLOHostData	Gets the host data displayed on the Server Information page.	iLO 4, iLO 5
225	Read-HPEiLOSMBIOSRecord	Decodes the server data and displays SMBIOS information.	iLO 4, iLO 5
226	Get-HPEiLOInvalidFirmwareImage	Gets the Invalid firmware image details.	iLO 5
227	Get-HPEiLOInfo	Gets the unauthenticated details of the target iLO.	iLO 5
228	Get-HPEiLOChassisInfo	Gets the comprehensive target chassis information.	iLO 5
229	Get-HPEiLOSystemInfo	Gets the comprehensive target system information.	iLO 5
230	Get-HPEiLOMaxThreadLimit	Gets the maximum thread that can be spawned by the cmdlets of the HPEiLOCmdlets module.	iLO 4, iLO 5
231	Set-HPEiLOMaxThreadLimit	Sets the maximum thread that can be spawned by the cmdlets of the HPEiLOCmdlets module.	iLO 4, iLO 5

Table Continued

232	Get-HPEiLOPerformanceMonitoringData	Gets complete performance monitoring data from all sensors of iLO.	iLO 5
233	Get-HPEiLOPerformanceAlertSetting	Gets the set of thresholds defined for a sensor.	iLO 5
234	Set-HPEiLOPerformanceAlertSetting	Modifies the lower or upper threshold for the specified sensor.	iLO 5
235	Get-HPEiLOServerWorkloadInfo	Gets the list of workload characteristics for metrics.	iLO 5
236	Get-HPEiLOSecurityDashboardInfo	Gets security dashboard information of an iLO server.	iLO 5
237	Enable-HPEiLOSecurityDashboardSetting	Enables Security dashboard Ignore flags.	iLO 5
238	Disable-HPEiLOSecurityDashboardSetting	Disables Security dashboard Ignore flags.	iLO 5
239	Get-HPEiLOPerformanceTuningSetting	Gets a list of BIOS attribute settings with recommended values as a part of workload performance advisor.	iLO 5
240	Invoke-HPEiLOSecureSystemErase	Resets the system BIOS settings and iLO to manufacturing defaults. It also erases the user data on the system.	iLO 5
241	Get-HPEiLOSecureSystemEraseStatus	Gets the overall erase status of System ROM, iLO, and User Data.	iLO 5
Backup and Restore cmdlets			
242	Backup-HPEiLOSetting	Backs up the current iLO setting in a .bak file or on a NAND.	iLO 5
243	Restore-HPEiLOSetting	Restores the iLO setting from a .bak file or from NAND.	iLO 5
Firmware Verification Scan cmdlets			
244	Get-HPEiLOFirmwareVerificationScanSetting	Gets the firmware verification scan setting.	iLO 5
245	Set-HPEiLOFirmwareVerificationScanSetting	Modifies the firmware verification scan setting.	iLO 5

Table Continued

246	Invoke-HPEiLOFirmwareVerificationScan	Invokes the firmware verification scan run.	iLO 5
247	Get-HPEiLOFirmwareVerificationLastScanResult	Gets the firmware verification scan result.	iLO 5
248	Send-HPEiLOSystemRecoveryEvent	Generates a recovery event that triggers a separate management tool to initiate a full system recovery.	iLO 5
Maintenance Window Cmdlets			
249	Add-HPEiLOMaintenanceWindow	Adds the maintenance window with time period for installation task.	iLO 5
250	Get-HPEiLOMaintenanceWindow	Gets the maintenance window list.	iLO 5
251	Set-HPEiLOMaintenanceWindow	Sets the maintenance window.	iLO 5
252	Remove-HPEiLOMaintenanceWindow	Deletes the selected maintenance window.	iLO 5
253	Clear-HPEiLOMaintenanceWindow	Deletes all the maintenance windows.	iLO 5
IRC Setting Cmdlets			
254	Get-HPEiLOIntegratedRemoteConsoleTrustSetting	Gets the Integrated Remote Console (IRC) settings.	iLO 5
255	Set-HPEiLOIntegratedRemoteConsoleTrustSetting	Sets the Integrated Remote Console (IRC) settings.	iLO 5
Chassis Power Setting cmdlets			
256	Get-HPEiLOPowerMeter	Gets both power meter and fast power meter sample details with count.	iLO 5
257	Get-HPEiLOChassisPowerCalibrationData	Gets the detailed calibration data of the Apollo chassis server.	iLO 5
258	Get-HPEiLOChassisPowerCapSetting	Gets the Apollo chassis power capping settings.	iLO 5
259	Set-HPEiLOChassisPowerCapSetting	Sets the Apollo chassis power capping settings.	iLO 5

Table Continued

260	Get-HPEiLOChassisPowerNodeInfo	Gets the node configuration for each node of an Apollo server.	iLO 5
261	Get-HPEiLOChassisPowerZoneConfiguration	Gets the power zone configuration of an Apollo server.	iLO 5
262	Set-HPEiLOChassisPowerZoneConfiguration	Sets the Apollo chassis server power zone configuration.	iLO 5
263	Get-HPEiLOChassisPowerRegulatorSetting	Gets Apollo chassis power regulator settings.	iLO 5
264	Set-HPEiLOChassisPowerRegulatorSetting	Sets Apollo chassis power regulator settings.	iLO 5
265	Start-HPEiLOChassisPowerCalibrationConfiguration	Starts the Apollo chassis power calibration configuration	iLO 5
System Diagnostics Cmdlets			
266	Reset-HPEiLOSystemManufacturingDefault	Resets all BIOS configuration settings to default manufacturing values.	iLO5 Gen10 Plus
267	Reset-HPEiLOSystemDefault	Resets all BIOS configuration settings to their default values.	iLO5 Gen10 Plus
268	Enable-HPEiLOSystemIntelligentDiagnosticsMode	Enables the Intelligent Diagnostics Mode.	iLO5 Gen10 Plus
269	Enable-HPEiLOSystemSafeMode	Enables the server to boot in safe mode.	(iLO5 Gen10 Plus
270	Get-HPEiLOSystemDiagnosticsModeSetting	Gets the system diagnostics mode setting.	(iLO5 Gen10 Plus

Using the `Find-HPEiLO` cmdlet

When learning about the iLO cmdlets, a good place to start is with the `Find-HPEiLO` cmdlet. This cmdlet scans IP addresses and finds iLOs that exist within the specified range. The `Range` parameter can be a single IP address, a subnet list, or a range of IP addresses. When the command finds an iLO, it obtains basic information about the iLO without requiring a username or password. This can be useful for performing a quick inventory within a data center, or perhaps determining what firmware versions exist. The information is returned as a single object or as an array of objects of iLOs found.

Example 1

The following is an example of using `Find-HPEiLO`.

```
PS C:\> Find-HPEiLO 10.20.30.1
IP                    : 10.20.30.1
Hostname              : abc.domain.com
SPN                   : ProLiant DL380 Gen8
```

```
FWRI                    : 2.55
PN                      : Integrated Lights-Out 4 (iLO 4)
SerialNumber            : CN761109QP
cUUID                   : 37303137-3332-4E43-3736-313130395150
```

Example 2

The following is an example of using `Find-HPEiLO` with single IP and the `Full` parameter.

```
PS C:\> $output = Find-HPEiLO 10.20.30.1 -Full

PS C:\> $output

IP : 10.20.30.1
Hostname                : abc.domain.com
HostSystemInformation : HPE.iLO.Response.HSI
ManagementProcessor   : HPE.iLO.Response.MP
BladeSystem             : HPE.iLO.Response.BladeSystem
Spatial                 : HPE.iLO.Response.RIMPSpatial
Health                  : HPE.iLO.Response.RIMPHealth

PS C:\> $output.HostSystemInformation

PS C:\> $output.HostSystemInformation
SerialNumber            : CN761109QP
SPN                     : ProLiant DL385p Gen8
UUID                    : 710723CN761109QP
SP                      : 1
cUUID                   : 37303137-3332-4E43-3736-313130395150
Virtual                 : HPE.iLO.Response.RIMP_HSI_Virtual
ProductID               : 710723-371
```

Example 3

The following is an example of using `Find-HPEiLO` with a search range which checks 11 addresses, in which 3 iLOs are found:

```
PS C:\> Find-HPEiLO 10.20.30.1-11
IP                      : 10.20.30.1
Hostname                : abc.domain.com
SPN                     : ProLiant DL380 Gen8
FWRI                    : 2.55
PN                      : Integrated Lights-Out 4 (iLO 4)
SerialNumber            : CN761109QS
cUUID                   : 37303137-3332-4E43-3736-31313039515


IP                      : 10.20.30.5
Hostname                : abc5.domain.com
SPN                     : ProLiant ML350 Gen9
FWRI                    : 2.44
PN                      : Integrated Lights-Out 4 (iLO 4)
SerialNumber            : SGH611WL3S
cUUID                   : 35343537-3433-4753-4836-3131574C335


IP                      : 10.20.30.6
Hostname                : abc6.domain.com
SPN                     : ProLiant DL360 Gen10
FWRI                    : 1.20
PN                      : Integrated Lights-Out 5 (iLO 5)
```

```
SerialNumber          : CN76370GNS
cUUID                 : 36334C44-2D30-4E43-3736-333730474E4
```

Example 4

The following is a script that pipes output from `Find-HPEiLO` through Add-Member to add two required fields, and then to `Connect-HPEiLO` that connects to the reachable iLOs.

```
PS C:\>$connection = Find-HPEiLO 10.20.30.1-15 |

• {Add-Member -PassThru -InputObject $_ Username admin}|

% {Add-Member -PassThru -InputObject $_ Password admin123}| Connect-HPEiLO

PS C:/>$connection


IP                    : 10.20.30.40
Hostname              :
Timeout               : 30
DisableCertificate
Authentication        : True
TargetInfo            : @{ProductName=ProLiant ML350 Gen9;
                        ServerFamily=ProLiant;
                        ServerGeneration=Gen9;
                        ServerModel=ML350;
                        iLOGeneration=iLO4;
                        iLOFirmwareVersion=2.62;
                        SystemROM=P92v2.72 (03/25/2019);
                        ProcessorName=Intel(R) Xeon(R)
                        CPU E5-2603 v3 @ 1.60GHz}
ExtendedInfo          : @{UserSuppliedAddress=10.20.30.40;
                        HttpConnectAddress=10.20.30.40;
                        UserName=admin;
                        Modifier1=vyUuq0Oh/
                        XOc1n+Vv4w1YcFF+FvdoVk/FUEH2cdAD/4=;
                        Modifier2=4eA+yvA4kyRK8WYwIeT7zA==}
ConnectionInfo        : @{RIBCL=; REST=}


IP                    : 10.20.30.40
Hostname              : xyz.abcd.com
Timeout               : 30
DisableCertificate
Authentication        : True
TargetInfo            : @{ProductName=ProLiant DL580 Gen10;
                        ServerFamily=ProLiant;
                        ServerGeneration=Gen10;
                        ServerModel=DL580;
                        iLOGeneration=iLO5;
                        iLOFirmwareVersion=2.1;
                        SystemROM=U34 v2.16 (09/12/2019);
                        ProcessorName=Intel(R) Genuine processor}
ExtendedInfo          : @{UserSuppliedAddress=10.20.30.40;
                        HttpConnectAddress=10.20.30.40;
                        UserName=admin;
                        Modifier1=b3JJfwxwbZSKKyjpgKmn3
                        QPj16vDZ7Q3oq9ejPZCOPk=;
                        Modifier2=cqEhobrR75LcegppbRmPWQ==}
ConnectionInfo        : @{Redfish=}
```

To monitor the operation of the `Find-HPEiLO` cmdlet, use the `Verbose` parameter. The default value for the `Timeout` parameter is 30 seconds. If the timeout value is not long enough for iLOs to respond, try using a `Timeout` parameter with a larger value.

In the preceding 4 commands, no double quotes are required around the `Range` parameter. If a comma is included in the range, double quotes are required. PowerShell interprets a comma (,) as a list separator. Without double quotes ("), part of what should be a string is interpreted by PowerShell as a number. The operation of combined ranges is defined as creating a combination of each subnet address with each other subnet. The following are examples of input range parameters using double quotes.

Range Parameter	Description
"10.20.30.1,15"	Specifies two addresses to check: 10.20.30.1 and 10.20.30.15.
"10.20.30,29.93,103"	Specifies four addresses to check: 10.20.30.93,10.20.30.103,10.20.29.93, and 10.20.29.103.
"10.20.30,29.93-103"	Specifies 22 addresses to check: 10.20.30.93 through 10.20.30.103 and 10.20.29.93 through 10.20.29.103.

Establishing an iLO connection

The iLO cmdlets supports the following generations of HPE ProLiant servers. Each generation of servers has a different connection mechanism.

- iLO 4 servers
- iLO 5 servers

Prerequisites

- Install iLO cmdlets on the Windows management client.
- Make sure that the iLO IP address of the target server is able to ping from the management client where the iLO cmdlets are installed.
- The target iLO must have a valid server certificate. If a valid certificate is not available, use the `DisableCertificateAuthentication` switch parameter to establish the connection.

Establishing a connection to an iLO 4 server

Use this mode of connection for iLO 4 servers. It uses the iLO IP address and iLO user credentials.

Procedure

1. Execute `Connect-HPEiLO` with the iLO IP address of the Gen8 or Gen9 target server. A successful connection will return the connection object.
2. Use the object from the previous step to run iLO cmdlets.

Example: Connecting to an iLO 4 server that does not have a valid server certificate

```
PS C:\> $connection = Connect-HPEiLO -IP 10.20.30.40 -Username admin -
Password admin123 -DisableCertificateAuthentication
PS C:\> $connection


IP                  : 10.20.30.40
Hostname            :
Timeout             : 30
DisableCertificate
Authentication      : True
TargetInfo          : @{ProductName=ProLiant ML350 Gen9;
                      ServerFamily=ProLiant;
                      iLOFirmwareVersion=2.62; SystemROM=P92
                      v2.72 (03/25/2019);
                      ProcessorName=Intel(R) Xeon(R)
                      CPU E5-2603 v3 @ 1.60GHz}
ExtendedInfo        : @{UserSuppliedAddress=10.20.30.40;
                      HttpConnectAddress=10.20.30.40;
                      UserName=admin;
                      Modifier1=vyUuq0Oh/XOc1n+Vv4w1YcFF+FvdoVk/
                      FUEH2cdAD/4=;
                      Modifier2=4eA+yvA4kyRK8WYwIeT7zA==}
ConnectionInfo      : @{RIBCL=; REST=}


PS C:\> $connection.TargetInfo


ProductName          : ProLiant ML350 Gen9
ServerFamily         : ProLiant
ServerGeneration     : Gen9
ServerModel          : ML350
iLOGeneration        : iLO4
iLOFirmwareVersion   : 2.62
SystemROM            : P92 v2.72 (03/25/2019)
ProcessorName        : Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60GHz


PS C:\> $connection.ExtendedInfo


UserSuppliedAddress : 10.20.30.40
HttpConnectAddress  : 10.20.30.40
UserName            : admin
Modifier1           : vyUuq0Oh/XOc1n+Vv4w1YcFF+FvdoVk/FUEH2cdAD/4=
Modifier2           : 4eA+yvA4kyRK8WYwIeT7zA==


PS C:\> $connection.ConnectionInfo | fl


RIBCL : @{AuthToken=ciNRS6xqkr9VO6JV7f1LQQ==; iLOResetWaitTimeout=0}
REST  : @{RootUrl=https://10.20.30.40/rest/v1;
        XAuthToken=iOIck9yMeVbdoNJaD4Zh1xUHMa1CsiQpP1
        MHZls5BpfkG1yMFYgjEAEdZFfUm0Ng;
```

```
Location=https://10.20.30.40/rest/v1/
SessionService/Sessions/admin5e385e3ea6a7ef9e;
BIOSAdminPassword=; BIOSAttributeRegistryName
=HpBiosAttributeRegistryP92.1.1.72}
```

Establishing a connection to an iLO 5 server

Use this mode of connection for iLO 5 servers. It uses the iLO IP address and iLO user credentials.

Procedure

1. Execute `Connect-HPEiLO` with the iLO IP address of the Gen10 target server. A successful connection will return the session object.
2. Use the session object from the previous step to run the iLO cmdlets.

Example: Connecting to an iLO 5 server that has a valid server certificate

```
PS C:\> $connection = Connect-HPEiLO -IP 10.20.30.40
-Username admin -Password admin123
PS C:\> $connection

IP                  : 10.20.30.40
Hostname            : xyz.abcd.com
Timeout             : 30
DisableCertificate
Authentication      : True
TargetInfo          : @{ProductName=ProLiant DL580 Gen10;
                      ServerFamily=ProLiant;
                      ServerGeneration=Gen10;
                      ServerModel=DL580;
                      iLOGeneration=iLO5;
                      iLOFirmwareVersion=2.1;
                      SystemROM=U34 v2.16 (09/12/2019);
                      ProcessorName=Intel(R) Genuine processor}
ExtendedInfo        : @{UserSuppliedAddress=10.20.30.40;
                      HttpConnectAddress=10.20.30.40;
                      UserName=admin;
                      Modifier1=b3JJfwxwbZSKKyjpg
                      Kmn3QPj16vDZ7Q3oq9ejPZCOPk=;
                      Modifier2=cqEhobrR75LcegppbRmPWQ==}
ConnectionInfo      : @{Redfish=}

PS C:\> $connection.TargetInfo

ProductName         : ProLiant DL580 Gen10
ServerFamily        : ProLiant
ServerGeneration    : Gen10
ServerModel         : DL580
iLOGeneration       : iLO5
iLOFirmwareVersion  : 2.1
SystemROM           : U34 v2.16 (09/12/2019)
ProcessorName       : Intel(R) Genuine processor

PS C:\> $connection.ExtendedInfo

UserSuppliedAddress : 192.168.10.55
HttpConnectAddress  : xyz.abcd.com
```

```
UserName             : admin
Modifier1            : b3JJfwxwbZSKKyjpgKmn
                       3QPj16vDZ7Q3oq9ejPZCOPk=
Modifier2            : cqEhobrR75LcegppbRmPWQ==

PS C:\> $connection.ConnectionInfo

Redfish


-------


@{ResourceDirectoryJSON={"@odata.context":"/redfish/v1/
 $metadata#HpeiLOResourceDirectory.HpeiLOResourceDirectory",
 "@odata.etag":"W/\"9B6CB9C6\"", "@odata.id":"/...
```

IPv6 support

Consider the following when using IPv6.

- 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 are supported with the comma character. For example, `1a00,1b00::1fe8` resolves to two addresses, `1a00::1fe8` and `1b00::1fe8`.

Example: Connect-HPEiLO using IPv6 address

```
PS C:\> $connection = Connect-HPEiLO
-IP  FE80::9618:82FF:FE03:3A70 -Username
admin -Password aduser1234 -DisableCertificateAuthentication
PS C:\> $connection

IP                   : fe80:0:0:0:9618:82ff:fe03:3a70
Hostname             :
Timeout              : 30
DisableCertificate
Authentication       : True
TargetInfo           : @{ProductName=ProLiant ML350 Gen9;
                       ServerFamily=ProLiant;
                       ServerGeneration=Gen9;
                       ServerModel=ML350; iLOGeneration=iLO4;
                       iLOFirmwareVersion=2.62;
                       SystemROM=P92v2.72 (03/25/2019);
                       ProcessorName=Intel(R) Xeon(R)
                       CPU E5-2603 v3 @ 1.60GHz}
ExtendedInfo         : @{UserSuppliedAddress
                       =fe80:0:0:0:9618:82ff:fe03:3a70;
                       HttpConnectAddress
                       =fe80:0:0:0:9618:82ff:fe03:3a70;
                       UserName=admin;
```

```
                      Modifier1=vyUuq0Oh/XOc1n+Vv4w1YcFF+FvdoVk/
                      FUEH2cdAD/4=;
                      Modifier2=4eA+yvA4kyRK8WYwIeT7zA==}
ConnectionInfo      : @{RIBCL=; REST=}

PS C:\> $connection.TargetInfo

ProductName         : ProLiant ML350 Gen9
ServerFamily        : ProLiant
ServerGeneration    : Gen9
ServerModel         : ML350
iLOGeneration       : iLO4
iLOFirmwareVersion  : 2.62
SystemROM           : P92 v2.72 (03/25/2019)
ProcessorName       : Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60GHz

PS C:\> $connection.ExtendedInfo

UserSuppliedAddress : fe80:0:0:0:9618:82ff:fe03:3a70
HttpConnectAddress  : fe80:0:0:0:9618:82ff:fe03:3a70
UserName            : admin
Modifier1           : vyUuq0Oh/XOc1n+Vv4w1YcFF+FvdoVk/FUEH2cdAD/4=
Modifier2           : 4eA+yvA4kyRK8WYwIeT7zA==

PS C:\> $connection.ConnectionInfo | fl

RIBCL : @{AuthToken=ciNRS6xqkr9VO6JV7f1LQQ==; iLOResetWaitTimeout=0}
REST  : @{RootUrl=https://fe80:0:0:0:9618:82ff:fe03:3a70/rest/v1;
        XAuthToken=iOIck9yMeVbdoNJaD4Zh1xUHMa1CsiQpP1
        MHZls5BpfkG1yMFYgjEAEdZFfUm0Ng;
        Location=https://fe80:0:0:0:9618:82ff:fe03:3a70/
        rest/v1/SessionService/Sessions/
        admin5e385e3ea6a7ef9e; BIOSAdminPassword=;
        BIOSAttributeRegistryName=HpBiosAttributeRegistryP92.1.1.72}
```

Example: Connect-HPEiLO using IPv6 address with port number

```
PS C:\> $connection = Connect-HPEiLO
-IP [fe80:0:0:0:3ea8:2aff:fe19:abc2]:
100 -Username admin -Password admin123
PS C:\> $connection

IP                  : [fe80:0:0:0:3ea8:2aff:fe19:abc2]:100
Hostname            :
Timeout             : 30
DisableCertificate
Authentication      : True
TargetInfo          : @{ProductName=ProLiant DL580 Gen10;
                      ServerFamily=ProLiant;
                      ServerGeneration=Gen10;
                      ServerModel=DL580; iLOGeneration=iLO5;
                      iLOFirmwareVersion=2.1;
                      SystemROM=U34 v2.16 (09/12/2019);
                      ProcessorName=Intel(R)
                      Genuine processor}
ExtendedInfo        : @{UserSuppliedAddress
                      =[fe80:0:0:0:3ea8:2aff:fe19:abc2]:100;
                      HttpConnectAddress
```

```
                      =[fe80:0:0:0:3ea8:2aff:fe19:abc2]:100;
                      UserName=admin;
                      Modifier1=b3JJfwxwbZSKKyjpg
                      Kmn3QPj16vDZ7Q3oq9ejPZCOPk=;
                      Modifier2=cqEhobrR75LcegppbRmPWQ==}
ConnectionInfo      : @{Redfish=}


PS C:\> $connection.TargetInfo

ProductName         : ProLiant DL580 Gen10
ServerFamily        : ProLiant
ServerGeneration    : Gen10
ServerModel         : DL580
iLOGeneration       : iLO5
iLOFirmwareVersion  : 2.1
SystemROM           : U34 v2.16 (09/12/2019)
ProcessorName       : Intel(R) Genuine processor


PS C:\> $connection.ExtendedInfo

UserSuppliedAddress : [fe80:0:0:0:3ea8:2aff:fe19:abc2]:100
HttpConnectAddress  :
UserName            : admin
Modifier1           : b3JJfwxwbZSKKyjpgKmn3QPj16vDZ7Q3oq9ejPZCOPk=
Modifier2           : cqEhobrR75LcegppbRmPWQ==

PS C:\> $connection.ConnectionInfo

Redfish
-------
@{ResourceDirectoryJSON={"@odata.context":"/redfish/v1/
 $metadata#HpeiLOResourceDirectory.HpeiLOResourceDirectory",
 "@odata.etag":"W/\"9B6CB9C6\"","@odata.id":"/...
```

For more information on IPv6, see the following website or the references it links to: **http://en.wikipedia.org/wiki/IPv6**

Hostname support

Example: Connect-HPEiLO using hostname

```
PS C:\> $connection = Connect-HPEiLO
-IP dl360gen9-01.abcd.com -Username admin
-Password aduser1234 -DisableCertificateAuthentication
PS C:\> $connection

IP                                : 10.20.30.40
Hostname                          : dl360gen9-01.abcd.com
Timeout                           : 30
DisableCertificateAuthentication : True
TargetInfo                        : @{ProductName=ProLiant ML350 Gen9;
                                    ServerFamily=ProLiant;
                                    ServerGeneration=Gen9;
                                    ServerModel=ML350;
                                    iLOGeneration=iLO4;
                                    iLOFirmwareVersion=2.62;
```

```
                                    SystemROM=P92 v2.72
                                    (03/25/2019);
                                    ProcessorName=Intel(R)
                                    Xeon(R) CPU E5-2603 v3 @ 1.60GHz}
ExtendedInfo                      : @{UserSuppliedAddress
                                    =dl360gen9-01.abcd.com;
                                    HttpConnectAddress
                                    =dl360gen9-01.abcd.com;
                                    UserName=admin;
                                    Modifier1=vyUuq0Oh/
                                    XOc1n+Vv4w1YcFF+FvdoVk/
                                    FUEH2cdAD/4=;
                                    Modifier2=4eA+yvA4kyRK8WYwIeT7zA==}
ConnectionInfo                    : @{RIBCL=; REST=}

PS C:\> $connection.TargetInfo

ProductName                       : ProLiant ML350 Gen9
ServerFamily                      : ProLiant
ServerGeneration                  : Gen9
ServerModel                       : ML350
iLOGeneration                     : iLO4
iLOFirmwareVersion                : 2.62
SystemROM                         : P92 v2.72 (03/25/2019)
ProcessorName                     : Intel(R) Xeon(R) CPU
                                    E5-2603 v3 @ 1.60GHz

PS C:\> $connection.ExtendedInfo

UserSuppliedAddress               : dl360gen9-01.abcd.com
HttpConnectAddress                : dl360gen9-01.abcd.com
UserName                          : admin
Modifier1                         : vyUuq0Oh/XOc1n+Vv4w1YcFF+FvdoVk/
                                    FUEH2cdAD/4=
Modifier2                         : 4eA+yvA4kyRK8WYwIeT7zA==

PS C:\> $connection.ConnectionInfo | fl

RIBCL : @{AuthToken=ciNRS6xqkr9VO6JV7f1LQQ==;
        iLOResetWaitTimeout=0}
REST  : @{RootUrl=https://fe80:0:0:0:9618:82ff:fe03:3a70/rest/v1;
        XAuthToken=iOIck9yMeVbdoNJaD4Zh1xUHMa1CsiQpP1MHZ
        ls5BpfkG1yMFYgjEAEdZFfUm0Ng;
        Location=https://fe80:0:0:0:9618:82ff:fe03:3a70/
        rest/v1/SessionService/Sessions/admin5e385e3ea6a7ef9e;
        BIOSAdminPassword=;
        BIOSAttributeRegistryName
        =HpBiosAttributeRegistryP92.1.1.72}
```

Example: Connect-HPEiLO using hostname with port number

```
PS C:\> $connection

IP                                : 10.20.30.40:100
Hostname                          : dl360gen10-01.abcd.com:100
Timeout                           : 30
DisableCertificateAuthentication : True
TargetInfo                        : @{ProductName=ProLiant
```

```
                                    DL580 Gen10;
                                    ServerFamily=ProLiant;
                                    ServerGeneration=Gen10;
                                    ServerModel=DL580;
                                    iLOGeneration=iLO5;
                                    iLOFirmwareVersion=2.1;
                                    SystemROM=U34 v2.16 (09/12/2019);
                                    ProcessorName=Intel(R)
                                    Genuine processor}
ExtendedInfo                      : @{UserSuppliedAddress
                                    =dl360gen10-01.abcd.com:100;
                                    HttpConnectAddress
                                    =dl360gen10-01.abcd.com:100;
                                    UserName=admin;
                                    Modifier1=b3JJfwxwbZSKKyjpgKmn
                                    3QPj16vDZ7Q3oq9ejPZCOPk=;
                                    Modifier2=cqEhobrR75LcegppbRmPWQ==}
ConnectionInfo                    : @{Redfish=}

PS C:\> $connection.TargetInfo

ProductName                       : ProLiant DL580 Gen10
ServerFamily                      : ProLiant
ServerGeneration                  : Gen10
ServerModel                       : DL580
iLOGeneration                     : iLO5
iLOFirmwareVersion                : 2.1
SystemROM                         : U34 v2.16 (09/12/2019)
ProcessorName                     : Intel(R) Genuine processor

PS C:\> $connection.ExtendedInfo

UserSuppliedAddress               : dl360gen10-01.abcd.com:100
HttpConnectAddress                : dl360gen10-01.abcd.com:100
UserName                          : admin
Modifier1                         : b3JJfwxwbZSKKyjpgKmn3QPj1
                                    6vDZ7Q3oq9ejPZCOPk=
Modifier2                         : cqEhobrR75LcegppbRmPWQ==

PS C:\> $connection.ConnectionInfo

Redfish


-------


@{ResourceDirectoryJSON={"@odata.context":"/redfish/v1/
$metadata#HpeiLOResourceDirectory.HpeiLOResourceDirectory",
"@odata.etag":"W/\"9B6CB9C6\"","@odata.id":"/...
```

XAuthToken support

NOTE: XAuthToken is supported on HPE ProLiant Gen10 servers only.

Example: Connect-HPEiLO using Xauthtoken parameter (SSO token)

```
PS C:\> $connection = Connect-HPEiLO
-IP 192.168.10.34 -XAuthToken
f8febab0d8c218372b9aa74e97d17af7
-DisableCertificateAuthentication
PS C:\> $connection

IP                  : 10.20.30.40:100
Hostname            : dl360gen10-01.abcd.com:100
Timeout             : 30
DisableCertificate
Authentication      : True
TargetInfo          : @{ProductName=ProLiant DL580 Gen10;
                      ServerFamily=ProLiant;
                      ServerGeneration=Gen10;
                      ServerModel=DL580;
                      iLOGeneration=iLO5;
                      iLOFirmwareVersion=2.1;
                      SystemROM=U34 v2.16
                      (09/12/2019);
                      ProcessorName=Intel(R)
                      Genuine processor}
ExtendedInfo        : @{UserSuppliedAddress
                      =dl360gen10-01.abcd.com:100;
                      HttpConnectAddress
                      =dl360gen10-01.abcd.com:100;
                      UserName=admin;
                      Modifier1=b3JJfwxwbZSKKyjpg
                      Kmn3QPj16vDZ7Q3oq9ejPZCOPk=;
                      Modifier2=cqEhobrR75LcegppbRmPWQ==}
ConnectionInfo      : @{Redfish=}

PS C:\> $connection.TargetInfo

ProductName         : ProLiant DL580 Gen10
ServerFamily        : ProLiant
ServerGeneration    : Gen10
ServerModel         : DL580
iLOGeneration       : iLO5
iLOFirmwareVersion  : 2.1
SystemROM           : U34 v2.16 (09/12/2019)
ProcessorName       : Intel(R) Genuine processor

PS C:\> $connection.ExtendedInfo

UserSuppliedAddress : dl360gen10-01.abcd.com:100
HttpConnectAddress  : dl360gen10-01.abcd.com:100
UserName            : admin
Modifier1           : b3JJfwxwbZSKKyjpg
                      Kmn3QPj16vDZ7Q3oq9ejPZCOPk=
Modifier2           : cqEhobrR75LcegppbRmPWQ==

PS C:\> $connection.ConnectionInfo

Redfish
```

```
-------


@{ResourceDirectoryJSON={"@odata.context":"/redfish/v1/
 $metadata#HpeiLOResourceDirectory.HpeiLOResourceDirectory",
 "@odata.etag":"W/\"9B6CB9C6\"","@odata.id":"/...
```

Example: Connecting to iLO5 servers using XAuthToken (via oneView)

XauthToken can be generated from Redfish API for the given iLO IP or via One view using OV cmdlets.

Use either of the following OV cmdlets to generate XauthToken:

- ```
  PS C:\>Connect-HPOVMgmt -Hostname <IP>
  -UserName <username> -Password <password>

  PS C:\>$result                = Get-HPOVServer
  PS C:\>$remoteConsoleUrl      = "$($result[1].uri)/remoteConsoleUrl"
  PS C:\>$resp                  = Send-HPOVRequest $remoteConsoleUrl
  PS C:\>$URL,$session          = $resp.remoteConsoleUrl.Split("&")
  PS C:\>$http, $iLOIP          = $URL.split("=")
  PS C:\>$sName,$xAuthToken     = $session.split("=")

  $connection = Connect-HPEiLO -IP $iLOIP -XAuthToken $xAuthToken
  PS C:\> $connection

  IP                  : 10.20.30.40
  Hostname            : dl360gen10-01.abcd.com
  Timeout             : 30
  DisableCertificate
  Authentication      : True
  TargetInfo          : @{ProductName=ProLiant
                        DL580 Gen10;
                        ServerFamily=ProLiant;
                        ServerGeneration=Gen10;
                        ServerModel=DL580;
                        iLOGeneration=iLO5;
                        iLOFirmwareVersion=2.1;
                        SystemROM=U34 v2.16 (09/12/2019);
                        ProcessorName=Intel(R)
                        Genuine processor}
  ExtendedInfo        : @{UserSuppliedAddress=10.20.30.40;
                        HttpConnectAddress=10.20.30.40;
                        UserName=admin;
                        Modifier1=b3JJfwxwb
                        ZSKKyjpgKmn3QPj16vDZ7Q3oq9ejPZCOPk=;
                        Modifier2=cqEhobrR75LcegppbRmPWQ==}
  ConnectionInfo      : @{Redfish=}

  PS C:\> $connection.TargetInfo

  ProductName         : ProLiant DL580 Gen10
  ServerFamily        : ProLiant
  ServerGeneration    : Gen10
  ServerModel         : DL580
  iLOGeneration       : iLO5
  iLOFirmwareVersion  : 2.1
  ```
  ```

```
SystemROM           : U34 v2.16 (09/12/2019)
ProcessorName       : Intel(R) Genuine processor

PS C:\> $connection.ExtendedInfo

UserSuppliedAddress : 10.20.30.40
HttpConnectAddress  : 10.20.30.40
UserName            : admin
Modifier1           : b3JJfwxwbZSKKyjpg
                      Kmn3QPj16vDZ7Q3oq9ejPZCOPk=
Modifier2           : cqEhobrR75LcegppbRmPWQ==

PS C:\> $connection.ConnectionInfo

Redfish


-------


@{ResourceDirectoryJSON={"@odata.context":"/redfish/v1/
 $metadata#HpeiLOResourceDirectory.HpeiLOResourceDirectory",
 "@odata.etag":"W/\"9B6CB9C6\"","@odata.id":"/...
```

- `Get-HPOVIloSso` cmdlet available from HPOneView version 4.x module:

```
PS C:\>Connect-HPOVMgmt -Hostname <IP>
-UserName <username> -Password <password>

PS C:\>$result       = Get-HPOVServer
PS C:\>$xAuthToken   = $result | Get-HPOVIloSso -IloRestSession

$connection = Connect-HPEiLO -IP $iLOIP -XAuthToken $xAuthToken
PS C:\> $connection

IP                  : 10.20.30.40
Hostname            : dl360gen10-01.abcd.com
Timeout             : 30
DisableCertificate
Authentication      : True
TargetInfo          : @{ProductName=ProLiant DL580 Gen10;
                      ServerFamily=ProLiant;
                      ServerGeneration=Gen10;
                      ServerModel=DL580;
                      iLOGeneration=iLO5;
                      iLOFirmwareVersion=2.1;
                      SystemROM=U34 v2.16 (09/12/2019);
                      ProcessorName=Intel(R) Genuine processor}
ExtendedInfo        : @{UserSuppliedAddress=10.20.30.40;
                      HttpConnectAddress=10.20.30.40;
                      UserName=admin;
                      Modifier1=b3JJfwxwbZSKKyjpg
                      Kmn3QPj16vDZ7Q3oq9ejPZCOPk=;
                      Modifier2=cqEhobrR75LcegppbRmPWQ==}
ConnectionInfo      : @{Redfish=}

PS C:\> $connection.TargetInfo
```

```
ProductName          : ProLiant DL580 Gen10
ServerFamily         : ProLiant
ServerGeneration     : Gen10
ServerModel          : DL580
iLOGeneration        : iLO5
iLOFirmwareVersion   : 2.1
SystemROM            : U34 v2.16 (09/12/2019)
ProcessorName        : Intel(R) Genuine processor

PS C:\> $connection.ExtendedInfo

UserSuppliedAddress : 10.20.30.40
HttpConnectAddress  : 10.20.30.40
UserName            : admin
Modifier1           : b3JJfwxwbZSKKyjpg
                      Kmn3QPj16vDZ7Q3oq9ejPZCOPk=
Modifier2           : cqEhobrR75LcegppbRmPWQ==

PS C:\> $connection.ConnectionInfo

Redfish


-------


@{ResourceDirectoryJSON={"@odata.context":"/redfish/v1/
 $metadata#HpeiLOResourceDirectory.HpeiLOResourceDirectory",
 "@odata.etag":"W/\"9B6CB9C6\"","@odata.id":"/...
```

Credential Support

Example: Connecting to iLO 4 server or iLO 5 server with credential input

```
PS C:\>$User = "User01"
PS C:\>$PWord = ConvertTo-SecureString
-String "P@sSwOrd" -AsPlainText -Force
PS C:\> $Credential = New-Object
-TypeName "System.Management.Automation
.PSCredential" -ArgumentList $User, $PWord

PS C:\> $connection = Connect-HPEiLO
-Credential $credential -IP 10.20.30.40
PS C:\> $connection

IP                  : 10.20.30.40
Hostname            : dl360gen10-01.abcd.com
Timeout             : 30
DisableCertificate
Authentication      : True
TargetInfo          : @{ProductName=ProLiant DL580 Gen10;
                      ServerFamily=ProLiant;
                      ServerGeneration=Gen10;
```

```
                        ServerModel=DL580;
                        iLOGeneration=iLO5;
                        iLOFirmwareVersion=2.1;
                        SystemROM=U34 v2.16 (09/12/2019);
                        ProcessorName=Intel(R)
                        Genuine processor}
ExtendedInfo          : @{UserSuppliedAddress=10.20.30.40;
                        HttpConnectAddress=10.20.30.40;
                        UserName=admin;
                        Modifier1=b3JJfwxwb
                        ZSKKyjpgKmn3QPj16vDZ7Q3oq9ejPZCOPk=;
                        Modifier2=cqEhobrR75LcegppbRmPWQ==}
ConnectionInfo        : @{Redfish=}

PS C:\> $connection.TargetInfo

ProductName           : ProLiant DL580 Gen10
ServerFamily          : ProLiant
ServerGeneration      : Gen10
ServerModel           : DL580
iLOGeneration         : iLO5
iLOFirmwareVersion    : 2.1
SystemROM             : U34 v2.16 (09/12/2019)
ProcessorName         : Intel(R) Genuine processor

PS C:\> $connection.ExtendedInfo

UserSuppliedAddress : 10.20.30.40
HttpConnectAddress  : 10.20.30.40
UserName            : admin
Modifier1           : b3JJfwxwbZSKKyjpgKmn3QPj16vDZ7Q3oq9ejPZCOPk=
Modifier2           : cqEhobrR75LcegppbRmPWQ==

PS C:\> $connection.ConnectionInfo

Redfish


-------
@{ResourceDirectoryJSON={"@odata.context":"/redfish/v1/
 $metadata#HpeiLOResourceDirectory.HpeiLOResourceDirectory",
 "@odata.etag":"W/\"9B6CB9C6\"","@odata.id":"/...
```

NOTE:

Other examples in this document use IPv4, but could use IPv6 or hostname instead, if supported on the network.

IPv4, IPv6, or hostname addresses can be used to establish the connection.

The `iLOResetWaitTime` parameter can be supplied with the `Connect-HPEiLO` cmdlet. `iLOResetWaitTime` is the maximum time in seconds until the cmdlet waits for iLO to return, post iLO reset. In case of any cmdlet that resets iLO, the cmdlet waits for the iLO to reset until the time specified in `iLOResetWaitTime`. For example, `Update-HPEiLOFirmware` or `Add-HPEiLOLanguagePack`.

Using the `Disconnect-HPEiLO` cmdlet

Use the `Disconnect-HPEiLO` cmdlet to disconnect the connection object when you are finished using the iLO settings.

For a connection to iLO 5 servers, if you do not use `Disconnect-HPEiLO`, there will be an open session that will not be disconnected until a session inactivity timer expires. Connection expire time is based on iLO session timeout.

```
PS C:\> Disconnect-HPEiLO -Connection $connection
```

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

Using the `Test-HPEiLOConnection` cmdlet

Use the `Test-HPEiLOConnection` cmdlet to test the connection object before executing any operation.

```
PS C:\> $output = $connection | Test-HPEiLOConnection
PS C:\> $output


IP                   : 10.20.30.40:443
Hostname             : abc.domain.com:443
IsConnected          : True
TestConnectionDetail : {HPE.Framework.Connector.Common.TestConnectionStatus}
StatusMessage        : OK



PS C:\> $output.TestConnectionDetail | fl


ConnectionType : RIBCL
IsConnected    : True
StatusMessage  : Connection is valid.
```

If the cmdlet is successful, a return object with the `IsConnected` property, True is returned. If an error occurs, an appropriate output message is displayed.

Connecting to multiple targets

In early releases of iLO Cmdlets 1.x, running one cmdlet that sent data to multiple targets resulted in a significant amount of time spent waiting for responses. This time was a result of normal network delays and device response and data collection delays. But when added together in performing each operation serially, it resulted in a significant amount of time to perform operations on many targets. To avoid this situation, multithreading has been implemented. When using multithreading, commands are sent to each target in parallel during the operation of one cmdlet and responses are waited for in parallel. Multithreading provides a significant performance improvement. Most commands that support multiple targets use multithreading for iLO 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.

- Default value is 256 threads.
- Set thread value cautiously depending upon the client system configuration to avoid deadlocks and race conditions. The value range is between 1 and 4096.

NOTE:

- Use the `Get-HPEiLOMaxThreadLimit` cmdlet to get the maximum number of parallel threads to execute the iLO cmdlets.
- Use the `Set-HPEiLOMaxThreadLimit` cmdlet to set the maximum number of parallel threads to execute the iLO cmdlets.

Performance of the cmdlets depends on the following factors:

- Current system load
- Available memory
- Number of processors
- Network configuration
- Other systems in the network
- Other network traffic

To take advantage of multithreading, a single cmdlet is used. However, it is directed to multiple targets in a single invocation by passing parameter values as an array.

Establishing connection to multiple targets

A connection can be established to multiple targets by providing a range of IP addresses and the related username and password. `Connect-HPEiLO` returns a list of connection objects which can be used to execute any iLO cmdlets.

Example 1

The following is an example of using `Connect-HPEiLO` with multiple IP addresses. The count 3 shows that three connection objects are created.

```
PS C:\> $connection = Connect-HPEiLO -IP 10.20.30.1,10.20.30.2,10.20.30.3 -
Username admin -Password admin123 PS C:\> $connection.Count
3
```

Example 2

The following is an example of using `Connect-HPEiLO` with a range of IP addresses that connects to all the available IP addresses.

```
PS C:\> $connection = Connect-HPEiLO -IP 10.20.30.1-30 -Username admin -
Password admin123
```

Example 3

The following is an example of using `Connect-HPEiLO` using CSV input that contains the iLO IP and its corresponding username and password. Consider a CSV file with IP, Username, and Password columns.

```
$IP=@()
$Username=@()
$Password=@()
Import-Csv "C:\iLOInput.CSV" |`
ForEach-Object {
$IP += $_.IP
$Username += $_.Username
```

```
$Password += $_.Password
}
PS C:\> $connection = Connect-HPEiLO -IP $IP -Username $Username -Password
$Password
```

Example 4

The following is an example of using `Get-HPEiLOMaxThreadLimit` to get the current default MAX thread limit and `Set-HPEiLOMaxThreadLimit` to modify the value and cross check the modified value.

After this, in the same PowerShell session the connection cmdlet will use the 512 modified value to launch parallel threads. Subsequent Get or Set cmdlets will use the same max thread limit until you modify or close the session.

```
PS C:\> Get-HPEiLOMaxThreadLimit
256

PS C:\> Set-HPEiLOMaxThreadLimit -MaxThreadLimit 512

PS C:\> Get-HPEiLOMaxThreadLimit
512

PS C:\> $cons = Connect-HPEiLO 192.168.10-11 admin admin123 -DisableCertificateAuthentication
-Verbose
VERBOSE: Performing the operation "Connect-HPEiLO" on target "IP: 192.168.10-11".
VERBOSE: [Connect-HPEiLO] Executing the cmdlet with 512 tasks using 512 concurrent threads.
.
.

PS C:\> Find-HPEiLO 192.168.10-11 -Verbose
VERBOSE: Performing the operation "Find-HPEiLO" on target "Range: 192.168.10-11".
WARNING: It might take a while to search for all the iLOs if the input is a
very large range.
Use Verbose for more information.
VERBOSE: [Find-HPEiLO] Executing the cmdlet with 512 tasks using 512 concurrent threads.
```

Executing a cmdlet using multiple connections

Example 1

The following example executes `Get-HPEiLOAssetTag` to multiple targets.

```
PS C:\> Get- HPEiLOAssetTag -Connection $connection
AssetTag   : TestAssetTag
IP         : 10.20.30.1
Hostname   : ab.domain.com
Status     : OK
StatusInfo :

AssetTag   : Test2AssetTag
IP         : 10.20.30.2
Hostname   : ab2.domain.com
Status     : OK
StatusInfo :

AssetTag   : Test3AssetTag
IP         : 10.20.30.3
Hostname   : ab3.domain.com
Status     : OK
StatusInfo :
```

Example 2

The following example executes `Set-HPEiLOAssetTag` on multiple servers. The asset tag will be changed correspondingly on each server.

```
PS C:\> $result = Set-HPEiLOAssetTag -Connection $connection -AssetTag
@("TestMultiConAssetTag"," TestMultiCon2AssetTag"," TestMultiCon3AssetTag")
```

Executing a cmdlet with multiple connection objects passed as pipeline input.

When there are multiple connections being passed as pipeline input, use the comma operator (",") to pass the list of connections as an array. If the comma unary operator is not used, then the parameter unary value at index 0 will be set for all the targets.

In this example, `$connection` has three connections that are passed as pipeline input to the `Set-HPEiLOAssetTag` cmdlet using the comma unary operator (","). `AssetTag` will be changed in each of the corresponding target.

```
PS C:\> $result = ,$connection | Set-HPEiLOAssetTag -AssetTag
@("TestMultiConAssetTag"," TestMultiCon2AssetTag"," TestMultiCon3AssetTag")
```

Example 2

In this example, `$connection` has three connections that are passed as pipeline input to the `Set-HPEiLOAssetTag` cmdlet without the comma unary operator (","). The `AssetTag` parameter array value at index 0 (TestMultiConAssetTag) will be set for all the targets.

```
PS C:\> $connection | Set-HPEiLOAssetTag -AssetTag
@("TestMultiConAssetTag"," TestMultiCon2AssetTag"," TestMultiCon3AssetTag")

WARNING: Some values of the command-line parameter 'AssetTag' have been
ignored for the pipeline objects(s) at index: [0], [1], [2].
```

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-HPEiLO` to `Get-HPEiLOAHSStatus` to get the AHS status of those connected servers. The `-Verbose` parameter can be used to view more information.

PowerShell script:

Example 1

```
PS C:\> Connect-HPEiLO -IP 10.20.30.100-102 -Username "username" -Password
"password" | Get-HPEiLOAHSStatus
AHSEnabled              : Yes
AHSHardwareEnabled      : Yes
AHSTemporaryHoldEnabled : No
IP                      : 10.20.30.100
Hostname                : abc.domain.com
Status                  : OK
StatusInfo              :

AHSEnabled              : Yes
AHSHardwareEnabled      : Yes
AHSTemporaryHoldEnabled : No
IP                      : 10.20.30.101
Hostname                : abc2.domain.com
Status                  : OK
StatusInfo              :
```

```
AHSEnabled              : Yes
AHSHardwareEnabled      : Yes
AHSTemporaryHoldEnabled : No
IP                      : 10.20.30.102
Hostname                : abc3.domain.com
Status                  : OK
StatusInfo              :
```

This threading enables multiple commands to multiple servers to be sent at the same time. `Connect-HPEiLO` makes the connection object array of the three servers. Those connections are in turn passed through to `Get-HPEiLOAHSStatus`, which uses those connections and requests the AHS status information from each server.

Example 2

The following example shows how the Alert Mail Settings of one server can be replicated to other servers.

```
PS C:\> $alertMailSetting = Connect-HPEiLO –IP 10.20.30.1 -Username
"username" -Password "password" | Get-HPEiLOAlertMailSetting
PS C:\> $alertMailSetting
AlertMailEmail        : test@abc.com
AlertMailEnabled      : Yes
AlertMailSenderDomain : powershvpn
AlertMailSMTPPort     : 28
AlertMailSMTPServer   : smtpabc.domain.com
IP                    : 10.20.30.1
Hostname              : abc.domain.com
Status                : OK
StatusInfo            :

PS C:\> $connection = Connect-HPEiLO –IP 10.20.30.12-14 –Username “username”
–Password “password”
PS C:\> $alertMailSetting | Set-HPEiLOAlertMailSetting –Connection
$connection
PS C:\> $output = Get-HPEiLOAlert
PS C:\> $output

AlertMailEmail        : test@abc.com
AlertMailEnabled      : Yes
AlertMailSenderDomain : powershvpn
AlertMailSMTPPort     : 28
AlertMailSMTPServer   : smtpabc.domain.com
IP                    : 10.20.30.12
Hostname              : abc12.domain.com
Status                : OK
StatusInfo            :

AlertMailEmail        : test@abc.com
AlertMailEnabled      : Yes
AlertMailSenderDomain : powershvpn
AlertMailSMTPPort     : 28
AlertMailSMTPServer   : smtpabc.domain.com
IP                    : 10.20.30.14
Hostname              : abc14.domain.com
Status                : OK
StatusInfo            :
```

Using the `Update-HPEiLOFirmware` cmdlet

The `Update-HPEiLOFirmware` cmdlet is used to update the firmware.

In case of iLO 4, the following firmware types can be updated:

1. iLO (.bin)
2. Complex programmable logic device (.vme)
3. Power PIC (.hex)
4. ROM firmware image (.flash and .full)
5. Other formats supported by the target platform

In case of iLO 5, the following firmware types can be updated:

1. iLO (.bin)
2. ROM firmware image (.flash and .full)
3. Power Management Controller
4. Server Platform Services Firmware
5. Smart Array drivers
6. Intelligent Platform Abstraction Data
7. Smart Storage Battery
8. TPM or TM firmware
9. SAS Programmable Logic Device
10. System Programmable Logic Device
11. Intelligent Provisioning
12. Networking adapters
13. NVMe Backplane firmware
14. Innovation Engine (IE) firmware
15. Drive firmware
16. Power Supply firmware
17. Embedded Video Controller
18. Other formats supported by the target platform

Procedure

1. Locate and download the server firmware package from **http://www.hpe.com/info/ilo**.
2. Execute the downloaded firmware package `CPxxxxxx.exe` and extract the package to a local folder.

3. Execute `Update-HPEiLOFirmware` with the `Location` parameter set to the full path of the image that was extracted from the download.
4. Reboot the server for changes to take effect.

Example 1

The following is an example of using `Update-HPEiLOFirmware` with a single IP and firmware file in the local folder.

```
PS C:\> $output = Update-HPEiLOFirmware –Connection $connection –Location C:
\Firmwares\ilo4_255.bin
PS C:\> $output
IP         : 10.20.30.1
Hostname   : abc.domain.com
Status     : WARNING
StatusInfo : HPE.iLO.Response.StatusInfo

PS C:\> $output.StatusInfo.Message
Server will need to be reset for changes to be applied.
```

Example 2

The following is an example of using `Update-HPEiLOFirmware` in iLO 5 with a single IP and firmware file located in the web server with `iLOWaitResetTime` in the connection object. `iLOWaitResetTime` is the maximum time in seconds until the cmdlet waits for iLO to return, post iLO reset.

```
PS C:\> $output = Update-HPEiLOFirmware –Connection $connection –Location
https://10.20.30.100/webfiles/iLO5/iLO5_120.bin

PS C:\> $output
```

Example 3

The following is an example of using `Update-HPEiLOFirmware` in iLO 5 to update the firmware located in the web server and upload it to the repository.

```
PS C:\> $output = Update-HPEiLOFirmware –Connection $connection –Location –
UpdateRepository https://10.20.30.100/webfiles/iLO5/
U15_2.30_08_06_2016.signed.flash

PS C:\> $output
IP         : 10.20.30.3
Hostname   : abc3.domain.com
Status     : WARNING
StatusInfo : HPE.iLO.Response.StatusInfo

PS C:\> $output.StatusInfo.Message
ResetRequired
```

NOTE: An Https URL is supported only in case of iLO 5.

Using the `Backup-HPEiLOSetting` and `Restore-HPEiLOSetting` cmdlets

The Backup & Restore feature in iLO 5 allows you to restore the iLO configuration on a system with the same hardware configuration as the system that was backed up. This feature is not meant to duplicate a configuration and apply it to a different iLO system. The cmdlets `Backup-HPEiLOSetting` and `Restore-HPEiLOSetting` can be used to perform these operations.

You might want to restore the iLO configuration in the following situations:

- Battery failure or removal
- Reset to factory defaults
- Accidental or incorrect configuration change
- System board replacement
- Lost license key

Example 1

The following example shows the usage of the `Backup-HPEiLOSetting` cmdlet that saves the existing iLO settings in a .bak file in the location specified.

```
PS C:\> $connection = Connect-HPEiLO 10.20.30.1,10.20.30.2 -Username admin -
Password admin123

PS C:\> Backup-HPEiLOSetting -Connection $connection -BackupFileLocation C:
\Users\admin\test.bak
```

Example 2

The following example shows the usage of the `Backup-HPEiLOSetting` cmdlet with password.

```
PS C:\> $connection = Connect-HPEiLO 10.20.30.1,10.20.30.2 -Username admin -
Password admin123

PS C:\> Backup-HPEiLOSetting -Connection $connection -BackupFileLocation C:
\Users\admin\test.bak -BackupFilePassword "bakFilePassword"
```

Example 3

The following example shows the usage of the `Backup-HPEiLOSetting` cmdlet that saves the iLO settings on NAND.

```
PS C:\> $connection = Connect-HPEiLO 10.20.30.1,10.20.30.2 -Username admin -
Password admin123

PS C:\> Backup-HPEiLOSetting -Connection $connection -BackupOnNAND
```

Example 4

The following example shows the usage of the `Restore-HPEiLOSetting` cmdlet to restore the setting.

```
PS C:\> $connection = Connect-HPEiLO 10.20.30.1,10.20.30.2 -Username admin -
Password admin123

PS C:\> Restore-HPEiLOSetting -Connection $connection -BackupFileLocation C:
\Users\admin\test.bak
```

Example 5

The following example shows the usage of the `Restore-HPEiLOSetting` cmdlet that restores the password protected file.

```
PS C:\> $connection = Connect-HPEiLO 10.20.30.1,10.20.30.2 -Username admin -
Password admin123

PS C:\> Backup-HPEiLOSetting -Connection $connection -BackupFileLocation C:
\Users\admin\test.bak -BackupFilePassword "bakFilePassword"
```

Example 6

The following example shows the usage of the `Restore-HPEiLOSetting` cmdlet that restores the settings from NAND.

```
PS C:\> $connection = Connect-HPEiLO 10.20.30.1,10.20.30.2 -Username admin -
Password admin123

PS C:\> Restore-HPEiLOSetting -Connection $connection -RestoreFromNAND
```

Using the `Get-HPEiLORedfishMessageInfo` cmdlet

The `Get-HPEiLORedfishMessageInfo` cmdlet is used to get the detail of the redfish message. This cmdlet is supported only on iLO 5. If you want to know more about the error message or any message that is returned by the target while executing the cmdlet, use this cmdlet by supplying the message that is obtained to the `MessageID` parameter.

Example 1

The following is an example of using `Get-HPEiLORedfishMessageinfo` without the `MessageID` parameter. When `MessageID` is not provided, the cmdlet lists out all the possible messages.

```
PS C:\> $output = Get-HPEiLORedfishMessageInfo -Connection $connection

PS C:\> $output
AccessDenied                      : HPE.iLO.Response.Redfish.MessageIDInfo
AccountForSessionNoLongerExists   : HPE.iLO.Response.Redfish.MessageIDInfo
AccountModified                   : HPE.iLO.Response.Redfish.MessageIDInfo
AccountNotModified                : HPE.iLO.Response.Redfish.MessageIDInfo
AccountRemoved                    : HPE.iLO.Response.Redfish.MessageIDInfo
ActionNotSupported                : HPE.iLO.Response.Redfish.MessageIDInfo
ActionParameterDuplicate          : HPE.iLO.Response.Redfish.MessageIDInfo
ActionParameterMissing            : HPE.iLO.Response.Redfish.MessageIDInfo
.
.
.
.
.
.
IP                                : 10.20.30.1
Hostname                          : abcd.com
Status                            : OK
StatusInfo                        :

PS C:\> $output.AccessDenied
Description                       : While attempting to access, connect to, or
                                    transfer to/from another resource,
                                    the service was denied access.
Message                           : While attempting to establish a connection to %1,
                                    the service was denied access.
NumberOfArgs                      : 1
ParamTypes                        : {string}
Resolution                        : Verify that the URI is correct and that
                                    the service has the
                                    appropriate credentials
Severity                          : Critical
Oem                               :
```

Example 2

The following is an example of using `Get-HPEiLORedfishMessageInfo` with the `MessageID` value that was returned while executing `Reset-HPEiLO`.

```
PS C:\> $output = Reset-HPEiLO -Connection $connection
-Device iLO -Confirm: $false
```

```
PS C:\> $output.StatusInfo
Category    Message                AffectedAttribute
--------    -------                -----------------
General     ResetInProgress

PS C:\> $output = Connect-HPEiLO -IP 10.20.30.1 -Username "username" -
Password "password" | Get-HPEiLORedfishMessageInfo -MessageID
ResetInProgress

PS C:\> $output
MessageInfo  : {ResetInProgress,ImportCertSuccessfuliLOResetinProgress}
IP           : 10.20.30.1
HostName     : abc.domain.com
Status       : OK
StatusInfo   :

PS C:\> $output.MessageInfo
MessageID    : ResetInProgress
Description  : A management processor reset is in progress.
Message      : A management processor reset is in progress.
NumberOfArgs : 0
ParamTypes   : {}
Resolution   : Wait for management processor reset to complete,
               and then retry the operation.
Severity     : Warning
Oem          :

MessageID    : ImportCertSuccessfuliLOResetinProgress
Description  : Import Certificate was successful and the management
               processor is being reset.
Message      : Import Certificate was successful. Management Processor
               reset is in progress to enable the new certificate.
NumberOfArgs : 0
ParamTypes   : {}
Resolution   : None
Severity     : Warning
Oem          :
```

Using the `Get-HPEiLOModuleVersion` cmdlets

These cmdlets determine the current version of the iLO Cmdlets installed and update the module if necessary.

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

```
PS C:\> Get-HPEiLOModuleVersion
Name                              : HPEiLOCmdlets
Path                              : C:\Program Files\Hewlett-Packard\PowerShell\
                                    Modules\HPEiLOCmdlets\HPEiLOCmdlets.dll
Description                       : Scripting Tools for Windows PowerShell :
                                    iLO Cmdlets
                                    uses the RIBCL and Redfish interface to
                                    communicate to iLO. These cmdlets can be used to
                                    configure and manage iLO on HPE ProLiant Gen8,
                                    Gen9 or Gen10 servers.
```

```
GUID                               : 4d93d4cf-72b1-4d1a-a247-bef198bebf96
CurrentUICultureName               : en-US
CurrentUICultureVersion            :
CurrentModuleVersion               : 3.0.0.0
LatestAvailableModuleVersion       : 3.0.0.0
ProductModuleDownloadURL           : https://www.hpe.com/servers/powershell
PowerShellGalleryModuleDownloadURL : www.powershellgallery.com/packages/HPEiLOCmdlets
                                     /3.0.0.0
DotNetVersion                      : 4.7
DotNetFrameworkDescription         : 4.7.02053
PSVersion                          : @{PSVersion=5.0.10586.117;
                                     PSCompatibleVersions=System.Version[];
                                     BuildVersion=10.0.10586.117;
                                     CLRVersion=4.0.30319.42000;
                                     WSManStackVersion=3.0;
                                     PSRemotingProtocolVersion=2.3;
                                     SerializationVersion=1.1.0.1}
OSVersion                          : @{Caption=Microsoft Windows 8.1 Enterprise;
                                     CSDVersion=; Version=6.3.9600; BuildNumber=9600}
CCGVersion                         : 2.0.0.0
AvailableUICulture                 : {}
Status                             : OK
StatusInfo                         :
```

Logging

The iLO cmdlet provides necessary logging for the cmdlets that are executed in the PowerShell session. Hewlett Packard Enterprise recommends logging. Logging can be enabled or disabled using the cmdlets `Enable-HPEiLOLog` and `Disable-HPEiLOLog`. The log files must be deleted using the cmdlet `Clear-HPEiLOLog`. When logging is enabled using the cmdlet `Enable-HPEiLOLog`, the log file gets created in the installed module path within the logs folder. By default logging is not enabled. The detailed information logged helps in troubleshooting.

Example 1

The following is an example of enabling logging.

```
PS C:\> Enable-HPEiLOLog

LogFilePath
-----------
C:\Program Files\Hewlett-Packard\PowerShell\Modules
\HPEiLOCmdlets\Logs\HPEiLOCmdlets_2142018_1593332.log
```

Example 2

The following is an example of disabling logging.

```
PS C:\> Disable-HPEiLOLog
```

Example 2

The following is an example of deleting the last 2 log files from the Logs folder. All the log files will be deleted even though the parameter `Last` is not supplied.

```
PS C:\> Clear-HPEiLOLog –Last 2
```

NOTE: When the log file size exceeds more than 10MB, then a new log file will be created with the same name appended with numbers. For example, if the initial log file name is HPEiLOCmdlets_2142018_1593332.log and has exceeded 10 MB, then a new file with name HPEiLOCmdlets_2142018_1593332_1.log will be created. The log configuration can be modified and obtained using the Get/Set-HPEiLOLogConfig. These cmdlets must be executed after enabling the log.

Error Handling - Cmdlet, parameter, and parameter value supportability on target servers

All the iLO cmdlets are not supported on all generations of servers. When a cmdlet that is not supported is used, an error message is returned. Familiarize yourself with the cmdlets that are supported on the target server before executing any cmdlets.

The following connection object is used for all the scenarios:

```
PS C:\> $connection = Connect-HPEiLO -IP 10.20.30.40
-Username admin -Password admin123
PS C:\> $connection
IP                                  : 10.20.30.40
Hostname                            : dl360gen9-01.abcd.com
Timeout                             : 30
DisableCertificateAuthentication    : True
TargetInfo                          : @{ProductName=ProLiant ML350 Gen9;
                                      ServerFamily=ProLiant;
                                      ServerGeneration=Gen9;
                                      ServerModel=ML350;
                                      iLOGeneration=iLO4;
                                      iLOFirmwareVersion=2.62;
                                      SystemROM=P92 v2.72 (03/25/2019);
                                      ProcessorName=Intel(R) Xeon(R)
                                      CPU E5-2603 v3 @ 1.60GHz}
ExtendedInfo                        : @{UserSuppliedAddress
                                      =dl360gen9-01.abcd.com;
                                      HttpConnectAddress
                                      =dl360gen9-01.abcd.com;
                                      UserName=admin;
                                      Modifier1=vyUuq0Oh/
                                      XOc1n+Vv4w1YcFF+FvdoVk/FUEH2cdAD/4=;
                                      Modifier2=4eA+yvA4kyRK8WYwIeT7zA==}
ConnectionInfo                      : @{RIBCL=; REST=}
```

Example 1: The following is an example of an object returned when the cmdlet is not supported on the iLO.

```
PS C:\> $output = $connection | Get-HPEiLODeviceInventory
PS C:\> $output | fl
IP                                  : 10.20.30.40
Hostname                            : dl360gen9-01.abcd.com
Status                              : ERROR
StatusInfo                          : HPE.iLO.Response.StatusInfo

PS C:\> $output.StatusInfo | fl
Category                            : FeatureSupportability
Message                             : Feature not supported on iLO 3,
                                      iLO 4 and iLO 5
                                      (FW Ver: 1.10,1.11,1.15,1.17).
AffectedAttribute                   : {}
```

Example 2: The following is an example of an object returned when the parameter of a cmdlet is not supported on the iLO.

```
PS C:\> $output = $c | Set-HPEiLOAccessSetting
-SSDPProtocolEnabled Yes
PS C:\> $output | fl
IP                                  : 10.20.30.40
Hostname                            : dl360gen9-01.abcd.com
```

```
Status                              : ERROR
StatusInfo                          : HPE.iLO.Response.StatusInfo

PS C:\> $output.StatusInfo | fl
Category                            : ParameterSupportability
Message                             : The parameter(s)
                                      listed in AffectedAttribute are
                                      not supported on the target iLO.
                                      For more details about
                                      supported parameters,
                                      see the cmdlet help.
AffectedAttribute                   : {SSDPProtocolEnabled}
```

Example 3: The following is an example of an object returned when the parameter value of a cmdlet is not supported on the iLO.

```
PS C:\> $output = $connection | Set-HPEiLOSNMPv3User
-ID 5 -SecurityName testUser -AuthenticationProtocol MD5
-AuthenticationPassphrase admin123 - PrivacyProtocol
AES -PrivacyPassphrase admin123

PS C:\> $output | fl
IP                                  : 10.20.30.40
Hostname                            : dl360gen9-01.abcd.com
Status                              : ERROR
StatusInfo                          : HPE.iLO.Response.StatusInfo

PS C:\> $output.StatusInfo | fl
Category                            : ParameterValueDependency
Message                             : The parameter value(s)
                                      listed in AffectedAttribute
                                      are not supported
                                      on the target iLO.
                                      For more details about
                                      supported parameter values,
                                      see the cmdlet help.
AffectedAttribute                   : {[ID, 4,5,6,7,8]}
```

Example 4: The following is an example of an object returned when one parameter value depends upon the other parameter value.

```
PS C:\> $key = ,@("L_ALT","L_CTRL")
PS C:\> $output = $connection | Set-HPEiLOComputerLockConfiguration -
LockType Windows -KeySequence $key

PS C:\> $output.StatusInfo | fl

PS C:\> $output.StatusInfo | fl
Category                            : ParameterValueDependency
Message                             : The parameter(s) listed
                                      in AffectedAttribute cannot be
                                      set on the target iLO when
                                      LockType is Disabled,
                                      Windows. For more details
                                      about supported parameters,
                                      see the cmdlet help.
AffectedAttribute                   : {[KeySequence, ]}
```

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. Along the way, you can test code and view objects to understand better how to accomplish your goals.

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

Procedure

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 iLO by different cmdlets.
5. Summarize or output the data in the desired format.

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

When using data sources such as CSV files, XML files, 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 organization's security policies and code accordingly.

NOTE: To clear the string parameter value for any parameter, provide "". For clearing IPv4 values, provide "0.0.0.0" and for IPv6 provide "::".

Security Encryption Setting in iLO

Security Encryption settings for iLO 4

AES/3DES - When AES/3DES encryption is enabled, you must use a cipher strength equal to or greater than AES/3DES to connect to iLO through these secure channels. The AES/3DES encryption enforcement setting does not affect communications and connections over less-secure channels.

FIPS - iLO operates in a mode intended to comply with the requirements of FIPS 140-2 level 1. **FIPS** is a set of computer security standards, mandated for use by United States government agencies and contractors. The FIPS security state is not the same as FIPS validated. FIPS validated refers to software that has received validation by completing the Cryptographic Module Validation Program.

Security Encryption settings for iLO 5

Production (default) - iLO uses the factory default encryption settings.

HighSecurity - iLO enforces the use of AES ciphers over the secure channels, including secure HTTP transmissions through the browser, SSH port, iLO RESTful API, and RIBCL. When **HighSecurity** is enabled, you must use a supported cipher to connect to iLO through these secure channels. This security state does not affect communications and connections over less-secure channels.

FIPS - iLO operates in a mode intended to comply with the requirements of FIPS 140-2 level 1. **FIPS** is a set of computer security standards, mandated for use by United States government agencies and contractors. The FIPS security state is

not the same as FIPS validated. FIPS validated refers to software that has received validation by completing the Cryptographic Module Validation Program.

SuiteB - The **SuiteB** security state (also called CNSA mode) is available only when the FIPS security state is enabled. iLO operates in a mode intended to comply with the **SuiteB** requirements defined by the NSA, and intended to secure systems used to hold United States government top secret classified data.

Setting the security state using the `Set-HPEiLOEncryptionSetting` cmdlet

```
PS C:\> $connection = Connect-HPEiLO -IP 10.20.30.40 -Username admin -Password admin123

PS C:\> $output = $connection | Set-HPEiLOEncryptionSetting -SecurityState FIPS
```

For more information regarding security state, see the **iLO security states** section in the iLO 4 and iLO 5 User Guides.

Special characters supportability

For **iLO 5** servers, the following characters are not supported in the corresponding firmware version:

Firmware Version	Not Supported Character
1.10, 1.15	`~!@#$%^&*()_-+={}{}|;',<>?
1.20	#'?

NOTE: Both single quote (') and double quote (") characters in combination are not supported as a parameter value in any cmdlet.

For example: `Set-HPEiLOAssetTag -Server 10.20.30.40 -Username "username" -Password "password" -AssetTag "asset'"tag"`

Troubleshooting

General issues

Verifying iLO firmware versions

If a problem occurs, follow the below course of action:

1. Verify the most current versions of iLO firmware installed in the iLO. Updating to the most current firmware might solve the problem. For information on updating iLO firmware, see the iLO 4 or iLO 5 user guide.
2. Verify the cmdlet module version and Update to latest cmdlet module version if available.
3. Enable logging and capture the logs. Contact the support team with the captured logs.

iLO protocols and ports

The following table identifies the protocols and ports for iLO:

Product	Protocol	Port
iLO	HTTP	80
	HTTPS	443

The iLO cmdlets do not work after enabling the "Enforce AES/3DES Encryption" setting in HPE iLO 4

If the iLO cmdlets do not work after enabling the "Enforce AES/3DES Encryption" setting in iLO 4, use the following table to verify that your environment has the correct versions.

iLO Firmware	.NET Framework version	Windows PowerShell version	Transport Layer Security (TLS) version
Any version of iLO 4 or iLO 5	4.7.1 or later	3.0, 4.0, 5.0, or 5.1	TLS 1.0, 1.1, or 1.2

Find-HPEiLO cmdlet response time is longer when client proxy server setting is invalid

If there is an invalid proxy server Internet setting on the client environment, the Find-HPEiLO cmdlet response time is longer.

Find-HPEiLO does not find the iLO even when iLO is reachable through browser

The XML reply setting might be turned off. The XML reply setting must be turned on before executing the cmdlet. To enable the setting, navigate to **Security > Access Settings > XML Reply** through an iLO GUI.

Smart components do not install and remain stalled in the iLO installation queue

To install smart components, the operating system must be available on the target server. In addition, Agentless Management Service (AMS) must be running on the target server.

Cmdlets throw the exception `Target url(s) could not be located on Gen10 servers`

If a cmdlet throws the exception `Target url(s) could not be located on Gen10 servers`, it could be a scenario where License Key is missing. Adding License Key might solve the problem.

Usage tips

Sample scripts

In iLO Cmdlets v3.0.0.0 several sample scripts are included in a folder named Samples, which is located in the directory specified by the user while installing. The primary purpose of these sample scripts is to provide instructions on the following scripting aspects:

- Samples for providing parameters for a cmdlet (such as named parameters, pipeline parameters, imported parameters from CSV).
- Samples for handling with errors or exceptions in the script.
- Samples for implementing scripts for complicated functions, which need several cmdlets. Examples of the functions are generating and importing iLO certificate settings, iLO IP to use, and Active Directory authentication.

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

Memory Optimization

Use `[System.GC]::Collect()` to optimize client system memory consumption during script execution.

Websites

General websites

Hewlett Packard Enterprise Information Library

www.hpe.com/info/EIL

Single Point of Connectivity Knowledge (SPOCK) Storage compatibility matrix

www.hpe.com/storage/spock

Storage white papers and analyst reports

www.hpe.com/storage/whitepapers

Windows PowerShell websites

The following websites provide useful information for using PowerShell.

Microsoft Script Center

http://technet.microsoft.com/en-us/scriptcenter/default

Windows PowerShell Blog

http://blogs.msdn.com/b/powershell/

PowerShell.org

http://powershell.org/

PowerShell Magazine

http://www.powershellmagazine.com/

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:

  Hewlett Packard Enterprise Support Center

  www.hpe.com/support/hpesc

  Hewlett Packard Enterprise Support Center: Software downloads

  www.hpe.com/support/downloads

  Software Depot

  www.hpe.com/support/softwaredepot

- To subscribe to eNewsletters and alerts:

  www.hpe.com/support/e-updates

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

  www.hpe.com/support/AccessToSupportMaterials

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

Customer self repair

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

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

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

Remote support

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

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

Remote support and Proactive Care information

HPE Get Connected

www.hpe.com/services/getconnected

HPE Proactive Care services

www.hpe.com/services/proactivecare

HPE Proactive Care service: Supported products list

www.hpe.com/services/proactivecaresupportedproducts

HPE Proactive Care advanced service: Supported products list

www.hpe.com/services/proactivecareadvancedsupportedproducts

Proactive Care customer information

Proactive Care central

www.hpe.com/services/proactivecarecentral

Proactive Care service activation

www.hpe.com/services/proactivecarecentralgetstarted

Warranty information

To view the warranty for your product or to view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products* reference document, go to the Enterprise Safety and Compliance website:

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

Additional warranty information

HPE ProLiant and x86 Servers and Options

www.hpe.com/support/ProLiantServers-Warranties

HPE Enterprise Servers

www.hpe.com/support/EnterpriseServers-Warranties

HPE Storage Products

www.hpe.com/support/Storage-Warranties

HPE Networking Products

www.hpe.com/support/Networking-Warranties

Regulatory information

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

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

Additional regulatory information

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

www.hpe.com/info/reach

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

www.hpe.com/info/ecodata

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

www.hpe.com/info/environment

Documentation feedback

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