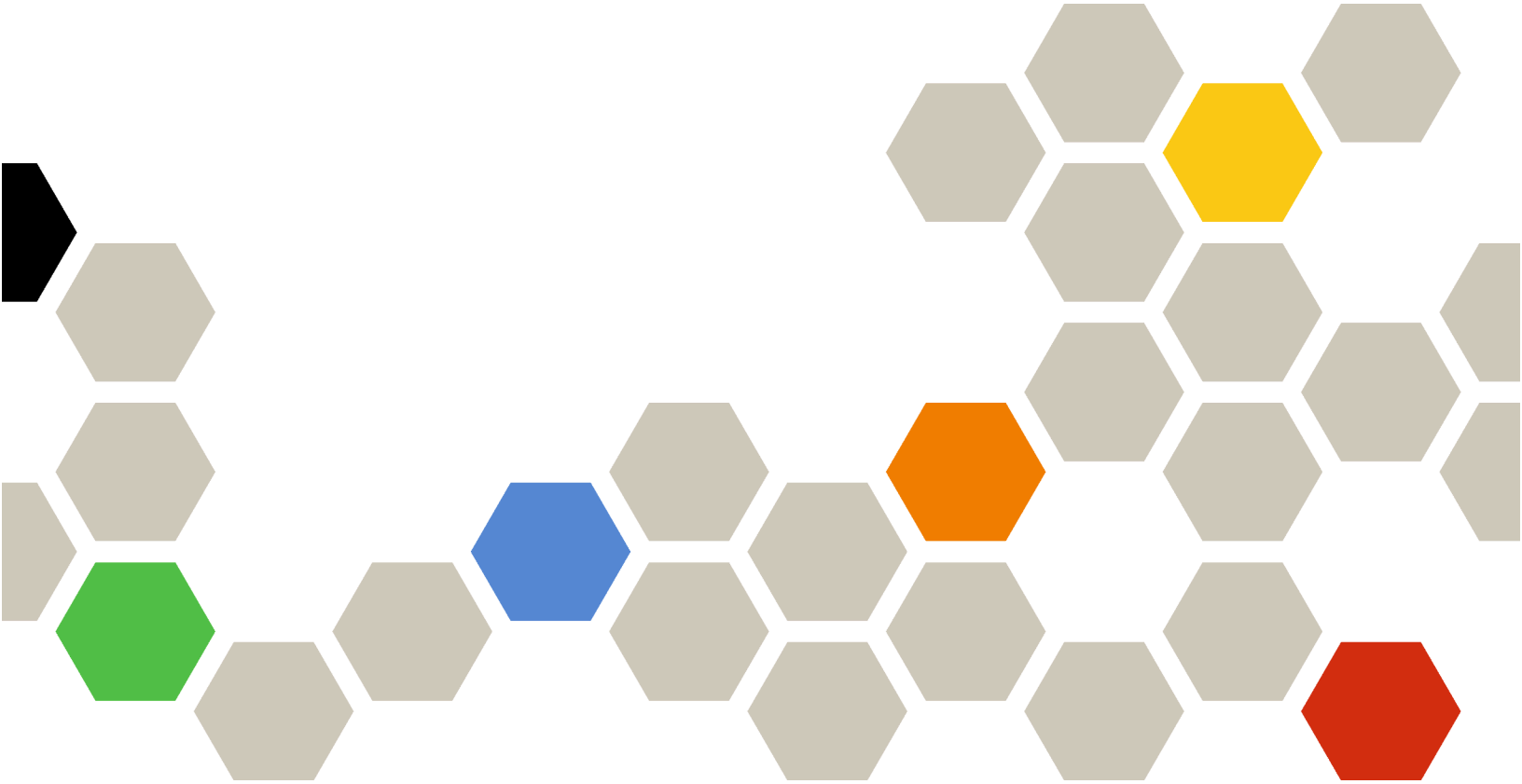




Lenovo XClarity Administrator PowerShell Toolkit Reference



Version 4.0.0

First Edition (February 2023)

© Copyright Lenovo 2015, 2023.

LIMITED AND RESTRICTED RIGHTS NOTICE: If data or software is delivered pursuant a General Services Administration "GSA" contract, use, reproduction, or disclosure is subject to restrictions set forth in Contract No. GS-35F-05925.

Contents

Contents	i	ScheduleTriggerAction class	32
Summary of changes in the PowerShell toolkit	v	ScheduleShowDTO class	33
Chapter 1. PowerShell (LXCAPSTool) toolkit.	1	ServersGlobalSettings class	33
Chapter 2. Installing the Lenovo XClarity Administrator PowerShell toolkit	3	SingleSignOnStatus class	34
Chapter 3. Lenovo XClarity Administrator PowerShell toolkit examples.	5	StoredCredentialData class	35
Chapter 4. XClarity Administrator PowerShell toolkit data types	7	StoredCredentialType enum	36
BackupInfo class	7	SwitchConfigData class	36
BulletinData class	9	WarrantyContractStatus enum	37
LXCABackupsInfo class	10	WarrantyInfo class	37
ConfigData class	11	Chapter 5. PowerShell cmdlet reference.	39
DeviceMaintenanceStatus class	11	Virtual appliance and PowerShell management cmdlets	39
DriveSecureEraseResult class	12	Add-LXCAPSLogHeader	39
FirmwarePackageData class	13	Connect-LXCA	39
FirmwareRepositoryData class	14	Disconnect-LXCA	41
IdentityManagementSystems class	15	Get-LXCACConnection	41
IdentityManagementSystemsPaths class	15	Get-LXCACInformation	42
JobNotes class	16	Invoke-LXCARestMethod	43
LicenseComplianceData class	18	Set-LXCAPSLogLevel	44
LicenseComplianceStatus enum	19	Management-server cmdlets	45
LicenseWarningPeriod class	19	Backup-LXCA	45
LXCABackupRestoreRequest class	20	Copy-LXCABackup	46
LXCABackupRestoreResult class	21	Get-LXCABackups	48
LXCABackupResult class	21	Get-LXCALicenseCompliance	49
LXCACConnection class	22	Get-LXCALicensableDeviceCount	50
LXCACFileDownloadResult class	23	Get-LXCALicenseWarningPeriod	51
LXCASchedule class	24	Get-LXCAManagementServerPackage	51
QuantityLicense class	25	Get-LXCAQuantityLicense	52
QuantityLicenseData class	26	Get-LXCASingleSignon	54
QuantityLicenseFileInstallInfo class	27	Install-LXCAQuantityLicense	55
QuantityLicenseInstallResult class	28	Install-LXCAUpdateManagementServer	56
QuantityLicenseStatus enum	28	Remove-LXCABackup	57
RequestResult class	29	Remove-LXCAQuantityLicense	58
RequestResultStatus enum	29	Restore-LXCABackup	59
ScheduleObject class	30	Set-LXCALicenseWarningPeriod	61
ScheduleRule class	30	Set-LXCASingleSignon	62
		Discovery and management cmdlets	63
		Add-LXCAManagedDevice	63
		Invoke-LXCASystemAction	70
		Remove-LXCAManagedDevice	73
		Set-LXCASystemIPInterface	75
		Set-LXCASystemProperty	77
		Inventory cmdlets	79
		Add-LXCARack	79

Add-LXCARackItem	80	Update-LXCASFirmwareRepository.	132
Get-LXCAActiveStatus	83	Operating-system deployment cmdlets	135
Get-LXCAAvailableMachineType	85	Get-LXCADeployableServer	135
Get-LXCACHassis	86	Get-LXCADeployGlobalSetting	136
Get-LXCACHassisInventory	87	Get-LXCAOSImage	138
Get-LXCACHassisMember.	87	Import-LXCAOSImage	138
Get-LXCAComputeNode	89	Install-LXCAOSImage	140
Get-LXCADeviceMaintenance	90	New-LXCADeployTask	142
Get-LXCAIOModule	90	Remove-LXCAOSImage.	145
Get-LXCARack	91	Set-LXCADeployGlobalSetting	146
Get-LXCARackServer	92	Event, alert, and audit log cmdlets	147
Get-LXCARackSwitch	93	Add-LXCAEventExclusionFilter	147
Get-LXCAScalableComplexSystem	94	Backup-LXCAEventLog	148
Get-LXCAServer.	95	Get-LXCAActiveAlert	149
Get-LXCAServersGlobalSettings	96	Get-LXCAAuditLog	151
Get-LXCASStorage	96	Get-LXCAEvent	152
Get-LXCASStorageNode	97	Get-LXCAEventExclusionFilter	154
Get-LXCAThinkServer	98	Remove-LXCAEventExclusionFilter	155
Remove-LXCARack	99	Start-LXCAEventMonitor	156
Remove-LXCARackItem	100	Job cmdlets	158
Set-LXCADeviceMaintenance	101	Add-LXCAGJobNote	158
Set-LXCARack	102	Get-LXCAGJobNotes	159
Set-LXCARackItem	103	Get-LXCASchedule	159
Set-LXCAServersGlobalSettings	104	Get-LXCATask	161
Resource group cmdlets	105	New-LXCAScheduleObj.	162
Add-LXCAResourceGroupMembers	105	New-LXCAScheduleRule	164
Get-LXCAResourceGroups	106	Remove-LXCAGJobNotes	167
Server configuration cmdlets	107	Remove-LXCATask	168
Get-LXCAConfigPattern.	107	Remove-LXCASchedule	169
Get-LXCAConfigPatternDeployableServer	108	Stop-LXCATask	170
Get-LXCAConfigProfile	109	Security cmdlets	171
Import-LXCAConfigPattern	110	Add-LXCARoleGroup.	171
Install-LXCAConfigPattern	111	Add-LXCASStoredCredential	172
Install-LXCAConfigProfile	112	Add-LXCAUserAccount	174
Remove-LXCAConfigProfileFromSystem	113	Get-LXCAActiveSession	175
Switch configuration cmdlets	114	Get-LXCAEncapsulation	175
Backup-LXCASwitchConfig	114	Get-LXCAIdentityManagementSystems	176
Get-LXCASwitchConfigData	115	Get-LXCARole	177
Get-LXCASwitchConfigFile	117	Get-LXCARoleGroup	178
Import-LXCASwitchConfig.	118	Get-LXCASStoredCredential	178
Remove-LXCASwitchConfigFile	119	Get-LXCAUserAccount	179
Restore-LXCASwitchConfig	120	Install-LXCASStoredCredential	180
Set-LXCASwitchPort	120	Install-LXCATrustedCertificate	181
Firmware update cmdlets	122	Remove-LXCARoleGroup	181
Get-LXCASystemCompliance	122	Remove-LXCASStoredCredential	182
Get-LXCAUpdateCompliancePolicy	123	Remove-LXCAUserAccount	183
Get-LXCAUpdatePackage.	124	Set-LXCAEncapsulation.	184
Import-LXCAUpdatePackage	125	Set-LXCAResourceAccessControl	184
Install-LXCAUpdatePackage	127	Set-LXCAResourceView	185
Join-LXCAUpdateCompliancePolicy	131	Set-LXCARoleGroup	186

Set-LXCASStoredCredential	188	Get-LXCASServiceFile	195
Set-LXCAUserAccount	189	Get-LXCASServiceTickets	196
Set-LXCAUserAccountPassword	191	Get-LXCABulletins	198
Service and support cmdlets.	192	Set-LXCABulletinService	199
Get-LXCADevicesWarranty	192	Notices	cci
Get-LXCAFFDCFile	194	Trademarks	ccii

Summary of changes in the PowerShell toolkit

Lenovo XClarity Administrator supports enhancements to the PowerShell toolkit.

This documentation includes new cmdlets and parameters that apply to the current XClarity Administrator release and later. For information about enhancements to the PowerShell toolkit in other releases, see [Summary of changes in the PowerShell toolkit](#) in the XClarity Administrator online documentation.

Version 4.0.0

The following cmdlets were added or modified.

- [Connect-LXCA](#). Added the **-ChangeExpiredPassword** parameter to prompts for the current password, new password, and confirm password if the password is expired.
- [Get-LXCAServer](#). Added the **-Type** parameter to specify the server type.
- [Get-LXCASystemCompliance](#). Added the EdgeServer value for the **-SystemType** parameter.
- [Get-LXCAUpdateCompliancePolicy](#). Added the EdgeServer value for the **-SystemType** parameter.
- [Install-LXCAUpdatePackage](#). Added the EdgeServer value for the **-SystemType** parameter.
- [Invoke-LXCASystemAction](#). Starts a job that runs in the background to perform the requested power operation. Added the **-Schedule** parameter to create a schedule for performing the power operation. Added the **-DeviceUUID** parameter to replace the following deprecated attributes: **-CMMUuid**, **-ComputeNodeUuid**, **-CanisterUuid**, **-IOSwitchUuid**, **-RackServerUuid**, **-ThinkServerUuid**, **-RackSwitchUuid**, **-StorageUuid**. Added the **-ServerAction** parameter to replace the following deprecated attributes: **-ComputeNodeAction**, **-RackServerAction**, **-ThinkServerAction**.
- [Join-LXCAUpdateCompliancePolicy](#). Added the EdgeServer value for the **-SystemType** parameter.
- (New) [New-LXCAScheduleObj](#). Creates a local schedule object that can be used with certain cmdlets to create a scheduled job on XClarity Administrator.
- (New) [New-LXCAScheduleRule](#). Creates a rule for a recurring schedule object that is to be created using [New-LXCAScheduleObj](#) cmdlet.
- [Set-LXCASystemIPInterface](#). Added the **-ServerUuid** parameter to specify the UUID of a server for which you want to set the IP interface. Deprecated **-ServerUuid** and **-ComputeNodeUuid** parameter; however, these aliases are supported for backward compatibility
- [Set-LXCASystemProperty](#). Added the **-ServerUuid** parameter to specify the UUID of a server for which you want to modify system properties. Deprecated **-ServerUuid**, **-ThinkServerUuid** and **-ComputeNodeUuid** parameter; however, these aliases are supported for backward compatibility

The following data objects were added or modified.

- [LXCASchedule class](#). Contains information about a scheduled job.
- [ScheduleObject class](#). Contains information about the schedule to be used for creating a XClarity Administrator scheduled job.
- [ScheduleRule class](#). Contains information about the scheduling rule.
- [ScheduleTriggerAction class](#). Contains information about update compliance for a specific resource manager.

Chapter 1. PowerShell (LXCAPSTool) toolkit

Lenovo XClarity Administrator PowerShell (LXCAPSTool) toolkit provides a library of cmdlets to automate provisioning and resource management from a Microsoft PowerShell session.

The XClarity Administrator PowerShell toolkit cmdlets provide an interface to XClarity Administrator REST APIs to automate functions such as:

- Logging in to XClarity Administrator
- Managing and unmanaging chassis, servers, storage devices, and top-of-rack switches (devices)
- Collecting inventory data for devices and components
- Deploying an operating-system image to one or more servers
- Configuring servers through the use of Configuration Patterns
- Applying firmware updates to devices
- Managing user accounts
- Monitoring events

Chapter 2. Installing the Lenovo XClarity Administrator PowerShell toolkit

To use the Lenovo XClarity Administrator PowerShell (LXCAPSTool) toolkit, install the toolkit, and start a Microsoft PowerShell session. The PowerShell toolkit is included with the XClarity Administrator appliance and can be downloaded as described below.

Before you begin

Before installing the XClarity Administrator PowerShell toolkit, ensure that the local Microsoft Windows system meets the following prerequisites:


- Microsoft .NET Framework 4.5 is installed. You can download Microsoft .NET Framework 4.5 from the [Microsoft .NET Framework 4.5 webpage](#).
- Transport Layer Security (TLS) v1.2 or v1.3 is enabled on the system. For more information, see [Enable Transport Layer Security \(TLS\) 1.2 overview - Configuration Manager | Microsoft Docs](#) webpage
- The local system has network connectivity to the XClarity Administrator management server.
- You must be logged in as a user with administrator authority to install and uninstall the PowerShell toolkit
- The XClarity Administrator PowerShell toolkit supports the following PowerShell versions (see the [Installing PowerShell on Windows](#) webpage).
 - XClarity Administrator v3.1 and later requires PowerShell 4, 5.1 or PowerShell Core 7.0.3 or later.
 - XClarity Administrator v3.0 requires PowerShell 3, 4, 5.1 or PowerShell Core 7.0.1 or later.

Note: PowerShell Core 7.0 is updated frequently by Microsoft. The XClarity Administrator PowerShell toolkit is aligned with the latest stable version of PowerShell that is available when the XClarity Administrator PowerShell toolkit version is released.

Procedure

- **Using PowerShell Get**Run the following command from a PowerShell session to download and install the toolkit. For more information, see the [PowerShell Gallery – LXCAPSTool webpage](#).
`Install-Module -Name LXCAPSTool`
- **Using the InstallLXCAPSTool.bat file:**

Complete the following steps to download and install the toolkit.

1. Download the toolkit by clicking the user-actions menu ( ADMIN_USER) on the XClarity Administrator title bar, clicking **Resources**, and then clicking **PowerShell toolkit Download** from the dialog.
2. Unzip the package into a local directory.
3. Run the installation .BAT file as administrator to install the tool.
`unzip_directory\InstallLXCAPSTool.bat`

When the installation script is started, you are prompted to choose whether to install for the current user or for all users.

4. Choose the user that you want to use to install the toolkit. Enter **C** for the current user or **A** for all users.
5. Start a PowerShell session.
6. Validate that the module was imported by running the following cmdlet.

```
Get-Help *LXCA*
```

A list of all of the XClarity Administrator cmdlets is returned.

```
PS C:\> get-Help *LXCA*
```

Name	Category	Module	Synopsis
Get-LXCALicenseCompliance	Cmdlet	LXCAPSTool	Get-LXC...
Get-LXCALicensableDeviceCount	Cmdlet	LXCAPSTool	Get-LXC...
Get-LXCALicenseWarningPeriod	Cmdlet	LXCAPSTool	Get-LXC...
Get-LXCAManagementServerPackage	Cmdlet	LXCAPSTool	Get-LXC...
Get-LXCAQuantityLicense	Cmdlet	LXCAPSTool	Get-LXC...
Install-LXCAQuantityLicense	Cmdlet	LXCAPSTool	Install...
Install-LXCAUpdateManagementServer	Cmdlet	LXCAPSTool	Install...
Remove-LXCAQuantityLicense	Cmdlet	LXCAPSTool	Remove-...
Set-LXCALicenseWarningPeriod	Cmdlet	LXCAPSTool	Set-LXC...

- Use the `Connect-LXCA` cmdlets to connect to your XClarity Administrator server, for example:

```

$user = "USERID"
$password = ConvertTo-SecureString "Password" -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential($user, $password)
Connect-LXCA 192.0.2.0 -Credential $cred -SkipCertificateCheck

```

After you finish

- You can get help for any PowerShell cmdlets using the `Get-Help` cmdlet.
 - `Get-Help *LXCA*`. Lists all PowerShell cmdlets
 - `Get-Help <cmdlet_name>`. Shows brief help for the specified cmdlet
 - `Get-Help <cmdlets_name> -full`. Shows detailed help for the specified cmdlet
- To uninstall the PowerShell toolkit, run the following command as administrator:

```

unzip_directory\RemoveLXCAPSTool.bat

```
- To install a later version, uninstall the PowerShell toolkit, and then install the later version.

Chapter 3. Lenovo XClarity Administrator PowerShell toolkit examples

Several sample scripts are available to help you to begin using the Lenovo XClarity Administrator PowerShell toolkit to manage devices.

The sample scripts are location in the following directory:
`unzip_directory\Sample`

The following scripts are included in the Sample directory:

- `End_to_End.ps1`
Demonstrates how to set up a single script to manage rack servers, update firmware, deploy Configuration Patterns, and deploy operating systems to those rack servers.
- `BulkImport_And_Configuration.ps1`
Demonstrates how to use the [Add-LXCAManagedDevice -BulkImportFile](#) cmdlet to manage multiple systems using bulk import and then use the [Install-LXCAConfigPattern](#) cmdlet configure managed systems using Configuration Patterns
- `Manage_Systems.ps1`
Demonstrates how to use the [Add-LXCAManagedDevice](#) cmdlet to manage multiple devices and then use the [Invoke-LXCASystemAction](#) cmdlet to restart the managed devices.
- `View_Systems.ps1`
Demonstrates how to use the [Get-LXCACHassis](#), [Get-LXCAComputeNode](#), and [Get-LXCARackServer](#) cmdlets to view the managed devices.
- `View_Events.ps1`
Demonstrates how to use the [Get-LXCAEvent](#) cmdlet to view the management server and hardware events.
- `Monitor_Events.ps1`
Demonstrates how to use the [Start-LXCAEventMonitor](#) cmdlet to monitor the critical, fatal and warning events and to send an email when a new event occurs.
- `Config_Systems.ps1`
Demonstrates how to use the [Install-LXCAConfigPattern](#) cmdlet to deploy Configuration Patterns to target devices.
- `Firmware_Update.ps1`
Demonstrates how to use the [Install-LXCAUpdatePackage](#) cmdlet to install firmware updates on selected systems based on compliance policies.
`Management_Server_Update_Update.ps1`
Demonstrates how to use the [Import-LXCAUpdatePackage](#) and [Invoke-LXCARestMethod](#) cmdlets to install management-server updates.
- `OSDeployment\OS_Deployment.ps1`
Demonstrates how to use the [New-LXCADeployTask](#) and [Install-LXCAOSImage](#) cmdlets to install operating systems on managed servers.
- `OSDeployment\OS_Deployment_custoImage.ps1`

Demonstrates how to use the [New-LXCADeployTask](#) and [Install-LXCAOSImage](#) cmdlets to deploy a custom operating-system image to a managed server.

- OSDeployment\OS_Deployment_MultipleTargets_MultipleJobs.ps1

Demonstrates how to use the [New-LXCADeployTask](#) and [Install-LXCAOSImage](#) cmdlets to deploy an operating-system image to multiple managed servers using multiple jobs.

- OSDeployment\OS_Deployment_MultipleTargets_SingleJob.ps1

Demonstrates how to use the [New-LXCADeployTask](#) and [Install-LXCAOSImage](#) cmdlets to deploy an operating-system image to multiple managed servers using a single job.

Other sample scripts are included as well.

Chapter 4. XClarity Administrator PowerShell toolkit data types

The following data types (classes) are used by the Lenovo XClarity Administrator PowerShell toolkit cmdlets.

BackupInfo class

This data object contains information about a specific Lenovo XClarity Administrator backup.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public DateTime BackupDate { get; set; }

Date and time when the backup was created

public string BackupRequestedBy { get; set; }

Name of the user that created the backup

public int Chassis { get; set; }

Number of managed chassis in the backup

public string Checksum { get; set; }

Checksum that is used to verify the backup content

public bool IsDhcpEnabled { get; set; }

Indicates whether DHCP mode is enabled on the XClarity Administrator instance that was backed up.

public string DhcpEth { get; set; }

Network interface that uses DHCP (for example, "eth0")

If DHCP mode is disabled on the XClarity Administrator instance that was backed up, this value is empty.

public string DhcpMac { get; set; }

MAC address of the network interface that uses DHCP

If DHCP mode is disabled on the XClarity Administrator instance that was backed up, this value is empty.

public string Eth0IPv4 { get; set; }

IPv4 address of the eth0 interface

public string Eth0IPv6 { get; set; }

IPv6 address of the eth0 interface

public string Eth1IPv4 { get; set; }

IPv4 address of the eth1 interface

public string Eth1IPv6 { get; set; }

IPv6 address of the eth1 interface

public string FileName { get; set; }

Name of the XClarity Administrator backup

public long FirmwareSize { get; set; }

Total size of all firmware-update packages that are included in the backup

public string IpRedirect { get; set; }

IP address that is used to access the source XClarity Administrator

This IP address is used to redirect the browser after import is completed.

public string Label { get; set; }

User-defined label for the backup

public string[] Location { get; set; }

Location of the XClarity Administrator backup. This can be one or more of the following values.

- **internal**. XClarity Administrator local repository
- **{mount_point}**. Mount point for the remote share that is already configured in the XClarity Administrator instance.

public string ManagementServerUuid { get; set; }

UUID of the target XClarity Administrator instance that was backed up

public string ManagementServerVersion { get; set; }

Version of the XClarity Administrator instance that was backed up

public long OslmagesSize { get; set; }

Total size of all operating-system images that are included in the backup

public int OslmagesQty { get; set; }

Number of operating-system images in the backup

public int Patterns { get; set; }

Number of configuration patterns in the backup

public int Racks { get; set; }

Number of managed racks in the backup

public int Servers { get; set; }

Number of managed servers in the backup

public long Size { get; set; }

Total size of the backup package

public int Storages { get; set; }

Number of managed store devices in the backup

public int Switches { get; set; }

Number of managed switches in the backup

public string UriRedirect { get; set; }

DNS address that is used to access the source XClarity Administrator, if applicable

If the DNS address exists, it is used to redirect the browser after import is completed.

public int Users { get; set; }

Number of XClarity Administrator users in the backup

public string Uuid { get; set; }

UUID of the backup file

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

BulletinData class

This data object contains information about a Lenovo bulletin.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public DateTime? PublishingDate { get; set; }

Date when the bulletin was published

public string Id { get; set; }

Bulletin ID

public string ValidDateRange { get; set; }

Interval when this bulletin comes into action

public DateTime? StartDate { get; set; }

Date when this bulletin comes into action

public DateTime? EndDate { get; set; }

Date when this bulletin ends

public string Message { get; set; }

Bulletin message

public string DocLink { get; set; }

URL to detailed information about the bulletin

public string Severity { get; set; }

Bulletin severity

public string Category { get; set; }

Bulletin category

public string Component { get; set; }

Component associated with this bulletin

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

LXCABackupsInfo class

This data object contains information about all Lenovo XClarity Administrator backups.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Inheritance > Object > [RequestResult class](#)

Properties

public BackupInfo[] BackupInfos { get; set; }

Contains information about all XClarity Administrator backups.

If the retrieving backups process failed, the property's value is null.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

ConfigData class

This data object contains information about each switch-configuration data files.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string FileId { get; set; }

ID of the configuration-data file

public string FileName { get; set; }

Name of the configuration-data file

public string Date { get; set; }

Date when the configuration-data file was created

public string Comment { get; set; }

Comment about the configuration-data file

public string Size { get; set; }

Size of the configuration-data file

public string SwitchUuid { get; set; }

UUID of the switch

public string FWVersion { get; set; }

Version of firmware that is running on the switch

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function.

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance.

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

DeviceMaintenanceStatus class

This data object contains information about maintenance Status for a device.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string UUID { get; set; }

UUID of the device.

public string State { get; set; }

Maintenance mode. This can be one of the following values.

- **Enabled**
- **Disabled**
- **Unknown**

public DateTime? StartDate { get; set; }

If **State** is set to “Enabled,” this property specifies when the maintenance starts.

This property is translated into the time zone of the local system.

public DateTime? EndDate { get; set; }

If **State** is set to “Enabled,” this property specifies when the maintenance ends.

This property is translated into the time zone of the local system.

public string Message { get; set; }

If **State** is set to “Unknown,” this property contains an explanation of this state; otherwise, it is an empty string.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function.

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance.

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

DriveSecureEraseResult class

This data object contains information about the status of the drives secure erase request.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Inheritance -> Object -> [RequestResult](#)

Properties

public string TaskID { get; set; }

If the request was accepted by the Lenovo XClarity Administrator, this property contains the ID of the job that was created, null otherwise.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

FirmwarePackageData class

This data object contains information about a firmware package.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string MachineType { get; set; }

Machine type that applies to this firmware package

public string Name { get; set; }

Readable name of this firmware package

public DateTime ReleaseDate { get; set; }

Release date for this firmware-updates package

public string BuildNumber { get; set; }

Build number of this firmware-updates package

public int TotalSize { get; set; }

Total size, in bytes, of this firmware-updates package

public int DownloadedSize { get; set; }

Size, in bytes, of the downloaded updates in this firmware-updates package

public string FixId { get; set; }

ID of this firmware-updates package

public string PayloadFilename { get; set; }

File name of this firmware-updates package

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

FirmwareRepositoryData class

This data object contains information about the firmware-updates repository.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string Action { get; set; }

Action that was run on the repository.

public int Progress { get; set; }

Progress of the action was run on the repository.

public FirmwarePackageData[] FirmwarePackages { get; set; }

Packages that are contained by the repository.

Methods

public string[] FixIds();

Retrieves all distinct firmware packages IDs (fixIDs) from the repository.

public string[] FixIds(string machineType);

Retrieves all distinct firmware packages IDs (fixIDs) from the repository that are mapped to a specific machine type.

public string[] FixIds(string machineType, string fixIdPattern);

Retrieves all distinct firmware packages IDs (fixIDs) from the repository that are mapped to a specific machine type and contains a package ID substring.

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();
Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();
Returns a string that represents the current object.

(Inherited from System.Object)

IdentityManagementSystems class

This data object contains information about the identity-management systems configuration.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public int Port { get; set;}
Port for the identity-management system.

public string Name { get; set;}
Type of identity-management system.

public string Host { get; set;}
Hostname of the identity-management systems.

public IdentityManagementSystemsPaths[] Paths { get; set;}
Information about each path to the identity-management system.

Methods

public virtual bool Equals (object obj);
Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();
Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();
Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();
Returns a string that represents the current object.

(Inherited from System.Object)

IdentityManagementSystemsPaths class

This data object contains a path of the identity-management system.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string Id { get; set;}

Path ID.

public string ApplicationId { get; set;}

Application ID.

public string Safe { get; set;}

Path's safe.

public string Folder { get; set;}

Path's folder.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

JobNotes class

This data object contains information about a note that is attached to a Lenovo XClarity Administrator job.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string CreatedBy { get; set; }

Name of the user that added the note or changed the job state.

public string NoteId { get; set; }

ID of this note.

public int JobUID { get; set; }

UID of the job that has attached this note.

public string NewState { get; set;}

Current Job state. This can be one of the following values.

- **Aborted**
- **Blocked**

- Cancelled
- CancelledWithError
- CancelledWithWarning
- Cancelling
- Complete
- CompleteWithError
- CompleteWithWarning
- Expired
- Initializing
- Interrupted
- InterruptedWithError
- InterruptedWithWarning
- Investigating
- Pending
- Resolved
- Running
- RunningWithError
- RunningWithWarning
- Skipped
- Stopped
- StoppedWithError
- StoppedWithWarning
- Unknown
- Uploading
- Validating
- Waiting

public string PreviousState { get; set;}

Previous state of the job. This can be one of the following values.

- Aborted
- Blocked
- Cancelled
- CancelledWithError
- CancelledWithWarning
- Cancelling
- Complete
- CompleteWithError
- CompleteWithWarning
- Expired
- Initializing
- Interrupted
- InterruptedWithError
- InterruptedWithWarning
- Investigating
- Pending
- Resolved
- Running
- RunningWithError
- RunningWithWarning
- Skipped
- Stopped
- StoppedWithError
- StoppedWithWarning
- Unknown
- Uploading

- **Validating**
- **Waiting**

public string Text { get; set; }

Note text.

public DateTime? TimeStamp { get; set; }

Time stamp when the note was entered or the state was changed.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function.

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance.

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

LicenseComplianceData class

This data object contains information about whether XClarity Administrator is license compliant.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Inheritance > Object > [RequestResult](#)

Properties

public LicenseComplianceStatus Status { get; set; }

License compliance status.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function.

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance.

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

LicenseComplianceStatus enum

This data object contains the license-compliance status.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Fields

NotCompliant

The number of active licenses is less than the number of managed devices that require licenses for advanced functions for more than 90 days. Advanced functions are disabled.

Compliant

The number of active licenses is greater than or equal to the number of managed devices that require licenses for advanced functions, or the 90-day grace period is enabled.. Advanced functions are enabled.

Registered

Lenovo XClarity Administrator is registered. The number of active licenses is less than the number of allowed managed devices that require licenses for advanced functions. Advanced functions are disabled.

Unknown

The license-compliance status is unknown.

LicenseWarningPeriod class

This data object contains information about the license warning period.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Inheritance -> Object -> [RequestResult](#)

Properties

public int WarningPeriod { get; set;}

The license warning period value, which determines the number of days before license expires when Lenovo XClarity Administrator triggers a warning.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function.

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance.

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

LXCABackupRestoreRequest class

This data object contains the operational status when a backup is being restored or imported.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public bool RestoreOS { get; set;}

Indicates whether to restore operating-system images.

public bool RestoreManagedDevices { get; set;}

Indicates whether to restore device inventory.

public bool RestoreFirmware { get; set;}

Indicates whether to restore firmware and OS device-driver updates.

public bool RestoreNetworkSettings { get; set;}

Indicates whether to restore network settings.

public string FileName { get; set;}

Name of the XClarity Administrator backup.

public string Uuid { get; set;}

UUID of the backup file.

public string Status { get; set;}

Status of the restore/import operation.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

LXCABackupRestoreResult class

This data object contains information about the results of a Lenovo XClarity Administrator restore or import process.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Inheritance > Object > [RequestResult class](#)

Properties

public BackupRestoreRequest RestoreInfo { get; set;}
Operational status when a backup is restored or imported

Methods

public virtual bool Equals (object obj);
Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();
Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();
Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();
Returns a string that represents the current object.

(Inherited from System.Object)

LXCABackupResult class

This data object contains information about the results of a Lenovo XClarity Administrator backup process.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Inheritance > Object > [RequestResult](#)

Properties

public String BackupId { get; set;}
UUID of the backup package.

If the backup process failed, the property's value is null.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

LXCAConnection class

This data object contains information about the current connection to Lenovo XClarity Administrator.

This object is read-only and can be instantiated only using the [Connect-LXCA](#) cmdlet.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Common

Assembly: Lenovo.SysMgmt.LXCA.Integration.Common

Properties

public string Id { get; }

Session ID of the current connection to XClarity Administrator

public bool IsConnected { get; }

Indicates whether this user session is connected to XClarity Administrator. This can be one of the following values.

- **true.** This user session is connected to XClarity Administrator.
- **false.** This user session is dropped.

public string Host { get }

XClarity Administrator IP address or hostname

public int Port { get; }

XClarity Administrator port that is used for this connection

public int Timeout { get; }

Number of seconds that the cmdlet waits for an HTTPS response from XClarity Administrator

public string UserName { get; }

Name of the user that is authenticated to this session

public DateTime? LoginTime { get; }

Timestamp when the connection to XClarity Administrator was established

public DateTime? LogoutTime { get; }

Timestamp when the connection to XClarity Administrator was dropped

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

LXCAFileDownloadResult class

This data object contains the full path to the file that was downloaded from Lenovo XClarity Administrator to the local system.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Inheritance > Object > [RequestResult](#)

Properties

public String FileName{ get; set;}

Full path of the downloaded file.

If the download process fails, the property value is null.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function.

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance.

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

LXCASchedule class

This data object contains information about a scheduled job.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string Id { get; set; }

ID of the scheduled job.

public string Name { get; set; }

Name of the scheduled job.

public string CreationDate { get; set; }

Timestamp when the scheduled job was created.

public string ExecDate { get; set; }

Timestamp when the scheduled job is to run next.

public string NextExecDate { get; set; }

Timestamp for the next scheduled run time of the job.

public string State { get; set; }

State of the scheduled job. This can be one of the following values.

- **ACTIVE**
- **ENDED**
- **PAUSED**

public string Type { get; set; }

Type of schedule. This can be one of the following values.

- **ONE_TIME**. The job runs one time on all target devices. If the specified start and end date are in the past, the job runs imminently.
- **RECURRING**. The job runs on the specified dates and times on all target devices.
- **EVENT_TRIGGERED**. The job runs when a specified event occurs. This job runs only on the device that generated the event.

public string CreatedBy { get; set; }

Username that created the scheduled job.

public string[] ComponentIDs { get; set; }

List of UUIDs of the devices or resource groups that are the target for the action.

public string Description { get; set; }

Description of the scheduled job.

public string[] JobIDs { get; set; }

List of IDs for all jobs that were started from the schedule.

public string LastExecDate { get; set; }

Timestamp when the scheduled job was last run.

public bool? MatchEverything { get; set; }

Indicates whether the action is to be run against all managed devices. This can be one of the following values.

- **true**. The action is to be run against all managed devices
- **false**. The action is run against only the managed device that is specified by the target attribute.

public string Targets { get; set; }

Device or resource group name that is the target for the job.

If not applicable, the value is NONE.

If more, there is more than one target, the value is Multiple Targets.

public ScheduleTriggerAction TriggerAction { get; set; }

Information about the action that you want to schedule.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

QuantityLicense class

This data object contains information about all installed licenses on a Lenovo XClarity Administrator server.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Common

Assembly: Lenovo.SysMgmt.LXCA.Integration.Common

Inheritance > Object > [RequestResult](#)

Properties

public QuantityLicenseData[] Licenses { get; set; }

Information about all installed licenses.

This field is an empty array if no license is installed on XClarity Administrator or if the request failed.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function.

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance.

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

QuantityLicenseData class

This data object contains information about a license.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Common

Assembly: Lenovo.SysMgmt.LXCA.Integration.Common

Properties

public DateTime StartDate { get; set; }

Date when this license starts.

public DateTime EndDate { get; set; }

Date when this license ends.

public int RemainingDays { get; set; }

Number of days until this license expires.

public bool IsEnterprise { get; set; }

Indicates whether this is an enterprise license. This can be one of the following values.

- **true**. This is an enterprise license.
- **false**. This is a standard license.

public string Quantity { get; set; }

Number of devices that can be managed under this license. For the Enterprise license, this value is "Unlimited."

public string ProductName { get; set; }

Product name of this license.

public string ProductDescription { get; set; }

Product description of this license.

public int Id { get; set; }

License ID.

public QuantityLicenseStatus Status { get; set; }

License status.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function.

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance.

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

QuantityLicenseFileInstallInfo class

This data object contains information about the installation result of a license-activation key.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string Result { get; set; }

Result of installation process. This can be one of the following values.

- OK
- DUPLICATE
- EXPIRED
- INVALID
- NOTLXCA

public string Reason { get; set; }

Short explanation of the results

public string FileName { get; set; }

File name of the license which was installed or tried to be installed.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

QuantityLicenseInstallResult class

This data object contains information about the installation result of one or more license-activation keys.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Inheritance > Object > [RequestResult](#)

Properties

public QuantityLicenseFileInstallInfo[] InstallInfo{ get; set}

Information about the installation result for each license.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

QuantityLicenseStatus enum

This data object contains the license status.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Fields

Invalid

The license is not valid.

Valid

The license is valid.

AboutToExpire

The license is about to expire.

Unknown

The license status is unknown.

RequestResult class

This data object contains information about the result of the request.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public RequestResultStatus Result { get; set; }

Results of the request. This can be one of the following values.

- **Success.** The request completed successfully.
- **Failed.** The request failed. A descriptive error message was returned.
- **Warning.** The request completed with a warning. A descriptive error message was returned.

public string Message { get; set; }

Additional information to clarify the reason for request result. If the value of **Result** is “Success,” then this field is empty.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function.

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance.

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

RequestResultStatus enum

This data object contains the result of the request.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data.EnumResources

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Fields

Success

The request was completed successfully.

Failed

The request failed.

Warning

The request completed with a warning.

ScheduleObject class

This data object contains information about the schedule to be used for creating a XClarity Administrator scheduled job.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string Name { get; set; }

Schedule name.

public string Type { get; }

Type of schedule. This can be one of the following values. Note that this value is read-only.

- **ONE_TIME**. The job runs one time on all target devices. If the specified start and end date are in the past, the job runs imminently.
- **RECURRING**. The job runs on the specified dates and times on all target devices.
- **EVENT_TRIGGERED**. The job runs when a specified event occurs. This job runs only on the device that generated the event.

public DateTime OccurrenceExecDate { get; set; }

Timestamp when the scheduled job is to run next. This property is required when schedule type is **ONE_TIME**.

public ScheduleRule Rule { get; set; }

Information about the scheduling rules. This property is required when schedule type is **RECURRING**.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

ScheduleRule class

This data object contains information about the scheduling rule.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string Type { get; }

Type of recurring schedule. This can be one of the following values.

- **daily**
- **weekly**
- **monthly**
- **yearly**

public string StartDate { get; set; }

Date and time when the job starts running.

public string EndDate { get; set; }

Date and time when the job stops running.

public int EndAfter { get; set; }

End after a number of occurrences

public bool NoEnd { get; }

Indicates whether the schedule has no end date. This can be one of the following values.

- **true.** The schedule has no end date.
- **false.** (default) The schedule has an end date.

public string[] Days { get; set; }

Days when the job is to run. This can be one or more of the following values.

- **monday**
- **tuesday**
- **wednesday**
- **thursday**
- **friday**
- **saturday**
- **sunday**

public int RecurEvery { get; set; }

Interval for running the job (for example, specify 2 for every two weeks).

public string DayOfWeek { get; set; }

Day of each week when the job is to run. This property is required when rule type is weekly, monthly, or yearly. This can be one of the following values.

- **monday**
- **tuesday**
- **wednesday**
- **thursday**
- **friday**
- **saturday**
- **sunday**

public int WeekOfMonth { get; set; }

Week of each month when the job is to run. This property is required when rule type is monthly. This can take values between 1 - 5.

public string MonthOfYear { get; set; }

Month of each year when the job is to run. This property is required when rule type is yearly. This can be one of the following values.

- **january**
- **february**
- **march**
- **april**

- **june**
- **july**
- **august**
- **september**
- **october**
- **november**
- **december**

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

ScheduleTriggerAction class

This data object contains information about update compliance for a specific resource manager.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string Title { get; set; }

Type of action to be performed. This can be one of the following value.

- **CUSTOM_REST**
- **REST**

public string RestURL { get; set; }

REST URL of the job to be run.

public string RestMethod { get; set; }

REST method of the action to be run.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

ScheduleShowDTO class

This data object contains information about.

Namespace: Lenovo.XClarityOrchestrator.PowerShell.Data.Alerts

Assembly: Lenovo.XClarityOrchestrator.PowerShell.Data

Properties

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

ServersGlobalSettings class

This data object contains global inventory-configuration settings for servers.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string SMARTDataRetrievalPreference { get; set;}

Indicates whether the collection of SMART data for ThinkAgile and ThinkSystem servers is enabled or not. This can be one of the following values.

- Enabled. SMART data collection is enabled.

- Disabled. SMART data collection is disabled.

public int SecureEraseBatchSize { get; set;}

Maximum number of servers on which the secure-erase operation can be performed at one time.

Note: the value is in 3 - 100 range.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

SingleSignOnStatus class

This data object contains information about the single sign-on status.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Inheritance > Object > [RequestResult class](#)

Properties

public string Uuid { get; set;}

Device or appliance UUID.

If reading of the single sign-on status fails, the value of this field is null.

public bool IsSsoEnabled { get; set;}

Indicates whether single sign-on is enabled or disabled.

If reading of the single sign-on status fails, the value of this field is always false and must be ignored.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

StoredCredentialData class

This data object contains information about the stored credentials.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string Id { get; set; }

ID of the stored credential.

In case of error, the **Id** is null, and the **Description** contains the failure explanation.

public string UserName { get; set; }

User name for this stored credential.

public int Description { get; set; }

Description of this stored credential.

In case of error, the **Description** contains the failure explanation and **Id** is null.

public StoredCredentialType Type { get; set; }

Type of this stored credential.

public bool IsValid { get; }

Returns true if this stored credential's **Id** is not null.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function.

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance.

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

StoredCredentialType enum

This data object specifies the stored-credential types.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Fields

Management

The stored credential is used as a management credential.

Recovery

The stored credential is used as a recovery credential.

Unknown

The stored credential's type is not defined.

SwitchConfigData class

This data object contains information about switch-configuration backup files.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public ConfigData ConfigDatas { get; set; }

Information about each configuration-data file

public string HostName { get; set; }

Hostname of the switch

public int IpAddress { get; set; }

IP address of the switch

public string Type { get; set; }

Type of the switch

public string Uuid { get; set; }

UUID of the switch

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function.

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance.

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

WarrantyContractStatus enum

This data object specifies the warranty contract status.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data.EnumResources

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Fields

Active

The warranty contract is active.

Expired

The warranty contract is expired.

NotAvailable

The warranty contract information is not available.

WarrantyInfo class

This data object contains information about warranty data for a managed device.

Namespace: Lenovo.SysMgmt.LXCA.Integration.Data

Assembly: Lenovo.SysMgmt.LXCA.Integration.Data

Properties

public string DeviceName { get; set; }

Name that is assigned to the device.

public string ProductName { get; set; }

Commercial name of the device.

public string MachineTypeAndModel { get; set; }

Machine type and model of the device.

public string SerialNumber { get; set; }

Serial number of the device.

public string MachineUUID { get; set; }

Device UUID.

public string WarrantyNumber { get; set; }

Warranty contract ID.

public DateTime? WarrantyStartDate { get; set; }

Warranty start date. If the start date is not available, then this property is not defined.

public DateTime? WarrantyExpirationDate { get; set; }

Warranty expiration date. If the expiration date is not available, then this property is not defined.

public [WarrantyContractStatus](#) WarrantyStatus { get; set; }

Warranty contract status. This can be one of the following values.

- **Active**. The warranty contract is active
- **Expired**. The warranty contract is expired
- **NotAvailable**. The warranty contract information is not available

public string[] Groups { get; set; }

List of names of the resource groups to which the device belongs.

public string[] GroupsUUIDS { get; set; }

List of UUIDs of the resource groups to which the device belongs.

public string DeviceURI { get; set; }

Device URI.

public bool IsWarrantyAvailable { get; set; }

Indicates whether the device has a warranty contract (active or expired). This can be one of the following values.

- **true**. The device has a warranty contract
- **false**. The device does not have a warranty contract

public string ParentUUID { get; set; }

Parent UUID of the parent device. If the device does not have a parent, this value is null.

Methods

public virtual bool Equals (object obj);

Determines whether the specified object is equal to the current object.

(Inherited from System.Object)

public virtual int GetHashCode ();

Serves as the default hash function

(Inherited from System.Object)

public Type GetType ();

Gets the System.Type of the current instance

(Inherited from System.Object)

public virtual string ToString ();

Returns a string that represents the current object.

(Inherited from System.Object)

Chapter 5. PowerShell cmdlet reference

These are the cmdlets that are provided by the Lenovo XClarity Administrator PowerShell toolkit.

Virtual appliance and PowerShell management cmdlets

The following PowerShell cmdlets are available for managing the Lenovo XClarity Administrator virtual appliance and PowerShell sessions.

Add-LXCAPSLogHeader

This cmdlet writes a user message as a header in the PS log file, which can be used to mark the start and end of a block in a script.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCAPSLogHeader [-UserMessage String]  
                    [-MachineId]  
                    [-Identity]  
                    [CommonParameter]
```

Parameters

-UserMessage *String*

Specifies the message in the header that the user wants to add in PS log file.

-MachineId

If specified, adds the machine identity (machine name) in the header that is to be written in PS log file.

-Identity

If specified, adds the user identity (user name and domain if is the case) in the header that is to be written in PS log file.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return any data.

Examples

The following example writes a full message header in PS log file. **-UserMessage** contains the full name of the current script

```
d-LXCAPSLogHeader -UserMessage "Starting '$PSCmdPath'" -Identity -MachineId
```

Connect-LXCA

This cmdlet creates a connection to the Lenovo XClarity Administrator server that can be used by other cmdlets.

Use the [Disconnect-LXCA](#) cmdlet to disconnect from XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Connect-LXCA -Host String
             -Credential PSCredential
             [-Port Int]
             [-Proxy WebProxy]
             [-Timeout Int]
             [-SkipCertificateCheck]
             [-ChangeExpiredPassword]
             [CommonParameter]
```

Parameters

-Host *String*

Specifies the HTTPS address (IP address or hostname) for the XClarity Administrator server.

-Credential *PSCredential*

Specifies a credential object that includes the credentials for logging in to the XClarity Administrator server.

Tip: To obtain the credential object, use the `Get-Credential` cmdlet.

-Port *Int*

Specifies the port on the XClarity Administrator server to be used for the connection. The default value is 443.

-Proxy *WebProxy*

Specifies a Web service proxy object that includes proxy connection and authentication information.

Tip: To obtain a proxy object, use the `New-WebServiceProxy` cmdlet.

-Timeout *Int*

Specifies the number of seconds that the cmdlet waits for an HTTPS response from the XClarity Administrator server. By default, the time-out value is 600 seconds.

-SkipCertificateCheck

If specified, this cmdlet does not check for server certificates. Otherwise, the connection is made only if the XClarity Administrator server contains a valid certificate that is signed by a trusted certification authority.

-ChangeExpiredPassword

If specified and the password is expired, this cmdlet prompts for the current password, new password, and confirm password.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns an [LXCAClass](#) object that contains the information for the XClarity Administrator server connection.

Examples

The following example retrieves the credentials that are then passed to this cmdlet to connect to the XClarity Administrator server with IP address 192.0.2.0.

```
$cred = Get-Credential
$connection = Connect-LXCA -Host 192.0.2.0 -Credential $cred
```


The following example changes the expired password.

```
$LxcaIP = "192.0.2.0"
```

```
# get credential for user with expired password  
$cred = Get-Credential
```

```
Connect-LXCA -Host $LxcaIP -Credential $cred -SkipCertificateCheck -ChangeExpiredPassword
```

Disconnect-LXCA

This cmdlet disconnects from the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Disconnect-LXCA [-Connection LXCAConnection]  
                [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

The XClarity Administrator management server is no longer connected through PowerShell. This cmdlet does not return an object.

Examples

The following example connects and then disconnects from a XClarity Administrator server.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
#Other actions
```

```
Disconnect-LXCA
```

Get-LXCAConnection

This cmdlet retrieves all available PowerShell connections with the Lenovo XClarity Administrator server that can be used by other cmdlets. Use `Disconnect-LXCA` to disconnect from the XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAConnection [CommonParameter]
```

Parameters

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more [LXCAClass](#) objects.

Examples

The following example retrieves a list of all available PowerShell sessions and then disconnects them from the XClarity Administrator server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$connections = Get-LXCAClass
foreach ($conn in $connections)
{
    Disconnect-LXCA -Connection $conn
}
```

Get-LXCAClass

This cmdlet retrieves information about Lenovo XClarity Administrator and PowerShell tool, including the product name, version, and build number.

This cmdlet gets information about the Lenovo XClarity Administrator server, including product name, version, and build number.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAClass [-Connection LXCAClass]
               [CommonParameter]
```

Parameters

-Connection *LXCAClass*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns an [LXCAClass](#) object.

Examples

The following example retrieves basic information about the XClarity Administrator and PowerShell tool.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred
```

```
$info = Get-LXCAInformation
```

```
Disconnect-LXCA
```

Invoke-LXCARestMethod

This cmdlets invokes a Lenovo XClarity Administrator REST API directly.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Invoke-LXCARestMethod -Connection LXCACConnection  
                    -ResourceUrl String  
                    -Method String  
                    [-AdditionalHeaders Hashtable]  
                    [-RequestBody String]  
                    [-OutFile String]  
                    [CommonParameter]
```

```
Invoke-LXCARestMethod -Connection LXCACConnection  
                    -ResourceUrl String  
                    -Method String  
                    -UploadFilePath String[]  
                    -UploadFormName String  
                    [-AdditionalHeaders Hashtable]  
                    [-OutFile String]  
                    [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-AdditionalHeaders *Hashtable*

Specifies the additional request headers.

The following headers *are not* supported:

- Accept
- Connection
- Conect-Length
- Content-Type
- Date
- Expect
- Host
- If-Modified-Since
- Range
- Referer
- Transfer-Encoding
- User-Agent

-Method *String*

Specifies the HTTP request method. You can specify one of the following methods:

- DELETE
- GET
- PATCH
- POST
- PUT

-OutFile *String*

Specifies the output directory and file name in which to return the response body. If you do not specify a directory, the output file is saved in the current location.

-RequestBody *String*

Specifies the request body.

-ResourceUrl *String*

Specifies the resource URL.

-UploadFilePath *String[]*

Specifies one or more file paths from which a file is to be uploaded.

-UploadFormName *String*

Specifies the field name in the POST method when uploading a file.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

If you specify the **-OutFile** parameter, this cmdlet returns an `LXCARestResponseOutFile` object that includes the HTTP response headers, HTTP status code, and full directory and file name of the output file.

If you *do not* specify the **-OutFile** parameter, this cmdlet returns an `LXCARestResponse` object that contains the HTTP response headers, HTTP status code, and response body.

Examples

- The following example sends a GET /nodes request with an additional header (Item=10).

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$url = "/nodes"
$headers = @{'Item'='10'};
$restResult = Invoke-LXCARestMethod -ResourceUrl $url -Method GET -AdditionalHeaders $additionalHeader
$restResult.HttpStatusCode
$restResult.Headers
$restResult.Body
```

```
Disconnect-LXCA
```

- The following example uploads an update policy file `cmm.xml` to the management server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$url = "compliancePolicies?action=import"
$restResult = Invoke-LXCARestMethod -Method POST -ResourceUrl $url -UploadFilePath C:\cmm.xml`
                    -UploadFormName "uploadedfiles[]"

$restResult.HttpStatusCode
$restResult.Headers
$restResult.Body
```

```
Disconnect-LXCA
```

Set-LXCAPSLogLevel

This cmdlet modifies the log level of the PowerShell cmdlets for the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCAPSLogLevel -LogLevel String  
                  [CommonParameter]
```

Parameters

-LogLevel *String*

Specifies the type of errors that are added to the log. You can specify one of the following log levels:

- **Error.** Messages with a severity of Error are added to the log.
- **Warn.** Messages with a severity of Error and Warning are added to the log.
- **Info.** Messages with a severity of Error, Warning, and Info are added to the log. This is the default level.
- **Debug.** Messages of all severities and additional information that is required to debug issues are added to the log.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet modifies the level of errors that are added to the PowerShell log. No objects are returned.

Examples

The following example adds messages of all severities and additional debug information to the log.

```
Set-LXCAPSLogLevel -LogLevel Debug
```

Management-server cmdlets

The following PowerShell cmdlets are available for performing management-server update functions and retrieving licenses.

Backup-LXCA

Use this cmdlet to create a backup a Lenovo XClarity Administrator system data, settings, and imported files such as operating-system images, firmware updates, and OS device drivers.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Backup-LXCA [-Connection LXCACConnection]  
            -Label SecureString  
            -Password SecureString  
            [-RemoteShareDestination String]  
            [-IncludeOS]  
            [-IncludeFirmware]  
            [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Label *SecureString*

Name of the backup

-Password *SecureString*

Passphrase that is required for decrypting and restoring the backup.

Attention: If you forget the passphrase, it cannot be recovered.

-RemoteShareDestination *String*

Remote share location where the backup will be created.

If not specified, the backup is created in the XClarity Administrator internal repository.

-IncludeOS

If specified, the backup includes operating system images.

-IncludeFirmware

If specified, the backup includes firmware and OS device-driver updates.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- [Lenovo.SysMgmt.LXCA.Integration.Data.LXCABackupResult](#)

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.LXCABackupResult](#) object that contains information about the backup process result. If the backup process succeeds, the backup id is also returned.

Examples

The following example creates a full backup, including OS images, firmware updates, and OS device driver updates.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $cred -SkipCertificateCheck

$backupPwd = ConvertTo-SecureString "password" -AsPlainText -Force
$result = Backup-LXCA -Label backupLabel -Password $backupPwd -IncludeFirmware -IncludeOS

# Show all the information about the request, including result value, message (if any), and brief
# description of each backup (if any)
$result

Disconnect-LXCA
```

Copy-LXCABackup

Use this cmdlet to copy a Lenovo XClarity Administrator backup from the local repository to a remote share, copy a backup from a remote share to the local repository, or to move a backup from the local repository to a remote share.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Copy-LXCABackup [-Connection LXCACConnection]  
                -Uuid String  
                -ToLocalRepository  
                [CommonParameter]
```

```
Copy-LXCABackup [-Connection LXCACConnection]  
                -Uuid String  
                -SharedLocation String  
                [-Move]  
                [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String*

Specifies the UUID of the backup to copy to the specified destination.

-ToLocalRepository *String*

Specifies the local repository to use as the backup destination.

-SharedLocation *String*

Specifies the location to save the backup. The location must be the mount point for a remote share that is configured in the XClarity Administrator instance.

-Move

If specified, the backup is copied to the shared location and the original copy is deleted.

If not specified, the backup is copied to the shared location and the original copy is left as is.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- [Lenovo.SysMgmt.LXCA.Integration.Data.RequestResult](#)

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.RequestResult](#) object that contains information about the backup copy process

Examples

The following example copies a backup from a shared location to the XClarity Administrator local repository.

```
$cred = Get-Credential  
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck  
  
$(Copy-LXCABackup -Uuid c4e24aab-7f56-4474-99f4-a81c676df128 -ToLocalRepository)  
  
Disconnect-LXCA
```

The following example move a backup from the XClarity Administrator local repository to a shared location.

```
$cred = Get-Credential  
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck  
  
$result = Copy-LXCABackup -Move -Uuid c4e24aab-7f56-4474-99f4-a81c676df128 `  
                -SharedLocation /mnt/sharedLocation
```

Get-LXCABackups

Use this cmdlet to retrieve information about all Lenovo XClarity Administrator backups or export (download) an XClarity Administrator backup (.tar file) to the local system.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCABackups [-Connection LXCACConnection]
                [CommonParameters]
```

```
Get-LXCABackups [-Connection LXCACConnection]
                -Uuid String
                [-Destination String]
                [-Overwrite]
                [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Destination *String*

Specifies the directory and file name where the backup is to be saved on the local system.

- If you specify a directory but not a file name (in other words, if the string ends with a back slash "\"), the file name that is returned by the XClarity Administrator is used.
- If you specify a string that does not end with a backslash "\", the name after the last backslash "\" is used as the file name.
- If the directory does not exist, it is automatically created.

-Overwrite

When specified, if the file already exists in the specified directory, the file is overwritten. The default value is **false**.

If not specified, and the file already exists in the specified directory, the file is not overwritten, and the System.IO.IOException: The _full_file_path_ file already exists exception is thrown.

-Uuid *String*

UUID of the backup to be downloaded.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- [Lenovo.SysMgmt.LXCA.Integration.Data.LXCACFileDownloadResult](#)

When the **-Uuid** parameter is specified, this cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.LXCACFileDownloadResult](#) object, which contains the full path of the downloaded backup file on the local system.

- [Lenovo.SysMgmt.LXCA.Integration.Data.LXCABackupsInfo](#)

When the **-Uuid** parameter is not specified, this cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.LXCABackupsInfo](#) object that contains information about all XClarity Administrator backups.

Examples

The following example retrieves info about all backups existing on the XClarity Administrator server.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck
```

```
# Get information about existing backups
$backups = Get-LXCABackups
```

```
# Show all the information about the request, including result value, message (if any), and brief
# description of each backup (if any)
$backups
```

```
# Show detailed information about the existing backups.
$backups.BackupInfos
```

```
Disconnect-LXCA
```

The following example exports a specific backup (using the **-Uuid** parameter) to the local system and returns the fully-qualified path of the file. If a file with same name already exists, the existing file is replaced. The file name used for the exported file is the one that is returned by XClarity Administrator.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck
```

```
# Get information about existing backups
$backups = Get-LXCABackups
```

```
# Use the UUID of the first backup in the BackupInfos
$file = Get-LXCABackups -Uuid $backups.BackupInfos[0].Uuid -Destination C:\LXCA\ -Overwrite
$file
```

```
Disconnect-LXCA
Connect-LXCA -Host 1.111.11.11 -Credential $cred
```

Get-LXCALicenseCompliance

This cmdlet retrieves information about whether Lenovo XClarity Administrator is license compliant.

License compliance is determined based on the number of managed devices that support the advanced functions. The number of managed devices must not exceed the total number of licenses in all active license keys.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCALicenseCompliance [-Connection LXCAConnection]
                           [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For

detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.LicenseComplianceData](#) object, that contains information about the license compliance status.

Examples

The following example returns the license compliance status of the XClarity Administrator server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$compliance = Get-LXCALicenseCompliance

# Show all information about the request:
$compliance

# Show compliance status.
$compliance.Status

Disconnect-LXCA
```

Get-LXCALicensableDeviceCount

This cmdlet retrieves the number of managed devices that require licenses for advanced functions.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCALicensableDeviceCount [-Connection LXCAConnection]
                               [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns the number of the managed devices that are subject to licensing.

For XClarity Administrator v2.7.0 and earlier, the `NotSupportedException` exception is thrown.

Examples

The following example returns the number of managed devices that are subject to licensing.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$deviceCount = Get-LXCALicensableDeviceCount
```

Disconnect-LXCA

Get-LXCALicenseWarningPeriod

This cmdlet retrieves the license warning period, which determines the number of days before a license expires when Lenovo XClarity Administrator triggers a warning.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCALicenseWarningPeriod [-Connection LXCACConnection]  
                             [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.LicenseWarningPeriod](#) object, that contains information about the license warning period on the XClarity Administrator server.

Examples

The following example retrieves the license warning period on the XClarity Administrator server.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
$warning = Get-LXCALicenseWarningPeriod  
  
# Show all information about the request:  
$warning  
  
# Show warning period value.  
$warning.WarningPeriod  
  
Disconnect-LXCA
```

Get-LXCAManagementServerPackage

This cmdlet retrieves information about one or all management-server update packages that are in the Lenovo XClarity Administrator updates repository.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAManagementServerPackage [-Connection LXCACConnection]  
                                 -FixId String  
                                 [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-FixId *String*

Specifies the ID of the management-server update package. If an ID is not specified, the cmdlet returns information about all management-server update packages in the repository.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more UpdatePackage objects.

Examples

- The following example retrieves information about all management-server update packages.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$packages = Get-LXCAManagementServerPackage
```

```
Disconnect-LXCA
```

- The following example retrieves information about a specific management-server update packages.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$fixId = lnvgv_utl_lxca-core_2.1.0-2018xxxx_anyos_noarch
Get-LXCAManagementServerPackage $fixId
```

```
Disconnect-LXCA
```

Get-LXCAQuantityLicense

This cmdlet retrieves all installed licenses or export a specified list of the installed licenses to the local system.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCAQuantityLicense [-Connection LXCAConnection]
                        [CommonParameter]
```

```
Add-LXCAQuantityLicense [-Connection LXCAConnection]
                        -LicenseIds String[]
                        [-Destination String]
                        [-Overwrite]
                        [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Destination String

Specifies the directory and file name where the file is to be saved on the local system.

- If you specify a directory but not a file name (in other words, if the string ends with a back slash “\”), the file name that is returned by the XClarity Administrator is used.

If the directory does not exist, it is automatically created.

- If you specify a string that does not end with a backslash “\”, the name after the last backslash “\” is used as the file name.

Notes: The cmdlet might modify the file extension to reflect the actual file type. The type depends on how many installed licenses can be exported.

- If only one license can be exported, the exported file is the license key, and the file extension is “key”.
- If multiple licenses can be exported, the exported file is an archive license keys, and the file extension is “zip”.

-LicenseIds String[]

Specifies a list of IDs of licenses to be exported, separated by a comma.

Notes: The list is filtered to match the IDs of the installed licenses. Therefore:

- Only founded licenses are exported.
- If no license is found, the ERROR: Licenses with IDs <licenses_ids> are not installed on LXCA message is displayed.

-Overwrite

When specified, if the file already exists in the specified directory, the file is overwritten. The default value for Overwrite is false.

If not specified, and the file already exists in the specified directory, the file is not overwritten, and the System.IO.IOException: The _full_file_path_ file already exists exception is thrown.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- **Lenovo.SysMgmt.LXCA.Integration.Data.QuantityLicense**

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.QuantityLicense](#) object that contains information about all installed licenses on the XClarity Administrator server.

- **System.String**

When **-LicenseIds** parameter is specified, this cmdlet returns a System.String object, which contains the full path of the exported licenses file on the local system.

Examples

- The following example retrieves all licenses that are installed on the XClarity Administrator server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$license = Get-LXCAQuantityLicense
```

```
# show all the information about the request:
```

```
# - Result value,  
# - Message if any,  
# - brief description of the licenses data if any.  
$license
```

```
# show detailed information about the installed licenses.  
$license.Licenses
```

```
Disconnect-LXCA
```

- The following example exports specific licenses (using the **-LicenseIds** parameter) to the local system and returns the fully-qualified path of the file. If the file with same name already exists, the old file is replaced. The file name used for the exported file is the one that is returned by XClarity Administrator.

```
$cred = Get-Credential
```

```
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$file = Get-LXCAQuantityLicense -LicenseIds 1,5 -Destination C:\LXCA\ -Overwrite  
$file
```

```
Disconnect-LXCA
```

Get-LXCASingleSignon

Use this cmdlet to return the single sign-on setting for Lenovo XClarity Administrator management server or for a specific ThinkSystem or ThinkAgile server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCASingleSignon [-Connection LXCAConnection]  
                    [-Uuid String]  
                    [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String*

Specifies the UUID of a ThinkSystem or ThinkAgile server for which the single sign-on status is returned.

If not specified, the single sign-on setting for XClarity Administrator is returned.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- **Lenovo.SysMgmt.LXCA.Integration.Data.SingleSignOnStatus** This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.SingleSignOnStatus](#) object that contains information about the single sign-on status.

Examples

The following example returns the single sign-on setting for a XClarity Administrator.

```
$cred = Get-Credential
```

```
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

$(Get-LXCASingleSignOn)

Disconnect-LXCA
```

The following example returns the single sign-on setting for a specific server.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

$(Get-LXCASingleSignOn -Uuid AB58B2DD17E604572A4679E24BE2938DE)

Disconnect-LXCA
```

Install-LXCAQuantityLicense

This cmdlet uploads and installs one or more license activation keys in the Lenovo XClarity Administrator server. You can import and install each license-activation key individually or multiple key bundled in a zip archive.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Install-LXCAQuantityLicense [-Connection LXCAConnection]
                             -LicenseFile string
                             [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-LicenseFile *string*

Specifies the full path and file name of the license activation key or zip archive to be uploaded and installed.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- [Lenovo.SysMgmt.LXCA.Integration.Data.QuantityLicenseInstallResult](#)

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.QuantityLicenseInstallResult](#) object, that contains information about the operation result on the XClarity Administrator server.

Examples

The following example uploads and installs a license activation key C:\LXCA\lxcaLicense.key in the XClarity Administrator server

```
$cred = Get-Credential
Connect-LXCA -Host 1.111.11.11 -Credential $cred

$result = Install-LXCAQuantityLicense -LicenseFile C:\LXCA\lxcaLicense.key
```

```
# show all information about the request:
$result
$result.InstallInfo
```

```
Disconnect-LXCA
```

Install-LXCAUpdateManagementServer

This cmdlet installs a management-server update.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Install-LXCAUpdateManagementServer [-Connection LXCAConnection]
                                     -FixId String
                                     [-AsJob]
                                     [-WaitForJob]
                                     [-Schedule ScheduleObject]
                                     [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-FixId *String*

Specifies the ID of the management-server update package to be applied. The ID can be retrieved from cmdlet [Get-LXCAManagementServerPackage](#).

-AsJob

If specified, the cmdlet runs as background job and returns a Job object immediately.

-WaitForJob

When specified with the **-AsJob** parameter, PowerShell waits for this background job in XClarity Administrator to either start or complete before running the next cmdlet in the script. This can be one of the following values.

- **ToStart.** PowerShell waits until this background job starts in XClarity Administrator, so multiple jobs can run synchronously.
- **ToEnd.** PowerShell waits until this background job completes in XClarity Administrator, so that one only job runs at a time.

If this parameter not specified, this cmdlet does not wait. The next cmdlet in the script runs regardless of whether this background job starts or completes.

If **-AsJob** is not specified, this parameter is ignored.

-Schedule *ScheduleObject*

When specified, the cmdlet runs based on the specified schedule.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

If you specify the **-Asjob** parameter, this cmdlet returns a Job object. You can retrieve the UpdateResult object from Job.Output when the job is complete. You can retrieve the SubJob from Job.GetSubJobStatus() to get the status of each update.

If you do not specify the **-Asjob** parameter, this cmdlet does not return an object.

Examples

- The following example installs an existing management-server update package at a specific time, using a schedule object.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

# Only one package can be applied at a time!
$fixId = lnvgy_utl_lxca-core_2.4.0-2019xxxx_anyos_noarch
$schedule = New-LXCAScheduleObj -JobName $name -OccurrenceDate ('2019-05-22 4:35 PM')
Install-LXCAUpdateManagementServer -FixId $fixId -Schedule $schedule

Disconnect-LXCA
```

- The following example installs an existing management-server update package. PowerShell waits until the background job completes before running the next cmdlet.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

# Only one package can be applied at a time!
$fixId = lnvgy_utl_lxca-core_2.4.0-2019xxxx_anyos_noarch
Install-LXCAUpdateManagementServer -FixId $fixId -WaitForJob ToEnd

Disconnect-LXCA
```

- The following example imports the management-server update package that is located in the c:\updates folder to the XClarity Administrator server and then applies the updates. The import and installation both run as jobs in the background. PowerShell waits until the import-update job completes before applying the update, and then waits for the apply-update job to start before disconnecting from XClarity Administrator. Note that the installation-job continues to run in XClarity Administrator after the script disconnects PowerShell from XClarity Administrator.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

Import-LXCAUpdatePackage -ApplianceUpdate -Folder c:\updates -AsJob -WaitForJob ToEnd

$fixID = "lnvgy_sw_lxca_thinksystemrepo3-2.4.0_anyos_noarch"
Install-LXCAUpdateManagementServer -FixId $fixID -AsJob -WaitForJob ToStart

Disconnect-LXCA
```

Remove-LXCABackup

Use this cmdlet to delete an Lenovo XClarity Administrator backup from a specific location.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Remove-LXCABackup [-Connection LXCACONNECTION]
                  -Uuid String
                  -Location String
                  [CommonParameters]
```

```
Remove-LXCABackup [-Connection LXCACONNECTION]  
                -Uuid String  
                -SharedLocation String  
                [CommonParameter]
```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Location *String*

Location from where to delete the backup. This can be one of the following values:

- **All**. Deletes the backup from all locations (XClarity Administrator local repository and remote shares).
- **Internal**. Deletes the backup from only the XClarity Administrator local repository.

-SharedLocation *String*

Mounting point of the configured remote share from where to delete the backup.

-Uuid *String*

UUID of the backup to be deleted.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- [Lenovo.SysMgmt.LXCA.Integration.Data.RequestResult](#)

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.RequestResult](#) object that contains information about the delete-backup process.

Examples

The following example deletes an existing backup from the XClarity Administrator local repository.

```
$cred = Get-Credential  
Connect-LXCA -Host "192.0.2.0" -Credential $cred -SkipCertificateCheck  
  
$result = Remove-LXCABackup -Uuid c4e24aab-7f56-99f4-a81c676df128 -Location Internal  
  
# Show all the request information, including result values and messages, if any  
$result | Format-List  
  
Disconnect-LXCA
```

The following example deletes an existing backup from a configured remote share.

```
$cred = Get-Credential  
Connect-LXCA -Host "192.0.2.0" -Credential $cred -SkipCertificateCheck  
  
$result = Remove-LXCABackup -Uuid c4e24aab-a81c676df128 -SharedLocation '/mnt/sharedFolder'  
  
# Show all the request information, including result values and messages, if any  
$result | Format-List  
  
Disconnect-LXCA
```

Remove-LXCAQuantityLicense

This cmdlet removes a specific license activation key from the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Remove-LXCAQuantityLicense [-Connection LXCAConnection]  
                           -LicenseId int  
                           [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-LicenseId *int*

Specifies the ID of the license activation key to be removed.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.RequestResult](#) object, that contains information about the operation result on the XClarity Administrator server.

Examples

- The following example removes the license with id 3 from the XClarity Administrator server.

```
$cred = Get-Credential  
Connect-LXCA -Host 1.111.11.11 -Credential $cred  
  
$result = Remove-LXCAQuantityLicense -LicenseId 3  
  
# show all information about the request:  
$result  
  
Disconnect-LXC
```

- The following example removes all installed licenses from the XClarity Administrator server.

```
$cred = Get-Credential  
Connect-LXCA -Host 1.111.11.11 -Credential $cred  
  
$license = Get-LXCAQuantityLicense  
  
foreach($item in $license.Licenses)  
{  
    $(Remove-LXCAQuantityLicense -LicenseId $item.Id)  
}  
  
Disconnect-LXCA
```

Restore-LXCABackup

Use this cmdlet to restore Lenovo XClarity Administrator system data, settings, and imported files such as operating-system images, firmware updates, and OS device drivers.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Restore-LXCABackup [-Connection LXCACConnection]  
                  -Uuid String  
                  -Password SecureString  
                  [-RestoreFirmware]  
                  [-RestoreManagedDevice]  
                  [-RestoreNetworkSettings]  
                  [-RestoreOS]  
                  [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String*

Specifies the UUID of the backup to be restored.

-Password *String*

Specifies the passphrase that is required for decrypting and restoring the backup.

-RestoreFirmware *String*

When specified, firmware and OS device-driver updates are restored.

When not specified, firmware and OS device-driver updates are not restored.

-RestoreManagedDevices *String*

When specified, device inventory is restored.

When not specified, device inventory is not restored.

-RestoreNetworkSettings *String*

When specified, network settings are restored.

When not specified, network settings are not restored.

-RestoreOS *String*

When specified, operating-system images are restored.

When not specified, operating-system images are not restored.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- **[Lenovo.SysMgmt.LXCA.Integration.Data.LXCABackupRestoreResult](#)**

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.LXCABackupRestoreResult](#) object that contains information about the backup-restore result.

Examples

The following example restores a backup.

```
$cred = Get-Credential  
Connect-LXCA -Host "192.0.2.0" -Credential $cred -SkipCertificateCheck
```

```

$backupPwd = ConvertTo-SecureString "password" -AsPlainText -Force

$result = Restore-LXCABackup -Password $backupPwd -Uuid f56382e0-e39e-4cac-8124-9b96cdc52dbf -RestoreFirmware `
-RestoreManagedDevices -RestoreNetworkSettings -RestoreOS

# Show all the information about the request, including result value and message (if any)
$result | Format-List

# Show detailed backup restore information, if the restore process completed successfully
$result.RestoreInfo

Disconnect-LXCA

```

Set-LXCALicenseWarningPeriod

This cmdlet modifies the license warning period, which determines the number of days before a license expires when Lenovo XClarity Administrator triggers a warning.

Note: You cannot pipe objects to this cmdlet.

Syntax

```

Set-LXCALicenseWarningPeriod [-Connection LXCACONNECTION]
                             -WarningPeriod int
                             [CommonParameter]

```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-WarningPeriod *int*

Specifies the value for the warning period to set on the XClarity Administrator server. The value can be from 1 - 270.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.RequestResult](#) object, that contains information about the operation result on the XClarity Administrator server.

Examples

The following example sets the license warning period on the XClarity Administrator server.

```

$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$result = Set-LXCALicenseWarningPeriod -WarningPeriod 149

# Show all information about the request:
$result

Disconnect-LXCA

```

Set-LXCASingleSignon

Use this cmdlet to enable or disable single sign-on for Lenovo XClarity Administrator management server or for a specific ThinkSystem or ThinkAgile server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCASingleSignon [-Connection LXCAConnection]  
                    [-Uuid String]  
                    -SSOValue String  
                    [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String*

Specifies the UUID of a ThinkSystem or ThinkAgile server for which the single sign-on status is returned.

If not specified, the single sign-on setting for XClarity Administrator is modified.

-SSOValue *String*

Enables or disables single sign-on. This can be one of the following.

- **Enable.** Enables single sign-on.
- **Disable:** Disables single sign-on.

Note: Single sign-on is disabled automatically when using the CyberArk identity-management system for authentication.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.RequestResult](#) object that contains information about the operation result.

Examples

The following example modifies the single sign-on setting for a XClarity Administrator.

```
$cred = Get-Credential  
Connect-LXCA -Host "192.0.2.0" -Credential $cred -SkipCertificateCheck
```

```
$(Set-LXCASingleSignon -SSOValue Enable)
```

```
Disconnect-LXCA
```

The following example modifies the single sign-on setting for a specific server.

```
$cred = Get-Credential  
Connect-LXCA -Host "192.0.2.0" -Credential $cred -SkipCertificateCheck  
$(Set-LXCASingleSignon -SSOValue Disable -Uuid AB58B2DD17E604572A4679E24BE2938DE)
```

Discovery and management cmdlets

The following PowerShell cmdlets are available for performing discovery, manage, and unmanage functions.

Add-LXCAManagedDevice

This cmdlet adds one or more chassis, rack and tower servers, RackSwitch switches, and rack storage devices for management by the Lenovo XClarity Administrator server (manages a device).

This cmdlet does not support managing RackSwitch switches by specifying a UUID; however, you can specify an IP address.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCAManagedDevice [-Connection LXCACConnection]
    -ChassisUuid String
    -Username String-Password SecureString | -StoredCredentialId String
    [-RecoveryPassword SecureString]
    -DisableManagedAuth
    [-ForceManage]
    [-AsJob]
    [CommonParameter]
```

```
Add-LXCAManagedDevice [-Connection LXCACConnection]
    -RackServerUuid String
    -Username String-Password SecureString | -StoredCredentialId String
    -RecoveryPassword SecureString
    -DisableManagedAuth
    [-ForceManage]
    [-AsJob]
    [CommonParameter]
```

```
Add-LXCAManagedDevice [-Connection LXCACConnection]
    -ThinkServerUuid String
    -Username String-Password SecureString | -StoredCredentialId String
    -DisableManagedAuth
    [-ForceManage]
    [-AsJob]
    [CommonParameter]
```

```
Add-LXCAManagedDevice [-Connection LXCACConnection]
    -StorageUuid String
    -Username String-Password SecureString | -StoredCredentialId String
    -DisableManagedAuth
    [-ForceManage]
    [-AsJob]
    [CommonParameter]
```

```
Add-LXCAManagedDevice [-Connection LXCACConnection]
    -IPAddress String
    -Username String-Password SecureString | -StoredCredentialId String
    [-RecoveryPassword SecureString]
    -DisableManagedAuth
    [-ForceManage]
    [-AsJob]
```

[*CommonParameter*]

Add-LXCAManagedDevice [-Connection *LXCACConnection*]
-FQDN *String*]
-Username *String*-Password *SecureString* | -StoredCredentialId *String*
[-RecoveryPassword *SecureString*]
-DisableManagedAuth
[-ForceManage]
[-AsJob]
[*CommonParameter*]

Add-LXCAManagedDevice [-Connection *LXCACConnection*]
-IPv4Range *String*]
-Username *String*-Password *SecureString* | -StoredCredentialId *String*
[-RecoveryPassword *SecureString*]
-DisableManagedAuth
[-ForceManage]
[-AsJob]
[*CommonParameter*]

Add-LXCAManagedDevice *LXCACConnection*
-ServerAddress *String*
-UserName *String*
[-ApplicationId *String*]
[-Folder *String*]
[-Safe *String*]
[-DisableManagedAuth]
[-ForceManage]
[-AsJob]
[*CommonParameter*]

Add-LXCAManagedDevice *LXCACConnection*
-DeviceAddress *String*
-DeviceType *String*
-DisableDeviceDiscovery
-StoredCredentialId *String*
[-ForceManage]
[-AsJob]
[*CommonParameter*]

Add-LXCAManagedDevice [-Connection *LXCACConnection*]
-BulkImportFile *String*
-DisableManagedAuth
-ServerStoredCredentialsId *String*
[-CurrentRackServerUsername *String*]
[-CurrentRackServerPassword *SecureString*]
[-NewRackServerPassword *SecureString*]
-ChassisStoredCredentialsId *String*
[-CurrentChassisUsername *String*]
[-CurrentChassisPassword *SecureString*]
[-NewChassisPassword *SecureString*]
[-RecoveryPassword *SecureString*]
[-CurrentStorageUserName *SecureString*]
[-CurrentStoragePassword *SecureString*]
[-CurrentSwitchUserName *SecureString*]
[-CurrentSwitchPassword *SecureString*]
[-SwitchEnablePassword *SecureString*]
[-AsJob]
[*CommonParameter*]

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-DeviceAddress *String*

Specifies the IP address of the device to manage.

-DeviceType *String*

Specifies the type of device to manage. This can be the following value.

- **ThinkSystemDBSwitch**
- **NVIDIAMellanoxSwitch**

-DisableDeviceDiscovery

If specified, devices are managed using the specified IP address without first discovering the device.

-ChassisUuid *String[]*

Specifies one or more UUIDs of chassis to be managed by the XClarity Administrator server. The chassis must have already been discovered.

Tip: Use the [Get-LXCAThinkServerChassis -Unmanaged](#) cmdlet to retrieve the UUIDs of discovered but not managed chassis.

-RackServerUuid *String[]*

Specifies one or more UUIDs of Converged, NeXtScale, and System x servers to be managed by the XClarity Administrator server. The servers must have already been discovered.

Tip: Use the [Get-LXCARackServer -Unmanaged](#) cmdlet to retrieve the UUIDs of discovered but not managed Converged, NeXtScale, and System x servers.

-ThinkServerUuid *String[]*

Specifies one or more UUIDs of ThinkServer servers to be managed by the XClarity Administrator server. The servers must have already been discovered.

Tip: Use the [Get-LXCAThinkServer -Unmanaged](#) cmdlet to retrieve the UUIDs of discovered but not managed ThinkServer servers.

-StorageUuid *String[]*

Specifies one or more UUIDs for rack storage devices to be managed by the XClarity Administrator server. The storage device must have already been discovered.

Tip: Use the [Get-LXCAStroage -Unmanaged](#) cmdlet to retrieve the UUIDs of discovered but not managed storage devices.

-FQDN *String[]*

Specifies one or more FQDN addresses of chassis, rack servers and rack storage devices to be managed by the XClarity Administrator server.

-IPAddress *String[]*

Specifies one or more IPv4 and IPv6 addresses for chassis, rack and tower servers, and rack storage devices, and RackSwitch switches to be managed by the XClarity Administrator server.

-IPv4Range *String[]*

Specifies one or more ranges of IPv4 addresses. Use a format similar to 192.168.9.1-192.168.9.10 to specify a range.

-StoredCredentialId *String*

Specifies the stored credential to use for this managed device.

Use the [Get-LXCASStoredCredential](#) cmdlet to retrieve stored credential ID.

If you specify **-DisableManageAuth**, you must specify a stored credential to manage the device; otherwise, you must specify either **-Username** or **--StoredCredentialId**, but not both.

-Username *String*

Specifies a user account with **lxc-supervisor** authority. If the device to be managed is a chassis, specify a user account for the CMM in that chassis.

You must specify either **-Username** or **--StoredCredentialId**, but not both.

-Password *SecureString*

Specifies the password for the user account.

-RecoveryPassword *SecureString*

Specifies the recovery-ID password for the device.

When you manage a chassis, the CMM is configured to authenticate users with XClarity Administrator. Local CMM user accounts are no longer valid. If there are issues with XClarity Administrator, you can use the `RECOVERY_ID` to access the CMM directly. Ensure that you write down the password that you specify and store it in a secure location.

When you manage a Converged, NeXtScale, or System x server, the baseboard management controller is configured to authenticate users with the XClarity Administrator server. Local management-controller user accounts are no longer valid. If there are issues with the XClarity Administrator server, you can use the `RECOVERY_ID` to access the management controller directly. Ensure that you write down the password that you specify and store it in a secure location.

-BulkImportFile *String*

Specifies the path of the bulk-import file to use to manage multiple devices. The file template in CSV format can be downloaded from the XClarity Administrator web interface.

-ServerStoredCredentialsId *String*

Specifies the stored credentials for all servers if you specify the **-BulkImportFile** parameter.

-CurrentRackServerUsername *String*

Specifies the user account for the rack server if you specify the **-BulkImportFile** parameter. This name is used if a user name is not defined in the bulk-import file.

-CurrentRackServerPassword *SecureString*

Specifies the password for the user account that is specified using the **-CurrentRackServerUsername** parameter. This password is used if a password is not defined in the bulk-import file.

-NewRackServerPassword *SecureString*

Specifies the new password for the user account that is specified using the **-CurrentRackServerUsername** parameter. This password is used only if the current password is expired.

-ChassisStoredCredentialsId *String*

Specifies the stored credentials for all chassis and chassis devices if you specify the **-BulkImportFile** parameter.

-CurrentChassisUsername *String*

Specifies the user account for the chassis if you specify the **-BulkImportFile** parameter. This name is used if there a user name is not defined in the bulk-import file.

-CurrentChassisPassword *SecureString*

Specifies the password for the user account that is specified using the **-CurrentChassisUsername** parameter. This password is used if a password is not defined in the bulk-import file.

-NewChassisPassword *String*

Specifies the new password for the user account that is specified using the **-CurrentChassisUsername** parameter. This password is used only if the current password is expired.

-CurrentStorageUserName *String*

Specifies the user account for the storage device if you specify the **-BulkImportFile** parameter. This name is used if a user name is not defined in the bulk-import file.

-CurrentStoragePassword *SecureString*

Specifies the password for the user account that is specified using the **-CurrentStorageUserName** parameter. This password is used if a password is not defined in the bulk-import file.

-CurrentSwitchUserName *String*

Specifies the user account for the RackSwitch switch if you specify the **-BulkImportFile** parameter. This name is used if a user name is not defined in the bulk-import file.

-CurrentSwitchPassword *SecureString*

Specifies the password for the user account that is specified using the **-CurrentSwitchUserName** parameter. This password is used if a password is not defined in the bulk-import file.

-SwitchEnablePassword *SecureString*

Specifies the password to enable Privileged Exec Mode, if the switch requires it. This password is used if a password is not defined in the bulk-import file.

-ServerAddress *String*

Specifies the IP address or hostname where the user account is stored, as defined in CyberArk. This is typically the IP address of the Lenovo XClarity Controller or external LDAP server (if applicable).

-UserName *String*

Specifies the name of the user account to use to access the device.

-ApplicationId *String*

Specifies the application ID from CyberArk

Notes:

- If you specify the **-ApplicationId**, you must also specify **-Safe** and **-Folder**.
- If you do not specify **-ApplicationId**, XClarity Administrator uses the paths that are already defined to identify the onboarded accounts in CyberArk.

-Folder *String*

Specifies the folder from CyberArk.

Notes:

- If you specify the **-ApplicationId**, you must also specify **-Safe** and **-Folder**.
- If the onboarded account is not in a Folder, specify an empty string.

-Safe *String*

Specifies the safe from CyberArk.

Note: If you specify the **-ApplicationId**, you must also specify **-Safe** and **-Folder**.

-DisableManagedAuth

If specified, devices are managed using local authentication. If not specified, devices are managed using XClarity Administrator managed authentication.

Note: Managed and local authentication are not supported for ThinkServer and System x M4 servers.

-ForceManage

If specified, the devices are force managed by this XClarity Administrator server.

Notes: Use this force-management option only if you previously attempted to manage the device and management was not successful due to one of the following error conditions.

- If the managing XClarity Administrator failed and cannot be recovered.

Note: If the replacement XClarity Administrator instance uses the same IP address as the failed XClarity Administrator, you can manage the device again using the RECOVERY_ID account and password (if applicable) and the **Force management** option.

- If the managing XClarity Administrator was taken down before the devices were unmanaged.
- If the devices were not unmanaged successfully.

Attention: Devices can be managed by only one XClarity Administrator instance at a time. Management by multiple XClarity Administrator instances is not supported. If a device is managed by one XClarity Administrator, and you want to manage it with another XClarity Administrator, you must first unmanage the device from the original XClarity Administrator, and then manage it with the new XClarity Administrator.

-AsJob

When specified, the cmdlet runs as background job, and returns a Job object immediately. If not specified, the cmdlets runs in synchronous mode.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlets returns one of the following objects:

- If the **-AsJob** parameter is specified, a Job object is returned.
- If no UUID is specified, a ManagedDeviceResult object is returned.
- If the **-ChassisUUID** is specified, a ChassisManageResult object is returned.
- If the **-RackServerUUID** is specified, a RackServerManageResult object is returned.
- If the **-ThinkServerUUID** is specified, a ThinkServerManageResult object is returned.
- If the **-StorageUUID** parameter is specified, a StroageManageResult object is returned.

Examples

- The following example manages a Brocade switch without first discovering the device.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$job = Add-LXCAManagedDevice -DeviceAddress 192.0.2.125 -DeviceType ThinkSystemDBSwitch`
    -DisableDeviceDiscovery -StoredCredentialId 11524 -AsJob -ForceManage
while($job.State -eq "Running")
{
    Start-Sleep -Seconds 5
}

Disconnect-LXCA
```

- The following example retrieves all chassis that have been discovered but that are currently unmanaged and then manages those chassis using managed authentication.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$unmanagedchassis = Get-LXCACHassis -Unmanaged
$password = ConvertTo-SecureString "Password" -AsPlainText Force
$recoveryPwd = ConvertTo-SecureString "Password2" -AsPlainText Force
foreach ($chassis in $unmanagedchassis)
{
    Add-LXCAManagedDevice -ChassisUuid $chassis.uuid -Username ADMIN -Password $password`
        -RecoveryPassword $recoveryPwd
}
}
```

Disconnect-LXCA

- The following example retrieves all chassis that have been discovered but that are currently unmanaged and then manages those chassis using local authentication.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$unmanagedchassis = Get-LXCACHassis -Unmanaged
$password = ConvertTo-SecureString "Password" -AsPlainText -Force
$recoveryPwd = ConvertTo-SecureString "Password2" -AsPlainText -Force
foreach ($chassis in $unmanagedchassis)
{
    Add-LXCAManagedDevice -ChassisUuid $chassis.uuid -StoredCredentialId $SCID -DisableManagedAuth
}
```

Disconnect-LXCA

- The following example manages an undiscovered chassis as a job that runs in the background.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$password = ConvertTo-SecureString "Password" -AsPlainText -Force
$recoveryPwd = ConvertTo-SecureString "Password2" -AsPlainText -Force
$job = Add-LXCAManagedDevice -IpAddress 1.111.11.11 -Username USERID -Password $password`
    -RecoveryPassword $recoveryPwd -AsJob
while($job.State -eq "Running")
{
    Start-Sleep -Seconds 5
}
```

Disconnect-LXCA

- The following example manages multiple devices using an IPv4 address range.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$password = ConvertTo-SecureString "Password" -AsPlainText -Force
$recoveryPwd = ConvertTo-SecureString "Password2" -AsPlainText -Force
Add-LXCAManagedDevice -Ipv4Range 1.111.11.11-1.111.11.20 -Username USERID -Password $password`
    -RecoveryPassword $recoveryPwd
```

Disconnect-LXCA

- The following example uses a bulk-import file to manage multiple devices using a user name and password and managed authentication.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$password = ConvertTo-SecureString "Password" -AsPlainText -Force
Add-LXCAManagedDevice -BulkImportFile c:\bulkimport.csv -CurrentRackServerUsername USERID`
    -CurrentRackServerPassword $password
```

Disconnect-LXCA

- The following example uses a bulk-import file to manage multiple devices using a stored credential and local authentication.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$password = ConvertTo-SecureString "Password" -AsPlainText -Force
Add-LXCAManagedDevice -BulkImportFile c:\bulkimport.csv -ServerStoredCredentialsId $scId -DisableManagedAuth
```

```
Disconnect-LXCA
```

- The following example discovers and manages a server when managed authentication is enabled and uses CyberArk for authentication.

```
$cred = Get-Credential
```

```
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$job = Add-LXCAManagedDevice -ServerAddress 192.0.2.125 `
    -UserName UserName -ForceManage -AsJob
```

```
while($job.State -eq "Running")
{
    Start-Sleep -Seconds 5
}
```

```
Disconnect-LXCA
```

Invoke-LXCASystemAction

This cmdlet powers on, powers off, restarts, or virtually reseats a target device (CMM, server, switch, or storage controller).

This cmdlet starts a job that runs in the background to perform the operation. The cmdlet response is a URI in the form `/tasks/{task_id}` (for example, `/tasks/12`) that represents the job that is created to perform this request. You can use [Get-LXCATask -TaskUIDs {task_uuid_list}](#) to monitor the status and progress of the job.

Notes:

- You cannot pipe objects to this cmdlet.
- The following parameters are deprecated and will be removed at a future date. Use **-DeviceUUID** instead.
 - **-CMMUuid**
 - **-ComputeNodeUuid**
 - **-CanisterUuid**
 - **-IOSwitchUuid**
 - **-RackServerUuid**
 - **-ThinkServerUuid**
 - **-RackSwitchUuid**
 - **-StorageUuid**
- The following parameters are deprecated and will be removed at a future date. Use **-ServerAction** instead.
 - **-ComputeNodeAction**
 - **-RackServerAction**
 - **-ThinkServerAction**

Syntax

```
Invoke-LXCASystemAction [-Connection LXCACConnection]
    -DeviceUUID String
    [-Force]
    [-Schedule ScheduleObject]
    -CMMAction String
    [-FirmwareUpdate]
    [-WhatIf]
    [-Confirm]
    [CommonParameter]
```

```
Invoke-LXCASystemAction [-Connection LXCACConnection]
    -DeviceUUID String
```

```

        [-Force]
        [-Schedule ScheduleObject]
        -ServerAction String
        [-FirmwareUpdate]
        [-WhatIf]
        [-Confirm]
        [CommonParameter]

Invoke-LXCASystemAction [-Connection LXCACConnection]
        -DeviceUUID String
        [-Force]
        [-Schedule ScheduleObject]
        -CanisterAction String
        [-WhatIf]
        [-Confirm]
        [CommonParameter]

Invoke-LXCASystemAction [-Connection LXCACConnection]
        -DeviceUUID String
        [-Force]
        [-Schedule ScheduleObject]
        -IOSwitchAction String
        [-FirmwareUpdate]
        [-WhatIf]
        [-Confirm]
        [CommonParameter]

Invoke-LXCASystemAction [-Connection LXCACConnection]
        -DeviceUUID String
        [-Force]
        [-Schedule ScheduleObject]
        -RackSwitchAction String
        [-FirmwareUpdate]
        [-WhatIf]
        [-Confirm]
        [CommonParameter]

Invoke-LXCASystemAction [-Connection LXCACConnection]
        -DeviceUUID String
        [-Force]
        [-Schedule ScheduleObject]
        -StorageAction String
        -ControllerId String
        [-WhatIf]
        [-Confirm]
        [CommonParameter]

```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-DeviceUUID *String*

Specifies the UUID of a device on which to perform the system action.

-ControllerId *String*

Specifies the controller on which you want to take action. You can specify one of the following values:

- **A.** Top controller
- **B.** Bottom controller

-CanisterAction *String*

Specifies the action to be performed on the specified storage controller (canister). You can specify one of the following actions:

- PowerOn
- PowerOff
- Restart

-CMMAction *String*

Specifies the action to be performed on the specified CMM. You can specify the following actions:

- **Restart**

-IOSwitchAction *String*

Specifies the action to be performed on the specified Flex switch. You can specify one of the following actions:

- **PowerOn**
- **PowerOff**
- **Restart**

-RackSwitchAction *String*

Specifies the action to be performed on the specified RackSwitch switch. You can specify one of the following actions:

- **Restart**

-ServerAction *String*

Specifies the action to be performed on the specified ThinkAgile, ThinkEdge, ThinkSystem, Converged, NeXtScale, or System x server. You can specify one of the following actions:

- **PowerOn**
- **PowerOff**
- **Restart**
- **ShutdownOSandPowerOff**
- **ShutdownOSandRestart**
- **RestartWithNMI** (non-maskable Interrupt)
- **ResetBMC**
- **VirtualReseat** (for device in a chassis)

-StorageAction *String*

Specifies the action to be performed on the specified storage device. You can specify one of the following actions:

- **PowerOn**
- **PowerOff**
- **Restart**

-FirmwareUpdate

When specified, the cmdlets perform power actions during a firmware update.

-Force

When specified, the cmdlet performs the system action without first prompting for confirmation.

-Schedule *ScheduleObject*

Creates a scheduled job (task) for this cmdlet on the XClarity Administrator server.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

This cmdlet supports the **-WhatIf** and **-Confirm** risk management parameters.

Results

- **System.String**

If `-Schedule` is not specified, this cmdlet return a string that contains the URI in the form `/tasks/{task_id}` (for example, `/tasks/12`) of the job that it was created to perform the request. You can use [Get-LXCATask -TaskUIDs {task_uuid_list}](#) to monitor the status and progress of the job.

- **Lenovo.SysMgmt.LXCA.Integration.Data.LXCASchedule**

If `-Schedule` is specified, this cmdlet returns a `LXCASchedule` object that contains details about the schedule just created.

Examples

- The following example powers on a specific server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$taskUri = Invoke-LXCASystemAction -DeviceUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA `
    -ServerAction PowerOn `
    -Force

# show task URI
$taskUri

Disconnect-LXCA
```

- The following example restarts a device during a firmware update.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$taskUri = Invoke-LXCASystemAction -DeviceUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA `
    -ServerAction Restart `
    -FirmwareUpdate `
    -Force

# show task URI
$taskUri

Disconnect-LXCA
```

- The following example schedules a power on action on a specific server.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $cred -SkipCertificateCheck

$schedule = New-LXCAScheduleObj -ScheduleName "server schedule" `
    -OccurrenceDate (Get-Date).AddMinutes(15)

$result = Invoke-LXCASystemAction -DeviceUuid AAAAAAAAAAAAA `
    -ServerAction PowerOn `
    -Schedule $schedule
    -Force

# show created schedule
$result

Disconnect-LXCA
```

Remove-LXCAManagedDevice

This cmdlet removes one or more devices (including chassis, rack and tower servers, rack switches, and rack storage devices) from management by the Lenovo XClarity Administrator server (unmanages the device).

Chassis components (CMMs, compute nodes, storage devices, and Flex switches) are unmanaged when you unmanage the chassis.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Remove-LXCAManagedDevice [-Connection LXCAConnection]  
                        [-ChassisUuid String[]]  
                        [-ServerUuid String[]]  
                        [-ThinkServerUuid String[]]  
                        [-RackSwitchUuid String[]]  
                        [-StorageUuid String[]]  
                        [-ForceUnmanage]  
                        [-AsJob]  
                        [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ChassisUuid *String[]*

Specifies the UUIDs of one or more chassis to be unmanaged. The chassis must be managed by XClarity Administrator.

-ServerUuid *String[]*

Specifies the UUIDs of one or more Converged, NeXtScale, and System x servers to be unmanaged. The server must be managed by XClarity Administrator.

-ThinkServerUuid *String[]*

Specifies the UUIDs of one or more ThinkServer servers to be unmanaged. The server must be managed by XClarity Administrator.

-RackSwitchUuid *String[]*

Specifies the UUIDs of one or more RackSwitch switches to be unmanaged. The RackSwitch switch must be managed by XClarity Administrator.

-StorageUuid *String[]*

Specifies the UUIDs of one or more rack storage devices to be unmanaged. The storage device must be currently managed by XClarity Administrator.

-ForceUnmanage

If specified, the cmdlet unmanages the device even if the device is not reachable.

-Asjob

If specified, the cmdlet runs as a background job and returns a Job object immediately. Otherwise, the cmdlet runs in synchronous mode.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet unmanages the specified devices. If the `-Asjob` parameter is specified, this cmdlet returns a Job object. If the `--Asjob` parameter is not specified, no object is returned.

Examples

- The following example unmanages all managed chassis.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```

$managedchassis = Get-LXCACHassis
$chassisUuids = @()
Foreach($chassis in $managedchassis)
{
    $chassisUuids += $chassis.Uuid
}
Remove-LXCAManagedDevice -ChassisUuid $chassisUuids

Disconnect-LXCA

```

- The following example unmanages a specific managed chassis as a job that runs in the background.

Note: You can view the status of the job as it is running.

```

$cred = Get-Credential
Connect-LXCA -Host192.0.2.0 -Port 8044 -Credential $cred

$chassis = Get-LXCACHassis
$job = Remove-LXCAManagedDevice -ChassisUuid $chassis[0].Uuid -AsJob
while ($job.State -eq "Running")
{
    Start-Sleep -Seconds 5
}

Disconnect-LXCA

```

Set-LXCASystemIPInterface

This cmdlet modifies the IP interface configuration of a chassis, server, or Flex switch. You can specify settings such as DHCP mode, gateway, and address.

Note: You cannot pipe objects to this cmdlet.

Syntax

```

Set-LXCASystemIPInterface [-Connection LXCACConnection]
    -CMMUuid String
    -InterfaceName String
    [-IPv4Address String]
    [-IPv4DHCPMode String]
    [-IPv4Gateway String]
    [-IPv4Subnet String]
    [-IPv6DHCPEnabled Boolean]
    [-IPv6StatelessEnabled Boolean]
    [-IPv6StaticEnabled Boolean]
    [-StaticIPv6Address String]
    [-StaticIPv6Gateway String]
    [-StaticIPv6Prefix Int]
    [CommonParameter]

```

```

Set-LXCASystemIPInterface [-Connection LXCACConnection]
    -ServerUuid String
    -InterfaceName String
    [-IPv4Address String]
    [-IPv4DHCPMode String]
    [-IPv4Gateway String]
    [-IPv4Subnet String]
    [-IPv6DHCPEnabled Boolean]
    [-IPv6StatelessEnabled Boolean]
    [-IPv6StaticEnabled Boolean]
    [-StaticIPv6Address String]

```

[-StaticIPv6Gateway *String*]
[-StaticIPv6Prefix *Int*]
[*CommonParameter*]

Set-LXCASystemIPInterface [-Connection *LXCACConnection*]
-SwitchUuid *String*
-InterfaceName *String*
[-IPv4Address *String*]
[-IPv4DHCPMode *String*]
[-IPv4Gateway *String*]
[-IPv4Subnet *String*]
[-IPv6DHCPEnabled *Boolean*]
[-IPv6StatelessEnabled *Boolean*]
[-IPv6StaticEnabled *Boolean*]
[-StaticIPv6Address *String*]
[-StaticIPv6Gateway *String*]
[-StaticIPv6Prefix *Int*]
[*CommonParameter*]

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-CMMUuid *String*

Specifies the UUID of a primary CMM for which you want to set the IP interface.

-ServerUuid *String*

Specifies the UUID of a server for which you want to set the IP interface.

Note: The **-RackServerUuid** and **-ComputeNodeUuid** aliases are supported for backward compatibility.

-SwitchUuid *String*

Specifies the UUID of a Flex switch for which you want to set IP interface.

-InterfaceName *String*

Specifies the name of the IP interface to be modified (for example, eth0 or eth1).

Tip: Use a device inventory cmdlet to retrieve the current interface name, such as [Get-LXCACHassisInventory](#) or [Get-LXCARackServer](#).

-IPv4DHCPMode *String*

Specifies the IPv4 DHCP mode. You can specify one of the following values:

- **DHCP_only.** Obtain the IP address from a DHCP server only.
- **DHCP_then_Static.** Attempt to obtain the IP address from a DHCP server. If not successful, use a static IP configuration.
- **Static.** Use a static IP configuration.

-IPv4Subnet *String*

Specifies the IPv4 subnet mask to be used when DHCP mode is set to *Static* or *DHCP_then_Static*.

-IPv4Gateway *String*

Specifies the IPv4 gateway address to be used when DHCP mode is set to *Static* or *DHCP_then_Static*.

-IPv4Address *String*

Specifies the IPv4 address to be used when DHCP mode is set to *Static* or *DHCP_then_Static*.

-IPv6DHCPEnabled *Boolean*

Specifies whether to use stateful address configuration (DHCPv6).

-IPv6StatelessEnabled *Boolean*

Specifies to use stateless address auto-configuration.

-IPv6StaticEnabled *Boolean*

Specifies whether to use a statically assigned IP address.

-StaticIPv6Address *String*

Specifies the IPv6 address to be used when IPv6StaticEnabled is set to true.

-StaticIPv6Gateway *String*

Specifies the IPv6 gateway address to be used when **-IPv6StaticEnabled** is set to true.

-StaticIPv6Prefix *Int*

Specifies the IPv6 prefix length to be used when **-IPv6StaticEnabled** is set to true.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet modifies the IP address configuration. No object is returned.

Examples

The following example modifies the static IPv4 configuration for a rack server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Set-LXCASystemIPInterface -RackServerUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA -InterfaceName eth0`
    -IPv4DHCPmode Static -IPv4Subnet 255.255.255.0 -IPv4Gateway 192.0.2.1`
    -IPv4Address 192.0.2.121
```

```
Disconnect-LXCA
```

Set-LXCASystemProperty

This cmdlet modifies the properties for a chassis, rack or tower server, RackSwitch switch, or rack storage device. You can set properties such as hostname, description, contact, and display name.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCASystemProperty [-Connection LXCAConnection]
    -ChassisUuid String
    [-DisplayName String]
    [-Hostname String]
    [-Description String]
    [-Contact String]
    [-EncapsulationMode String]
    [-Location String]
    [CommonParameter]

Set-LXCASystemProperty [-Connection LXCAConnection]
    -ServerUuid String
    [-DisplayName String]
    [-Hostname String]
    [-Description String]
```

[-Contact *String*]
[-EncapsulationMode *String*]
[-Location *String*]
[*CommonParameter*]

Set-LXCASystemProperty [-Connection *LXCAConnection*]
 -RackSwitchUuid *String*
 [-DisplayName *String*]
 [-Description *String*]
 [-Contact *String*]
 [-Location *String*]
 [*CommonParameter*]

Set-LXCASystemProperty [-Connection *LXCAConnection*]
 -StorageUuid *String*
 [-DisplayName *String*]
 [-Description *String*]
 [-Contact *String*]
 [*CommonParameter*]

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ChassisUuid *String*

Specifies the UUID of a chassis for which you want to set the properties.

-ServerUuid *String*

Specifies the UUID of a server for which you want to set the properties.

Note: The **-RackServerUuid**, **-ThinkServerUuid** and **-ComputeNodeUuid** aliases are supported for backward compatibility.

-RackSwitchUuid *String*

Specifies the UUID of a RackSwitch switch for which you want to set the properties.

-StorageUuid *String*

Specifies the UUID of a rack storage device for which you want to set the properties.

-DisplayName *String*

Specifies the display name value to be set.

-Hostname *String*

Specifies the hostname value to be set.

-Description *String*

Specifies the description value to be set.

-Contact *String*

Specifies the contact value to be set.

-EncapsulationMode *String*

Specifies the Encapsulation mode to be set. You can specify one of the following values.

- **normal.** Encapsulation is disabled for this node. The global encapsulation setting is disabled by default. When disabled, the device encapsulation mode is set to “normal” and the firewall rules are not changed as part of the management process.
- **encapsulationLite.** Encapsulation is enabled for this node. When the global encapsulation setting is enabled and the device supports encapsulation, XClarity Administrator communicates with the device during the management process to change the device encapsulation mode to “encapsulationLite”

and to change the firewall rules on the device to limit incoming requests to those only from XClarity Administrator.

-Location String

Specifies the device location value to be set.

The location value can be 1 - 47 characters. You can use upper and lower case letter, number, and the following special characters: dash (-), underscore (_), and period (.). The first and last character cannot be a dash.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet sets the specified properties. No object is returned.

Examples

The following example modifies the display name, contact, and location values for a chassis.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Set-LXCASystemProperty -ChassisUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA -DisplayName "Display Name" `
    -Contact "Admin"

Disconnect-LXCA
```

Inventory cmdlets

The following PowerShell cmdlets are available for performing inventory functions.

Add-LXCARack

This cmdlet creates a rack.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCARack [-Connection LXCAConnection]
    -Name String
    -Height Int
    -Location String
    -Room String
    [CommonParameter]
```

Parameters

-Connection LXCAConnection

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Name String

Specifies the name of rack. The name must be unique.

The name can be 1 - 256 characters. You can use upper and lower case letter, number, and the following special characters: dash (-), underscore (_), and period (.). The first and last character cannot be a dash.

-Height *Int*

Specifies the height of the rack, in rack units. The available values are 6, 12, 18, 25, 37, 42, 45, 46, 48, 50, and 52.

-Location *String*

Specifies the location of rack.

The location can be 1 - 256 characters. You can use upper and lower case letter, number, and the following special characters: dash (-), underscore (_), and period (.). The first and last character cannot be a dash.

-Room *String*

Specifies the room where the rack is located.

The room can be 1 - 256 characters. You can use upper and lower case letter, number, and the following special characters: dash (-), underscore (_), and period (.). The first and last character cannot be a dash.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a Rack object.

Examples

The following example creates a rack.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$rack = Add-LXCARack -Name MyRack -Height 42 -Location Lab-1 -Room 101

Disconnect-LXCA
```

Add-LXCARackItem

This cmdlet adds a managed device to a specific rack. The managed device can be a chassis, rack server, a scalable complex, or a filler. The rack location information is applied to the specified device.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCARackItem [-Connection LXCACONNECTION]
    -Chassis
    -RackUuid String
    -ItemUuid String
    -ItemName String
    -ItemLowestRackUnit Int
    [CommonParameter]

Add-LXCARackItem [-Connection LXCACONNECTION]
    -ComplexSystem
```


-ComplexSystemId *String*
-RackUuid *String*
-ItemLowestRackUnit *Int*
[CommonParameter]

Add-LXCARackItem [-Connection *LXCACConnection*]
-Server
-RackUuid *String*
-ItemUuid *String*
-ItemName *String*
-ItemLowestRackUnit *Int*
[CommonParameter]

Add-LXCARackItem [-Connection *LXCACConnection*]
-Placeholder
-RackUuid *String*
-ItemName *String*
-ItemHeight *String*
-ItemLowestRackUnit *Int*
[CommonParameter]

Add-LXCARackItem [-Connection *LXCACConnection*]
-PlaceholderType *String*
-RackUuid *String*
-ItemName *String*
-ItemLowestRackUnit *Int*
[CommonParameter]

Add-LXCARackItem [-Connection *LXCACConnection*]
-Storage
-RackUuid *String*
-ItemUuid *String*
-ItemName *String*
-ItemLowestRackUnit *Int*
[CommonParameter]

Add-LXCARackItem [-Connection *LXCACConnection*]
-Switch
-RackUuid *String*
-ItemUuid *String*
-ItemName *String*
-ItemLowestRackUnit *Int*
[CommonParameter]

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Chassis

If specified, this cmdlets adds a chassis to the rack.

-ComplexSystem

If specified, this cmdlets adds a scalable-complex device to the rack.

-ComplexSystemId *String*

Specifies the ID of a scalable complex system to be added to the rack.

-ItemLowestRackUnit *Int*

Specifies the lowest rack unit position of the managed device in the target rack.

-ItemHeight *String*

Specifies the height of the managed device in rack units.

-ItemName *String*

Specifies the name of the managed device. The name is displayed in the rack.

-ItemUuid *String*

Specifies the UUID of the managed device to be added to the rack.

Note: The **-ItemUuid** is not required for a placeholder.

-Placeholder

If specified, this cmdlets adds a generic placeholder to the rack.

-PlaceholderType PlaceholderType

If specified, this cmdlets adds a placeholder of specified type to the rack. This can be one of the following values.

- 1U_Filler
- 2U_Filler
- 4U_Filler
- 3U_PureStorage
- IBM_Storwize_Control_Enclosure
- IBM_Storwize_Expansion_Enclosure
- Lenovo_RackSwitch
- Juniper_RackSwitch
- Lenovo_B300_FC_SAN_Switch
- Lenovo_6505_FC_SAN_Switch
- Lenovo_6510_FC_SAN_Switch
- ThinkSystem_DB610S_32Gb_FC_SAN_Switch
- ThinkSystem_DB620S_32Gb_FC_SAN_Switch
- ThinkSystem_DB400D_32Gb_FC_SAN_Director
- ThinkSystem_DB800D_32Gb_FC_SAN_Director
- 2U_Lenovo_Storage_Controller_Enclosure
- 2U_Lenovo_Storage_Expansion_Enclosure
- 2U_NetApp_FAS_Disk_Shelf
- 2U_NetApp_FAS_Controller
- 3U_NetApp_FAS_Controller
- 4U_NetApp_FAS_Disk_Shelf
- 4U_NetApp_FAS_Controller
- 6U_NetApp_FAS_Controller
- 1U_EMV_VNX_Standbay_Power_Supply
- 1U_EMV_VNX_Control_Station
- 2U_EMV_VNX_Data_Mover_Enclosure
- 2U_EMV_VNX_Disk_Array_Enclosure
- 2U_EMV_VNX_Disk_Processor_Enclosure
- 2U_EMV_VNX_Standby_Power_Supply
- 2U_EMV_VNX_Storage_Processor_Enclosure
- 3U_EMV_VNX_Disk_Array_Enclosure
- 3U_EMV_VNX_Disk_Processor_Enclosure
- 4U_EMV_VNX_Disk_Array_Enclosure
- 4U_EMV_VNX_Storage_Processor_Enclosure

-RackUuid c

Specifies the UUID of target rack.

-Server

If specified, this cmdlets adds a rack server to the rack.

-Storage

If specified, this cmdlets adds a storage device to the rack.

-Switch

If specified, this cmdlets adds a top-of-rack switch to the rack.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

- The following example adds a System x server into the rack.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

Add-LXCARackItem -RackUuid "EEEEEEEE-1111-2222-3333-FFFFFFFFFFFF" -Server -ItemLowestRackUnit 10`
                -ItemUuid "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"

Disconnect-LXCA
```

- The following example adds a generic placeholder into the rack.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

Add-LXCARackItem -RackUuid "EEEEEEEE-1111-2222-3333-FFFFFFFFFFFF" -Placeholder -ItemName "Switch" `
                -ItemLowestRackUnit 19 -ItemHeight 1

Disconnect-LXCA
```

- The following example adds a placeholder for a ThinkSystem DB800D 32Gb FC SAN Director to the rack.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

Add-LXCARackItem -RackUuid "EEEEEEEE-1111-2222-3333-FFFFFFFFFFFF"
                -PlaceholderType ThinkSystem_DB800D_32Gb_FC_SAN_Director -ItemName "DB800D SAN Director" `
                -ItemLowestRackUnit 20

Disconnect-LXCA
```

Get-LXCAActiveStatus

This cmdlet retrieves the status for one or more managed devices.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAActiveStatus [-Connection LXCAConnection]
                    [-ComponentID String]
                    [CommonParameter]

Get-LXCAActiveStatus [-Connection LXCAConnection]
                    [-Severity EventSeverity]
                    [-EventClass EventClass]
                    [CommonParameter]
```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Component ID *String[]*

Specifies one or more UUIDs of the components to which the status applies.

-Severity *EventSeverity[]*

Specifies the event severity. You can specify one or more of the following severities, separated by a comma.

- **Unknown**
- **Informational**
- **Warning**
- **Minor**
- **Major**
- **Critical**
- **Fatal**

Note: The severities that are listed through this cmdlet differ from those listed in the Lenovo XClarity Administrator web interface.

Lenovo XClarity Administrator web interface severity	Severity parameter
Critical	Fatal, Critical, Major
Warning	Minor, Warning
Informational	Informational
Unknown	Unknown

-EventClass *EventClass[]*

Specifies the event source type. You can specify one or more of the following event classes, separated by a comma.

- **ADAPTOR** (adapter card)
- **AUDIT**
- **BLADE** (Flex System server)
- **COOLING**
- **DISKS** (storage)
- **EXPANSION** (expansion card)
- **IOMODULE** (Flex System switch)
- **MEMORY**
- **OTHER**
- **POWER**
- **PROCESSOR**
- **SWITCH** (RackSwitch switch)
- **SYSTEM** (rack or tower server)
- **TEST**
- **UNKNOWN**

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more `ActiveStatus` objects.

Examples

- The following example retrieves the status for all managed devices.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$status = Get-LXCAActiveStatus
```

```
Disconnect-LXCA
```

- The following example retrieves a list of devices that have a status with a severity of Critical or Fatal.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$status = Get-LXCAActiveStatus -Severity Critical,Fatal
```

```
Disconnect-LXCA
```

Get-LXCAAvailableMachineType

This cmdlet retrieves all available machine types that are supported by the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAAvailableMachineType [-Connection LXCACConnection]
                             [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a list of `MachineType` objects.

Examples

The following example retrieves all available machine types.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$mts = Get-LXCAAvailableMachineType
```

```
Disconnect-LXCA
```

Get-LXCACHassis

This cmdlet retrieves system information for one or more chassis that have been discovered by Lenovo XClarity Administrator, including managed and unmanaged chassis.

This cmdlet returns system information (such as location, health, and machine type) about the chassis and CMMs that are installed in the chassis. It does not include detailed information about other chassis components, such as compute nodes, Flex switches, or fans. For inventory information for the chassis and chassis components, use the [Get-LXCACHassisInventory](#) cmdlet.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCACHassis [-Connection LXCACConnection]  
                [-ChassisUuid String[]]  
                [CommonParameter]
```

```
Get-LXCACHassis [-Connection LXCACConnection]  
                [-Unmanaged]  
                [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ChassisUuid *String[]*

Specifies the UUID of the chassis. If no UUID is specified and you specify the **-Unmanaged** parameter, all unmanaged chassis are returned. If no UUID is specified and the **-Unmanaged** parameter is *not* specified, this cmdlet returns all managed chassis.

-Unmanaged

If specified, this cmdlet returns a list of unmanaged chassis. Otherwise, this cmdlet returns a list of managed chassis.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more Chassis objects that contain information about each chassis and the CMMs that are installed in the chassis.

Examples

The following example retrieves information about all managed chassis.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$chassis = Get-LXCACHassis
```

```
Disconnect-LXCA
```

Get-LXCACHassisInventory

This cmdlet retrieves inventory data about one or more chassis from the Lenovo XClarity Administrator server.

Note: You can pipe Chassis objects to this cmdlet.

Syntax

```
Get-LXCACHassisInventory [-Connection LXCACConnection]  
                        -Chassis Chassis[]  
                        [CommonParameter]
```

```
Get-LXCACHassisInventory [-Connection Connection]  
                        -ChassisUuid String[]  
                        [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Chassis *Chassis[]*

Specifies a list of chassis. The cmdlet returns the inventory data for each chassis in the list.

Tip: To obtain the chassis list, use [Get-LXCACHassis](#).

-ChassisUuid *String[]*

Specifies one or more UUIDs of chassis. The cmdlet returns the inventory data from the specified chassis.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more ChassisInventory objects that contain the chassis inventory data.

Examples

The following example retrieves inventory details for all managed chassis.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
$chassis = Get-LXCACHassis  
$chassisInventory= Get-LXCACHassisInventory -Chassis $chassis  
  
Disconnect-LXCA
```

Get-LXCACHassisMember

This cmdlet retrieves information about the components (members) of one or more managed chassis.

Chassis components include:

- Flex System compute nodes
- Storage devices
- Fans

- Fan logic modules (fan muxes)
- Power supplies
- I/O modules (pass-thru modules and switches)

Note: You can pipe Chassis objects to this cmdlet.

Syntax

```
Get-LXCAChassisMember [-Connection LXCACConnection]
                    [-Chassis Chassis[]]
                    [CommonParameter]
```

```
Get-LXCAChassisMember [-Connection LXCACConnection]
                    [-ChassisUuid String[]]
                    [CommonParameter]
```

```
Get-LXCAChassisMember [-Connection LXCACConnection]
                    [-MemberUuid String[]]
                    [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Chassis *Chassis[]*

Specifies a list of chassis. The cmdlet returns the information about all components in each chassis in the list. If no chassis is specified, all components in all chassis are returned.

Tip: To obtain the chassis list, use [Get-LXCAChassis](#).

-ChassisUuid *String[]*

Specifies one or more chassis UUIDs. The cmdlet returns the information about all components in each chassis in the list. If no UUID is specified, all components in all chassis are returned.

-MemberUuid *String[]*

Specifies the UUID of one or more specific components in the chassis. The cmdlet returns the component with the specified UUID. If no UUID is specified, all components in all chassis are returned.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more ChassisMember objects that contain information about each installed component.

Examples

The following example retrieves details about all installed components for a specified chassis.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$chassis = Get-LXCAChassis
$chassismember = Get-LXCAChassisMember -Chassis $chassis[0]

Disconnect-LXCA
```


Get-LXCAComputeNode

This cmdlet retrieves inventory data for one or more servers in Flex System chassis.

Note: You can pipe Chassis objects to this cmdlet.

Syntax

```
Get-LXCAComputeNode [-Connection LXCACConnection]  
                  [-Chassis Chassis[]]  
                  [CommonParameter]
```

```
Get-LXCAComputeNode [-Connection LXCACConnection]  
                  [-ChassisUuid String[]]  
                  [CommonParameter]
```

```
Get-LXCAComputeNode [-Connection Connection]  
                  [-NodeUuid String[]]  
                  [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Chassis *Chassis[]*

Specifies a list of Flex System chassis. The cmdlet returns all servers in each chassis in the list. If no chassis is specified, the cmdlet returns all servers in all managed Flex System chassis.

Tip: To obtain the list of chassis, use [Get-LXCACHassis](#).

-ChassisUuid *String[]*

Specifies one or more Flex System chassis chassis UUIDs. The cmdlet returns all servers in the specified chassis. If no chassis UUID is specified, the cmdlet returns all servers in all managed Flex System chassis.

-NodeUuid *String[]*

Specifies one or more UUIDs for servers in Flex System chassis. If no server UUID is specified, the cmdlet returns all servers in all managed Flex System chassis.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more ComputeNode objects with information about the server in the Flex System chassis, such as firmware levels, installed processors, and installed memory.

Examples

The following example retrieves details about all servers in all managed Flex System chassis.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$node = Get-LXCAComputeNode
```

```
Disconnect-LXCA
```

Get-LXCADeviceMaintenance

This cmdlet retrieves maintenance status and schedules for a specific device.

Notes:

- Schedules (start and end dates) that are returned for the device are translated to the time zone of the local system.
- You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCADeviceMaintenance [-Connection LXCACConnection]  
                           -UUID String  
                           [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-UUID *String*

Specifies the UUID of the device to check the maintenance status.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns [Lenovo.SysMgmt.LXCA.Integration.Data.DeviceMaintenanceStatus](#) object.

If the device is not in maintenance mode, an exception with message 404 - Not Found is thrown, and the DeviceMaintenanceStatus object has **State** property set to "Unknown."

Examples

The following example retrieves maintenance status of a specific device.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
$deviceUUID = "000123444"  
$(Get-LXCADeviceMaintenance -UUID $deviceUUID)  
  
Disconnect-LXCA
```

Get-LXCAIOModule

This cmdlet retrieves information about one or more Flex System switches from the Lenovo XClarity Administrator server.

Note: You can pipe Chassis objects to this cmdlet.

Syntax

```
Get-LXCAIOModule [-Connection LXCACConnection]  
                 [-Chassis Chassis[]]  
                 [CommonParameter]
```

```
Get-LXCAIOModule [-Connection LXCAConnection]  
                [-ChassisUuid String]  
                [CommonParameter]
```

```
Get-LXCAIOModule [-Connection LXCAConnection]  
                [-IOModuleUuid string]  
                [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Chassis *Chassis* []

Specifies one or more managed chassis. The cmdlet returns all Flex System switches in each specified chassis. If no chassis is specified, the cmdlet returns all Flex System switches from all managed chassis.

Tip: To obtain the chassis list, use `Get-LXCACHassis`.

-ChassisUuid *string*[]

Specifies the UUID of one or more managed chassis. The cmdlet returns all Flex System switches from the specified chassis. If no UUID is specified, the cmdlet returns all Flex System switches from all managed chassis.

-IOModuleUuid *string*[]

Specifies the UUID of one or more Flex System switches. If no UUID is specified, the cmdlet returns all managed Flex System switches.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns IOModule objects.

Examples

The following example retrieves details about all Flex System switches in the specified chassis.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
$iomodules = Get-LXCAIOModule -ChassisUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
  
Disconnect-LXCA
```

Get-LXCARack

This cmdlet retrieves information about one or more racks.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCARack [-Connection LXCAConnection]  
            [-RackUuid String []]  
            [CommonParameter]
```

Parameters

-Connection *LXCACconnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-RackUuid *String []*

Specifies the UUIDs of one or more racks. If no UUID specified, this cmdlet returns information about all racks.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more Rack objects.

Examples

The following example retrieves information about the rack with UUID AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$rack = Get-LXCARack -RackUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Disconnect-LXCA
```

Get-LXCARackServer

This cmdlet retrieves inventory data for one or more Converged, System x, and NeXtScale servers from the Lenovo XClarity Administrator server.

To retrieve information about ThinkServer servers, use the [Get-LXCAThinkServer](#) cmdlet.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCARackServer [-Connection LXCACconnection]
                  [-Uuid String ]
                  [-Unmanaged]
                  [CommonParameter]
```

Parameters

-Connection *LXCACconnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String*

Specifies one or more UUIDs of rack servers. If no UUID is specified, the cmdlet returns all managed rack x servers unless you also specify the **-Unmanage** parameter.

-Unmanaged

If specified, the cmdlet returns all rack servers that are not currently managed. If not specified, this cmdlet returns all managed rack servers.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more RackServer objects.

Examples

The following example retrieves a list of all managed rack servers.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$rackservers = Get-LXCARackServer
```

```
Disconnect-LXCA
```

Get-LXCARackSwitch

This cmdlet retrieves inventory data for one or more RackSwitch switches from the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCARackSwitch [-Connection LXCACONNECTION]
                  [-Uuid String]
                  [-Unmanaged]
                  [CommonParameter]
```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String*

Specifies the UUIDs of one or more RackSwitch switches. If no UUID is specified, the cmdlet returns all managed RackSwitch switches unless you also specify the **-Unmanage** parameter.

-Unmanaged

If specified, the cmdlet returns all RackSwitch switches that are not currently managed. If not specified, this cmdlet returns all managed RackSwitch switches.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more RackSwitch objects.

Examples

The following example returns a list of all managed RackSwitch switches.

```
$cred = Get-Credential
```

```
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$rackswitches = Get-LXCARackSwitch
```

```
Disconnect-LXCA
```

Get-LXCAScalableComplexSystem

This cmdlet retrieves inventory data for one or more scalable-complex devices that Lenovo XClarity Administrator manages.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAScalableComplexSystem -Connection LXCACONNECTION  
                               [-ID String[]]  
                               -RackServer  
                               [CommonParameter]
```

```
Get-LXCAScalableComplexSystem -Connection LXCACONNECTION  
                               [-ID String[]]  
                               -ComputeNode  
                               [CommonParameter]
```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ID *String[]*

Specifies one or more scalable complex system IDs.

-RackServer

Specifies that the scalable system was created from rack servers.

-ComputeNode

Specifies that the scalable system was created from servers in Flex System chassis.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more `ComplexForComputeNode` or `ComplexForRackServer` objects.

Examples

The following example retrieves information about all scalable complex systems created from System x servers.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
Get-LXCAScalableComplexSystem -RackServer
```

```
Disconnect-LXCA
```

Get-LXCAServer

This cmdlet retrieves inventory data for one or more servers (including rack servers and compute nodes) from Lenovo XClarity Administrator.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAServer [-Connection LXCAConnection]  
               [-Uuid String[]]  
               [-Type SystemType]  
               [-Unmanaged]  
               [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String[]*

Specifies the UUIDs of one or more servers, separated by a comma, for which information is to be returned. If no UUID is specified, the cmdlet returns information for all servers.

-Type *SystemType*

Specifies the type of server to return. Can have one of the following values.

- **ComputeNode**
- **EdgeServer**
- **RackServer**
- **ThinkServer**

-Unmanaged

If specified, the cmdlet returns servers that are not currently managed. If not specified, this cmdlet returns managed servers.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more Server objects.

Examples

- The following example retrieves inventory data for two managed servers.

```
$cred = Get-Credential  
Connect-LXCA -Host192.0.2.0 -Credential $cred  
  
$inventory_data = Get-LXCAServer -UUID AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB  
  
Disconnect-LXCA
```

- The following example retrieves a list of all managed servers.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
$servers = Get-LXCAServer  
  
Disconnect-LXCA
```

- The following example retrieves a list of all unmanaged servers.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$unmanaged_servers = Get-LXCAServer -Unmanaged

Disconnect-LXCA
```

Get-LXCAServersGlobalSettings

This cmdlet returns global-configuration settings for ThinkAgile and ThinkSystem servers.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAServersGlobalSettings [-Connection LXCACConnection]
                               [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- [Lenovo.SysMgmt.LXCA.Integration.Data.ServersGlobalSettings](#)

If the request was successful, this cmdlet returns a [ServerGlobalSettings](#) object.

- [Lenovo.SysMgmt.LXCA.Integration.Data.RequestResult](#)

If an error occurred, this cmdlet returns a [RequestResult](#) object that contains information about the status of this request.

Examples

The following example retrieves global-configuration settings for ThinkAgile and ThinkSystem servers.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $cred -SkipCertificateCheck

$result = Get-LXCAServersGlobalSettings
$result

#Disconnect from LXCA
Disconnect-LXCA
```

Get-LXCASStorage

This cmdlet retrieves inventory data for one or more rack storage devices from the Lenovo XClarity Administrator server.

To retrieve inventory data Flex System storage devices, see the [Get-LXCASStorageNode](#) cmdlet.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCASStorage [-Connection LXCACConnection]  
                [-Uuid String[]]  
                [-Unmanaged]  
                [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String[]*

Specifies the UUIDs of one or more rack storage devices. If no UUID is specified, the cmdlet returns all managed rack storage devices unless you also specify the **-Unmanage** parameter.

-Unmanaged

If specified, the cmdlet returns information for all rack storage devices that are not currently managed. If not specified, this cmdlet returns all managed rack storage devices.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more Storage objects.

Examples

The following example retrieves a list of all managed rack storage devices.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred  
  
$storage = Get-LXCASStorage  
  
Disconnect-LXCA
```

Get-LXCASStorageNode

This cmdlet retrieves inventory data for one or more storage devices in a chassis.

To retrieve inventory data for rack storage devices, see the [Get-LXCASStorage](#) cmdlet.

Note: You can pipe Chassis objects to this cmdlet.

Syntax

```
Get-LXCASStorageNode [-Connection LXCACConnection]  
                   [-Chassis Chassis[]]  
                   [CommonParameter]  
  
Get-LXCASStorageNode [-Connection LXCACConnection]  
                   [-ChassisUuid String[]]  
                   [CommonParameter]
```

```
Get-LXCASStorageNode [-Connection LXCACConnection]  
                    [-NodeUuid String[]]  
                    [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Chassis *Chassis[]*

Specifies one or more chassis. The cmdlet returns all storage devices in each chassis in the list. If no chassis is specified, the cmdlet returns all storage devices in all managed chassis.

Tip: To obtain the chassis list, use [Get-LXCACHassis](#).

-ChassisUuid *String[]*

Specifies one or more chassis UUIDs. The cmdlet returns all storage devices in the specified chassis. If no chassis UUID is specified, the cmdlet returns all storage devices in all managed chassis.

-NodeUuid *String[]*

Specifies one or more storage-device UUIDs. If no UUID is specified, the cmdlet returns all storage devices in all managed chassis.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more StorageNode objects.

Examples

The following example retrieves information about all storage devices that are installed in all discovered chassis.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
$nodes = Get-LXCASStorageNode  
  
Disconnect-LXCA
```

Get-LXCAThinkServer

This cmdlet retrieves inventory data for one or more ThinkServer servers from the Lenovo XClarity Administrator server.

To retrieve information about Converged, System x, and NeXtScale servers, use the [Get-LXCARackServer](#) cmdlet.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAThinkServer [-Connection LXCACConnection]  
                   [-Uuid String[]]  
                   [-Unmanaged]
```

[CommonParameter]

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String[]*

Specifies the UUIDs of one or more ThinkServer servers. If no UUID is specified, the cmdlet returns all managed ThinkServer servers unless you also specify the **-Unmanage** parameter.

-Unmanaged

If specified, the cmdlet returns all ThinkServer servers that are not currently managed. If not specified, this cmdlet returns all managed ThinkServer servers.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more ThinkServer objects.

Examples

The following example retrieves a list of all managed ThinkServer servers.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

$thinkservers = Get-LXCAThinkServer

Disconnect-LXCA
```

Remove-LXCARack

This cmdlet deletes a specific rack.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Remove-LXCARack [-Connection LXCACConnection]
                -RackUuid String
                [CommonParameter]
```

```
Remove-LXCARack [-Connection LXCACConnection]
                -Rack Rack
                [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-RackUuid *String*

Specifies the UUID of the rack to be deleted.

-Rack *Rack*

Specifies the rack object to be deleted.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

The following example deletes the rack with UUID AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Remove-LXCARack -RackUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Disconnect-LXCA
```

Remove-LXCARackItem

This cmdlet removes managed devices from a specific rack.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Remove-LXCARackItem [-Connection LXCACConnection]
                    -RackUuid String
                    -RackItemUuid String
                    [CommonParameter]

Remove-LXCARackItem [-Connection LXCACConnection]
                    -RackUuid String
                    -ComplexId String
                    [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ComplexId *String*

Specifies the ID of the scalable-complex system to be removed.

-RackItemUuid *String*

Specifies the UUID of the managed device to be removed.

-RackUuid *String*

Specifies the UUID of target rack.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

The following example removes a managed device with UUID `BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB` from rack with UUID `AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA`.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Remove-LXCARack -RackUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA -RackItemUuid BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

Disconnect-LXCA
```

Set-LXCADeviceMaintenance

This cmdlet modifies maintenance status and schedules for a specific device.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCADeviceMaintenance [-Connection LXCACconnection]
                           -UUID String
                           -Maintenance String
                           [-EndTime DateTime]
                           [-StartTime DateTime]
                           [CommonParameter]
```

Parameters

-Connection *LXCACconnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-UUID *String*

Specifies the UUID of the device to modify the maintenance status.

-Maintenance *String*

Specifies the new maintenance status of the device. You can specify one of the following values.

- **Enabled**
- **Disabled**

-EndTime *DateTime*

Specifies when the device maintenance will end.

- If **-EndTime** is not specified and new maintenance value is “Enabled”, the device remains in maintenance until the maintenance will be disabled.
- If new maintenance value is “Disabled”, this parameter is ignored.

-StartTime *DateTime*

Specifies when the device maintenance will start.

- If **-StartTime** is not specified and new maintenance value is “Enabled”, the device goes in maintenance immediately.
- If new maintenance value is “Disabled”, this parameter is ignored.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For

detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return data.

Examples

The following example put one device in maintenance for 3 hours, starting 2 hours from now.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$deviceUUID = "000123444"
Set-LXCADeviceMaintenance -UUID $deviceUUID -Maintenance Enabled `
    -StartTime (Get-Date).AddHours(2) `
    -EndTime (Get-Date).AddHours(5)

Disconnect-LXCA
```

Set-LXCARack

This cmdlet modifies properties for a specific rack.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCARack [-Connection LXCAConnection]
             -RackUuid String
             [-Name String]
             [-Location String]
             [-Room String]
             [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-RackUuid *String*

Specifies the UUID of the rack to be modified.

-Name *String*

Specifies the name of rack. The name must be unique.

The name can be 1 - 256 characters. You can use upper and lower case letter, number, and the following special characters: dash (-), underscore (_), and period (.). The first and last character cannot be a dash.

-Location *String*

Specifies the location of rack.

The location can be 1 - 256 characters. You can use upper and lower case letter, number, and the following special characters: dash (-), underscore (_), and period (.). The first and last character cannot be a dash.

-Room *String*

Specifies the room where the rack is located.

The room can be 1 - 256 characters. You can use upper and lower case letter, number, and the following special characters: dash (-), underscore (_), and period (.). The first and last character cannot be a dash.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

The following example modifies the name, location, and room for a rack with UUID AAAAAAAAAAAAAA.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Set-LXCARack -RackUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA -Name NewMyRack -Location Lab-2 -Room 201

Disconnect-LXCA
```

Set-LXCARackItem

This cmdlet modifies properties for a managed device in a rack.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCARackItem [-Connection LXCAConnection]
                 -RackUuid String
                 -ItemUuid String
                 [-ItemName String]
                 [-ItemLowestRackUnit Int]
                 -ItemType String
                 [CommonParameter]
```

```
Set-LXCARackItem [-Connection LXCAConnection]
                 -RackUuid String
                 -ComplexSystemId String
                 [-ItemLowestRackUnit Int]
                 [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ComplexSystemId *String*

Specifies the ID of the scalable-complex device when the target is a scalable-complex device.

-ItemLowestRackUnit *Int*

Specifies the lowest rack unit position of the managed device in the target rack.

-ItemName *String*

Specifies the name of the managed device to be displayed in the rack.

-ItemType *String*

Specifies the managed-device type. You can specify one of the following values:

- **Chassis**
- **Server**
- **Switch**
- **Storage**
- **Placeholder.** (filler)

-ItemUuid *String*

Specifies the UUID of the managed device.

-RackUuid *String*

Specifies the UUID of target rack.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

The following example modifies the lowest rack unit value of server with UUID BBBB BBBB BBBB.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

Set-LXCARackItem -RackUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA -ItemUuid BBBB BBBB BBBB BBBB BBBB BBBB BBBB`
                -ItemLowestRackUnit 10 -ItemType Server

Disconnect-LXCA
```

Set-LXCAServersGlobalSettings

This cmdlet modifies global-configuration settings for ThinkAgile and ThinkSystem servers.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCAServersGlobalSettings [-Connection LXCACConnection]
                               [-SMARTDataRetrieval String]
                               [-BatchSize Integer]
                               [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-SMARTDataRetrieval *String*

Specifies if the collection of SMART data for ThinkAgile and ThinkSystem servers will be enabled or not. This can be one of the following values.

- **enable.** SMART data collection will be enabled.
- **disable.** SMART data collection will be disabled.

-BatchSize *Integer*

Specifies the maximum number of servers on which the secure-erase operation can be performed at one time. This value can be from **3 – 100**.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- [Lenovo.SysMgmt.LXCA.Integration.Data.RequestResult](#)

This cmdlet returns a [RequestResult](#) object that contains information about the status of this request.

Examples

The following example enables the SMARTData retrieval and sets the maximum number of servers on which the secure-erase operation can be performed at one time to 15.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

$result = Set-LXCAServersGlobalSettings -SMARTDataRetrieval enable -BatchSize 15
$result

#Disconnect from LXCA
Disconnect-LXCA
```

Resource group cmdlets

The following PowerShell cmdlets are available for managing resource groups.

Add-LXCAResourceGroupMembers

This cmdlet adds devices to a static resource group.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCAResourceGroupMembers [-Connection LXCAConnection]
                               -GroupUuid String
                               -DeviceType String
                               -DevicesUuid String[]
                               [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-GroupUuid *String*

Specifies the UUID of the resource group to which you want to add devices.

-DeviceType *String*

Specifies the type of the devices to be added. You can specify one of the following values.

- **Chassis**
- **Nodes**
- **Switches**

- **Storage**

-DevicesUuid *String[]*

Specifies the UUID of the devices to be added.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a `Lenovo.SysMgmt.LXCA.Integration.Data.ResourceGroup` object for the resource group with specified UUID. The returned object contains all the devices (including devices that were just added) in the resource group.

Note: This cmdlet does not returns any error if one or more devices were not added to the group.

Examples

The following example adds three switches to a specific resource group.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

$groupUuid = "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
$switchesUuids = @("EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE", "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF", `
"CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC")
$result = Add-LXCAResourceGroupMembers -GroupUuid $groupUuid -DevicesUuid $switchesUuids -DeviceType Switches
$result

Disconnect-LXCA
```

Get-LXCAResourceGroups

This cmdlet retrieves information about all resource groups or a specific resource group.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAResourceGroups [-Connection LXCAConnection]
                        -Type String
                        [-GroupUuid String]
                        [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Type *String*

Specifies the type of resource group. This can be the following value.

- **static**

-GroupUuid *String*

Specifies the UUID of the resource group to be retrieved. If not specified, information about all static resource groups are returned.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one of the following objects:

- `Lenovo.SysMgmt.LXCA.Integration.Data.ResourceGroup`
If **-GroupUuid** is specified, this cmdlet returns a `Lenovo.SysMgmt.LXCA.Integration.Data.ResourceGroup` object for the group with specified UUID.
- `Lenovo.SysMgmt.LXCA.Integration.Data.ResourceGroup[]`
If **-GroupUuid** is not specified, this cmdlet returns an array of `Lenovo.SysMgmt.LXCA.Integration.Data.ResourceGroup` objects.

Examples

- The following example returns the information about a specific resource group.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck
```

```
$uuid = "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
$result = Get-LXCAResourceGroups -Type Static -GroupUuid $uids
$result
```

```
# Group members can be founded in:
$result.Members
```

```
Disconnect-LXCA
```

- The following example returns the information about all static resource groups.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck
```

```
$result = Get-LXCAResourceGroups -Type Static
$result
```

```
Disconnect-LXCA
```

Server configuration cmdlets

The following PowerShell cmdlets are available for performing server configuration (Configuration Patterns) functions.

Get-LXCAConfigPattern

This cmdlet returns information about one or more server patterns from the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAConfigPattern [-Connection LXCAConnection]
                    [-PatternID String[]]
                    [IncludeSettings]
                    [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-PatternID *String[]*

Specifies one or more server pattern IDs. If a pattern ID is not specified, all server patterns are returned.

-IncludeSettings

If specified, the cmdlets returns the detailed definition (settings) of the server pattern.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more ConfigPattern objects.

Examples

- The following example retrieves information about all server patterns defined on XClarity Administrator.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$patterns = Get-LXCAConfigPattern
```

```
Disconnect-LXCA
```

- The following example retrieves information for a specified server pattern, including information about referenced category patterns.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$patterns = Get-LXCAConfigPattern PatternId 33 IncludeSettings
$pattern.Settings
```

```
Disconnect-LXCA
```

Get-LXCAConfigPatternDeployableServer

This cmdlet retrieves a list of deployable rack servers and Flex System compute nodes that meet the criteria for a server pattern or profile.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAConfigPatternDeployableServer [-Connection LXCACConnection]
                                         -PatternID String
                                         [CommonParameter]
```

```
Get-LXCAConfigPatternDeployableServer [-Connection LXCACConnection]
                                         -ProfileID String
                                         [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-PatternID *String*

Specifies a server pattern ID. You can obtain a list of all server pattern IDs using the [Get-LXCAConfigPattern](#) cmdlet.

-ProfileID *String*

Specifies a server profile ID. You can obtain a list of all server profile IDs using the [Get-LXCAConfigProfile](#) cmdlet.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more `LXCAConfigPatternDeployableTarget` objects.

Examples

The following example retrieves a list of deployable servers based on a server pattern ID.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$patterns = Get-LXCAConfigPattern
$chassis = Get-LXCAConfigPatternDeployableServer -PatternID $patterns[0].Id
$chassis[0].Servers

Disconnect-LXCA
```

Get-LXCAConfigProfile

This cmdlet retrieves one or more server profiles from the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAConfigProfile [-Connection LXCAConnection]
                    [-ProfileID String[]]
                    [CommonParameter]

Get-LXCAConfigProfile [-Connection LXCAConnection]
                    [-PatternID String[]]
                    [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ProfileID *String[]*

Specifies one or more server profile IDs. If a server profile ID is not specified, all server profiles are returned.

-PatternID *String[]*

Specifies one or more the server pattern IDs. If a server pattern ID is not specified, all server profiles for the specified patterns are returned.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more ConfigProfile objects.

Examples

The following example retrieves information about all server profiles defined in XClarity Administrator.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

$profiles = Get-LXCAConfigProfile

Disconnect-LXCA
```

Import-LXCAConfigPattern

This cmdlet imports configuration-pattern settings into Lenovo XClarity Administrator.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Import-LXCAConfigPattern [-Connection Connection]
                        -Settings String
                        [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Settings *String*

Specifies the configuration-pattern settings. The settings must be in the correct JSON format.

To ensure the correct format, use an existing configuration pattern as a template.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns the ID of the imported configuration pattern.

Examples

The following example imports the configuration-pattern settings from a local file to the XClarity Administrator server

```

$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$settings = Get-Content Path c:\pattern_setting.txt
Import-LXCAConfigPattern Setting $settings

Disconnect-LXCA

```

Install-LXCAConfigPattern

This cmdlet deploys a server pattern to one or more target servers.

Note: You cannot pipe objects to this cmdlet.

Syntax

```

Install-LXCAConfigPattern [-Connection LXCAConnection]
                        -PatternID String
                        -TargetComputeNodeId String[]
                        -Restart String
                        [-AsJob]
                        [CommonParameter]

Install-LXCAConfigPattern [-Connection LXCAConnection]
                        -PatternID String
                        -TargetRackServerId String[]
                        -Restart String
                        [-AsJob]
                        [CommonParameter]

```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-PatternID *String*

Specifies the ID of the configuration pattern to be deployed.

-TargetComputeNodeid *String[]*

Specifies one or more IDs of the target Flex System servers. If a target is an empty bay, specify the location ID; otherwise, specify the server UUID. To obtain the value, use the [Get-LXCAConfigPatternDeployableServer](#) cmdlet.

-TargetRackServerId *String[]*

Specifies one or more UUIDs of target Converged, NeXtScale, and System x servers. The ID is the server location ID. To obtain the value, use the [Get-LXCAConfigPatternDeployableServer](#) cmdlet.

-Restart *String*

Specifies when to restart target server. You can specify one of the following values:

- **Immediate.** The server is powered on or restarted immediately, which activates all baseboard management controller, UEFI, and server configuration changes.
- **Defer.** Management controller changes are activated on the server immediately. However, UEFI and server configuration changes are not activated until the next restart of the server.
- **Pending.** A profile is generated for the server with the settings for review, but the settings are not activated on the server. To activate the settings, you must manually activate the server profile, and restart the server.

-AsJob

If specified, the cmdlet runs as background job and returns a Job object immediately.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

If you specify the **-AsJob** parameter, this cmdlet returns a Job object. The ConfigPatternDeployResult[] can be retrieved from Job.Output.

If you *do not* specify the **-AsJob** parameter, this cmdlet returns one or more ConfigPatternDeployResult objects.

Examples

The following example deploys a server pattern to a target Flex System server and defers the activation of that pattern until the next restart of the server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$patterns = Get-LXCAConfigPattern
$targetchassis = Get-LXCAConfigPatternDeployableServer -PatternId $patterns[0].Id
Install-LXCAConfigPattern -PatternId $patterns[0].Id -TargetComputeNodeId $targetchassis[0].Servers[0].Id`
    -restart defer

Disconnect-LXCA
```

Install-LXCAConfigProfile

This cmdlet deploys a server profile to a specific server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Install-LXCAConfigProfile [-Connection LXCAConnection]
    -ProfileID String
    -TargetComputeNodeId String[]
    -Restart String
    [-AsJob]
    [CommonParameter]
```

```
Install-LXCAConfigProfile [-Connection LXCAConnection]
    -ProfileID String
    -TargetRackServerId String[]
    -Restart String
    [-AsJob]
    [CommonParameter]
```

Parameters

-Connection LXCAConnection

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ProfileID String

Specifies the ID of the configuration profile to be deployed.

-TargetComputeNodeid String

Specifies a target Flex System compute node ID. If a target is an empty bay, specify the location ID; otherwise, specify the server UUID. To obtain the value, use the [Get-LXCAConfigPatternDeployableServer](#) cmdlet.

-TargetRackServerId *String*

Specifies a target Converged, NeXtScale, or System x server UUID. The ID is the server location ID. To obtain the value, use the [Get-LXCAConfigPatternDeployableServer](#) cmdlet.

-Restart *String*

Specifies when to restart the target server. You can specify one of the following values:

- **Immediate.** The server is powered on or restarted immediately, which activates all baseboard management controller, UEFI, and server configuration changes.
- **Defer.** Management controller changes are activated on the server immediately. However, UEFI and server configuration changes are not activated until the next restart of the server.

-AsJob

When specified, the cmdlet runs as background job and returns a Job object immediately.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

If you specify the **-AsJob** parameter, this cmdlet returns a Job object. The ConfigPatternDeployResult objects can be retrieved from Job.Output.

If you *do not* specify the **-AsJob** parameter, this cmdlet, it returns one or more ConfigPatternDeployResult objects.

Examples

The following example retrieves a server profile from one server and deploys it to another server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$profile = Get-LXCAConfigProfile | where {$_.ServerUuid -ieq AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA}
Remove-LXCAConfigProfileFromSystem -Profile $profile
Install-LXCAConfigProfile -ProfileId $profile.Id -TargetComputeNodeId BBBB`
    -restart defer

Disconnect-LXCA
```

Remove-LXCAConfigProfileFromSystem

This cmdlet removes (deactivates) assignment of a server profile from the associated server. The server profile is still available on the Lenovo XClarity Administrator server.

Note: You can pipe a ConfigProfile object to this cmdlet.

Syntax

```
Remove-LXCAConfigProfileFromSystem [-Connection LXCAConnection]
    -Profile ConfigProfile
    [-Force]
    [CommonParameter]
```

```
Remove-LXCAConfigProfileFromSystem [-Connection LXCAConnection]  
                                     -ProfileID String  
                                     [-Force]  
                                     [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Profile *ConfigProfile*

The server profile object. You can obtain a server profile object using the [Get-LXCAConfigProfile](#) cmdlet.

-ProfileID *String*

The server profile ID. You can obtain a list of all server profile IDs using the [Get-LXCAConfigProfile](#) cmdlet.

-Force

Forces removal (deactivation) of the profile even if the target device is not reachable.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet removes the server profile from assignment. No objects are returned.

Examples

The following example deactivates the server profile from a specified server.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
$profiles = Get-LXCAConfigProfile | where{$_status -eq "ASSIGNED"}  
Remove-LXCAConfigProfileFromSystem -ProfileId $profiles[0].Id  
  
Disconnect-LXCA
```

Switch configuration cmdlets

The following PowerShell cmdlets are available for performing switch configuration functions.

Backup-LXCASwitchConfig

This cmdlet backs up configuration data for a RackSwitch or Flex System switch. When backing up configuration data, the switch configuration data is imported into Lenovo XClarity Administrator from the target switch as a configuration-data file.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Backup-LXCASwitchConfig [-Connection LXCAConnection]  
                        -FileName String  
                        -Uuid String  
                        [-Description String]
```

[*CommonParameter*]

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-FileName *String*

Specifies the name of the configuration-data file.

-Uuid *String*

Specifies the UUID of the switches to be backed up. To obtain the switch UUIDs, use the [Get-LXCARackSwitch](#) or [Get-LXCACHassisMember](#) cmdlet.

-Description *String*

Specifies the description of the configuration-data.

If a description is not specified, the following default text is used: “*switch_name* configuration at *timestamp*.”

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- System.String

This cmdlet returns a string with result of the backup operation.

Examples

The following example backs up configuration data for a specific switch to a file name Switch1.cfg.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Backup-XCASwitchConfig -UUID AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA -FileName Switch1.cfg `
    -Description "Switch1 configuration"

Disconnect-LXCA
```

Get-LXCASwitchConfigData

This cmdlet retrieves information about configuration-data files for RackSwitch and Flex System switches.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCASwitchConfigData [-Connection LXCACConnection]
                        [-Uuid String]
                        [-ConfigData]
                        [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid String

Specifies the UUID of the switch for which information is to be retrieved.

If not specified, this cmdlet retrieves information about configuration-data files for all managed RackSwitch and Flex System switches.

-ConfigData

If specified, detailed information about the configuration-data file is returned. This includes the switch UUID, file ID, file name, date, size, and comments.

Tip: If not specified, you can retrieve detailed information using the **ConfigDatas** property in the [Lenovo.SysMgmt.LXCA.Integration.Data.SwitchConfigData](#) object (see the examples below).

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- **[Lenovo.SysMgmt.LXCA.Integration.Data.SwitchConfigData\[\]](#)**

If **-Uuid** and **-ConfigData** are not specified, the cmdlet returns an array of [Lenovo.SysMgmt.LXCA.Integration.Data.SwitchConfigData](#) objects, for all managed switches.

- **[Lenovo.SysMgmt.LXCA.Integration.Data.SwitchConfigData](#)**

If **-Uuid** is specified and **-ConfigData** is not specified, the cmdlet returns a single [Lenovo.SysMgmt.LXCA.Integration.Data.SwitchConfigData](#) object, for the switch with the given UUID.

- **[Lenovo.SysMgmt.LXCA.Integration.Data.ConfigData\[\]](#)**

If **-ConfigData** is specified, the cmdlet returns an array of [Lenovo.SysMgmt.LXCA.Integration.Data.SwitchConfigData](#) objects.

In this case, if **-Uuid** is specified, the array contains data for the switch with the given UUID; otherwise, the array contains data for all managed switches.

Examples

- The following example returns information about configuration-data files for all managed switches, and then displays the detailed information.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

The following two blocks are equivalent in generated output.

```
# Block 1
$switchesConfig = Get-LXCASwitchConfigData
# $switchesConfig is an array of Lenovo.SysMgmt.LXCA.Integration.Data.SwitchConfigData objects.
$switchesConfig.ConfigDatas
```

```
# Block 2
Get-LXCASwitchConfigData -ConfigData
```

```
Disconnect-LXCA
```

- The following example returns information about configuration-data files for the switch with the specified UUID, and then displays the detailed information.

```

$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$switchUuid = "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"

# The following two blocks are equivalent in generated output.

# Block 1
$switchConfig = Get-LXCASwitchConfigData -Uuid $switchUuid
# $switchConfig is a single Lenovo.SysMgmt.LXCA.Integration.Data.SwitchConfigData object.
$switchConfig.ConfigDatas

# Block 2
Get-LXCASwitchConfigData -Uuid $switchUuid -ConfigData

Disconnect-LXCA

```

Get-LXCASwitchConfigFile

This cmdlet downloads (exports) one or more switch configuration-data files to the local system.

Note: You cannot pipe objects to this cmdlet.

Syntax

```

Get-LXCASwitchConfigFile [-Connection LXCAConnection]
                        [-FileName String]
                        [-Destination String]
                        [-Overwrite]
                        [CommonParameter]

```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-FileName *String*

Specifies the name of the configuration-data file.

-Destination *String*

Specifies the local directory where the file is to be saved.

If not specified, this cmdlet downloads a file named LXCAConfig_file.zip to the current PowerShell working directory.

-Overwrite

If specified, the cmdlet overwrites the local file when a file with that same name exists already.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a string that contains the full name of the retrieved file.

Examples

The following example downloads a single configuration-data file to the default directory on the local system, and overwrites the file if it already exists.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
Get-LXCASwitchConfigFile -FileName Switch1.cfg -Overwrite
```

```
Disconnect-LXCA
```

The following example downloads all configuration-data files to the current PowerShell working directory on the local system.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $cred -SkipCertificateCheck
$configData = Get-LXCASwitchConfigData -ConfigData
$files = @()
foreach ($config in $configData)
{
    $files += $config.FileName
}
$configFilesArchive = Get-LXCASwitchConfigFile -FileName $files
$configFilesArchive
Disconnect-LXCA
```

Import-LXCASwitchConfig

This cmdlet imports a switch configuration-data file from the local system into Lenovo XClarity Administrator.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Import-LXCASwitchConfig [-Connection Connection]
                        -FilePath String[]
                        [-Uuid String]
                        [-Comment String]
                        [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-FilePath *String[]*

Specifies the full directory and file name of the configuration-data file to import.

-Uuid *String*

Specifies the UUID of the switch to associate with the imported file. To obtain the switch UUIDs, use the [Get-LXCARackSwitch](#) or [Get-LXCACHassisMember](#) method

-Comment *String*

Specifies the description of the configuration-data file.

If a description is not specified, the following default text is used: "*switch_name* configuration at *timestamp*."

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For

detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlets imports the configuration-data file into the image repository.

Examples

The following example imports the Switch1.cfg file from the local system in the c:\configdata directory and associates it with a specific.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Import-LXCASwitchConfig -Uuid $switch -FilePath c:\configdata\Switch1.cfg -Comment "Switch1 configuration"

Disconnect-LXCA
```

Remove-LXCASwitchConfigFile

This cmdlet deletes a switch configuration-data file from Lenovo XClarity Administrator.

Note: You can pipe OSImage objects to this cmdlet.

Syntax

```
Remove-LXCASwitchConfigFile [-Connection LXCACConnection]
                             -FileName String]
                             [-Uuid String]
                             [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-FileName *String*]

Specifies the name of the configuration-data file to delete.

-Uuid *String*

Specifies the UUID of the switch that is associated with the configuration-data file. To obtain the switch UUIDs, use the [Get-LXCARackSwitch](#) or [Get-LXCAChassisMember](#) method.

Note: This parameter is required if the more than one switch have configuration-data files with the same name.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlets removes the configuration-data file from XClarity Administrator. No objects are returned.

Examples

The following example deletes a switch configuration-data file.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Remove-LXCASwitchConfigFile -UUID AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA -FileName Switch1.cfg

Disconnect-LXCA
```

Restore-LXCASwitchConfig

This cmdlet restores configuration data for a RackSwitch or Flex System switch. The switch configuration-data file is downloaded from Lenovo XClarity Administrator to the target switch, and the configuration takes effect automatically.

Note: You can pipe OSImage objects to this cmdlet.

Syntax

```
Restore-LXCASwitchConfig [-Connection LXCACConnection]
                        -FileName String
                        -Uuid String
                        [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-FileName *String*

Specifies the name of the configuration-data file to restore on the specified switch.

-Uuid *String*

Specifies the UUID of the switch to be restored. To obtain the switch UUIDs, use the [Get-LXCARackSwitch](#) or [Get-LXCACHassisMember](#) method.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlets removes the configuration-data file from XClarity Administrator. No objects are returned.

Examples

The following example restores configuration data on a specific switch using a compatible configuration-data file.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Restore-LXCASwitchConfig -UUID AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA -FileName Switch1.cfg

Disconnect-LXCA
```

Set-LXCASwitchPort

This cmdlet enables or disables ports on a managed RackSwitch or Flex System switch.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCASwitchPort [-Connection LXCAConnection]  
                  -Action String  
                  -Port String[]  
                  -Uuid String  
                  [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Action *String*

Specifies whether to enable or disable the specified ports. This can be of the following values:

- **enable**
- **disable**

To determine the current state of the port, use the [Get-LXCARackSwitch](#) cmdlet.

-Port *String[]*

Specifies the name of one or more ports. To obtain the port names, use the [Get-LXCARackSwitch](#) cmdlet.

-Uuid *String*

Specifies the UUID a managed RackSwitch or Flex System switch that contains the ports to be changed.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet modifies the port state. No object is returned.

Examples

- The following example disables specific ports in a managed switch.

```
$cred = Get-Credential  
Connect-LXCA -Host 1.111.11.11 -Port 443 -Credential $cred  
  
Set-LXCASwitchPort -Action disable -Port "Ethernet1/1","Ethernet1/3" -Uuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
  
Disconnect-LXCA
```

- The following example enables all ports in a managed switch.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred  
  
$rackSwitch = Get-LXCARackSwitch -Uuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
$portList=@();  
foreach($i in $rackSwitch.Ports) { $portList+=( , $i.Port)}  
Set-LXCASwitchPort -Action enable -Port $portList -Uuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
  
Disconnect-LXCA
```

Firmware update cmdlets

The following PowerShell cmdlets are available for performing firmware updates functions.

Get-LXCASystemCompliance

This cmdlet retrieves firmware-compliance information about one or more managed devices (including CMMs, switches, servers, and storage devices) to which a compliance policy is assigned.

Note: Firmware compliance policies are not supported for ThinkServer servers.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCASystemCompliance [-Connection LXCACConnection]  
                        -SystemType SystemType  
                        -SystemUuid String  
                        [CommonParameter]
```

```
Get-LXCASystemCompliance [-Connection LXCACConnection]  
                        -SystemType SystemType  
                        -Uuids String[]  
                        [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-SystemType *SystemType*

Specifies the type of the target device that is specified using the **-SystemUuid** parameter. Specify one of the following system types:

- **Chassis.** The specified device is a chassis and includes all CMMs, FlexSystem switches, and compute nodes in the chassis.
- **CMM.** The specified device is a CMM.
- **ComputeNode.** The specified device is a Flex System compute node.
- **EdgeServer.** The specified device is an edge server.
- **IOSwitch.** The specified device is a Flex System switch.
- **RackServer.** The specified device is a rack or tower server.
- **RackSwitch.** The specified device is a RackSwitch switch.
- **Storage.** The specified device is a storage device.

-SystemUuid *String*

Specifies the UUID of the target device. The target device can be a chassis, CMM, switch, server, or storage device. If you specify a chassis, information for all devices that are installed in the chassis are returned.

-Uuids *String[]*

Specifies UUIDs of one or more target devices of the same type. The target devices can be chassis, CMMs, Flex System switches, Flex System compute nodes, System x servers, RackSwitch switches or storage devices.

Note: Chassis UUIDs are not supported. If you specify chassis UUIDs, a null object is returned.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For

detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more SystemCompliance objects that contain firmware-compliance information.

Examples

- The following example retrieves firmware-compliance information for all devices in a chassis.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$compliance = Get-LXCASystemCompliance -SystemUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA -SystemType Chassis

Disconnect-LXCA
```
- The following example retrieves firmware-compliance information for multiple rack servers.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$compliance = Get-LXCASystemCompliance -SystemType RackServer`
                -Uids AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

Disconnect-LXCA
```

Get-LXCAUpdateCompliancePolicy

This cmdlet retrieves information about the firmware-compliance policy that is assigned to a managed device.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAUpdateCompliancePolicy [-Connection LXCAConnection]
                                [CommonParameter]
```

```
Get-LXCAUpdateCompliancePolicy [-Connection LXCAConnection]
                                -SystemUuid String
                                -SystemType String
                                [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-SystemUuid *String*

Specifies the UUID of the target device to which a firmware-compliance policy is assigned. The server can be a CMM, Flex System switch, or server.

If a firmware-compliance policy has been assigned to the specified device, the returned policy includes the UUID for that device. If no UUID is specified, the cmdlet returns information about all policies.

-SystemType *SystemType*

Specifies the type of device that was identified with the **-SystemUuid** parameter. Specify one of the following system types:

- **CMM.** The specified device is a CMM.
- **ComputeNode.** The specified device is a Flex System compute node.

- **EdgeServer.** The specified device is an edge server.
- **IOSwitch.** The specified device is a Flex System switch.
- **RackServer.** The specified device is a rack or tower server.
- **RackSwitch.** The specified device is a RackSwitch switch.
- **Storage.** The specified device is a storage device.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more CompliancePolicy objects that contain firmware-compliance policy information.

Examples

- The following example retrieves all firmware-compliance policies.

```
$cred = Get-Credential
$connection = Connect-LXCA -Host 192.0.2.0 -Credential $cred

$policy = Get-LXCAUpdateCompliancePolicy

Disconnect-LXCA
```

- The following example retrieves the firmware-compliance policy that is applicable to a specific server in a Flex System chassis.

```
$cred = Get-Credential
$connection = Connect-LXCA -Host 192.0.2.0 -Credential $cred

$policy = Get-LXCAUpdateCompliancePolicy -SystemUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA`
                                           -SystemType ComputeNode

Disconnect-LXCA
```

Get-LXCAUpdatePackage

This cmdlet retrieves a list of firmware-update packages in the firmware-updates repository.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAUpdatePackage [-Connection LXCACConnection]
                    [-MachineType String]]
                    [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-MachineType *String*]]

Specifies one or more machine types. If no machine type is specified, update packages for all machine types are returned.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For

detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more UpdatePackage objects.

Examples

The following example retrieves information about the update packages for all available machine types in the firmware-updates repository.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$packages = Get-LXCAUpdatePackage
```

```
Disconnect-LXCA
```

The following example retrieves information about the update packages for specific machine types in the firmware-updates repository.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$packages = Get-LXCAUpdatePackage -MachineType 7893,8724
```

```
Disconnect-LXCA
```

Import-LXCAUpdatePackage

This cmdlet imports one or more local firmware and management-server update packages to the firmware-updates repository.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Import-LXCAUpdatePackage [-Connection LXCACConnection]
                        -File String[]
                        [-ApplianceUpdate]
                        [-AsJob]
                        [-WaitForJob]
                        [CommonParameter]
```

```
Import-LXCAUpdatePackage [-Connection LXCACConnection]
                        -Folder String
                        [-FixId String[]]
                        [-ApplianceUpdate]
                        [-AsJob]
                        [-WaitForJob]
                        [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-File *String[]*

Specifies one or more file names of update packages to be imported.

-Folder String

Specifies the directory that contains the update packages to be imported. The packages must be in the specified folder; they cannot be in child folders.

-FixId String[]

Specifies one or more update fix IDs to be imported. If no fix ID is specified, all available update files are imported.

-ApplianceUpdate

When specified, the cmdlet uploads the update package to the management-server repository. If not specified, the cmdlet uploads the update package to the firmware-updates repository.

-AsJob

When specified, the cmdlet starts a background job in XClarity Administrator and returns a Job object immediately.

-WaitForJob

When specified with the **-AsJob** parameter, PowerShell waits for this background job in XClarity Administrator to either start or complete before running the next cmdlet in the script. This can be one of the following values.

- **ToStart.** PowerShell waits until this background job starts in XClarity Administrator, so multiple jobs can run synchronously.
- **ToEnd.** PowerShell waits until this background job completes in XClarity Administrator, so that one only job runs at a time.

If this parameter not specified, this cmdlet does not wait. The next cmdlet in the script runs regardless of whether this background job starts or completes.

If **-AsJob** is not specified, this parameter is ignored.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet adds the firmware-update packages to the appropriate repository. If you specify the **-AsJob** parameter, the cmdlet returns a Job object.

Examples

- The following example imports the firmware-update packages to the XClarity Administrator server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

Import-LXCAUpdatePackage -File c:\Lenovo_fw_bmc_gf6t60a_windows_i386.exe,`
                          c:\Lenovo_fw_bmc_gf6t60a_windows_i386.xml
```

```
Disconnect-LXCA
```

- The following example imports all firmware-update packages that are located in the c:\updates folder to the XClarity Administrator server as a job that runs in the background. A wait loop is enforced while the job is still in running state.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$job = Import-LXCAUpdatePackage -Folder c:\updates -AsJob
while ($job.State -eq "Running")
{
```

```

    Start-Sleep -Seconds 5
}

```

```
Disconnect-LXCA
```

- The following example imports all firmware-update packages that are located in the c:\updates folder to the XClarity Administrator server as a job that runs in the background. PowerShell waits until the import job completes before .ing fromXClarity Administrator.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
Import-LXCAUpdatePackage -Folder c:\updates -AsJob -WaitForJob ToEnd
```

```
Disconnect-LXCA
```

- The following example imports all management-server update packages that are located in the c:\updates folder the XClarity Administrator server as a job that runs in the background. PowerShell waits until the import-update job completes before disconnecting fromXClarity Administrator.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
Import-LXCAUpdatePackage -Folder c:\updates -AsJob -ApplianceUpdate -WaitForJob ToEnd
```

```
Disconnect-LXCA
```

- The following example imports the management-server update package that is located in the c:\updates folder to the XClarity Administrator server and then applies the updates. The import and installation both run as jobs in the background. PowerShell waits until the import-update job completes before applying the update, and then waits for the apply-update job to start before disconnecting fromXClarity Administrator.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
Import-LXCAUpdatePackage -ApplianceUpdate -Folder c:\updates -AsJob -WaitForJob ToEnd
```

```
$fixID = "lnvgy_sw_lxca_thinksystemrepo3-2.4.0_anyos_noarch"
Install-LXCAUpdateManagementServer -FixId $fixID -AsJob -WaitForJob ToEnd
```

```
Disconnect-LXCA
```

Install-LXCAUpdatePackage

This cmdlet applies one or more firmware updates to a target system with or without the compliance policy comparison result for the target system.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Install-LXCAUpdatePackage [-Connection LXCAConnection]
                        -Compliance SystemCompliance
                        -UpdateRule String
                        -ActivationMode String
                        -OnErrorMode String
                        [-ForceUpdate]
                        [-AsJob]
                        [-Schedule ScheduleObject]
                        [CommonParameter]
```

```
Install-LXCAUpdatePackage [-Connection LXCAConnection]
                        -Compliances SystemCompliance[]
```

```
-UpdateRule String
-ActivationMode String
-OnErrorMode String
[-ForceUpdate]
[-AsJob]
[-Schedule ScheduleObject]
[CommonParameter]
```

```
Install-LXCAUpdatePackage [-Connection LXCAConnection]
-Compliance SystemCompliance
-FixId String[]
-ActivationMode String
-OnErrorMode String
[-ForceUpdate]
[-AsJob]
[-Schedule ScheduleObject]
[CommonParameter]
```

```
Install-LXCAUpdatePackage [-Connection LXCAConnection]
-Uid String
-SystemType SystemType
-ComponentName String[]
-FixId String[]
-ActivationMode String
-OnErrorMode String
[-ForceUpdate]
[-AsJob]
[-Schedule ScheduleObject]
[CommonParameter]
```

```
Install-LXCAUpdatePackage [-Connection LXCAConnection]
-ApplyBundle
-Compliance SystemCompliance
[-AsJob]
[-Schedule ScheduleObject]
[CommonParameter]
```

```
Install-LXCAUpdatePackage [-Connection LXCAConnection]
-ApplyBundle
-Compliances SystemCompliance[]
[-AsJob]
[-Schedule ScheduleObject]
[CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-applyBundle

If specified, this cmdlet applies firmware updates to all components of specified ThinkSystem SR635 and SR655 devices according to the assigned firmware-compliance policy using a bundled image that contain all applicable firmware packages.

Attention: Before using the bundled-update function, review important information regarding ThinkSystem SR635 and SR655 devices and the bundle-update function limitations (see [Firmware update considerations](#) in the Lenovo XClarity Administrator online documentation).

-Compliance *SystemCompliance*

Specifies the firmware compliance-check results.

-Compliances *SystemCompliance[]*

Specifies one or more the firmware compliance-check results.

-UpdateRule *String*

Specifies which firmware updates in the firmware compliance-check result are to be applied to target device. You can specify one of the following values:

- **All**. All firmware updates in the compliance-check result are applied. To also apply to firmware updates that are in compliance, specify the **-ForceUpdate** parameter.
- **NotCompliantOnly**. Only those firmware updates in the compliance-check result that are not compliant are applied.

-Uuid *String*

Specifies the UUID of the target system.

-SystemType *SystemType*

Specifies the type of the target system. The type can be one of the following values.

- **Chassis**. The specified device is a chassis and includes all CMMs, FlexSystem switches, and compute nodes in the chassis.
- **CMM**. The specified device is a CMM.
- **ComputeNode**. The specified device is a Flex System compute node.
- **EdgeServer**. The specified device is an edge server.
- **IOSwitch**. The specified device is a Flex System switch.
- **RackServer**. The specified device is a rack or tower server.
- **RackSwitch**. The specified device is a RackSwitch switch.
- **Storage**. The specified device is a storage device.

-ComponentName *String[]*

Specifies the component name to be applied to the target device. The component name can be retrieved from cmdlet [Get-LXCAUpdatePackage](#).

The component name must be a one-to-one match to the FixId in the same order.

-FixId *String[]*

Specifies the ID of one or more firmware updates to be applied to target device. The component name can be retrieved from cmdlet [Get-LXCAUpdatePackage](#).

-ActivationMode *String*

Specifies when to restart the target device to activate new firmware. You can specify one of the following:

- **Immediate**. During the update process, the target device might be automatically restarted multiple times until the entire process is complete. Ensure that you quiesce all applications on the target device before you proceed.
- **Delayed**. Some but not all update operations are performed. The target device must be restarted manually to continue the update process. Additional restarts are then performed automatically until the update operation completes. This rule is supported only for servers and rack switches.
- **Prioritized**. Firmware updates on the baseboard management controller are activated immediately; all other firmware updates are activated the next time the device is restarted. Additional restarts are then performed until the update operation completes. This rule is supported only for servers.

-OnErrorMode *String*

Specifies how to handle an update error. You can specify one of the following values:

- **StopAllUpdatesOnError**. If an error occurs while updating any of the components in a target device (such as an adapter or baseboard management controller firmware), the firmware-updates process stops for all targeted devices in the current firmware-update job. In this case, none of the updates in the firmware-updates package for the device are activated. The current firmware that is installed on all targeted devices remains in effect.

- **ContinueOnError.** If an error occurs while updating one of the components in a target device (such as an adapter or management controller firmware), the firmware-updates process does not update the firmware for that specific component. However, the firmware-updates process continues to update other components for the device and continues with all other updates in the current update job.
- **ContinueToNextSystemOnError.** If an error occurs while updating one of the components in a target device (such as an adapter or management controller firmware), the firmware-updates process stops attempting to update the firmware for that specific device. The current firmware that is installed on that device remains in effect. However, if you have selected multiple devices as part of the current firmware-update job, the management server continues to update the firmware for the remaining devices selected as part of the current job.

-ForceUpdate

If specified, this cmdlet applies firmware updates to the specified devices even if the firmware is in compliance. Otherwise, if the firmware is already compliant, this cmdlet skips the firmware updates on the specified devices.

-AsJob

If specified, the cmdlet runs as background job and returns a Job object immediately.

-Schedule *ScheduleObject*

Creates a scheduled job (task) for this cmdlet on the XClarity Administrator server.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

If you specify the **-Asjob** parameter, this cmdlet returns a Job object. You can retrieve the UpdateResult object from Job.Output when the job is complete. You can retrieve the SubJob from Job.GetSubJobStatus() to get the status of each update.

If you do not specify the **-Asjob** parameter, this cmdlet returns an UpdateResult object.

Examples

- The following example retrieves a compliance-check result from a Flex System compute node and then applies all non-compliant firmware updates to that compute node.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred
$complianceResult = Get-LXCASystemCompliance -SystemUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA`
                    -SystemType ComputeNode
$updateResult = Install-LXCAUpdatePackage -Compliance $complianceResult -UpdateRule NotCompliantOnly`
                    -ActivationMode Immediate -OnErrorMode StopOnError -AsJob
```

```
Disconnect-LXCA
```

- The following example retrieves compliance-check results for a list of rack servers and then applies firmware updates to those servers.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred
$complianceResults = Get-LXCASystemCompliance -Uuids $nodes -SystemType RackServer
$updateResult = Install-LXCAUpdatePackage -Compliance $complianceResults -UpdateRule NotCompliantOnly`
                    -ActivationMode Immediate -OnErrorMode StopOnError -AsJob
```

```
Disconnect-LXCA
```

- The following example retrieves a compliance-check result from a Flex System compute node and then applies a firmware update to that compute node even if the firmware is already compliant.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred
$complianceResult = Get-LXCASystemCompliance -SystemUuid $node -SystemType ComputeNode
$updateResult = Install-LXCAUpdatePackage -Compliance $complianceResult -ActivationMode Immediate`
                -FixId $complianceResult.Firmwares[0].ComponentName`
                -OnErrorMode StopOnError -ForceUpdate -AsJob
```

```
Disconnect-LXCA
```

- The following example retrieves the compliance policy from a Flex System compute node and assigns it to another compute node to get the compliance-check result. The example then applies all available firmware updates to the compute node as a job that runs in the background.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$policy = Get-LXCAUpdateCompliancePolicy -SystemUuid $node -SystemType ComputeNode
$complianceResult = Join-LXCAUpdateCompliancePolicy -Policy $policy -SystemUuid $node
                  -SystemType ComputeNode
$job = Install-LXCAUpdatePackage -Compliance $complianceResult -UpdateRule All -ForceUpdate`
                -ActivationMode Immediate -OnErrorMode StopOnError -AsJob
```

```
while ($job.State -eq "Running")
{
    Start-Sleep -Seconds 5
    $job.GetSubJobStatus()
}
$updateResult = $job.Output
```

```
Disconnect-LXCA
```

- The following example creates a scheduled job (task) to apply two firmware updates to the target device, without a firmware compliance policy.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$fixid = "lnvgy_fw_uefi_tce123i-2.10_anyos_32-64","lnvgy_fw_dsa_dsyt98i-9.03_anyos_32-64"
$schedule = New-LXCAScheduleObj -JobName $name -OccurrenceDate ('2017-12-22 4:35 PM')
$ret = Install-LXCAUpdatePackage -Uuid $node -SystemType RackServer -OnErrorMode ContinueOnError`
        -ComponentName "UEFI","Diagnostics" -FixId $fixid -ActivationMode Delayed`
        -Schedule $schedule
```

```
Disconnect-LXCA
```

Join-LXCAUpdateCompliancePolicy

This cmdlet assigns a firmware-compliance policy to a specified device.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Join-LXCAUpdateCompliancePolicy [-Connection LXCAConnection]
                                -Policy CompliancePolicy
                                -SystemUuid String
                                -SystemType SystemType
                                [CommonParameter]
```

```
Join-LXCAUpdateCompliancePolicy [-Connection LXCAConnection]
                                -PolicyName String
                                -SystemUuid String
```

-SystemType *SystemType*
[*CommonParameter*]

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Policy *CompliancePolicy*

Specifies an firmware-compliance policy object.

-PolicyName *String*

Specifies the name of firmware-compliance policy.

-SystemUuid *String*

Specifies the UUID of the target device.

-SystemType *SystemType*

Specifies the type of the target device that is specified using the **-SystemUuid** parameter. You can specify one of the following system types:

- **CMM**. The specified device is a CMM.
- **ComputeNode**. The specified device is a Flex System compute node.
- **EdgeServer**. The specified device is an edge server.
- **IOSwitch**. The specified device is a Flex System switch.
- **RackServer**. The specified device is a rack or tower server.
- **RackSwitch**. The specified device is a RackSwitch switch.
- **Storage**. The specified device is a storage device.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a SystemCompliance object for the target device.

Examples

The following example retrieves a firmware-compliance policy and assigns it to a specific server in a Flex System chassis.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$policy= Get-LXCAUpdateCompliancePolicy
$complianceResult = Join-LXCAUpdateCompliancePolicy -Policy $policy[0] -SystemType ComputeNode`
                    -SystemUuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Disconnect-LXCA
```

Update-LXCASFirmwareRepository

This cmdlet refreshes the firmware catalog by retrieving information (metadata) about firmware updates and UpdateXpress System Packs (UXSPs) from the Lenovo Support website, or downloads specific firmware updates or UXSPs from Lenovo XClarity Support website and stores the files in the updates repository.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Update-LXCACFirmwareRepository [-Connection LXCACConnection]  
    -RefreshCatalog  
    -MachineTypes System.String[]  
    [-UXSPs]  
    [-LatestOnly]  
    [-AsJob]  
    [-WaitForJob System.String]  
    [CommonParameter]
```

```
Update-LXCACFirmwareRepository [-Connection LXCACConnection]  
    -RefreshCatalog  
    [-UXSPs]  
    [-LatestOnly]  
    [-AsJob]  
    [-WaitForJob System.String]  
    [CommonParameter]
```

```
Update-LXCACFirmwareRepository [-Connection LXCACConnection]  
    -AcquirePackages  
    -PackagesIDs System.String[]  
    [-AsJob]  
    [-WaitForJob System.String]  
    [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-RefreshCatalog

When specified, the cmdlet updates the firmware catalog by retrieving information (metadata) about firmware updates from the Lenovo Support website.

-UXSPs

When specified, the cmdlet updates the firmware catalog by retrieving information (metadata) about UXSPs from the Lenovo Support website.

-AcquirePackages

When specified, the cmdlet downloads the specified firmware updates from Lenovo XClarity Support website and stores the files in the updates repository.

-LatestOnly

When specified, retrieves information about the most current version of all firmware updates.

-AsJob

When specified, the cmdlet runs as background job, and returns a Job object immediately. If not specified, the cmdlet runs in synchronous mode.

-WaitForJob *System.String*

When specified with the **-AsJob** parameter, PowerShell waits for this background job in Lenovo XClarity Administrator to either start or complete before running the next cmdlet in the script. This can be one of the following values.

- **ToStart.** PowerShell waits until this background job starts in XClarity Administrator, so multiple jobs can run synchronously.
- **ToEnd.** PowerShell waits until this background job completes in XClarity Administrator, so that one only job runs at a time.

If this parameter is omitted, this cmdlet does not wait. The next cmdlet in the script runs regardless of whether this background job starts or completes.

If **-AsJob** is omitted, this parameter is ignored.

-MachineTypes *System.String*

Specifies the machine types for the firmware updates or UXSPs information to be retrieved.

-PackagesIDs *String*

Specifies the firmware packages IDs (fixIDs) to be downloaded.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- System.Management.Automation.Job

If you specify the **-AsJob** parameter, this cmdlets returns a System.Management.Automation.Job object.

- [Lenovo.SysMgmt.LXCA.Integration.Data.FirmwareRepositoryData](#)

This cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.FirmwareRepositoryData](#) object which contains information about the firmware repository.

Examples

- The following example downloads specified firmware packages. The cmdlet is run as a PowerShell job.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$packs = @(
    "lnvgy_fw_sraidmr35_930-50.0.1-0374-1_linux_x86-64",
    "lnvgy_fw_xcc_cdi346l-3.60_anyos_noarch",
    "lnvgy_utl_uxsp_c6sp03p-1.40_platform_32-64"
)

$result = Update-LXCAFirmwareRepository -AcquirePackages -PackagesIDs $packs -AsJob -WaitForJob ToEnd

# show job information.
$job|Format-List

# show general information about the downloaded packages contained in job object.
$job.Output|Format-List

# show the number of firmware packages included which were downloaded.
$job.Output.FirmwarePackages.Count

# show the details of one firmware package.
$job.Output.FirmwarePackages[0]

Disconnect-LXCA
```

- The following example retrieves the firmware catalog for all firmware. The cmdlet is run as a PowerShell job.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$job = Update-LXCAFirmwareRepository -RefreshCatalog -AsJob -WaitForJob ToEnd
```

```

# show job information.
$job|Format-List

# show general information about the firmware catalog contained in job object.
$job.Output|Format-List

# show the number of firmware packages included in this catalog.
$job.Output.FirmwarePackages.Count

# show the details of one firmware package.
$job.Output.FirmwarePackages[20]

Disconnect-LXCA

```

- The following example retrieves the firmware catalog for latest UXSPs packages.

```

$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$result = Update-LXCASoftwareRepository -RefreshCatalog -UXSPs -LatestOnly

# show general information about the UXSPs catalog.
$result|Format-List

# show the number of firmware packages included in this catalog.
$result.FirmwarePackages.Count

# show the details of one firmware package.
$result.FirmwarePackages[20]

Disconnect-LXCA

```

Operating-system deployment cmdlets

The following PowerShell cmdlets are available for performing operating-system deployment functions.

Get-LXCADeployableServer

This cmdlet retrieves a list of servers that are ready for operating-system deployment.

Use this cmdlet to determine if an operating system can be deployed to a server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCADeployableServer [-Connection LXCACONNECTION]
                        [-Uuid String[]]
                        [CommonParameter]
```

```
Get-LXCADeployableServer [-Connection LXCACONNECTION]
                        [-ImageID String]
                        [CommonParameter]
```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String[]*

Specifies of one or more server UUIDs. If no UUID is specified, all deployable servers are returned.

-ImageID *String*

Specifies the ID of an operating-system image that can be deployed to target servers. If no image ID is specified, all deployable servers are returned.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more DeployableServer objects.

Examples

The following example retrieves a list of all deployable servers.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$servers= Get-LXCADeployableServer

Disconnect-LXCA
```

Get-LXCADeployGlobalSetting

This cmdlet retrieves the global settings that are used to deploy operating systems, such as IP assignment type (static or DHCP), storage location, and deployment credentials.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCADeployGlobalSetting [-Connection LXCACConnection]
                             [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a DeployGlobalSetting object, for example:

```
public class DeployGlobalSetting
{
    public CredentialSetting[] Credentials;
    public string IpAssignment; //staticv4|staticv6|dhcpv4
    public ActiveDirectorySetting ActiveDirectory;
    public LicenseKeySetting LicenseKeys;

    public class CredentialSetting
```



```

{
    public string Type;        //"ESXi", "LINUX", or "WINDOWS", readonly
                              //"RHEL/ESXi" can also be used but will be deprecated in a future
                              //release.
    public string Name;       //"root" or "Administrator", readonly
    public string Password;   //Empty string
}

public class ActiveDirectorySetting
{
    public DomainSetting[] AllDomains;
    public string DefaultDomain;

    public class DomainSetting
    {
        public int ID;
        public string DomainName;
        public string OU;
    }
}

public class LicenseKeySetting
{
    public Win2022R1Key Win2022R1;
    public Win2019R1Key Win2019R1;
    public Win2016R1Key Win2016R1;
    public Windows10Key Windows10;
    public Windows11Key Windows11;

    public class Win2016R1Key
    {
        public string StandardLicenseKey;
        public string DataCenterLicenseKey;
    }
    public class Win2019R1Key
    {
        public string StandardLicenseKey;
        public string DataCenterLicenseKey;
    }
    public class Win2022R1Key
    {
        public string StandardLicenseKey;
        public string DataCenterLicenseKey;
    }
    public class Windows10Key
    {
        public string EnterpriseLicenseKey;
        public string WorkstationLicenseKey;
    }
    public class Windows11Key
    {
        public string EnterpriseLicenseKey;
        public string WorkstationLicenseKey;
    }
}
}

```

Examples

The following example retrieves the global settings that are used for operating-system deployment.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$setting = Get-LXCADeployGlobalSetting
```

```
Disconnect-LXCA
```

Get-LXCAOSImage

This cmdlet retrieves information about the operating-system images in the OS-image repository.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAOSImage [-Connection LXCACConnection]
                [-ImageID String]]
                [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ImageID *string*]

Specifies one or more operating-system image IDs. If no image ID is specified, all operating-system images in the OS-image repository are returned.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more OSImage objects with image information.

Examples

The following example retrieves information about all operating-systems images in the OS-images repository.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$images = Get-LXCAOSImage
```

```
Disconnect-LXCA
```

Import-LXCAOSImage

This cmdlet imports a local OS image or device driver into the OS-images repository.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Import-LXCAOSImage [-Connection LXCACConnection]
                  -ImagePath String
```

```
[-ChecksumType String]  
[-ChecksumValue String]  
[-AsJob]  
[-WaitForJob String]  
[CommonParameter]
```

```
Import-LXCAOSImage [-Connection LXCACConnection]  
-ImagePath String  
-DeviceDriver  
[-DeviceDriver]  
-OSImageName String  
[-AsJob]  
[-WaitForJob String]  
[CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ImagePath *String*

Specifies the file path and file name to be imported.

-ChecksumType *String*

Specifies the checksum type. You can specify one of the following values:

- MD5
- SHA1
- SHA256

If you specify the checksum types, you must also specify the **-ChecksumValue** parameter.

-ChecksumValue *String*

Specifies the checksum value to be used to check the integrity and security of the uploaded file.

-DeviceDriver

If specified, indicates that the import file is a device driver.

-OSImageName *String*

Specifies the name of the OS image to which you want to add the device driver (for example, redhat7.0).

Note: The OS image must exist in the OS images repository.

-AsJob

When specified, the cmdlet runs as background job and returns a Job object immediately.

-WaitForJob *String*

When specified with the **-AsJob** parameter, PowerShell waits for this background job in XClarity Administrator to either start or complete before running the next cmdlet in the script. This can be one of the following values.

- **ToStart.** PowerShell waits until this background job starts in XClarity Administrator, so multiple jobs can run synchronously.
- **ToEnd.** PowerShell waits until this background job completes in XClarity Administrator, so that only one job runs at a time.

If this parameter is not specified, this cmdlet does not wait. The next cmdlet in the script runs regardless of whether this background job starts or completes.

If **-AsJob** is not specified, this parameter is ignored.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlets imports the image into the image repository. If you specify the **-AsJob** parameter, this cmdlets returns a Job object.

Examples

- The following example imports an operating-system image into the OS-images repository in synchronous mode.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Import-LXCAOSImage -ImagePath c:\ESXi-5.1.1-1065491-RTM-Lenovo-20130607-GA.iso
$osimages = Get-LXCAOSImage

Disconnect-LXCA
```

- The following example imports a local OS image into the OS-images repository as a job that runs in the background and uses a loop to wait until the job completes.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$job = Import-LXCAOSImage -ImagePath c:\ESXi-5.1.1-1065491-RTM-Lenovo-20130607-GA.iso -AsJob
while($job.State -eq "Running")
{
    Start-Sleep -Seconds 5
}
$osimages = Get-LXCAOSImage

Disconnect-LXCA
```

- The following example imports a local OS image into the OS-images repository as a job that runs in the background and waits until the job completes.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$job = Import-LXCAOSImage -ImagePath c:\ESXi-5.1.1-1065491-RTM-Lenovo-20130607-GA.iso -AsJob -WaitForJob ToEnd

$osimages = Get-LXCAOSImage

Disconnect-LXCA
```

Install-LXCAOSImage

This cmdlet deploys an operating-system image to one or more target devices.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Install-LXCAOSImage [-Connection Connection]
                    -DeployTask DeployTask]
                    [-AsJob]
                    [CommonParameter]
```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-DeployTask *DeployTask[]*

Specifies one or more deployment tasks. The list is created from the [New-LXCADeployTask](#) cmdlet.

-AsJob

When specified, the cmdlet runs as background job and returns a Job object immediately.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

If the **-AsJob** parameter is specified, this cmdlet returns a Job object. The DeployResult objects can be retrieved from Job.Output.

If you *do not* specify the **-AsJob** parameter, this cmdlet returns one or more DeployResult objects.

Examples

- The following example installs VMware ESXi 5.5 on a target component in synchronous mode.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$nodes = Get-LXCADeployableServer
$images = Get-LXCAOSImage | where { $_.Name -like "*esxi5.5*" }
$deployTask = New-LXCADeployTask -ServerUid $nodes[0].Uid -ImageProfileID $images[0].Profiles[0].Id`
    -TargetDevice "SanDisk" -SanId "WWN123445" -IpAddress 9.5.12.122`
    -MacAddress $nodes[0].HostMacAddresses[0] -SubnetMask 255.255.255.0`
    -GateWay 9.5.12.1 -DNS1 9.10.244.100 -DNS2 9.10.244.200 -Mtu 1500
Install-LXCAOSImage -DeployTask $deployTask
```

```
Disconnect-LXCA
```

- The following example deploys multiple operating-system images to multiple target component as a job that runs in the background.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$nodes = Get-LXCADeployableServer
$esxi5 = Get-LXCAOSImage | where { $_.Name -like "*esxi5.5*" }
$rhel6 = Get-LXCAOSImage | where { $_.Name -like "*RHEL6*" }

#create deployment node list
$deployList = @()
$deployTask = New-LXCADeployTask -ServerUid $nodes[0].Uid -ImageProfileID $esxi5.Profiles[0].Id`
    -TargetDevice "SanDisk" -SanId "WWN123445" -IpAddress 9.5.12.122`
    -MacAddress $nodes[0].MacAddresses[0] -SubnetMask 255.255.255.0`
    -GateWay 9.5.12.1 -DNS1 9.10.244.100 -DNS2 9.10.244.200 -Mtu 1500

$deployList.Add($deployTask);
$deployTask = New-LXCADeployTask -ServerUid $nodes[1].Uid -ImageProfileID $rhel6.Profiles[0].Id`
    -TargetDevice "LocalDisk" -MacAddress $nodes[1].MacAddresses[0]`
    -IpAddress 9.5.12.123 -SubnetMask 255.255.255.0 -GateWay 9.5.12.1`
    -DNS1 9.10.244.100 -DNS2 9.10.244.200 -Mtu 1500

$deployList.Add($deployTask);
```

```

#deploy the node
$job = Install-LXCAOSImage -DeployTask $deployList -AsJob
while ($job.State eq "Running")
{
    Start-Sleep -Seconds 5
}
$job.Output

Disconnect-LXCA

```

New-LXCADeployTask

This cmdlet creates a task for operating-system deployment, including the target server, OS image profile, target storage device, and networking setting.

The DeployTask object that is returned by this cmdlet can be used by the [Install-LXCAOSImage](#) cmdlet.

Note: You cannot pipe objects to this cmdlet.

Syntax

```

New-LXCADeployTask [-Connection LXCACConnection]
                  -ServerUuid String
                  -ImageProfileID String
                  [-ConfigFileID String]
                  [-UnattendedFileID String]
                  -TargetDevice String
                  -MacAddress String
                  [-ADUserName String]
                  [-ADPassword SecureString]
                  [-DNS1 String]
                  [-DNS2 String]
                  [-GateWay String]
                  [-HostName String]
                  [-IpAddress String]
                  [-LicenseKey String]
                  [-Mtu String]
                  [-PrefixLength Int]
                  [-SanID String]
                  [-SubnetMask String]
                  [-VlanId String]
                  [-WindowsDomain String]
                  [-WindowsDomainBlob String]
                  [CommonParameter]

```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ServerUuid *String*

Specifies the UUID of the target server for operating-system deployment. The target server can be retrieved using the [Get-LXCADeployableServer](#) cmdlet.

-ImageProfileID *String*

Specifies the OS image profile ID. The profile ID can be retrieved using the [Get-LXCAOSImage](#) cmdlet.

-ConfigFileID *String*

ID of the custom configuration-settings file to use for this OS deployment.

This parameter is required if the OS image profile (that is specify by the **ImageProfileID** parameter) includes a custom configuration-settings file; otherwise, this parameter is ignored.

-UnattendedFileID *String*

ID of the unattended file to use for this OS deployment.

This parameter is required if OS image profile (that is specify by the **ImageProfileID** parameter) includes a custom unattended file; otherwise, this parameter is ignored.

-TargetDevice *String*

Specifies the location on target server where the OS image is to be deployed. You can specify one of the following values:

- **LocalDisk**. Uses the local disk drive.
- **EmbeddedHypervisor**. Uses embedded hypervisor (USB or SD media adapter) when ESXi is selected.

Note: For XClarity Administrator v1.0, use **usbdisk** instead of **EmbeddedHypervisor** for a USB Embedded Hypervisor when ESXi is selected.

- **M.2drive**. Uses first enumerated M.2 drive in the managed server.
- **SanDisk**. Uses SAN volumes if available. If specified, you must also specify the **-SanID** parameter.

-MacAddress *String*

Specifies the MAC address of the host server to which the IP address is to be bound.

For XClarity Administrator v1.2.0 and later, you can also specify **AUTO** to automatically detect the first Ethernet port that can be configured and used for deployment automatically.

-ADUserName *String*

Specifies the Active Directory user name that is required to join a domain.

-ADPassword *SecureString*

Specifies the Active Directory password that is required to join a domain.

-LicenseKey *String*

Specifies the license key that is needed for Windows and ESXi operating-system installation.

-DNS1 *String*

Specifies the first DNS server, in IPv4 format, on the network to be used after operating-system deployment. This is required if a static IP address was chosen when the global settings were defined.

-DNS2 *String*

Specifies the secondary DNS server, in IPv4 format, on the network to be used after operating system deployment. This is required if a static IP address was chosen when the global settings were defined.

-GateWay *String*

Specifies the gateway, in IPv4 format, of the network to be used after operating system deployment. This is required if a static IP address was chosen when the global settings were defined.

-HostName *String*

Specifies the hostname for the host server. If a hostname is not specified, a default hostname is assigned.

-IpAddress *String*

Specifies the IP address of the host server, in IPv4 format, to be used after operating-system deployment. This is required if a static IP address was chosen when the global settings were defined.

-Mtu *Int*

Specifies the maximum transmission unit (MTU) of the network to be used after operating system deployment. The default value is 1500. This is required if a static IP address was chosen when the global settings were defined.

-PrefixLength *Int*

Specifies the IPv6 prefix length to be used after operating-system deployment. This is required if a static IPv6 was chosen when the global settings were defined.

-SanID String

Specifies the SAN ID. This is required if the target device is a SAN disk. The SAN ID is a string starting with "lunpluswwn=", for example, lunpluswwn=2@50:05:07:68:05:0c:09:bb, where 2 is the LUN and 50:05:07:68:05:0c:09:bb is the WWN. Ensure that you use @ to join the two strings.

-SubnetMask String

Specifies the subnet mask, in IPv4 format, of the network to be used after operating-system deployment. This is required if a static IP address was chosen when the global settings were defined.

-VlanId String

Specifies the VLAN ID for operating-system VLAN tagging.

Important: This parameter requires that the VLAN mode setting is enabled in the global settings for operating-system deployment. To change the global settings, use the [Set-LXCADeployGlobalSetting](#) cmdlet.

-WindowsDomain String

(Windows only) Specifies the domain that the operating system is to join after it is installed successfully. If there is an organizational unit (OU), the domain must be a string like "domain/ou".

If the operating system will not join a domain, you can set this field to null.

Notes:

- Use the **ADUserName** and **ADPassword** parameters to specify the user name and password for the domain.
- To join an Active Directory domain, you must specify either the **WindowsDomain** or **WindowsDomainBlob** parameter. If both are specified, the **WindowsDomainBlob** parameter is used.

-WindowsDomainBlob String

(Windows only) Specifies the Active Directory Computer Account Metadata (in Base-64 encoded blob format) that can be used to join the Active Directory domain. For instructions for generating a file that contains the metadata blob data, see [Integrating with Windows Active Directory](#) in the Lenovo XClarity Administrator online documentation.

Notes:

- To join an Active Directory domain, you must specify either the **WindowsDomain** or **WindowsDomainBlob** parameter. If both are specified, the **WindowsDomainBlob** parameter is used.
- You can use metadata blob data when deploying any Windows operating system. However, this method must be used for Windows Nano Server. Specifying a domain in global settings is not supported for Windows Nano Server.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters](#) webpage.

Results

This cmdlet returns a DeployTask object that contains all information for operating-system deployment. The task can be used by [Install-LXCAOSImage](#) cmdlet.

Examples

The following example creates a deployment task and deploys an operating-system image.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$nodes = Get-LXCADeployableServer
$images = Get-LXCAOSImage | where {$_.Name -contains "esxi5.5"}
$deployTask = New-LXCADeployTask -ServerUuid $nodes[0].Uuid -ImageProfileID $images[0].Profiles[0].Id`
    -TargetDevice "SanDisk" -SanId "WWN123445" -IpAddress 9.5.12.122`
    -HostName "Server1" -MacAddress $nodes[0].MacAddresses[0]`
    -SubnetMask 255.255.255.0 -GateWay 9.5.12.1 -DNS1 9.10.244.100`
    -DNS2 9.10.244.200 -Mtu 1500
Install-LXCAOSImage -DeployTask $deploytask

Disconnect-LXCA
```

Remove-LXCAOSImage

This cmdlet removes an operating-system image and its image profiles from the OS-images repository.

Note: You can pipe OSImage objects to this cmdlet.

Syntax

```
Remove-LXCAOSImage [-Connection LXCACConnection]
                  -ImageID String
                  [-Force]
                  [CommonParameter]
```

```
Remove-LXCAOSImage [-Connection LXCACConnection]
                  -OSImage OSImage
                  [-Force]
                  [CommonParameter]
```

```
Remove-LXCAOSImage [-Connection LXCACConnection]
                  -ProfileID String
                  [-Force]
                  [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ImageID *String*

Specifies the OS image ID. The cmdlet removes the OS image and its image profiles.

-OSImage *OSImage*

Specifies the operating-system image. The cmdlet removes the image and its image profiles.

Tip: To obtain the OSImage object, use the [Get-LXCAOSImage](#) cmdlet.

-ProfileID *String*

Specifies the image profile ID. The cmdlet removes the image profile.

-Force

When specified, the cmdlet removes the image or imageprofile without prompting for confirmation about the removal.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlets removes the image or image profile (or both) from the OS-images repository. No objects are returned.

Examples

The following example removes an OS image from the OS image repository.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$images = Get-LXCAOSImage
Remove-LXCAOSImage -OSImage $images[0]
```

```
Disconnect-LXCA
```

Set-LXCADeployGlobalSetting

This cmdlet modifies the global settings that are used for operating-system deployment, such as IP assignment type and deployment credentials.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCADeployGlobalSetting [-Connection LXCACONNECTION]
                             -DeployGlobalSetting DeployGlobalSetting
                             [CommonParameter]
```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-DeployGlobalSetting *DeployGlobalSetting*[]

Specifies the global deployment settings, which include IP assignment type and deployment credentials.

Use the `Get-LXCADeployGlobalSetting` cmdlet to retrieve a list of the current settings.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet modifies the global settings for operating-system deployment. No objects are returned.

Examples

The following example retrieves and modifies global deployment settings.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$setting = Get-LXCADeployGlobalSetting
$setting.IpAssignment = "dhcpv4"
Set-LXCADeployGlobalSetting -DeployGlobalSetting $setting

Disconnect-LXCA
```

Event, alert, and audit log cmdlets

The following PowerShell cmdlets are available for performing monitoring and event functions, such as retrieving logs, retrieving event and alert status, and setting up event monitors.

Add-LXCAEventExclusionFilter

This cmdlet adds an event exclusion filter to exclude an event or active alert from a target device.

The [Get-LXCAEvent](#) cmdlet does not return excluded events. Use the [Get-LXCAEventExclusionFilter](#) cmdlet to get the exclusion filter. Use the [Add-LXCAEventExclusionFilter](#) and [Remove-LXCAEventExclusionFilter](#) cmdlets to add and remove the exclusion filter.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCAEventExclusionFilter [-Connection LXCACConnection]
                             -Name String
                             -Description String
                             -SourceID String
                             -Type String
                             [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Name *String*

Specifies the name of the exclusion filter. The name must be unique.

-Description *String*

Specifies a description for the exclusion filter.

-SourceID *String*

Specifies the UUID of the target device.

-Type *String*

Specifies the event type. You can specify one of the following event types:

- **event**
- **activeAlert**

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns an EventExclusionFilter object.

Examples

The following example excludes all events from a device.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Add-LXCAEventExclusionFilter -SourceID $node -Name "ExclusionFilterOnNode" -Type event `
    -Description "Suppress events from node"

Disconnect-LXCA
```

Backup-LXCAEventLog

This cmdlet retrieves the event log in CSV format and appends the log to a local file. You can use this cmdlet to back up the event log periodically to avoid event log loss.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAEvent [-Connection LXCAConnection]
              -Type String
              -StartSequenceNumber Long
              -Count Int
              -FileName String
              [-NoHead]
              [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Type *String*

Specifies the event type. You can specify one of the following event types:

- All
- AuditLog
- EventLog

-StartSequenceNumber *Long*

Specifies the starting event sequence number.

-Count *Int*

Specifies the number of events to be returned from the starting sequence number.

-FileName *String*

Specifies the file name to which the events are to be appended.

-NoHead

If specified, the CSV header is not saved to the local file.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a BackupEventResult object.

Note: The BackupEventResult object has a property called **NextSequenceNumber** that indicates the starting sequence number that can be used for the next backup.

Examples

The following example backs up the event log to a local file every hour.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$backupResult = Backup-LXCAEventLog -Type All -StartSequenceNumber 0 -Count 1000 -FileName c:\LXCALog.csv
While($true)
{
    Start-Sleep -Seconds (60*60*1)
    $backupResult = Backup-LXCAEventLog -Type All -StartSequenceNumber $backupResult.NextSequenceNumber`
        -Count 1000 -FileName c:\LXCALog.csv -NoHead
}

Disconnect-LXCA
```

Get-LXCAActiveAlert

This cmdlet retrieves all active alerts from the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAActiveAlert [-Connection LXCACConnection]
                    [-ComponentID String]
                    [CommonParameter]

Get-LXCAActiveAlert [-Connection LXCACConnection]
                    [-Severity EventSeverity]
                    [-EventClass EventClass]
                    [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Component ID *String*

Specifies one or more UUIDs of the components to which alerts apply.

-Severity *EventSeverity*

Specifies the alert severity. You can specify one or more of the following severities, separated by a comma.

- **Unknown**
- **Informational**
- **Warning**
- **Minor**
- **Major**
- **Critical**
- **Fatal**

Note: The severities that are listed through this cmdlet differ from those listed in the Lenovo XClarity Administrator web interface.

Lenovo XClarity Administrator web interface severity	Severity parameter
Critical	Fatal, Critical, Major
Warning	Minor, Warning
Informational	Informational
Unknown	Unknown

-EventClass *EventClass[]*

Specifies the alert source type. You can specify one or more of the following event classes, separated by a comma.

- **ADAPTOR** (adapter card)
- **AUDIT**
- **BLADE** (Flex System server)
- **COOLING**
- **DISKS** (storage)
- **EXPANSION** (expansion card)
- **IOMODULE** (Flex System switch)
- **MEMORY**
- **OTHER**
- **POWER**
- **PROCESSOR**
- **SWITCH** (RackSwitch switch)
- **SYSTEM** (rack or tower server)
- **TEST**
- **UNKNOWN**

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more ActiveAlert objects.

Examples

- The following example retrieves all active alerts from the XClarity Administrator server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$alerts = Get-LXCAActiveAlert

Disconnect-LXCA
```
- The following example retrieves a list of all critical and fatal active alerts for a Flex System server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$alerts = Get-LXCAActiveAlert -Severity Critical,Fatal -EventClass Blade

Disconnect-LXCA
|
```

Get-LXCAAuditLog

This cmdlet retrieves the audit log from the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAAuditLog [-Connection LXCACConnection]  
                [-EventID String[]]  
                [-SourceID String[]]  
                [-Severity EventSeverity[]]  
                [-PreviousHours Int]  
                [-StartSequenceNumber Long]  
                [-EndSequenceNumber Long]  
                [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-EventID *String[]*

Specifies one or more audit event IDs. If you do not specify an event ID, all events are returned.

Note: You can use the [Get-LXCAAuditLog](#) cmdlet to retrieve a list of audit event IDs.

-SourceID *String[]*

Specifies the UUID of the component that is the event source (raiser or owner).

-Severity *EventSeverity[]*

Specifies the event severity. You can specify one or more of the following severities, separated by a comma.

- **Unknown**
- **Informational**
- **Warning**
- **Minor**
- **Major**
- **Critical**
- **Fatal**

Note: The severities that are listed through this cmdlet differ from those listed in the Lenovo XClarity Administrator web interface.

Lenovo XClarity Administrator web interface severity	Severity parameter
Critical	Fatal, Critical, Major
Warning	Minor, Warning
Informational	Informational
Unknown	Unknown

-PreviousHours *Int*

Specifies the number of hours before the current time that is to be used for obtaining events.

-StartSequenceNumber *Long*

Specifies the start sequence number for obtaining events.

-EndSequenceNumber *Long*

Specifies the end sequence number for obtaining events.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more AuditLog objects

Examples

- The following example retrieves the entire audit log.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$auditlog = Get-LXCAAuditLog
```

```
Disconnect-LXCA
```

- The following example retrieves the critical and fatal audit log entries that have occurred in the last 12 hours.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$auditlog = Get-LXCAAuditLog -Severity Critical,Fatal, -PreviousHours 12
```

```
Disconnect-LXCA
```

Get-LXCAEvent

This cmdlet retrieves one or more management server and hardware events from the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAEvent [-Connection LXCAConnection]
              [-EventID String]
              [-SourceID String]
              [-Severity EventSeverity]
              [-EventClass EventClass]
              [-PreviousHours Int]
              [-StartSequenceNumber Long]
              [-EndSequenceNumber Long]
              [-All]
              [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-EventID *String*

Specifies one or more event IDs. If you do not specify an event ID, all events are returned.

Note: You can use the Get-LXCAEvent cmdlet to retrieve a list of event IDs.

-SourceID *String[]*

Specifies one or more UUIDs of the components that are the event source (raiser or owner).

-Severity *EventSeverity[]*

Specifies one or more event severities. You can specify one or more of the following severities, separated by a comma.

- **Unknown**
- **Informational**
- **Warning**
- **Minor**
- **Major**
- **Critical**
- **Fatal**

Note: The severities that are listed through this cmdlet differ from those listed in the Lenovo XClarity Administrator web interface.

Lenovo XClarity Administrator web interface severity	Severity parameter
Critical	Fatal, Critical, Major
Warning	Minor, Warning
Informational	Informational
Unknown	Unknown

-EventClass *EventClass[]*

Specifies the event source type. You can specify one or more of the following types, separated by a comma.

- **ADAPTOR** (adapter card)
- **AUDIT**
- **BLADE** (Flex System server)
- **COOLING**
- **DISKS** (storage)
- **EXPANSION** (expansion card)
- **IOMODULE** (Flex System switch)
- **MEMORY**
- **OTHER**
- **POWER**
- **PROCESSOR**
- **SWITCH** (RackSwitch switch)
- **SYSTEM** (rack or tower server)
- **TEST**
- **UNKNOWN**

-PreviousHours *Int*

Specifies the number hours before the current time in which events are to be obtained.

-StartSequenceNumber *Long*

Specifies the start sequence number for obtaining events.

-EndSequenceNumber *Long*

Specifies the end sequence number for obtaining events.

-All

If specified, the cmdlets also returns the event log entries.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more Event objects.

Examples

- The following example retrieves a list of all events with a severity of Critical.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$events = Get-LXCAEvent -Severity Critical
```

```
Disconnect-LXCA
```

- The following example retrieves a list of all of the Major, Critical, and Fatal events that have occurred for a server in the previous 12 hours.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$events = Get-LXCAEvent -Severity Fatal,Critical -EventClass Blade -PreviousHours 12
```

```
Disconnect-LXCA
```

- The following example retrieves a list of all of the Major, Critical, and Fatal events that are within the sequence number range from 100 to 1000.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$events = Get-LXCAEvent -Severity Major,Critical,Fatal -StartSequenceNumber 100 -EndSequenceNumber 1000
```

```
Disconnect-LXCA
```

Get-LXCAEventExclusionFilter

This cmdlet retrieves information about one or more event exclusion filters.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAEventExclusionFilter [-Connection LXCAConnection]
                             [-FilterID String]
                             [CommonParameter]
```

```
Get-LXCAEventExclusionFilter [-Connection LXCAConnection]
                             [-SourceID String]
                             [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-FilterID *String*

Specifies the ID of the exclusion filter. If no ID is specified, the cmdlet returns all exclusion filters.

-SourceID *String*

Specifies the UUID of the target device. If no UUID is specified, the cmdlet returns all exclusion filters.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more EventExclusionFilter objects.

Examples

The following example retrieves all event exclusion filters in place for a specified server:

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Get-LXCAEventExclusionFilter -SourceID AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Disconnect-LXCA
```

Remove-LXCAEventExclusionFilter

This cmdlet deletes an event exclusion filter.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Remove-LXCAEventExclusionFilter [-Connection LXCAConnection]
                               -FilterID String
                               [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-FilterID *String*

Specifies the ID of the exclusion filter to be deleted.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet deletes the exclusion filter. No object is returned.

Examples

The following example deletes the specified event exclusion filter from the XClarity Administrator server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Remove-LXCAEventExclusionFilter -FilterID "AUG:XFT:1"
```

Start-LXCAEventMonitor

This cmdlet creates an event monitor object to which you can subscribe by using the `Register-ObjectEvent` cmdlet.

You can use this cmdlet to monitor specific Lenovo XClarity Administrator events and trigger a defined script to handle the incoming event. It reconnects to XClarity Administrator when the network is not available or the management server is rebooted, and to continue monitoring events.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Start-LXCAEventMonitor [-Connection LXCAConnection]
                        [-EventID String[]]
                        [-SourceID String[]]
                        [-Severity EventSeverity[]]
                        [-EventClass EventClass[]]
                        [-MessageRegex String]
                        [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-EventID *String[]*

Specifies the event IDs to be monitored.

-SourceID *String[]*

Specifies the UUID of the component that is the event source (raiser or owner).

-Severity *EventSeverity[]*

Specifies the severity of the event to be monitored. You can specify one or more of the following severities, separated by a comma.

- **Unknown**
- **Informational**
- **Warning**
- **Minor**
- **Major**
- **Critical**
- **Fatal**

Note: The severities that are listed through this cmdlet differ from those listed in the Lenovo XClarity Administrator web interface.

Lenovo XClarity Administrator web interface severity	Severity parameter
Critical	Fatal, Critical, Major
Warning	Minor, Warning
Informational	Informational
Unknown	Unknown

-EventClass *EventClass[]*

Specifies the event source type. You can specify one or more of the following types, separated by a comma.

- **ADAPTOR** (adapter card)
- **AUDIT**
- **BLADE** (Flex System server)
- **COOLING**
- **DISKS** (storage)
- **EXPANSION** (expansion card)
- **IOMODULE** (Flex System switch)
- **MEMORY**
- **OTHER**
- **POWER**
- **PROCESSOR**
- **SWITCH** (RackSwitch switch)
- **SYSTEM** (rack or tower server)
- **TEST**
- **UNKNOWN**

-MessageRegex *String*

Specifies the event message regex expression.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns an `LXCAEventMonitor` object.

Examples

The following example creates an event monitor and the uses `Register-ObjectEvent` to print out the event ID and message when a monitored event is received.

Notes: In the `Register-ObjectEvent` cmdlet:

- **-EventName** must be `LXCAEvent`.
- **-SourceIdentifier** must be a unique string.
- **-Action** is the script that handles the incoming event.

In the script, `$EventArgs.Events` is for all incoming events at this moment. Each event has the same properties as the event returned by `Get-LXCAEvent` cmdlet.

To stop the event monitoring, invoke `$monitor.StopMonitor()`.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

$monitor = Start-LXCAEventMonitor -EventClass Audit,Power -Severity Critical,Fatal
Register-ObjectEvent -InputObject $monitor -EventName LXCAEvent -SourceIdentifier MonitorAUDITPOWER`
    -Action { Write-Host $EventArgs.Events[0].EventID -ForegroundColor Blue`
    Write-Host $EventArgs.Events[0].Message -ForegroundColor Blue }
... ..
$monitor.StopMonitor()

Disconnect-LXCA
```

Job cmdlets

The following PowerShell cmdlets are available for performing task and job management functions.

Add-LXCAJobNote

This cmdlet creates a note and change the state of a specific job (task).

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCAJobNote [-Connection LXCAConnection]  
                -JobUID String  
                -NewState String  
                -Text String  
                [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-JobUID *String*

Specifies the UID of the job.

If the job does not exist, the 404 - Not Found exception is thrown.

-NewState *String*

Specifies the new state of the job. You can specify one of the following values.

- **Aborted**
- **Investigating**
- **Resolved**
- **NoChange**

-Text *String*

Specifies the note text.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return data.

Examples

The following example add a net to a specific job, keeping the actual job state as is.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
$jobUID = 123444  
Add-LXCAJobNote -JobUID $jobUID -NewState NoChange -Text "No Change. Waiting for answer."  
  
Disconnect-LXCA
```

Get-LXCAJobNotes

This cmdlet retrieves the notes history for a specific job (task).

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAJobNotes [-Connection LXCACConnection]  
                -JobUID String  
                [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-JobUID *String*

Specifies the UID of the job.

If the job does not exist, the 404 - Not Found exception is thrown.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- [Lenovo.SysMgmt.LXCA.Integration.Data.JobNotes\[\]](#)

This cmdlet returns a `Lenovo.SysMgmt.LXCA.Integration.Data.JobNotes` array that contains information about all the notes for a specific job.

Examples

The following example retrieves all notes from a specific job

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$jobUID = 123444  
$(Get-LXCAJobNotes -JobUID $jobUID)
```

```
Disconnect-LXCA
```

Get-LXCASchedule

This cmdlet retrieves information about one or more job schedules.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCASchedule [-Connection LXCACConnection]  
                [-ScheduleName String]  
                [CommonParameter]
```

```
Get-LXCASchedule [-Connection LXCACConnection]  
                [-ScheduleType String]
```

[CommonParameter]

Parameters

-Connection *LXCACredential*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ScheduleName *String*

Specifies the name of the job schedule. If a name is not specified, the cmdlet returns information about all job schedules.

-ScheduleType *String*

Specifies the type of job schedule. If type is specified, the cmdlet returns information about all job schedule of the specified type. If a type is not specified, the cmdlet returns information about all job schedules.

You can specify one of the following values.

- **One Time.** Returns information about all job schedule that run one time (immediately or at a later time).
- **Recurring.** Returns information about all job schedule that run on a recurring basis.
- **Triggered.** Returns information about all job schedule that run are triggered when a specific event occurs.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more ScheduleObject objects

Examples

- The following example retrieves information about all job schedules.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$schedules = Get-LXCASchedule

Disconnect-LXCA
```
- The following example retrieves information about a specific job schedule.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Get-LXCASchedule -ScheduleName $name

Disconnect-LXCA
```
- The following example retrieves information about all reoccurring job schedule.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Get-LXCASchedule -ScheduleType Recurring

Disconnect-LXCA
```


Get-LXCATask

This cmdlet retrieves information about one or more tasks (jobs) from the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCATask [-Connection LXCACConnection]  
            [CommonParameter]
```

```
Get-LXCATask [-Connection LXCACConnection]  
            -TaskUIDs String[]  
            [CommonParameter]
```

```
Get-LXCATask [-Connection LXCACConnection]  
            [-TargetUIDs String[]]  
            [-Category String[]]  
            [-State String[]]  
            [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-TaskUIDs *String[]*

Specifies one or more task IDs.

-TargetUIDs *String[]*

Specifies one or more UUIDs of the components to which the task belongs.

-Category *String[]*

Specifies the categories to which the task belongs. You can specify one or more of the following values, separated by a comma.

- **Backup**
- **Configuration**
- **Custom**
- **Firmware**
- **Health**
- **Inventory**
- **Management**
- **OsDeployment**
- **OsDriverUpdates**
- **OsImport**
- **OsProfileExport**
- **Power**
- **RemoteAccess**
- **SelfMaintenance**
- **Service**
- **SwitchConfiguration**
- **SystemID**
- **Unknown**

-State *String[]*

Specifies the current states of the task. You can specify one or more of the following values, separated by a comma.

- **Aborted**

- **Blocked**
- **Cancelled**
- **CancelledWithError**
- **CancelledWithWarning**
- **Cancelling**
- **Complete**
- **CompleteWithError**
- **CompleteWithWarning**
- **Expired**
- **Initializing**
- **Interrupted**
- **InterruptedWithError**
- **InterruptedWithWarning**
- **Investigating**
- **Pending**
- **Resolved**
- **Running**
- **RunningWithError**
- **RunningWithWarning**
- **Skipped**
- **Stopped**
- **StoppedWithError**
- **StoppedWithWarning**
- **Unknown**
- **Uploading**
- **Validating**
- **Waiting**

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- `Lenovo.SysMgmt.LXCA.Integration.Data.LxcaJob[]`

This cmdlet returns `Lenovo.SysMgmt.LXCA.Integration.Data.LxcaJob` array that contains information about all the jobs in XClarity Administrator, all jobs with specified UIDs, or jobs that are filtered by target UUID, state and category.

Examples

The following example retrieves all tasks that are currently running.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$tasks = Get-LXCATask -State Running,RunningWithError,RunningWithWarning

Disconnect-LXCA
```

New-LXCAScheduleObj

This cmdlet creates a local schedule object that can be used with certain cmdlets to create a scheduled job on Lenovo XClarity Administrator.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
New-LXCAScheduleObj [-Connection LXCAConnection]
                   -ScheduleName String
                   -OccurrenceDate DateTime
                   [CommonParameter]
New-LXCAScheduleObj [-Connection LXCAConnection]
                   -ScheduleName String
                   -Rule ScheduleRule
                   [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ScheduleName *String*

Specifies schedule name.

-OccurrenceDate *DateTime*

Specifies the date and time when task is to run.

-Rule *ScheduleRule*

Specifies the rule for recurring schedule.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- **Lenovo.SysMgmt.LXCA.Integration.Data.ScheduleObject** This cmdlet returns a [ScheduleObject](#) that can be used with certain cmdlets to create a scheduled job on XClarity Administrator.

Examples

The following example creates a schedule to update the management server on specific date and time.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

$scheduleName = "Update-Package-Install"
$chedule = New-LXCAScheduleObj -ScheduleName $scheduleName `
                -OccurrenceDate '2022-2-22 02:00:00'

# Use the Schedule object in the Install-LXCAUpdateManagementServer cmdlet.
$fixId = lnvgu_utl_lxca-core_2.4.0-2019xxxx_aynos_noarch
Install-LXCAUpdateManagementServer -FixId $fixId -Schedule $chedule

Disconnect-LXCA
```

The following example creates a schedule to update the management server tomorrow, using the [Get-Date](#) cmdlet.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

$cheduleName = "Update-Package-Install"
$chedule = New-LXCAScheduleObj -ScheduleName $name -OccurrenceDate (Get-Date).AddDays(1)
```

```
# Use the Schedule object in the Install-LXCAUpdateManagementServer cmdlet.
$fixId = lnvgg_utl_lxca-core_2.4.0-2019xxxx_angos_noarch
Install-LXCAUpdateManagementServer -FixId $fixId -Schedule $schedule
```

```
Disconnect-LXCA
```

The following example creates a recurring schedule to restart a specific device on the third Wednesday of the month over a 2-year period, starting February 22, 2022.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

# Create a rule for the schedule.
$monthlyRule = New-LXCAScheduleRule -DayOfTheWeek Wednesday `
    -WeekOfTheMonth ThirdWeek `
    -RecurEvery 3 `
    -StartTimeStamp "2022-2-22 02:00:00" `
    -EndDate "2024-2-22"

$scheduleName = "server-restart"
$recurringSchedule = New-LXCAScheduleObj -ScheduleName $scheduleName -Rule $monthlyRule

# Use the Schedule object in Invoke-LXCASystemAction cmdlet to restart the server.
Invoke-LXCASystemAction -DeviceUuid AAAAAAAAAAAAAA `
    -ServerAction ShutdownOSandRestart `
    -Schedule $recurringSchedule `
    -Force
```

```
Disconnect-LXCA
```

New-LXCAScheduleRule

This cmdlet creates a rule for a recurring schedule object that is to be created using **New-LXCAScheduleObj** cmdlet.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
New-LXCAScheduleRule [-Connection LXCACConnection]
    -StartTimeStamp DateTime
    -EndDate Nullable[DateTime]
    -Days String[]
    [CommonParameter]
```

```
New-LXCAScheduleRule [-Connection LXCACConnection]
    -StartTimeStamp DateTime
    -EndAfter String
    -Days String[]
    [CommonParameter]
```

```
New-LXCAScheduleRule [-Connection LXCACConnection]
    -StartTimeStamp DateTime
    -EndDate Nullable[DateTime]
    [-RecurEvery Int32]
    -DayOfTheWeek String
    [CommonParameter]
```

```
New-LXCAScheduleRule [-Connection LXCACConnection]
    -StartTimeStamp DateTime
    -EndAfter String
```

```
[-RecurEvery Int32]  
-DayOfTheWeek String  
[CommonParameter]
```

```
New-LXCAScheduleRule [-Connection LXCACConnection]  
-StartTimeStamp DateTime  
-EndDate Nullable[DateTime]  
[-RecurEvery Int32]  
-DayOfTheWeek String  
-WeekOfTheMonth String  
[CommonParameter]
```

```
New-LXCAScheduleRule [-Connection LXCACConnection]  
-StartTimeStamp DateTime  
-EndAfter String  
[-RecurEvery Int32]  
-DayOfTheWeek String  
-WeekOfTheMonth String  
[CommonParameter]
```

```
New-LXCAScheduleRule [-Connection LXCACConnection]  
-StartTimeStamp DateTime  
-EndDate Nullable[DateTime]  
[-RecurEvery Int32]  
-DayOfTheWeek String  
-WeekOfTheMonth String  
-MonthOfTheYear String  
[CommonParameter]
```

```
New-LXCAScheduleRule [-Connection LXCACConnection]  
-StartTimeStamp DateTime  
-EndAfter String  
[-RecurEvery Int32]  
-DayOfTheWeek String  
-WeekOfTheMonth String  
-MonthOfTheYear String  
[CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-StartTimeStamp *DateTime*

Specifies the date and time when the schedule is to be triggered for the first time.

-EndDate *Nullable[DateTime]*

Specifies the date when the schedule is to end.

If set to \$null, the schedule is permanently Active until the schedule is manually ended.

-Days *String[]*

Specifies one or more days when scheduled job runs. This can be one of the following values

- **Monday**
- **Tuesday**
- **Wednesday**
- **Thursday**
- **Friday**
- **Saturday**
- **Sunday**

-EndAfter *String*

Specifies the number of times that the scheduled job runs before the schedule ends.

-RecurEvery *Int32*

Specifies the number of weeks, months, or years that the scheduled job runs.

-DayOfTheWeek *String*

Specifies the day of the week when the scheduled job runs. This can be one of the following value.

- **Monday**
- **Tuesday**
- **Wednesday**
- **Thursday**
- **Friday**
- **Saturday**
- **Sunday**

-WeekOfTheMonth *String*

Specifies the week of the month when the scheduled job runs. This can be one of the following values.

- **FirstWeek**
- **SecondWeek**
- **ThirdWeek**
- **ForthWeek**
- **FifthWeek**

-MonthOfTheYear *String*

Specifies the month of the year when the scheduled job runs. This can be one of the following values.

- **January**
- **February**
- **March**
- **April**
- **May**
- **June**
- **July**
- **August**
- **September**
- **October**
- **November**
- **December**

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- **[Lenovo.SysMgmt.LXCA.Integration.Data.ScheduleRule](#)**

This cmdlet returns a [ScheduleRule](#) object with data about the recurring rule to be used in a [LXCASchedule](#) object.

Examples

The following example creates a daily rule that runs on Mondays and Fridays for 22 occurrences.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck
```

```
$dailyEndAfter = New-LXCAScheduleRule -Days Monday, Friday `
    -EndAfter 22 `
```

```
-StartTimeStamp "2022-2-22 02:00:00"
```

```
Disconnect-LXCA
```

The following example creates a daily rule that is active until the schedule is manually ended. The **-EndDate** property is assigned with \$null value.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

$dailyNoEnd = New-LXCAScheduleRule -Days Friday `
    -StartTimeStamp "2022-2-22 02:00:00" `
    -EndDate $null
```

```
Disconnect-LXCA
```

The following example creates a monthly rule that occurs every three months, on the Wednesday of third week of that month, until the **-EndDate** is passed.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

$monthly = New-LXCAScheduleRule -DayOfTheWeek Wednesday `
    -WeekOfTheMonth ThirdWeek `
    -RecurEvery 3 `
    -StartTimeStamp "2022-2-22 02:00:00" `
    -EndDate "2024-2-22 02:00:00"
```

```
Disconnect-LXCA
```

Remove-LXCAJobNotes

This cmdlet removes all notes from a specific job (task).

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Remove-LXCAJobNotes [-Connection LXCACconnection]
                    -JobUID String
                    [CommonParameter]
```

Parameters

-Connection *LXCACconnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-JobUID *String*

Specifies the UID of the job.

If specified job does not exist, the 404 - Not Found exception is thrown.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return data.

Examples

The following example retrieves information about all job schedules

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$jobUID = 123444
Remove-LXCAJobNotes -JobUID $jobUID
```

```
Disconnect-LXCA
```

Remove-LXCATask

This cmdlet deletes one or more tasks (jobs) from the Lenovo XClarity Administrator server. Each task must be in a stopped, cancelled, or complete state before it can be removed.

Not all tasks can be deleted. This cmdlet deletes only tasks for which the **isDeletable** attribute is set to true. Task IDs are returned for only the tasks that were deleted.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Remove-LXCATask [-Connection LXCACConnection]
                -TaskUIDs String[]
                [CommonParameter]
```

```
Remove-LXCATask [-Connection LXCACConnection]
                -Tasks LxcaJob[]
                [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-TaskUIDs *String[]*

Specifies one or more task IDs.

-Tasks *LxcaJob[]*

Specifies one or more task instances.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- System.String[]

This cmdlet removes one or more tasks from the XClarity Administrator server and returns the task UIDs of the deleted jobs as an array of strings.

If none of the specified tasks can be deleted, the returned object is an empty array.

Examples

The following example deletes all tasks that are in Cancelled or Complete State on the XClarity Administrator server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$tasks = Get-LXCATask -State Cancelled,Complete
$(Remove-LXCATask -Task $tasks)

Disconnect-LXCA
```

Remove-LXCASchedule

This cmdlet deletes one or more job schedules.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Remove-LXCASchedule [-Connection LXCACONNECTION]
                   [-ScheduleName String[]]
                   [CommonParameter]
```

```
Remove-LXCASchedule [-Connection LXCACONNECTION]
                   [-Schedules ScheduleObject[]]
                   [CommonParameter]
```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ScheduleName *String[]*

Specifies the names of one or more job schedules to be deleted.

-Schedules *ScheduleObject[]*

Specifies the schedule object of one or more job schedules to be deleted.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

- The following example deletes the first job schedule in the list of returned job schedules.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$schedules = Get-LXCASchedule
Remove-LXCASchedule -Schedules $schedules[0]

Disconnect-LXCA
```

- The following example deletes a specific job schedule.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Remove-LXCASchedule -ScheduleName $name

Disconnect-LXCA
```

Stop-LXCATask

This cmdlet cancels one or more running or pending tasks (jobs) on the Lenovo XClarity Administrator server.

Not all tasks can be stopped. This cmdlet stop only tasks for which the **isStoppable** attribute is set to true. The task ID is returned for only the tasks that were stopped.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Stop-LXCATask [-Connection LXCACConnection]
              -TaskUIDs String[]
              [CommonParameter]

Stop-LXCATask [-Connection LXCACConnection]
              -Tasks LxcaJob[]
              [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-TaskUIDs *String[]*

Specifies one or more task IDs.

-Tasks *LxcaJob[]*

Specifies one or more task instances.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- System.String[]
- This cmdlet stops one or more tasks from the XClarity Administrator server and returns the task IDs of the stopped jobs as an array of strings. If none of the specified tasks can be stopped, the returned object is an empty array.

Examples

The following example cancels all running and pending tasks from the XClarity Administrator server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$tasks = Get-LXCATask -State Running,Pending
$(Stop-LXCATask -Tasks $tasks)
```

Security cmdlets

The following PowerShell cmdlets are available for performing security functions.

Add-LXCARoleGroup

This cmdlet creates a role group.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCARoleGroup [-Connection LXCAConnection]
                  [-Description String]
                  -RoleGroupName String
                  -RoleList String[]
                  [-UserList String[]]
                  [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Description *String*

Specifies the description for the new role group.

-RoleGroupName *String*

Specifies the name of the role group to be created. The name must be unique.

-RoleList *String[]*

One or more roles to be associated with the new role group. This can be one of the following values.

- **lxc-supervisor.** Users that are assigned this role can access, configure, and perform all available operations on the management server and all managed devices. Users that are assigned this role always have access to all managed devices. You cannot restrict access to devices for this role.
- **lxc-admin.** Users that are assigned this role can modify non-security related settings and perform all non-security related operations on the management server, including the ability to update and restart the management server. This role also provides the ability to view all configuration and status information about the management server and managed devices.
- **lxc-security-admin.** Users that are assigned this role can modify security settings and perform security-related operations on the management server and managed devices. This role also provides ability to view all configuration and status information about the management server and managed devices.

Users that are assigned this role always have access to all managed devices. You cannot restrict access to devices for this role.

- **lxc-hw-admin.** Users that are assigned this role can modify non-security settings and perform non-security related operations on managed devices, including the ability to update and restart managed devices. This role also provides the ability to view all configuration and status information about the management server and all managed devices.
- **lxc-fw-admin.** Users that are assigned this role can create firmware policies and deploy those policies to managed devices. Users that are not assigned this role can only view policy information.
- **lxc-os-admin.** Users that are assigned this role can download and deploy operating systems and device-driver updates to managed servers. Users that are not assigned this role can only view operating-system and device-driver information.

- **lxc-service-admin.** Users that are assigned this role can collect and download service files for XClarity Administrator and managed devices. Users that are not assigned this role can collect but not download service data.
- **lxc-hw-manager.** Users that are assigned this role can discover new devices and place those devices under the management control of the XClarity Administrator. This role prohibits users from performing operations or modifying configurations settings on the management server and managed devices beyond those operations that are necessary to discover and manage new devices.
- **lxc-operator.** Users that are assigned this role can view all configuration and status information about the management server and managed devices. This role prohibits users from performing operations or modifying configurations settings on the management server and managed devices.
- **lxc-recovery.** Users that are assigned this role can modify security settings and perform security-related operations on the management server. These users can also authenticate directly to the XClarity Administrator even if the authentication method is set to external LDAP server. This role provides a recovery mechanism in case a communication error occurs with the external LDAP server that uses the “Login Credentials” configuration.

Users that are assigned this role always have access to all managed devices. You cannot restrict access to devices for this role.

The following predefined roles are *reserved* and cannot be used to create new role groups or assigned to new users.

- **lxc-sysrdr**
- **lxc-sysmgr**

-UserList String[]

Specifies the IDs of one or more users to associate with the role group. You can use the [Get-LXCAUserAccount](#) cmdlet to obtain the user IDs.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does returns a RoleGroup object.

Examples

The following example creates a role group that is associated with two roles and two users.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
$role = Get-LXCARole
$user = Get-LXCAUserAccount -GetUserName
Add-LXCARoleGroup -RoleGroupName $name -Description $desc -RoleList $role[0],$role[2]
                -UserList $user[0],$user[1]
```

Add-LXCASharedCredential

This cmdlet creates a stored credential in the Lenovo XClarity Administrator server. There are two types of stored credentials: management and recovery. When you create a recovery stored credential, RECOVERY_ID is used as the user name.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCASStoredCredential [-Connection LXCAConnection]  
    [-Description String]  
    -Password SecureString  
    -UserName String  
    [CommonParameter]
```

```
Add-LXCASStoredCredential [-Connection LXCAConnection]  
    [-Description String]  
    -Password SecureString  
    -RecoveryCredential  
    [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Description *String*

Specifies the description of the new stored credential.

-Password *SecureString*

Specifies the password for the new stored credential.

-UserName *String*

Specifies the user name for the new stored credential.

-RecoveryCredential

Indicates that the stored credential that is designated for recovery purposes. In this case, credential's UserName is always *RECOVERY_ID*.

If not specified, the stored credential that is designated for management purposes

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a [StoredCredentialData](#) object.

Examples

The following example creates a management stored credential.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
$pwd = ConvertTo-SecureString "Passw0rd" -AsPlainText -Force  
$account = Add-LXCASStoredCredential -UserName USERID1 -Password $pwd  
  
Disconnect-LXCA
```

The following example creates a recovery stored credential.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
$pwd = ConvertTo-SecureString "Passw0rd" -AsPlainText -Force  
$account = Add-LXCASStoredCredential -Password $pwd -RecoveryCredential
```

Add-LXCAUserAccount

This cmdlet adds a user account to the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCAUserAccount [-Connection Connection]
                    -UserName String
                    -Password SecureString
                    -RoleGroups String[]
                    [-PasswordChangeFirstAccess Bool]
                    [-Description String]
                    [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Username *String*

Specifies the user ID of the new user account.

-Password *SecureString*

Specifies the password of the new user account.

-RoleGroups *String[]*

Specifies one or more role groups to which the new user account is to be assigned.

The following role groups are predefined in XClarity Administrator. You can also create customized role groups to manage authorization and access to resources on the web interface.

- **LXC-SUPERVISOR.** Includes the **lxc-supervisor** role.
- **LXC-ADMIN.** Includes the **lxca-admin** role.
- **LXC-SECURITY-ADMIN.** Includes the **lxc-security-admin** role.
- **LXC-HW-ADMIN.** Includes the **lxc-hw-admin** role.
- **LXC-FW-ADMIN.** Includes the **lxc-fw-admin** role.
- **LXC-OS-ADMIN.** Includes the **lxc-os-admin** role.
- **LXC-SERVICE-ADMIN.** Includes the **lxc-service-admin** role.
- **LXC-HW-MANAGER.** Includes the **lxc-hw-manager** role.
- **LXC-OPERATOR.** Includes the **lxc-operator** role.
- **LXC-RECOVERY.** Includes the **lxc-recovery** role.

-PasswordChangeFirstAccess *Bool*

Specifies that the password must be changed the first time it is used. The default value is True.

-Description *String*

Specifies a description for the new user account.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For

detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a `UserAccount` object.

Examples

The following example creates a user account.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$pwd = ConvertTo-SecureString "password" -AsPlainText Force
$account = Add-LXCAUserAccount -UserName USERID1 -Password $pwd -RoleGroups lxc-supervisor,lxc-admin`
          -PasswordChangeFirstAccess $true

Disconnect-LXCA
```

Get-LXCAActiveSession

This cmdlet retrieves a list of all currently active user sessions on the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAActiveSession [-Connection LXCAConnection]
                    [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns one or more `ActiveSession` objects.

Examples

The following example retrieves a list of all currently active user sessions.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$sessions = Get-LXCAActiveSession

Disconnect-LXCA
```

Get-LXCAEncapsulation

This cmdlet retrieves the global encapsulation setting on the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAEncapsulation [-Connection LXCAConnection]  
                    [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns the encapsulation setting as a Boolean value.

Examples

The following example retrieves the global encapsulation setting.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$setting = Get-LXCAEncapsulation
```

```
Disconnect-LXCA
```

Get-LXCAIdentityManagementSystems

This cmdlet returns information about all configured identity-management systems or a specific identity-management system configuration.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAIdentityManagementSystems [-Connection LXCAConnection]  
                                  [-Name String]  
                                  [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Name *String*

Type of identity-management system. This can be the following value.

- **CyberArk**.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- **Lenovo.SysMgmt.LXCA.Integration.Data.IdentityManagementSystems** This cmdlet returns an `IdentityManagementSystems` object, which contains information about all configured identity-management systems.
- Optionally use the **-Name** parameter to filter for a specific identity-management system configuration.

Examples

The following example retrieves all configured identity-management systems

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

$result = Get-LXCAIdentityManagementSystems

# show all the information about the request:
# - Port, Host, Name and Paths
$result

# show all the information about all paths:
$result.Paths | Format-List

Disconnect-LXCA
```

Get-LXCARole

This cmdlet retrieves information about all roles.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCARole [-Connection LXCAConnection]
             [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does returns a Role object.

Examples

The following example returns information for all role.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$roles = Get-LXCARole
```

Disconnect-LXCA

Get-LXCARoleGroup

This cmdlet retrieves information about all or a specific role group.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCARoleGroup [-Connection LXCACConnection]  
                  [-GroupName String]  
                  [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-GroupName *String*

Specifies the name of the role group.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does returns a RoleGroup object.

Examples

- The following example returns information for all role groups.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$roleGroups = Get-LXCARoleGroup
```

```
Disconnect-LXCA
```

- The following example returns information for a role group.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$roleGroup = Get-LXCARoleGroup -GroupName $name
```

```
Disconnect-LXCA
```

Get-LXCASharedCredential

This cmdlet retrieves information for stored credentials from the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCASharedCredential [-Connection LXCACConnection]  
                          [-Id String]  
                          [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Id *String*

Specifies the ID of a specific stored credential. If not specified, information about all stored credentials is returned.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- [Lenovo.SysMgmt.LXCA.Integration.Data.StoredCredentialData](#)

If **-Id** is specified, this cmdlet returns a single `StoredCredentialData` object, for the stored credential with specified ID.

- [Lenovo.SysMgmt.LXCA.Integration.Data.StoredCredentialData\[\]](#)

If **-Id** is not specified, this cmdlet returns an array of `StoredCredentialData` objects, for all stored credentials existing in XClarity Administrator.

Examples

The following example retrieves information for all stored credentials.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$accounts = Get-LXCASharedCredential
```

```
Disconnect-LXCA
```

The following example retrieves information for a specific stored credential.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
$accounts = Get-LXCASharedCredential -Id $id
```

```
Disconnect-LXCA
```

Get-LXCAUserAccount

This cmdlet returns information for all user accounts that were created in Lenovo XClarity Administrator.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAUserAccount [-Connection Connection]
                   [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a UserAccount object for each user account.

Examples

The following example retrieves all XClarity Administrator user accounts.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$accounts = Get-LXCAUserAccount

Disconnect-LXCA
```

Install-LXCASharedCredential

This cmdlet resolves stored credentials that are used to establish a secure connection to a managed device. Use this cmdlet when a stored credential becomes expired or inoperable on a device.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Install-LXCASharedCredential [-Connection LXCASharedCredential]
                             -Uuid String
                             -StoredCredentialId String
                             [CommonParameter]
```

Parameters

-Connection *LXCASharedCredential*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String*

Specifies the UUID of the device for which stored credentials are to be resolved.

-StoredCredentialId *String*

Specifies the ID of the stored credential to be resolved.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return object.

Examples

The following example The following example resolves stored credentials for a specific device.

```
$cred = Get-Credential
```

```
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
Install-LXCAStrustedCredential -Uuid $node -StoredCredentialId 1523
```

```
Disconnect-LXCA
```

Install-LXCATrustedCertificate

This cmdlet resolves untrusted certificates that are used to establish a secure connection to a managed device.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Install-LXCATrustedCertificate [-Connection LXCACConnection]  
                               [-Uuid String]  
                               [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String*

Specifies the UUID of the device for which untrusted certificates are to be resolved.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

The following example resolves untrusted certificates for a specific device.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
Get-LXCATrustedCertificate -UUID $node
```

```
Disconnect-LXCA
```

Remove-LXCARoleGroup

This cmdlet deletes a specific role group.

Notes:

- You cannot delete role groups that have members (users).
- You cannot delete the reserved role groups **lxc-sysrdr** and **lxc-sysmgr**.
- You cannot pipe objects to this cmdlet.

Syntax

```
Add-LXCACabinetItem [-Connection LXCACConnection]  
                    -RoleGroupName String  
                    [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-RoleGroupName *String*

Specifies the name of the role group to be deleted.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

The following example deletes a role group.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
Remove-LXCARoleGroup -RoleGroupName $name  
  
Disconnect-LXCA
```

Remove-LXCASharedCredential

This cmdlet deletes a shared credential from the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Remove-LXCASharedCredential [-Connection LXCACConnection]  
                             -Id String  
                             [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Id *String*

Specifies the ID of a specific shared credential to be deleted.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet removes a stored credential from the XClarity Administrator server. No object is returned.

Examples

The following example deletes a stored credential.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Remove-LXCASStoredCredential -Id $id

Disconnect-LXCA
```

Remove-LXCAUserAccount

This cmdlet removes an existing user account from the Lenovo XClarity Administrator server.

Note: You can pipe a `UserAccount` object to this cmdlet.

Syntax

```
Remove-LXCAUserAccount [-Connection LXCACONNECTION]
                        -UserAccount USERACCOUNT
                        [-Force]
                        [COMMONPARAMETER]
```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-UserAccount *USERACCOUNT*

Specifies the user account to be deleted.

-Force

When specified, removes the user account without first prompting for confirmation about the removal.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet removes the user account from the authentication server. No objects are returned.

Examples

The following example removes the `testuser` user account.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$account = Get-LXCAUserAccount | where { $_.UserName -eq "testuser" }
Remove-LXCAUserAccount -UserAccount $account -Force

Disconnect-LXCA
```

Set-LXCAEncapsulation

This cmdlet enables or disables the global encapsulation setting on the Lenovo XClarity Administrator server.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCAEncapsulation [-Connection LXCAConnection]  
                    -EnableEncapsulation Bool  
                    [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-EnableEncapsulation *Bool*

Specifies whether to enable the global encapsulation settings. You can specify one of the following values:

- **true**. Enables the global encapsulation settings
- **false**. Disables the global encapsulation settings

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

The following example enables the encapsulation setting.

```
$cred = Set-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred  
  
Set-LXCAEncapsulation -EnableEncapsulation $true  
  
Disconnect-LXCA
```

Set-LXCAResourceAccessControl

This cmdlet enables or disables the resource access control.

Notes:

- Only users with **lxc-supervisor** or **lxc-security-admin** authority can modify the access-control settings.
- You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCAResourceAccessControl [-Connection LXCAConnection]  
                             -enable Boolean  
                             [CommonParameter]
```


Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-enable *Boolean*

Specifies whether to enable access control for managed devices. This can be one of the following values.

- **true**. Enables access control.
- **false**. Disables access control.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

- The following example enables resource-access control.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
Set-LXCAResourceAccessControl -enable true
```

```
Disconnect-LXCA
```

- The following example disables resource-access control.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
Set-LXCAResourceAccessControl -enable false
```

```
Disconnect-LXCA
```

Set-LXCAResourceView

This cmdlet modifies the access-control settings for a specific device.

Notes:

- Only users with **lxc-supervisor** or **lxc-security-admin** authority can modify the access-control settings.
- You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCAResourceView [-Connection LXCAConnection]
                    [-RoleGroupNames String[]]
                    [-PublicAccess Boolean]
                    -Uuid String
                    {-Chassis | -Server | -Storage | -Switch}
                    [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-RoleGroupName *String[]*

Specifies the name of one or more role groups. You can use the [Get-LXCARoleGroup](#) cmdlet to obtain the role-group names.

-PublicAccess *Boolean*

Indicates whether the device can be accessed by all role groups. This can be one of the following values.

- **true**. The device is can be access by all role group.
- **false**. The device is restricted to specific role groups.

-Uuid *String*

Specifies the UUID of the device. You can use the [Get-LXCACHassis](#), [Get-LXCARackSwitch](#), [Get-LXCAServer](#), [Get-LXCAThinkServer](#), and [Get-LXCASStorageNode](#) cmdlets to obtain the device UUIDs.

-Chassis

Indicates that the specified UUID is for a managed chassis.

-Server

Indicates that the specified UUID is for a managed server.

-Storage

Indicates that the specified UUID is for a managed storage device.

-Switch

Indicates that the specified UUID is for a managed switch.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

The following example modifies public access and role groups for a specific managed server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

Set-LXCAResourceView -Uuid $uuid -server -PublicAccess $true -RoleGroupNames $roleGroupName1,$roleGroupName2

Disconnect-LXCA
```

Set-LXCARoleGroup

This cmdlet modifies a specific role group.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCARoleGroup [-Connection LXCAConnection]
                  -Id String
                  [-Description String]
                  [-RoleGroupName String]
```

[-RoleList *String*]
[-UserList *String*]
[*CommonParameter*]

Parameters

-Connection *LXCACconnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Id *String*

Specifies the unique ID of the role group to be changed.

-Description *String*

Specifies the description for the new role group.

-RoleGroupName *String*

Specifies the name of the role group. The name must be unique.

-RoleList *String*]

One or more roles to be associated with the new role group. This can be one of the following values.

- **lxc-supervisor**. Users that are assigned this role can access, configure, and perform all available operations on the management server and all managed devices. Users that are assigned this role always have access to all managed devices. You cannot restrict access to devices for this role.
- **lxc-admin**. Users that are assigned this role can modify non-security related settings and perform all non-security related operations on the management server, including the ability to update and restart the management server. This role also provides the ability to view all configuration and status information about the management server and managed devices.
- **lxc-security-admin**. Users that are assigned this role can modify security settings and perform security-related operations on the management server and managed devices. This role also provides ability to view all configuration and status information about the management server and managed devices.

Users that are assigned this role always have access to all managed devices. You cannot restrict access to devices for this role.

- **lxc-hw-admin**. Users that are assigned this role can modify non-security settings and perform non-security related operations on managed devices, including the ability to update and restart managed devices. This role also provides the ability to view all configuration and status information about the management server and all managed devices.
- **lxc-fw-admin**. Users that are assigned this role can create firmware policies and deploy those policies to managed devices. Users that are not assigned this role can only view policy information.
- **lxc-os-admin**. Users that are assigned this role can download and deploy operating systems and device-driver updates to managed servers. Users that are not assigned this role can only view operating-system and device-driver information.
- **lxc-service-admin**. Users that are assigned this role can collect and download service files for XClarity Administrator and managed devices. Users that are not assigned this role can collect but not download service data.
- **lxc-hw-manager**. Users that are assigned this role can discover new devices and place those devices under the management control of the XClarity Administrator. This role prohibits users from performing operations or modifying configurations settings on the management server and managed devices beyond those operations that are necessary to discover and manage new devices.
- **lxc-operator**. Users that are assigned this role can view all configuration and status information about the management server and managed devices. This role prohibits users from performing operations or modifying configurations settings on the management server and managed devices.
- **lxc-recovery**. Users that are assigned this role can modify security settings and perform security-related operations on the management server. These users can also authenticate directly to the XClarity Administrator even if the authentication method is set to external LDAP server. This role

provides a recovery mechanism in case a communication error occurs with the external LDAP server that uses the “Login Credentials” configuration.

Users that are assigned this role always have access to all managed devices. You cannot restrict access to devices for this role.

The following predefined roles are *reserved* and cannot be used to create new role groups or assigned to new users.

- **lxc-sysrdr**
- **lxc-sysmgr**

-UserList String[]

Specifies the IDs of one or more users to associate with the role group. You can use the [Get-LXCAUserAccount](#) cmdlet to obtain a user account.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet does not return an object.

Examples

- The following example modifies the role group name.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$group = Get-LXCARoleGroup -GroupName TEST
Set-LXCARoleGroup -Id $group.Id -RoleGroupName $name

Disconnect-LXCA
```
- The following example modifies the role group description and roles list.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$group = Get-LXCARoleGroup -GroupName TEST
$role = Get-LXCARole
Set-LXCARoleGroup -Id $group.Id -Description $desc -RoleList $role[1],$role[2]

Disconnect-LXCA
```
- The following example modifies the users that are associated with a role group.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$group = Get-LXCARoleGroup -GroupName TEST
$role = Get-LXCARole
Set-LXCARoleGroup -Id $group.Id -UserList $user[1],$user[2]

Disconnect-LXCA
```

Set-LXCASharedCredential

This cmdlet modifies properties for a specific shared credential.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCASStoredCredential [-Connection LXCAConnection]  
                          [-Description String]  
                          -Id String  
                          [-Password SecureString]  
                          [-UserName String]  
                          [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Description *String*

Specifies the new description of the stored credential.

-Id *String*

Specifies the ID of a specific stored credential to be modified.

-Password *SecureString*

Specifies the new password of the stored credential.

-UserName *String*

Specifies the new user ID of the stored credential.

Note: You cannot change a user name if the stored credential is associated with one or more devices. Any attempt to change the user name for a stored credential that is currently in use will throw an exception Internal Server Error.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a [StoredCredentialData](#) object.

Examples

- The following example modifies the user name and password for a specific stored credential.

```
$Password = ConvertTo-SecureString "curpwd" -AsPlainText -Force  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
Set-LXCASStoredCredential -Id $id -Password $Password -UserName ADMIN
```

```
Disconnect-LXCA
```

- The following example modifies the description for a specific stored credential.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred
```

```
Set-LXCASStoredCredential -Id $id -Description $description
```

```
Disconnect-LXCA
```

Set-LXCAUserAccount

This cmdlet modifies the properties for an existing user account on the Lenovo XClarity Administrator server.

To modify the password for an existing user account, use the `Set-LXCAUserAccountPassword` cmdlet.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCAUserAccount [-Connection Connection]  
                  -UserAccount UserAccount  
                  [-RoleGroups String[]]  
                  [-State String]  
                  [-Description String]  
                  [CommonParameter]
```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-UserAccount *UserAccount[]*

Specifies the user account to be modified. You can use the `Get-LXCAUserAccount` to obtain a user account.

-RoleGroups *String[]*

Specifies one or more role groups to which the new user account is to be assigned.

The following role groups are predefined in XClarity Administrator. You can also create customized role groups to manage authorization and access to resources on the web interface

- **LXC-SUPERVISOR.** Includes the **lxc-supervisor** role.
- **LXC-ADMIN.** Includes the **lxca-admin** role.
- **LXC-SECURITY-ADMIN.** Includes the **lxc-security-admin** role.
- **LXC-HW-ADMIN.** Includes the **lxc-hw-admin** role.
- **LXC-FW-ADMIN.** Includes the **lxc-fw-admin** role.
- **LXC-OS-ADMIN.** Includes the **lxc-os-admin** role.
- **LXC-SERVICE-ADMIN.** Includes the **lxc-service-admin** role.
- **LXC-HW-MANAGER.** Includes the **lxc-hw-manager** role.
- **LXC-OPERATOR.** Includes the **lxc-operator** role.
- **LXC-RECOVERY.** Includes the **lxc-recovery** role.

-State *String*

Specifies the current state for the user account. You can specify one of the following values:

- Active
- Inactive

-Description *String*

Specifies a description for the user account.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns a `UserAccount` object that contains the modified values.

Examples

The following example modifies properties for an existing user account.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$account = Get-LXCAUserAccount | where { $_.UserName -eq "testuser" }
$newaccount = Set-LXCAUserAccount -UserAccount $account -RoleGroups lxc-security-admin,lxc-admin`
              -Description "New full name"

Disconnect-LXCA
```

Set-LXCAUserAccountPassword

This cmdlet modifies the password for an existing user account on the Lenovo XClarity Administrator server.

Modifying the password of user account other than the currently logged in user is the same as resetting that user's password (without checking that user's password).

To modify other properties for an existing user account, use the `Set-LXCAUserAccount` cmdlet.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCAUserAccountPassword [-Connection Connection]
                             -CurrentPassword SecureString
                             -NewPassword SecureString
                             -ConfirmPassword SecureString
                             [CommonParameter]

Set-LXCAUserAccountPassword [-Connection Connection]
                             -Host String
                             [-Port Int]
                             [-Proxy WebProxy]
                             -UserName String
                             -CurrentPassword SecureString
                             -NewPassword SecureString
                             -ConfirmPassword SecureString
                             [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Host *String*

Specifies the HTTPS address (IP address or hostname) of the XClarity Administrator server.

-Port *Int*

Specifies the port on the XClarity Administrator server to be used for the connection. The default is 443.

-Proxy *WebProxy*

Specifies a Web service proxy object that includes proxy connection and authentication information.

-UserName *String*

Specifies the ID for the user account to be modified.

-CurrentPassword *SecureString*

Specifies the current password.

-NewPassword *SecureString*
Specifies the new password.

-ConfirmPassword *SecureString*
Confirms the new password.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlets modifies the password for the user account. No object is returned.

Examples

- The following example modifies the password for a currently authenticated user.

```
cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Port 443 -Credential $cred

$Password = ConvertTo-SecureString "curpwd" -AsPlainText -Force
$NewPassword = ConvertTo-SecureString "newpwd" -AsPlainText -Force
$ConfirmPassword = ConvertTo-SecureString "newpwd" -AsPlainText -Force
Set-LXCAUserAccountPassword -CurrentPassword $Password -NewPassword $NewPassword`
                           -ConfirmPassword $ConfirmPassword
```

```
Disconnect-LXCA
```

- The following example illustrates how to use this cmdlet to modify the password for a user that is not currently authenticated with the XClarity Administrator server.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred

$Password = ConvertTo-SecureString "curpwd" -AsPlainText -Force
$NewPassword = ConvertTo-SecureString "newpwd" -AsPlainText -Force
$ConfirmPassword = ConvertTo-SecureString "newpwd" -AsPlainText -Force
Set-LXCAUserAccountPassword -Host 192.0.2.0 -UserName "userid" -CurrentPassword $Password`
                           -NewPassword $NewPassword -ConfirmPassword $ConfirmPassword
```

```
Disconnect-LXCA
```

Service and support cmdlets

The following PowerShell cmdlets are available for performing service and support functions.

Get-LXCADevicesWarranty

This cmdlet retrieves or downloads information about warranties for managed devices.

Note: You cannot pipe objects to this cmdlet.

Syntax

To retrieve information about all warranties or all warranties in a specific state, use the following syntax.

```
Get-LXCADevicesWarranty [-Connection LXCACConnection]
                        [-WarrantyStatusType String]
                        [CommonParameter]
```

To download information about all warranties, use the following syntax.


```
Get-LXCADevicesWarranty [-Connection LXCACConnection]  
                        -Destination String  
                        [-Overwrite]  
                        [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-WarrantyStatusType *String*

Status of the warranties that you want to retrieve. This can be one of the following values.

- **All**. (default) Returns information about all warranties.
- **ActiveOnly**. Returns information about all active warranties.
- **ExpiredOnly**. Returns information about all expired warranties.
- **NotAvailableOnly**. Returns information about warranties for which the status is not available.

-Destination *String*

Directory and file name where the file is to be saved on the local system.

- If you do not specify a file name (in other words, if the destination ends in a back slash "\"), the default file name (`warrantyInfo-nn.csv.`) is used.
- If the directory does not exist, it is automatically created.

-Overwrite

When specified, if the file already exists in the specified directory, the file is overwritten.

If not specified, and the file already exists in the specified directory, the file is not overwritten and the `System.IO.IOException: The <full_csv_file_path> file already exists` exception is thrown

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

If **-Destination** is not specified, this cmdlet returns a [Lenovo.SysMgmt.LXCA.Integration.Data.WarrantyInfo](#) array that contains information for all warranties or for all warranties in a specific state.

If **-Destination** is specified, this cmdlet returns a `System.String` object that contains the full path of the saved csv file, and downloads the file to the specified directory on the local system.

Examples

- The following example returns information about all expired warranties.

```
$cred = Get-Credential  
Connect-LXCA -Host "192.0.2.0" -Credential $cred  
  
# $result contains an array of WarrantyInfo objects with expired Warranty Status.  
$result = Get-LXCADevicesWarranty -WarrantyStatusType ExpiredOnly  
$result  
  
Disconnect-LXCA
```

- The following example downloads all warranty information as a *csv file in the `$PSScriptRoot\.._lxca\warranties` directory using the default file name. If the file already exists, the file is overwritten.

```
$cred = Get-Credential  
Connect-LXCA -Host "192.0.2.0" -Credential $cred
```

```
# $result contains the full path of the saved csv file.
$result = Get-LXCADevicesWarranty -Destination "$PSScriptRoot\..\_lxca\warranties\" -Overwrite
$result
```

Disconnect-LXCA

Get-LXCAFFDCFile

This cmdlet downloads the first failure data capture (FFDC) file for Lenovo XClarity Administrator file to the local system.

Notes:

- You cannot pipe objects to this cmdlet.
- The returned FFDC file is a tar.gz archive.

Syntax

```
Get-LXCAFFDCFile [-Connection LXCACConnection]
                 -Destination String
                 [-Overwrite]
                 [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Destination *String*

Specifies the directory and file name where the file is to be saved on the local system.

- If you do not specify a file name (in other words, if the destination ends in a back slash "\"), the default file name **LXCA_<app-uuid>_<time>_<date>.tar.gz** is used.
- If the directory does not exist, it is automatically created.

-Overwrite

When specified, if the file already exists in the specified directory, the file is overwritten.

If not specified, and the file already exists in the specified directory, the file is not overwritten, and the System.IO.IOException: The <full_ffdc_file_path> file already exists exception is thrown.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet downloads the XClarity Administrator service-data file to the specified directory on the local system and returns a System.String object that contains the full path of the saved file

Examples

- The following example downloads the FFDC file, saves it on the local system, and prints the file's fully qualified path to the PS console. If a file with same name already exists, the file is overwritten.

```
$cred = Get-Credential
Connect-LXCA -Host 192.0.2.0 -Credential $cred -SkipCertificateCheck

$(Get-LXCAFFDCFile -Destination C:\LXCA\ffdc.tar.gz -Overwrite)
```

```
Disconnect-LXCA
```

- The following example downloads the FFDC file, saves it on the local system, and prints the file's fully qualified path to the PS console. The file name used is the one that is returned by XClarity Administrator.

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred -SkipCertificateCheck
```

```
$(Get-LXCAFFDCFile -Destination C:\LXCA\)
```

```
Disconnect-LXCA
```

Get-LXCAServiceFile

This cmdlet downloads the service First Failure Data Capture (service) file for a specified device (such as a chassis, rack server, rack switch, or storage device) to the local system.

Notes:

- This cmdlet downloads a service files that was previously collected. This cmdlet does not collect service data for the device.
- You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAServiceFile [-Connection LXCACConnection]  
                   -Uuid String[]  
                   -Destination String  
                   [-Overwrite]  
                   [CommonParameter]
```

```
Get-LXCAServiceFile [-Connection LXCACConnection]  
                   -All  
                   -Destination String  
                   [-Overwrite]  
                   [CommonParameter]
```

Parameters

-Connection *LXCACConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuid *String[]*

Specifies one or more UUIDs of managed devices.

-All

If specified, the cmdlet downloads FFDC data for all managed devices.

-Destination *String*

Specifies the local path where the file is to be saved.

The file is downloaded in a single .TAR.GZ archive file.

If a directory is specified, the cmdlet saves the FFDC file to a file using the file name that is returned from the XClarity Administrator server. If a directory and a file name are specified, the cmdlet saves the FFDC file using the specified file name.

Note: The full specified path must exist.

-Overwrite

If specified, the cmdlet overwrites the local file when a file with that same name exists.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns the path to the FFDC file on the local system.

Examples

- The following example downloads the FFDC file for a specified device to the C:\LXCA\ directory on the local system. If the file of the same name exists in the specified directory, the file is overwritten.

```
$cred = Get-Credential
$connection = Connect-LXCA -Host 192.0.2.0 -Credential $cred

Get-LXCAServiceFile -Uuid AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA -Destination C:\LXCA\ -Overwrite

Disconnect-LXCA
```

- The following example downloads the FFDC files for all managed devices in a compressed file named myFFDC.tar.gz in the C:\LXCA\ directory on the local system. If the file of the same name exists in the specified directory, the file is not overwritten.

```
$cred = Get-Credential
$connection = Connect-LXCA -Host 192.0.2.0 -Credential $cred

Get-LXCAServiceFile -All -Destination C:\LXCA\myFFDC.tar.gz

Disconnect-LXCA
```

Get-LXCAServiceTickets

This cmdlet retrieves information about service tickets.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCAServiceTickets [-Connection LXCACONNECTION]
                        [-Uuids String[]]
                        [-TicketState String]
                        [CommonParameters]
```

Parameters

-Connection *LXCACONNECTION*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Uuids *String[]*

Returns information about service tickets that are associated with one or more specific groups and devices, specified by UUID.

If not specified, information about all service tickets for all groups and devices are returned.

-TicketState *String*

Returns information about service tickets that are in a specific state. You can specify one of the following values.

- **Active**
- **Answered**
- **Cancelled**
- **Canceled**
- **Closed**
- **Created**
- **Error**
- **Initialized**
- **NewState**
- **Processing**
- **Rejected**
- **Resolved**
- **Submitted**
- **Unknown**
- **Waiting**

If not specified, information about all service tickets for all groups and devices are returned.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

This cmdlet returns an array of `Lenovo.SysMgmt.LXCA.Integration.Data.SwitchConfigData` objects (`Lenovo.SysMgmt.LXCA.Integration.Data.ServiceTicket[]`).

Examples

- The following example returns all service tickets that are associated with two specific devices.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

# $result contains all service tickets for the specified devices.
$uuids = @("AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA", "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB")
$result = Get-LXCAServiceTickets -Uuids $uuids
$result
```

```
Disconnect-LXCA
```

- The following example retrieves returns service tickets that are in the "processing" state for all devices in a specific group.

```
$cred = Get-Credential
Connect-LXCA -Host "192.0.2.0" -Credential $Cred -SkipCertificateCheck

# $result contains all service ticket that are in the "processing" state for devices in the
# specified group.
$group = CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
$result = Get-LXCAServiceTickets -Uuids $group -TicketState processing
$result
```

```
Disconnect-LXCA
```

Get-LXCABulletins

This cmdlet returns a list of bulletins from the last 30 days.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Get-LXCABulletins [-Connection LXCAConnection]  
                  [-Severity String]  
                  [-StartDate String]  
                  [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-Severity *String*

If specified, returns only bulletins with the specified severity. This can be one of following values.

- **Critical**
- **Informational**
- **Warning**

-StartDate *String*

If specified, returns only bulletins that were published after the specified timestamp.

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- [Lenovo.SysMgmt.LXCA.Integration.Data.BulletinData\[\]](#)

If the request was successful, this cmdlet returns an array of [BulletinData](#) objects that contains information about each bulletin.

- [Lenovo.SysMgmt.LXCA.Integration.Data.RequestResult](#)

If the bulletin service is not enable, no bulletin were returned, or an error occurred, this cmdlet returns a [RequestResult](#) object that contains information about the status of this request.

Examples

The following example retrieves all bulletins that match the specified parameters

```
$cred = Get-Credential  
Connect-LXCA -Host 1.111.11.11 -Credential $cred -SkipCertificateCheck
```

```
$result = Get-LXCABulletins -Severity Critical -StartDate 9/24/2021
```

```
# Show all bulletins:  
$result | Format-List
```

```
Disconnect-LXCA
```

Set-LXCABulletinService

This cmdlet enable or disable receiving Lenovo bulletins.

Note: You cannot pipe objects to this cmdlet.

Syntax

```
Set-LXCABulletinService [-Connection LXCAConnection]  
                        -ServiceStatus String  
                        [CommonParameter]
```

Parameters

-Connection *LXCAConnection*

Specifies the connection to the Lenovo XClarity Administrator server. If no connection is specified, the result from the last [Connect-LXCA](#) cmdlet is used.

-ServiceStatus *String*

Specifies whether the bulletin service is enabled or disabled. This can be one of following values.

- **Enabled**
- **Disabled**

CommonParameters

This cmdlet supports the following common parameters: **-Verbose**, **-Debug**, **-ErrorAction**, **-ErrorVariable**, **-WarningAction**, **-WarningVariable**, **-OutBuffer**, **-PipelineVariable**, **-OutVariable**. For detailed information about each common parameter, see the [Microsoft PowerShell Common Parameters webpage](#).

Results

- [Lenovo.SysMgmt.LXCA.Integration.Data.RequestResult](#)

This cmdlet returns a [RequestResult](#) object that contains information about the status of this request.

Examples

The following example enables bulletin service

```
$cred = Get-Credential  
Connect-LXCA -Host 192.0.2.0 -Credential $cred -SkipCertificateCheck  
  
$(Set-LXCABulletinService -ServiceStatus Enabled) | Format-List  
  
Disconnect-LXCA
```

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area.

Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document is not an offer and does not provide a license under any patents or patent applications. You can send inquiries in writing to the following:

*Lenovo (United States), Inc.
1009 Think Place
Morrisville, NC 27560
U.S.A.
Attention: Lenovo VP of Intellectual Property*

LENOVO PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Trademarks

LENOVO, SYSTEM, NEXTSCALE, SYSTEM X, THINKSERVER, THINKSYSTEM, and XCLARITY are trademarks of Lenovo.

Intel is a trademark of Intel Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds.

Microsoft, Windows, Windows Server, Windows PowerShell, Hyper-V, Internet Explorer, and Active Directory are registered trademarks of the Microsoft group of companies.

Mozilla and Firefox are registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Nutanix is a trademark and brand of Nutanix, Inc. in the United States, other countries, or both.

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

SUSE is a trademark of SUSE IP Development Limited or its subsidiaries or affiliates.

VMware vSphere is a registered trademark of VMware in the United States, other countries, or both.

All other trademarks are the property of their respective owners.

Lenovo