Cisco UCS Manager CLI Quick Reference Guide for Cisco UCS M-Series Modular Servers, Release 2.5

First Published: 2015-04-07

Last Modified: 2021-03-26

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

Audience

This guide is intended primarily for data center administrators with responsibilities and expertise in one or more of the following:

- Server administration
- Storage administration
- Network administration
- Network security

Conventions

Text Type	Indication
GUI elements	GUI elements such as tab titles, area names, and field labels appear in **this font**. Main titles such as window, dialog box, and wizard titles appear in **this font**.
Document titles	Document titles appear in *this font*.
TUI elements	In a Text-based User Interface, text the system displays appears in `this font`.
System output	Terminal sessions and information that the system displays appear in `this font`.
CLI commands	CLI command keywords appear in **this font**. Variables in a CLI command appear in *this font*.
[ ]	Elements in square brackets are optional.

Text Type	Indication
{x | y | z}	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the document.

Tip Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Caution Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

Warning IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Related Cisco UCS Documentation

Documentation Roadmaps

For a complete list of all B-Series documentation, see the *Cisco UCS B-Series Servers Documentation Roadmap* available at the following URL: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/overview/guide/UCS_roadmap.html

For a complete list of all C-Series documentation, see the *Cisco UCS C-Series Servers Documentation Roadmap* available at the following URL: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/overview/guide/ucs_rack_roadmap.html.

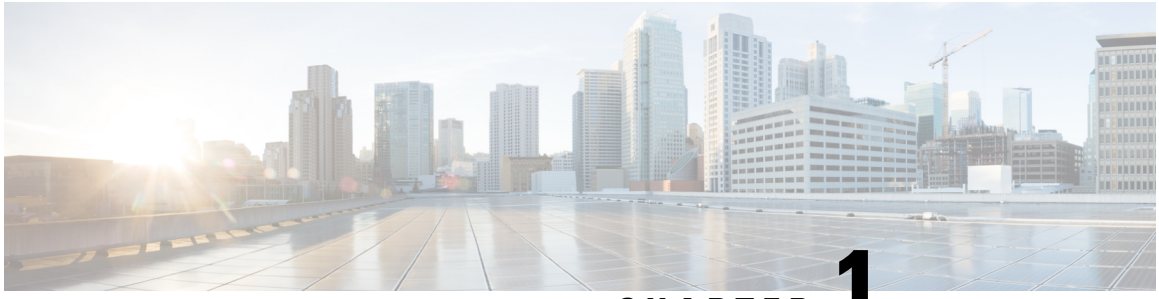
For information on supported firmware versions and supported UCS Manager versions for the rack servers that are integrated with the UCS Manager for management, refer to Release Bundle Contents for Cisco UCS Software.

Other Documentation Resources

Follow Cisco UCS Docs on Twitter to receive document update notifications.

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to ucs-docfeedback@external.cisco.com. We appreciate your feedback.



CHAPTER 1

Overview of Cisco UCS M-Series Modular Servers

This part contains the following chapters:

About Cisco UCS M-Series Modular Servers

A traditional server has captive resources, for example, CPU, network adapters, and local storage, which cannot be shared across multiple servers. Cisco UCS M-Series modular server is a new class of Cisco UCS servers that enables sharing some of these resources within a chassis. Resources, such as hard disk drives, I/O, and adapters, which have traditionally been dedicated to servers, are now aggregated at a chassis level and shared across multiple servers within that chassis.

The Cisco UCS M-Series modular server decouples the networking and storage components of the server and provides them as flexible, configurable resources that can be distributed as needed to the servers within the chassis. In Cisco UCS M-Series, the CPU and memory are local to the server. The server has standard PCIe connectivity to the chassis resources. The components that are shared in the chassis are power, management, cooling, storage, and networking.

What is New

Cisco UCS M-Series Modular Servers introduces the following in Cisco UCS Manager Release 2.5(2):

TPM and TXT Configuration through UCS Manager

The Trusted Platform Module (TPM) is a component that can securely store artifacts, such as passwords, certificates, or encryption keys, which are used to authenticate the server. A TPM can also be used to store platform measurements that help ensure that the platform remains trustworthy. Intel Trusted Execution Technology (TXT) provides greater protection for information that is used and stored on the server.

This release supports TPM and TXT configuration on Cisco UCSME-2814 compute cartridges through Cisco UCS Manager. TPM is enabled by default and TXT is disabled by default.

vNIC Placement on Host Ports

vNIC can be placed on one of the two host ports of the shared adapter. You can either explicitly specify the host port for placement, or allow Cisco UCS Manager to automatically assign vNICs to host ports. The host port placement of the vNIC determines the order of the vNIC on the adapter. The vNICs placed on the first host port will be enumerated first followed by the vNICs on the second host port. In this release, vNIC placement on host ports is applicable only to Cisco UCSME-1414 compute cartridges.

vNIC placement on host ports is detailed in *Cisco UCS Manager CLI Configuration Guide, Release 2.2.*

Support for up to 4 LUNs Per Server

The modular servers in Cisco UCSME-2814 compute cartridges include support for up to 4 LUNs per server, of which up to 2 LUNs are bootable. The modular servers in Cisco UCSME-1414 and UCSME-142 compute cartridges support up to 2 LUNs per server.

Cisco UCS M-Series Modular Servers introduces the following in Cisco UCS Manager Release 2.5(1):

Modular Server Architecture

Some of the main features of the modular server architecture are as follows:

- Compute cartridges—Each slot in the new modular chassis can contain a compute cartridge. Each compute cartridge can contain one or more servers.
- Three-tuple reference for modular servers—In the new architecture, compute servers are contained within cartridges, which in turn are contained in a chassis. This makes all references to Cisco M-Series servers take a three-tuple form—*chassis-id*/*cartridge-id*/*server-id*.
- Centralized resources—Resources such as storage, I/O, and storage controller are centralized and hosted in the chassis.
- Shared adapter—The shared adapter is contained in the modular chassis. It is the aggregate point for accessing network and storage resources for servers.

Storage Profiles

Storage profiles are introduced to provide flexibility in configuring the usage of centralized storage resources.

Cisco System Link Technology

The Cisco UCS Virtual Interface Card (VIC) used in the Cisco UCS M-Series modular servers provides multiple PCIe buses that connect to multiple servers simultaneously. It utilizes the System Link Technology, which extends a PCIe bus to each of the servers and creates a virtual device on the PCIe host interface for use by the local CPU. The OS sees this virtual device as a local PCIe device, and I/O traffic is passed up the host PCIe lanes, and mapped to the appropriate shared resource-the local storage or the networking interface.

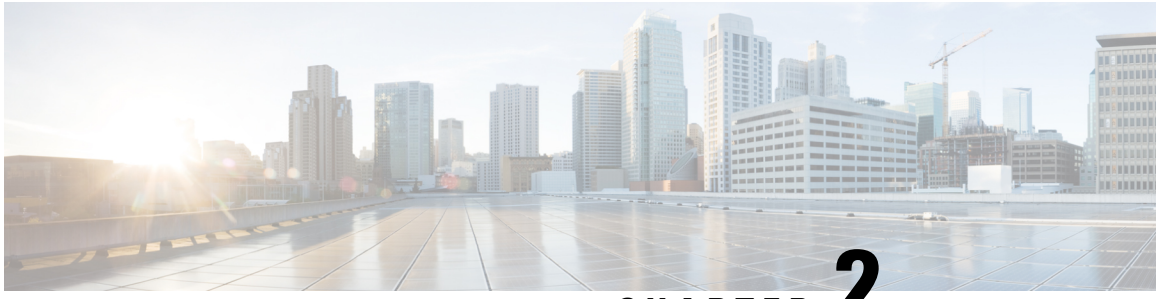
Virtual Storage Controller

The virtual storage controller provides access to virtual drives that are provided to the server through the shared storage controller and hard drives in the chassis. The virtual storage controller introduces a new PCIe device, known as a SCSI NIC (sNIC), which is presented to the OS. The OS views these items as locally-attached SCSI devices.

Guidelines and Limitations

- Because the connectivity between the Fabric Interconnect and the modular chassis is always in port channel mode, the chassis discovery policy is not applicable.
- Cisco UCS Manager Release 2.5 supports most of the features that are supported by Cisco UCS Manager Release 2.2. However, it does not support the following:
  - Cisco UCS B-Series Servers and Cisco C-Series Servers
  - Fiber Channel and Fiber Channel over Ethernet and associated configuration such as vHBA
  - Dynamic vNICs
  - usNICs
  - vMQ

All the features and configuration tasks that are supported by Cisco UCS Manager are detailed in *Cisco UCS Manager CLI Configuration Guide, Release 2.2*.



CHAPTER 2

Configuring Storage Profiles

This part contains the following chapters:

Storage Profiles

Unlike Cisco UCS B-Series and C-Series servers, the Cisco UCS M-Series modular servers do not have local storage. Instead, storage is centralized per chassis, and this centralized storage is shared by all servers in the chassis. To allow flexibility in defining the number of storage disks, roles and usage of these disks, and other storage parameters, you can create and use storage profiles. A storage profile encapsulates the storage requirements for one or more service profiles. LUNs configured in a storage profile can be used as boot LUNs or data LUNs, and can be dedicated to a specific server. You can also specify a local LUN as a boot device. However, LUN resizing is not supported. The introduction of storage profiles allows you to do the following:

- Configure multiple virtual drives and select the physical drives that are used by a virtual drive. You can also configure the storage capacity of a virtual drive.
- Configure the number, type and role of disks in a disk group.
- Associate a storage profile with a service profile.

You can create a storage profile both at an org level and at a service-profile level. A service profile can have a dedicated storage profile as well as a storage profile at an org level.

Disk Groups and Disk Group Configuration Policies

In UCS M-Series Modular Servers, servers in a chassis can use storage that is centralized in that chassis. You can select and configure the disks to be used for storage. A logical collection of these physical disks is called a disk group. Disk groups allow you to organize local disks. The storage controller controls the creation and configuration of disk groups.

A disk group configuration policy defines how a disk group is created and configured. The policy specifies the RAID level to be used for the disk group. It also specifies either a manual or an automatic selection of disks for the disk group, and roles for disks. You can use a disk group policy to manage multiple disk groups. However, a single disk group can be managed only by one disk group policy.

A hot spare is an unused extra disk that can be used by a disk group in the case of failure of a disk in the disk group. Hot spares can be used only in disk groups that support a fault-tolerant RAID level.

Virtual Drives

A disk group can be partitioned into virtual drives. Each virtual drive appears as an individual physical device to the Operating System.

All virtual drives in a disk group must be managed by using a single disk group policy.

Configuration States

Indicates the configuration states of a virtual drive. Virtual drives can have the following configuration states:

- Applying—Creation of the virtual drive is in progress.
- Applied—Creation of the virtual drive is complete, or virtual disk policy changes are configured and applied successfully.
- Failed to apply—Creation, deletion, or renaming of a virtual drive has failed due to errors in the underlying storage subsystem.
- Orphaned—The service profile that contained this virtual drive is deleted.
- Not in use—The service profile that contained this virtual drive is in the disassociated state.

Deployment States

Indicates the actions that you are performing on virtual drives. Virtual drives can have the following deployment states:

- No action—No pending work items for the virtual drive.
- Creating—Creation of the virtual drive is in progress.
- Deleting—Deletion of the virtual drive is in progress.
- Modifying—Modification of the virtual drive is in progress.
- Apply-Failed—Creation or modification of the virtual drive has failed.

Operability States

Indicates the operating condition of a virtual drive. Virtual drives can have the following operability states:

- Optimal—The virtual drive operating condition is good. All configured drives are online.
- Degraded—The virtual drive operating condition is not optimal. One of the configured drives has failed or is offline.
- Cache-degraded—The virtual drive has been created with a write policy of **write back** mode, but the BBU has failed, or there is no BBU.

  Note This state does not occur if you select the **always write back** mode.

- Partially degraded—The operating condition in a RAID 6 virtual drive is not optimal. One of the configured drives has failed or is offline. RAID 6 can tolerate up to two drive failures.
- Offline—The virtual drive is not available to the RAID controller. This is essentially a failed state.
- Unknown—The state of the virtual drive is not known.

Presence States

Indicates the presence of virtual drive components. Virtual drives have the following presence states:

- Equipped—The virtual drive is available.
- Mismatched—A virtual drive deployed state is different from its configured state.
- Missing—Virtual drive is missing.

RAID Levels

The RAID level of a disk group describes how the data is organized on the disk group for the purpose of ensuring availability, redundancy of data, and I/O performance.

The following are features provided by RAID:

- Striping—Segmenting data across multiple physical devices. This improves performance by increasing throughput due to simultaneous device access.
- Mirroring—Writing the same data to multiple devices to accomplish data redundancy.
- Parity—Storing of redundant data on an additional device for the purpose of error correction in the event of device failure. Parity does not provide full redundancy, but it allows for error recovery in some scenarios.
- Spanning—Allows multiple drives to function like a larger one. For example, four 20 GB drives can be combined to appear as a single 80 GB drive.

The supported RAID levels include the following:

- RAID 0 Striped—Data is striped across all disks in the array, providing fast throughput. There is no data redundancy, and all data is lost if any disk fails.
- RAID 1 Mirrored—Data is written to two disks, providing complete data redundancy if one disk fails. The maximum array size is equal to the available space on the smaller of the two drives.
- RAID 5 Striped Parity—Data is striped across all disks in the array. Part of the capacity of each disk stores parity information that can be used to reconstruct data if a disk fails. RAID 5 provides good data throughput for applications with high read request rates.

  RAID 5 distributes parity data blocks among the disks that are part of a RAID-5 group and requires a minimum of three disks.
- RAID 6 Striped Dual Parity—Data is striped across all disks in the array and two sets of parity data are used to provide protection against failure of up to two physical disks. In each row of data blocks, two sets of parity data are stored.

  Other than addition of a second parity block, RAID 6 is identical to RAID 5 . A minimum of four disks are required for RAID 6.
- RAID 10 Mirrored and Striped—RAID 10 uses mirrored pairs of disks to provide complete data redundancy and high throughput rates through block-level striping. RAID 10 is mirroring without parity and block-level striping. A minimum of four disks are required for RAID 10.

Automatic Disk Selection

When you specify a disk group configuration, and do not specify the local disks in it, Cisco UCS Manager determines the disks to be used based on the criteria specified in the disk group configuration policy. Cisco UCS Manager can make this selection of disks in multiple ways.

When all qualifiers match for a set of disks, then disks are selected sequentially according to their slot number. Regular disks and dedicated hot spares are selected by using the lowest numbered slot.

The following is the disk selection process:

1. Iterate over all local LUNs that require the creation of a new virtual drive. Iteration is based on the following criteria, in order:
   a. Disk type
   b. Minimum disk size from highest to lowest
   c. Space required from highest to lowest
   d. Disk group qualifier name, in alphabetical order
   e. Local LUN name, in alphabetical order
2. Select regular disks depending on the minimum number of disks and minimum disk size. Disks are selected sequentially starting from the lowest numbered disk slot that satisfies the search criteria.

Note If you specify **Any** as the type of drive, the first available drive is selected. After this drive is selected, subsequent drives will be of a compatible type. For example, if the first drive was SATA, all subsequent drives would be SATA. Cisco UCS Manager Release 2.5 supports only SATA and SAS.

Cisco UCS Manager Release 2.5 does not support RAID migration.

3. Select dedicated hot spares by using the same method as normal disks. Disks are only selected if they are in an **Unconfigured Good** state.
4. If a provisioned LUN has the same disk group policy as a deployed virtual drive, then try to deploy the new virtual drive in the same disk group. Otherwise, try to find new disks for deployment.

Supported LUN Modifications

Some modifications that are made to the LUN configuration when LUNs are already deployed on an associated server are supported.

The following are the types of modifications that can be performed:

- Creation of a new virtual drive.
- Deletion of an existing virtual drive, which is in the orphaned state.
- Non-disruptive changes to an existing virtual drive. These changes can be made on an existing virtual drive without loss of data, and without performance degradation:
  - Policy changes. For example, changing the write cache policy.
  - Modification of boot parameters

The removal of a LUN will cause a warning to be displayed. Ensure that you take action to avoid loss of data.

Unsupported LUN Modifications

Some modifications to existing LUNs are not possible without destroying the original virtual drive and creating a new one. All data is lost in these types of modification, and these modifications are not supported.

Disruptive modifications to an existing virtual drive are not supported. The following are unsupported disruptive changes:

- Any supported RAID level change that can be handled through reconstruction. For example, RAID0 to RAID1.
- Increasing the size of a virtual drive through reconstruction.
- Addition and removal of disks through reconstruction.

Destructive modifications are also not supported. The following are unsupported destructive modifications:

- RAID-level changes that do not support reconstruction. For example, RAID5 to RAID1.

- Shrinking the size of a virtual drive.
- RAID-level changes that support reconstruction, but where there are other virtual drives present on the same drive group.
- Disk removal when there is not enough space left on the disk group to accommodate the virtual drive.
- Explicit change in the set of disks used by the virtual drive.

Disk Insertion Handling

When the following sequence of events takes place:

1. The LUN is created in one of the following ways:
   a. You specify the slot specifically by using a local disk reference
   b. The system selects the slot based on criteria specified by you
2. The LUN is successfully deployed, which means that a virtual drive is created, which uses the slot.
3. You remove a disk from the slot, possibly because the disk failed.
4. You insert a new working disk into the same slot.

The following scenarios are possible:

- Non-Redundant Virtual Drives, on page 10
- Redundant Virtual Drives with No Hot Spare Drives, on page 10
- Redundant Virtual Drives with Hot Spare Drives, on page 11
- Replacing Hot Spare Drives, on page 11
- Inserting Physical Drives into Unused Slots, on page 11

Non-Redundant Virtual Drives

For non-redundant virtual drives (RAID 0), when a physical drive is removed, the state of the virtual drive is **Inoperable**. When a new working drive is inserted, the new physical drive goes to an **Unconfigured Good** state.

For non-redundant virtual drives, there is no way to recover the virtual drive. You must delete the virtual drive and re-create it.

Redundant Virtual Drives with No Hot Spare Drives

For redundant virtual drives (RAID 1, RAID 5, RAID 6, RAID 10) with no hot spare drives assigned, virtual drive mismatch, virtual drive member missing, and local disk missing faults appear until you insert a working physical drive into the same slot from which the old physical drive was removed.

If the physical drive size is greater than or equal to that of the old drive, the storage controller automatically uses the new drive for the virtual drive. The new drive goes into the **Rebuilding** state. After rebuild is complete, the virtual drive goes back into the **Online** state.

Redundant Virtual Drives with Hot Spare Drives

For redundant virtual drives (RAID 1, RAID 5, RAID 6, RAID 10) with hot spare drives assigned, when a drive fails, or when you remove a drive, the dedicated hot spare drive, if available, goes into the **Rebuilding** state with the virtual drive in the **Degraded** state. After rebuilding is complete, that drive goes to the **Online** state.

Cisco UCSM raises a disk missing and virtual drive mismatch fault because although the virtual drive is operational, it does not match the physical configuration that Cisco UCSM expects.

if you insert a new disk in the slot with the disk missing, automatic copy back starts from the earlier hot spare disk to the newly inserted disk. After copy back, the hot spare disk is restored. In this state all faults are cleared.

If automatic copy back does not start, and the newly inserted disk remains in the **Unconfigured Good**, **JBOD**, or **Foreign Configuration** state, remove the new disk from the slot, reinsert the earlier hot spare disk into the slot, and import foreign configuration. This initiates the rebuilding process and the drive state becomes **Online**. Now, insert the new disk in the hot spare slot and mark it as hot spare to match it exactly with the information available in Cisco UCSM.

Replacing Hot Spare Drives

If a hot spare drive is replaced, the new hot spare drive will go to the **Unconfigured Good**, **Unconfigured Bad**, **JBOD**, or **Foreign Configuration** state.

Cisco UCSM will raise a virtual drive mismatch or virtual drive member mismatch fault because the hot spare drive is in a state different from the state configured in Cisco UCSM.

You must manually clear the fault. To do this, you must perform the following actions:

1. Clear the state on the newly inserted drive to **Unconfigured Good**.
2. Configure the newly inserted drive as a hot spare drive to match what is expected by Cisco UCSM.

Inserting Physical Drives into Unused Slots

If you insert new physical drives into unused slots, neither the storage controller nor Cisco UCSM will make use of the new drive even if the drive is in the **Unconfigured Good** state and there are virtual drives that are missing good physical drives.

The drive will simply go into the **Unconfigured Good** state. To make use of the new drive, you will need to modify or create LUNs to reference the newly inserted drive.

Virtual Drive Naming

When you use UCSM to create a virtual drive, UCSM assigns a unique ID that can be used to reliably identify the virtual drive for further operations. UCSM also provides the flexibility to provide a name to the virtual

drive at the time of service profile association. Any virtual drive without a service profile or a server reference is marked as an orphan virtual drive.

In addition to a unique ID, a name is assigned to the drive. Names can be assigned in two ways:

- When configuring a virtual drive, you can explicitly assign a name that can be referenced in storage profiles.
- If you have not preprovisioned a name for the virtual drive, UCSM generates a unique name for the virtual drive.

You can rename virtual drives that are not referenced by any service profile or server.

LUN Dereferencing

A LUN is dereferenced when it is no longer used by any service profile. This can occur as part of the following scenarios:

- The LUN is no longer referenced from the storage profile
- The storage profile is no longer referenced from the service profile
- The server is disassociated from the service profile
- The server is decommissioned

When the LUN is no longer referenced, but the server is still associated, re-association occurs.

When the service profile that contained the LUN is disassociated, the LUN state is changed to **Not in use**.

When the service profile that contained the LUN is deleted, the LUN state is changed to **Orphaned**.

Guidelines and Limitations

- Cisco UCS Manager does not support initiating the following storage profile functions. However, you can monitor them through Cisco UCS Manager after they are performed:
  - Virtual Drive Rebuild
  - Virtual Drive Consistency Check
  - Virtual Drive Initialization
  - Patrol Read
  - BBU Relearning
  - Locator LED
  - BBU Configuration
  - Destructive LUN modifications
  - Automatic LUN creation
  - Disk replacement with hot spares

- JBOD mode
- Additional disk selection qualifiers

- Cisco UCS Manager does not support a combination of SAS and SATA drives in storage configurations.
- Cisco UCS Manager Release 2.5 only supports a stripe size of 64k and more. Having a stripe size of less than 64k will result in failure when a service profile is associated.

Controller Constraints and Limitations

In Cisco UCS Manager Release 2.5, the storage controller allows 64 virtual drives per controller, and up to 4 virtual drives per server, of which up to 2 virtual drives are bootable.

Note Only the modular servers in Cisco UCSME-2814 compute cartridges include support for up to 4 virtual drives per server.

Configuring Storage Profiles

Configuring a Disk Group Policy

You can choose to configure a disk group policy through automatic or manual disk selection. Configuring a disk group involves the following:

1. Setting the RAID Level, on page 13
2. Automatically Configuring Disks in a Disk Group, on page 14 or Manually Configuring Disks in a Disk Group, on page 16
3. Configuring Virtual Drive Properties, on page 17

Setting the RAID Level

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org# **create disk-group-config-policy** *disk-group-name*
3. UCS-A /org/disk-group-config-policy* # **set raid-level** *raid-level*
4. UCS-A /org/disk-group-config-policy* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.

	Command or Action	Purpose
Step 2	UCS-A /org# **create disk-group-config-policy** *disk-group-name*	Creates a disk group configuration policy with the specified name and enters disk group configuration policy mode.
Step 3	UCS-A /org/disk-group-config-policy* # **set raid-level** *raid-level*	Specifies the RAID level for the disk group configuration policy. The RAID levels that you can specify are: • raid-0-striped • raid-1-mirrored • raid-10-mirrored-and-striped • raid-5-striped-parity • raid-6-striped-dual-parity
Step 4	UCS-A /org/disk-group-config-policy* # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to set the RAID level for a disk group configuration policy.

```
UCS-A# scope org
UCS-A /org # create disk-group-config-policy raid5policy
UCS-A /org/disk-group-config-policy* # set raid-level raid-5-striped-parity
UCS-A /org/disk-group-config-policy* # commit-buffer
```

What to do next

Automatically or manually configure disks as part of the disk group configuration policy.

Automatically Configuring Disks in a Disk Group

You can allow UCSM to automatically select and configure disks in a disk group.

When you create a disk group with RAID 1 policy and configure four disks for it, a RAID1E configuration is created internally by the storage controller.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org# **enter disk-group-config-policy** *disk-group-name*
3. UCS-A /org/disk-group-config-policy* # **enter disk-group-qual**
4. UCS-A /org/disk-group-config-policy/disk-group-qual* # **set drive-type** *drive-type*
5. UCS-A /org/disk-group-config-policy/disk-group-qual* # **set min-drive-size** *drive-size*
6. UCS-A /org/disk-group-config-policy/disk-group-qual* # **set num-ded-hot-spares** *hot-spare-num*
7. UCS-A /org/disk-group-config-policy/disk-group-qual* # **set num-drives** *drive-num*
8. UCS-A /org/disk-group-config-policy/disk-group-qual* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org# **enter disk-group-config-policy** *disk-group-name*	Enters disk group configuration policy mode for the specified disk group name.
Step 3	UCS-A /org/disk-group-config-policy* # **enter disk-group-qual**	Enters disk group qualification mode. In this mode, UCSM automatically configures disks as part of the specified disk group.
Step 4	UCS-A /org/disk-group-config-policy/disk-group-qual* # **set drive-type** *drive-type*	Specifies the drive type for the disk group. You can select: • SSD • Unspecified **Note** If you specify **Unspecified** as the type of drive, the first available drive is selected. After this drive is selected, subsequent drives will be of a compatible type. For example, if the first was SSD, all subsequent drives would be SSD.
Step 5	UCS-A /org/disk-group-config-policy/disk-group-qual* # **set min-drive-size** *drive-size*	Specifies the minimum drive size for the disk group. Only disks that match this criteria will be available for selection. The range for minimum drive size is from 0 to10240 GB. You can also set the minimum drive size as **Unspecified**. If you set the minimum drive size as **Unspecified**, drives of all sizes will be available for selection.
Step 6	UCS-A /org/disk-group-config-policy/disk-group-qual* # **set num-ded-hot-spares** *hot-spare-num*	Specifies the number of dedicated hot spares for the disk group. The range for dedicated hot spares is from 0 to 24 hot spares. You can also set the number of dedicated hot spares as **Unspecified**. If you set the number of dedicated hot spares as **Unspecified**, the hot spares will be selected according to the disk selection process.
Step 7	UCS-A /org/disk-group-config-policy/disk-group-qual* # **set num-drives** *drive-num*	Specifies the number of drives for the disk group. The range for drives is from 0 to 24 drives. You can also set the number of drives as **Unspecified**. If you set the number of drives as **Unspecified**, the number of drives will be selected according to the disk selection process.
Step 8	UCS-A /org/disk-group-config-policy/disk-group-qual* # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to automatically configure disks for a disk group configuration policy.

```
UCS-A# scope org
UCS-A /org # enter disk-group-config-policy raid5policy
UCS-A /org/disk-group-config-policy* # enter disk-group-qual
UCS-A /org/disk-group-config-policy/disk-group-qual* # set drive-type ssd
UCS-A /org/disk-group-config-policy/disk-group-qual* # set min-drive-size 1000
UCS-A /org/disk-group-config-policy/disk-group-qual* # set num-ded-hot-spares 2
UCS-A /org/disk-group-config-policy/disk-group-qual* # set num-drives 7
UCS-A /org/disk-group-config-policy/disk-group-qual* # commit-buffer
```

What to do next

Configure Virtual Drives.

Manually Configuring Disks in a Disk Group

You can manually configure disks for a disk group.

When you create a disk group with RAID 1 policy and configure four disks for it, a RAID 1E configuration is created internally by the storage controller.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org# **enter disk-group-config-policy** *disk-group-name*
3. UCS-A /org/disk-group-config-policy* # **create local-disk-config-ref** *slot-num*
4. UCS-A /org/disk-group-config-policy/local-disk-config-ref *# **set role** *role*
5. UCS-A /org/disk-group-config-policy/local-disk-config-ref *# **set span-id** *span-id*
6. UCS-A /org/disk-group-config-policy/local-disk-config-ref *# **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org# **enter disk-group-config-policy** *disk-group-name*	Enters disk group configuration policy mode for the specified disk group name.
Step 3	UCS-A /org/disk-group-config-policy* # **create local-disk-config-ref** *slot-num*	Creates a local disk configuration reference for the specified slot and enters local disk configuration reference mode.
Step 4	UCS-A /org/disk-group-config-policy/local-disk-config-ref *# **set role** *role*	Specifies the role of the local disk in the disk group. You can select: • ded-hot-spare: Dedicated hot spare • normal

	Command or Action	Purpose
Step 5	UCS-A /org/disk-group-config-policy/local-disk-config-ref *# **set span-id** *span-id*	Specifies the ID of the span group to which the disk belongs. Disks belonging to a single span group can be treated as a single disk with a larger size. The values range from 0 to 8. You can also set the Span ID as **Unspecified** when spanning information is not required. **Note** In Cisco UCS Release 2.5, you can have a maximum of 4 span groups.
Step 6	UCS-A /org/disk-group-config-policy/local-disk-config-ref *# **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to manually configure disks for a disk group configuration policy.

```
UCS-A# scope org
UCS-A /org # enter disk-group-config-policy raid5policy
UCS-A /org/disk-group-config-policy* # create local-disk-config-ref 1
UCS-A /org/disk-group-config-policy/local-disk-config-ref *# set role ded-hot-spare
UCS-A /org/disk-group-config-policy/local-disk-config-ref* # set span-id 1
UCS-A /org/disk-group-config-policy/local-disk-config-ref *# commit-buffer
```

What to do next

Configure Virtual Drive Properties.

Configuring Virtual Drive Properties

All virtual drives in a disk group must be managed by using a single disk group policy.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org# **scope disk-group-config-policy** *disk-group-name*
3. UCS-A /org/disk-group-config-policy* # **create virtual-drive-def**
4. UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set access-policy** *policy-type*
5. UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set drive-cache** *state*
6. UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set io-policy** *policy-type*
7. UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set read-policy** *policy-type*
8. UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set strip-size** *strip-size*
9. UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set write-cache-policy** *policy-type*
10. UCS-A /org/disk-group-config-policy/virtual-drive-def* # **commit-buffer**
11. UCS-A /org/disk-group-config-policy/virtual-drive-def* # **show**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org# **scope disk-group-config-policy** *disk-group-name*	Enters disk group configuration policy mode for the specified disk group name.
Step 3	UCS-A /org/disk-group-config-policy* # **create virtual-drive-def**	Creates a virtual drive definition and enters the virtual drive definition mode.
Step 4	UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set access-policy** *policy-type*	Specifies the access policy. This can be one of the following: • blocked • platform-default • read-only: • read-write
Step 5	UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set drive-cache** *state*	Specifies the state of the drive cache. This can be one of the following: • enable • disable • no-change • platform-default **Important** In Cisco UCS Release 2.5, the drive cache state cannot be changed. It will remain as **platform-default**, irrespective of the drive cache state that you select.
Step 6	UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set io-policy** *policy-type*	Specifies the I/O policy. This can be one of the following: • cached • direct • platform-default
Step 7	UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set read-policy** *policy-type*	Specifies the read policy. This can be one of the following: • normal • platform-default • read-ahead

	Command or Action	Purpose
Step 8	UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set strip-size** *strip-size*	Specifies the strip size. This can be one of the following: • 64 KB • 128 KB • 256 KB • 512 KB • 1024 KB • platform-default
Step 9	UCS-A /org/disk-group-config-policy/virtual-drive-def* # **set write-cache-policy** *policy-type*	Specifies the write-cache-policy. This can be one of the following: • always-write-back • platform-default • write-back-good-bbu • write-through
Step 10	UCS-A /org/disk-group-config-policy/virtual-drive-def* # **commit-buffer**	Commits the transaction to the system configuration.
Step 11	UCS-A /org/disk-group-config-policy/virtual-drive-def* # **show**	Displays the configured virtual drive properties.

Example

This example shows how to configure virtual disk properties:

```
UCS-A# scope org
UCS-A /org # scope disk-group-config-policy raid0policy
UCS-A /org/disk-group-config-policy # create virtual-drive-def
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set access-policy read-write
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set drive-cache enable
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set io-policy cached
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set read-policy normal
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set strip-size 1024
UCS-A /org/disk-group-config-policy/virtual-drive-def* # set write-cache-policy write-through
UCS-A /org/disk-group-config-policy/virtual-drive-def* # commit-buffer
UCS-A /org/disk-group-config-policy/virtual-drive-def # show

Virtual Drive Def:
    Strip Size (KB): 1024KB
    Access Policy: Read Write
    Read Policy: Normal
    Configured Write Cache Policy: Write Through
    IO Policy: Cached
    Drive Cache: Enable
UCS-A /org/disk-group-config-policy/virtual-drive-def #
```

What to do next

Create a Storage Profile

Creating a Storage Profile

You can create a storage profile at the org level and at the service-profile level.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create storage-profile** *storage-profile-name*
3. UCS-A /org/storage-profile* # **commit-buffer**
4. (Optional) UCS-A /org* # **enter service-profile** *service-profile-name*
5. (Optional) UCS-A /org/service-profile* # **create storage-profile-def**
6. UCS-A /org/service-profile/storage-profile-def* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **create storage-profile** *storage-profile-name*	Creates a storage profile with the specified name at the org level and enters storage-profile configuration mode.
Step 3	UCS-A /org/storage-profile* # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	(Optional) UCS-A /org* # **enter service-profile** *service-profile-name*	Enters the specified service profile.
Step 5	(Optional) UCS-A /org/service-profile* # **create storage-profile-def**	Creates a storage profile at the service-profile level.
Step 6	UCS-A /org/service-profile/storage-profile-def* # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to create a storage profile at the org level.

```
UCS-A# scope org
UCS-A /org # create storage-profile stp2
UCS-A /org/storage-profile* # commit-buffer
```

This example shows how to create a storage profile at the service-profile level.

```
UCS-A# scope org
UCS-A /org* # enter service-profile sp1
UCS-A /org/service-profile* # create storage-profile-def
UCS-A /org/service-profile/storage-profile-def* # commit-buffer
```

What to do next

Create Local LUNs

Deleting a Storage Profile

You can delete a storage profile that was created at the org level or at the service-profile level.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **delete storage-profile** *storage-profile-name*
3. (Optional) UCS-A /org # **scope service-profile** *service-profile-name*
4. (Optional) UCS-A /org/service-profile # **delete storage-profile-def**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **delete storage-profile** *storage-profile-name*	Deletes the storage profile with the specified name at the org level.
Step 3	(Optional) UCS-A /org # **scope service-profile** *service-profile-name*	Enters the specified service profile.
Step 4	(Optional) UCS-A /org/service-profile # **delete storage-profile-def**	Deletes the dedicated storage profile at the service-profile level.

Example

This example shows how to delete a storage profile at the org level.

```
UCS-A # scope org
UCS-A /org #  delete storage-profile stor1
```

This example shows how to delete a storage profile at the service-profile level.

```
UCS-A # scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile # delete storage-profile-def
```

Creating Local LUNs

You can create local LUNs within a storage profile at the org level and within a dedicated storage profile at the service-profile level.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **enter storage-profile** *storage-profile-name*
3. UCS-A /org/storage-profile* # **create local-lun** *lun-name*
4. UCS-A /org/storage-profile/local-lun* # **set auto-deploy** {**auto-deploy** | **no-auto-deploy**}
5. UCS-A /org/storage-profile/local-lun* # **set disk-policy-name** *disk-policy-name*
6. UCS-A /org/storage-profile/local-lun* # **set order** *order-num*
7. UCS-A /org/storage-profile/local-lun* # **set size** *size*
8. UCS-A /org/storage-profile/local-lun* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **enter storage-profile** *storage-profile-name*	Enters storage-profile mode for the specified storage profile.
Step 3	UCS-A /org/storage-profile* # **create local-lun** *lun-name*	Creates a local LUN with the specified name.
Step 4	UCS-A /org/storage-profile/local-lun* # **set auto-deploy** {**auto-deploy** | **no-auto-deploy**}	Specifies whether the LUN should be auto-deployed or not.
Step 5	UCS-A /org/storage-profile/local-lun* # **set disk-policy-name** *disk-policy-name*	Specifies the name of the disk policy name for this LUN.
Step 6	UCS-A /org/storage-profile/local-lun* # **set order** *order-num*	Specifies the order of this LUN. The order can range from 1 to 64. You can also specify that the order should be **lowest-available** for the system to automatically assign the lowest available order to the LUN. Multiple LUNs referenced by a storage profile must have unique names and unique orders.
Step 7	UCS-A /org/storage-profile/local-lun* # **set size** *size*	Specifies the size of this LUN in GB. The size can range from 1 GB to 10240 GB. **Note** You do not need to specify a LUN size while claiming an orphaned LUN.
Step 8	UCS-A /org/storage-profile/local-lun* # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to configure a local LUN within a storage profile at the org level.

```
UCS-A# scope org
UCS-A /org # enter storage-profile stp2
UCS-A /org/storage-profile* # create local-lun lun2
UCS-A /org/storage-profile/local-lun* # set disk-policy-name dpn2
UCS-A /org/storage-profile/local-lun* # set order 2
UCS-A /org/storage-profile/local-lun* # set size 1000
```

```
UCS-A /org/storage-profile/local-lun* # commit-buffer
```

This example shows how to configure a local LUN within a dedicated storage profile at the service-profile level.

```
UCS-A# scope org
UCS-A /org* # enter service-profile stp1
UCS-A /org/service-profile* # enter storage-profile-def
UCS-A /org/service-profile/storage-profile-def # create local-lun lun1
UCS-A /org/service-profile/storage-profile-def/local-lun* # set disk-policy-name dpn1
UCS-A /org/service-profile/storage-profile-def/local-lun* # set order 1
UCS-A /org/service-profile/storage-profile-def/local-lun* # set size 1000
UCS-A /org/service-profile/storage-profile-def/local-lun* # commit-buffer
```

What to do next

Associate a Storage Profile with a Service Profile

Reordering Local LUNs In a Storage Profile

You can use the **set order** command to change the local LUN visibility order to the server. This operation will reboot the server.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **enter storage-profile** *storage-profile-name*
3. UCS-A /org/storage-profile # **enter local-lun** *lun-name*
4. UCS-A /org/storage-profile/local-lun* # **set disk-policy-name** *disk-policy-name*
5. UCS-A /org/storage-profile/local-lun # **set order** *order-num*
6. UCS-A /org/storage-profile/local-lun* # **set size** *size*
7. UCS-A /org/storage-profile/local-lun # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **enter storage-profile** *storage-profile-name*	Enters storage-profile mode for the specified storage profile.
Step 3	UCS-A /org/storage-profile # **enter local-lun** *lun-name*	Enters local-lun mode for the specified local LUN.
Step 4	UCS-A /org/storage-profile/local-lun* # **set disk-policy-name** *disk-policy-name*	Specifies the name of the disk policy name for this LUN.
Step 5	UCS-A /org/storage-profile/local-lun # **set order** *order-num*	Specifies the order of this LUN. The order can range from 1 to 64. You can also specify that the order should be **lowest-available** for the system to automatically assign the lowest available order to the LUN.

	Command or Action	Purpose
Step 6	UCS-A /org/storage-profile/local-lun* # **set size** *size*	Specifies the size of this LUN in GB. The size can range from 1 GB to 10240 GB.
Step 7	UCS-A /org/storage-profile/local-lun # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to reorder a local LUN within a storage profile at the org level.

```
UCS-A# scope org
UCS-A /org # enter storage-profile stp1
UCS-A /org/storage-profile* # enter local-lun lun1
UCS-A /org/storage-profile/local-lun* # set disk-policy-name dpn1
UCS-A /org/storage-profile/local-lun* # set order 1
UCS-A /org/storage-profile/local-lun* # set size 10
UCS-A /org/storage-profile/local-lun* # exit
UCS-A /org/storage-profile* # enter local-lun lun2
UCS-A /org/storage-profile/local-lun* # set disk-policy-name dpn2
UCS-A /org/storage-profile/local-lun* # set order 2
UCS-A /org/storage-profile/local-lun* # set size 10
UCS-A /org/storage-profile/local-lun* # exit
UCS-A /org/storage-profile* # commit-buffer

UCS-A /org/storage-profile # show configuration

 enter storage-profile stp1
     enter local-lun lun1
         set auto-deploy auto-deploy
         set disk-policy-name dpn1
         set order 1
         set size 10
     exit
     enter local-lun lun2
         set auto-deploy auto-deploy
         set disk-policy-name dpn2
         set order 2
         set size 10
     exit
     set descr ""
 exit

UCS-A /org/storage-profile # enter local-lun lun1
UCS-A /org/storage-profile/local-lun # set order 2
UCS-A /org/storage-profile/local-lun* # exit
UCS-A /org/storage-profile* # enter local-lun lun2
UCS-A /org/storage-profile/local-lun* # set order 1
UCS-A /org/storage-profile/local-lun* # exit
UCS-A /org/storage-profile* # commit-buffer
UCS-A /org/storage-profile # show configuration

 enter storage-profile stp1
     enter local-lun lun1
         set auto-deploy auto-deploy
         set disk-policy-name dpn1
         set order 2
         set size 10
     exit
     enter local-lun lun2
```

```
        set auto-deploy auto-deploy
        set disk-policy-name dpn2
        set order 1
        set size 10
    exit
    set descr ""
exit
```

Deleting Local LUNs In a Storage Profile

When a LUN is deleted, the corresponding virtual drive is marked as orphan after the virtual drive reference is removed from the server.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **enter storage-profile** *storage-profile-name*
3. (Optional) UCS-A /org/storage-profile* # **show local-lun**
4. UCS-A /org/storage-profile* # **delete local-lun** *lun-name*
5. UCS-A /org/storage-profile* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **enter storage-profile** *storage-profile-name*	Enters storage-profile mode for the specified storage profile.
Step 3	(Optional) UCS-A /org/storage-profile* # **show local-lun**	Displays the local LUNs in the specified storage profile.
Step 4	UCS-A /org/storage-profile* # **delete local-lun** *lun-name*	Deletes the specified LUN.
Step 5	UCS-A /org/storage-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to delete a LUN in a storage profile.

```
UCS-A # scope org
UCS-A /org # enter storage-profile stp2
UCS-A /org/storage-profile # show local-lun


Local SCSI LUN:

    LUN Name   Size (GB)   Order            Disk Policy Name Auto Deploy

    ---------- ----------- ---------------- ---------------- -----------

    luna       1           2                raid0            Auto Deploy

    lunb       1           1                raid0            Auto Deploy
```

```
UCS-A /org/storage-profile # delete local-lun luna
UCS-A /org/storage-profile* # commit-buffer
UCS-A /org/storage-profile* # show local-lun


Local SCSI LUN:

    LUN Name   Size (GB)   Order            Disk Policy Name Auto Deploy

    ---------- ----------- ---------------- ---------------- -----------

    lunb       1           1                raid0            Auto Deploy
```

Associating a Storage Profile with a Service Profile

A storage profile created under org can be referred by multiple service profiles, and a name reference in service profile is needed to associate the storage profile with a service profile.

Important Storage profiles can be defined under org and under service profile (dedicated). Hence, a service profile inherits local LUNs from both possible storage profiles. A service profile can have a maximum of two such local LUNs.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **scope service-profile** *service-profile-name*
3. UCS-A /org/service-profile # **set storage-profile-name** *storage-profile-name*
4. UCS-A /org/service-profile* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **scope service-profile** *service-profile-name*	Enters the specified service profile mode.
Step 3	UCS-A /org/service-profile # **set storage-profile-name** *storage-profile-name*	Associates the specified storage profile with the service profile. **Note** To dissociate the service profile from a storage profile, use the **set storage-profile-name** command and specify `""` as the storage profile name.
Step 4	UCS-A /org/service-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to associate a storage profile with a service profile.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile # set storage-profile-name stp2
```

This example shows how to dissociate a service profile from a storage profile.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile # set storage-profile-name ""
```

Displaying Details of All Local LUNs Inherited By a Service Profile

Storage profiles can be defined under org and as a dedicated storage profile under service profile. Thus, a service profile inherits local LUNs from both possible storage profiles. It can have a maximum of 2 such local LUNs. You can display the details of all local LUNs inherited by a service profile by using the following command:

SUMMARY STEPS

1. UCS-A /org/service-profile # **show local-lun-ref**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A /org/service-profile # **show local-lun-ref**	Displays the following detailed information about all the local LUNs inherited by the specified service profile: • **Name**—LUN name in the storage profile. • **Admin State**—Specifies whether a local LUN should be deployed or not. Admin state can be **Online** or **Undeployed**. When the local LUN is being referenced by a service profile, if the auto-deploy status is **no-auto-deploy** then the admin state will be **Undeployed**, else it will be **Online**. After the local LUN is referenced by a service profile, any change made to this local LUN's auto-deploy status is not reflected in the admin state of the LUN inherited by the service profile. • **RAID Level**—Summary of the RAID level of the disk group used. • **Provisioned Size (GB)**—Size, in GB, of the LUN specified in the storage profile.

	Command or Action	Purpose

		• **Assigned Size (MB)**—Size, in MB, assigned by UCSM.
		• **Config State**—State of LUN configuration. The states can be one of the following:
		  • **Applying**—Admin state is online, the LUN is associated with a server, and the virtual drive is being created.
		  • **Applied**—Admin state is online, the LUN is associated with a server, and the virtual drive is created.
		  • **Apply Failed**—Admin stage is online, the LUN is associated with a server, but the virtual drive creation failed.
		  • **Not Applied**—The LUN is not associated with a server, or the LUN is associated with a service profile, but admin state is undeployed.
		  • **Not In Use**—Service profile is using the virtual drive, but the virtual drive is not associated with a server.
		• **Reference LUN**—The preprovisioned virtual drive name, or UCSM-generated virtual drive name.
		• **Deploy Name**—The virtual drive name after deployment.
		• **ID**—Virtual drive ID.
		• **Drive State**—State of the virtual drive. The states are:
		  • **Unknown**
		  • **Optimal**
		  • **Degraded**
		  • **Inoperable**
		  • **Partially Degraded**

Example

```
UCS-A /org/service-profile # show local-lun-ref

Local LUN Ref:

    Profile LUN Name Admin State RAID Level             Provisioned Size (GB)  Assigned
```

```
Size (MB)   Config State Referenced Lun Deploy Name ID          Drive State

    ---------------- ----------- ---------------------- ----------------------
-------------------- ------------ -------------- ----------- ----------- -----------

    luna             Online      RAID 0 Striped         1
     1024 Applied      luna-1         luna-1      1003        Optimal

    lunb             Online      RAID 0 Striped         1
     1024 Applied      lunb-1         lunb-1      1004        Optimal

UCS-A /org/service-profile #



Local LUN Ref:
    Name             Admin State RAID Level             Provisioned Size (GB)  Assigned
Size (MB)   Config State Referenced Lun Deploy Name ID          Drive State
    ---------------- ----------- ---------------------- ----------------------
-------------------- ------------ -------------- ----------- ----------- -----------
    lun111           Online      RAID 0 Striped         30                     30720
          Applied      lun111-1       lun111-1    1001        Optimal
    lun201           Online      Unspecified            1                      0
          Not Applied
```

Displaying Detailed Information About LUNs Used By a Modular Server

SUMMARY STEPS

1. UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*
2. UCS-A /chassis/cartridge/server # **scope adapter** *adapter-id*
3. UCS-A /chassis/cartridge/server/adapter # **scope host-scsi** *host-scsi-id*
4. UCS-A /chassis/cartridge/server/adapter/host-scsi # **show detail expand**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*	Enters cartridge server mode for the specified modular server in the specified chassis and cartridge.
Step 2	UCS-A /chassis/cartridge/server # **scope adapter** *adapter-id*	Enters adapter mode for the specified adapter.
Step 3	UCS-A /chassis/cartridge/server/adapter # **scope host-scsi** *host-scsi-id*	Enters host-scsi mode for the specified host-scsi.
Step 4	UCS-A /chassis/cartridge/server/adapter/host-scsi # **show detail expand**	Displays detailed information about the LUNs that are being used by the specified modular server.

Example

This example shows how to display detailed information about the LUNs that are being used by modular server 1/1/1.

```
UCS-A # scope server 1/1/1
UCS-A /chassis/cartridge/server # scope adapter 1
UCS-A /chassis/cartridge/server/adapter # scope host-scsi 1
UCS-A /chassis/cartridge/server/adapter/host-scsi # show detail expand

Host Scsi Interface:
    ID: 1

    Local Lun Interface:
        Lun Name: lun-uefi
        Lun Id: 1008
        Lun Order: 1
        Oper Lun Id: 7
        Boot Dev: Disabled
        Lun Dn: sys/chassis-1/storage-SAS-1/vd-1008
```

Importing Foreign Configurations for a RAID Controller

SUMMARY STEPS

1. UCS-A# **scope chassis** *chassis-num*
2. UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}
3. UCS-A /chassis/raid-controller # **set admin-state import-foreign-configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}	Enters RAID controller chassis mode.
Step 3	UCS-A /chassis/raid-controller # **set admin-state import-foreign-configuration**	Allows import of configurations from local disks that are in the **Foreign Configuration** state.

Example

This example shows how to import foreign configurations from local disks that are in the **Foreign Configuration** state:

```
UCS-A# scope chassis 1
UCS-A /chassis # scope raid-controller 1 sas
UCS-A /chassis/raid-controller # set admin-state import-foreign-configuration
UCS-A /chassis/raid-controller* #
```

Configuring Local Disk Operations

SUMMARY STEPS

1. UCS-A# **scope chassis** *chassis-num*
2. UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}
3. UCS-A /chassis/raid-controller # **scope local-disk** *local-disk-id*
4. UCS-A /chassis/raid-controller/local-disk # **set admin-state** {**clear-foreign-configuration** | **dedicated-hot-spare** [*admin-vd-id*] | **prepare-for-removal** | **remove-hot-spare** | **unconfigured-good** | **undo-prepare-for-removal**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}	Enters RAID controller chassis mode.
Step 3	UCS-A /chassis/raid-controller # **scope local-disk** *local-disk-id*	Enters local disk configuration mode.
Step 4	UCS-A /chassis/raid-controller/local-disk # **set admin-state** {**clear-foreign-configuration** | **dedicated-hot-spare** [*admin-vd-id*] | **prepare-for-removal** | **remove-hot-spare** | **unconfigured-good** | **undo-prepare-for-removal**}	Configures the local disk to one of the following states: • **clear-foreign-configuration**—Clears any foreign configuration that exists in a local disk when it is introduced into a new configuration. • **dedicated-hot-spare**—Specifies the local disk as a dedicated hot spare. The admin virtual drive ID that you can assign ranges from 0 to 4294967295. • **prepare-for-removal**—Specifies that the local disk is marked for removal from the chassis. • **remove-hot-spare**—Specifies that the local disk is no longer a hot spare. Use this only to clear any mismatch faults. • **unconfigured-good**—Specifies that the local disk can be configured. • **undo-prepare-for-removal**—Specifies that the local disk is no longer marked for removal from the chassis.

Example

This example shows how to clear any foreign configuration from a local disk:

```
UCS-A /chassis/raid-controller/local-disk # set admin-state clear-foreign-configuration
```

This example shows how to specify a local disk as a dedicated hot spare:

```
UCS-A /chassis/raid-controller/local-disk* # set admin-state dedicated-hot-spare 1001
```

This example shows how to specify that a local disk is marked for removal from the chassis:

```
UCS-A /chassis/raid-controller/local-disk* # set admin-state prepare-for-removal
```

This example shows how to specify that a local disk is marked for removal as a hot spare:

```
UCS-A /chassis/raid-controller/local-disk* # set admin-state remove-hot-spare
```

This example shows how to specify that a local disk is working, but is unconfigured for use:

```
UCS-A /chassis/raid-controller/local-disk* # set admin-state unconfigured-good
```

This example shows how to specify that a local disk is no longer marked for removal from the chassis:

```
UCS-A /chassis/raid-controller/local-disk* # set admin-state undo-prepare-for-removal
```

Configuring Virtual Drive Operations

The following operations can be performed only on orphaned virtual drives:

- Delete an orphaned virtual drive
- Rename an orphaned virtual drive

Deleting an Orphaned Virtual Drive

SUMMARY STEPS

1. UCS-A# **scope chassis** *chassis-num*
2. UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}
3. (Optional) UCS-A /chassis/raid-controller # **delete virtual-drive id** *virtual-drive-id*
4. (Optional) UCS-A /chassis/raid-controller # **delete virtual-drive name** *virtual-drive-name*
5. (Optional) UCS-A /chassis/raid-controller # **scope virtual-drive** *virtual-drive-id*
6. UCS-A /chassis/raid-controller/virtual-drive # **set admin-state delete**
7. UCS-A /chassis/raid-controller/virtual-drive # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}	Enters RAID controller chassis mode.
Step 3	(Optional) UCS-A /chassis/raid-controller # **delete virtual-drive id** *virtual-drive-id*	Deletes the orphaned virtual drive with the specified virtual drive ID.
Step 4	(Optional) UCS-A /chassis/raid-controller # **delete virtual-drive name** *virtual-drive-name*	Deletes the orphaned virtual drive with the specified virtual drive name.

	Command or Action	Purpose
Step 5	(Optional) UCS-A /chassis/raid-controller # **scope virtual-drive** *virtual-drive-id*	Enters virtual drive mode for the specified orphaned virtual drive.
Step 6	UCS-A /chassis/raid-controller/virtual-drive # **set admin-state delete**	Deletes the orphaned virtual drive.
Step 7	UCS-A /chassis/raid-controller/virtual-drive # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to delete an orphan virtual drive by specifying the virtual drive ID.

```
UCS-A# scope chassis 1
UCS-A /chassis # scope raid-controller 1 sas
UCS-A /chassis/raid-controller # show virtual-drive

Virtual Drive:
    ID: 1001
    Name: lun111-1
    Block Size: 512
    Blocks: 62914560
    Size (MB): 30720
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 0
    Change Qualifier: No Change
    Config State: Applied
    Deploy Action: No Action

    ID: 1002
    Name: luna-1
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 1
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

    ID: 1003
    Name: lunb-1
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 2
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

    ID: 1004
    Name: lunb-2
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
```

```
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 3
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

    ID: 1005
    Name: luna-2
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 4
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

...

UCS-A /chassis/raid-controller # delete virtual-drive id 1002
Warning: When committed, the virtual drive will be deleted, which may result in data loss.

UCS-A /chassis/raid-controller # commit-buffer
```

This example shows how to delete an orphan virtual drive by specifying the virtual drive name.

```
UCS-A# scope chassis 1
UCS-A /chassis # scope raid-controller 1 sas
UCS-A /chassis/raid-controller # show virtual-drive

Virtual Drive:
    ID: 1001
    Name: lun111-1
    Block Size: 512
    Blocks: 62914560
    Size (MB): 30720
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 0
    Change Qualifier: No Change
    Config State: Applied
    Deploy Action: No Action

    ID: 1003
    Name: lunb-1
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 2
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

    ID: 1004
    Name: lunb-2
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 3
```

```
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

    ID: 1005
    Name: luna-2
    Block Size: 512
    Blocks: 2097152
    Size (MB): 1024
    Operability: Operable
    Presence: Equipped
    Oper Device ID: 4
    Change Qualifier: No Change
    Config State: Orphaned
    Deploy Action: No Action

...

UCS-A /chassis/raid-controller # delete virtual-drive name lunb-1
Warning: When committed, the virtual drive will be deleted, which may result in data loss.

UCS-A /chassis/raid-controller # commit-buffer
```

This example shows how to delete an orphan virtual drive by setting the admin-state.

```
UCS-A# scope chassis 1
UCS-A /chassis # scope raid-controller 1 sas
UCS-A /chassis/raid-controller # scope virtual-drive 1004
UCS-A /chassis/raid-controller/virtual-drive # set admin-state delete

Warning: When committed, the virtual drive will be deleted, which may result in data loss.

UCS-A /chassis/raid-controller/virtual-drive # commit-buffer
```

Renaming an Orphaned Virtual Drive

SUMMARY STEPS

1. UCS-A# **scope chassis** *chassis-num*
2. UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}
3. UCS-A /chassis/raid-controller # **scope virtual-drive** *virtual-drive-id*
4. UCS-A /chassis/raid-controller/virtual-drive # **set name** *virtual-drive-name*
5. UCS-A /chassis/raid-controller/virtual-drive # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}	Enters RAID controller chassis mode.
Step 3	UCS-A /chassis/raid-controller # **scope virtual-drive** *virtual-drive-id*	Enters virtual drive mode for the specified virtual drive.
Step 4	UCS-A /chassis/raid-controller/virtual-drive # **set name** *virtual-drive-name*	Specifies a name for the orphan virtual drive.

	Command or Action	Purpose
Step 5	UCS-A /chassis/raid-controller/virtual-drive # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to specify a name for an orphan virtual drive.

```
UCS-A /chassis # scope raid-controller 1 sas
UCS-A /chassis/raid-controller # scope virtual-drive 1060
UCS-A /chassis/raid-controller/virtual-drive* # set name vd1
UCS-A /chassis/raid-controller/virtual-drive* # commit-buffer
```

Configuring the Boot Policy for a Local LUN

Note In Cisco UCS Manager Release 2.5, you cannot configure JBOD as a boot device.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **scope boot-policy** *policy-name*
3. UCS-A /org/boot-policy # **create storage**
4. UCS-A /org/boot-policy/storage # **create local**
5. UCS-A /org/boot-policy/storage/local/ # **create local-lun**
6. UCS-A /org/boot-policy/storage/local/local-lun # **create local-lun-image-path** {**primary** | **secondary**}
7. UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path # **set lunname** *lun_name*
8. UCS-A /org/boot-policy/storage/local/*local-storage-device* # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 2	UCS-A /org # **scope boot-policy** *policy-name*	Enters organization boot policy mode for the specified boot policy.
Step 3	UCS-A /org/boot-policy # **create storage**	Creates a storage boot for the boot policy and enters organization boot policy storage mode.
Step 4	UCS-A /org/boot-policy/storage # **create local**	Creates a local storage location and enters the boot policy local storage mode.
Step 5	UCS-A /org/boot-policy/storage/local/ # **create local-lun**	Specifies a local hard disk drive as the local storage.

	Command or Action	Purpose
Step 6	UCS-A /org/boot-policy/storage/local/local-lun # **create local-lun-image-path** {**primary** | **secondary**}	Specifies the boot order for the LUN that you specify.
Step 7	UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path # **set lunname** *lun_name*	Specifies the name of the LUN that you want to boot from.
Step 8	UCS-A /org/boot-policy/storage/local/*local-storage-device* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a boot policy named lab1-boot-policy, create a local hard disk drive boot for the policy, specify a boot order and a LUN to boot from, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy lab1-boot-policy
UCS-A /org/boot-policy* # create storage
UCS-A /org/boot-policy/storage* # create local
UCS-A /org/boot-policy/storage/local* # create local-lun
UCS-A /org/boot-policy/storage/local/local-lun # create local-lun-image-path primary
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path # set lunname luna
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path # commit-buffer
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path #
```

What to do next

Include the boot policy in a service profile and template.

Local LUN Operations in a Service Profile

Although a service profile is derived from a service profile template, the following operations can be performed for each local LUN at the individual service profile level:

- Preprovisioning a LUN Name or Claiming an Orphan LUN, on page 38
- Deploying and Undeploying a LUN, on page 38
- Renaming a Service Profile Referenced LUN, on page 39

Note Preprovisioning a LUN name, claiming an orphan LUN, and deploying or undeploying a LUN result in server reboot.

Preprovisioning a LUN Name or Claiming an Orphan LUN

You can preprovision a LUN name or claim an orphan LUN by using the **set ref-name** command. Preprovisioning a LUN name or claiming an orphan LUN can be done only when the admin state of the LUN is **Undeployed**.

Important This operation will reboot the server.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org# **scope service-profile** *service-profile-name*
3. UCS-A /org/service-profile# **enter local-lun-ref** *lun-name*
4. UCS-A /org/service-profile/local-lun-ref# **set ref-name** *ref-lun-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org# **scope service-profile** *service-profile-name*	Enters the specified service profile mode.
Step 3	UCS-A /org/service-profile# **enter local-lun-ref** *lun-name*	Enters the specified LUN.
Step 4	UCS-A /org/service-profile/local-lun-ref# **set ref-name** *ref-lun-name*	Sets the referenced LUN name. If this LUN name exists and the LUN is orphaned, its is claimed by the service profile. If this LUN does not exist, a new LUN is created with the specified name.

- If the LUN exists and is not orphaned, a configuration failure occurs.
- If a LUN is already referred to and the ref-name is changed, it will release the old LUN and will claim or create a LUN with the ref-name. The old LUN is marked as an orphan after the LUN reference is removed from the server.

Example

This examples shows how to preprovision a LUN name.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile* # enter local-lun-ref lun1
UCS-A /org/service-profile/local-lun-ref* # set ref-name lun2
```

Deploying and Undeploying a LUN

You can deploy or undeploy a LUN by using the **admin-state** command. If the admin state of a local LUN is **Undeployed**, the reference of that LUN is removed and the LUN is not deployed.

Important This operation will reboot the server.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org# **scope service-profile** *service-profile-name*
3. UCS-A /org/service-profile# **enter local-lun-ref** *lun-name*
4. UCS-A /org/service-profile/local-lun-ref# **set admin-state** {**online** | **undeployed**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org# **scope service-profile** *service-profile-name*	Enters the specified service profile mode.
Step 3	UCS-A /org/service-profile# **enter local-lun-ref** *lun-name*	Enters the specified LUN.
Step 4	UCS-A /org/service-profile/local-lun-ref# **set admin-state** {**online** | **undeployed**}	Sets the admin state of the specified LUN to **online** or **undeployed**. If a LUN is already referred to and the admin state is set to **undeployed**, it will release the old LUN. The old LUN is marked as orphan after the LUN reference is removed from the server.

Example

This examples shows how to deploy a LUN.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile* # enter local-lun-ref lun1
UCS-A /org/service-profile/local-lun-ref* # set admin-state online
```

This examples shows how to undeploy a LUN.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile* # enter local-lun-ref lun1
UCS-A /org/service-profile/local-lun-ref* # set admin-state undeployed
```

Renaming a Service Profile Referenced LUN

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*

2. UCS-A /org# **scope service-profile** *service-profile-name*
3. UCS-A /org/service-profile# **enter local-lun-ref** *lun-name*
4. UCS-A /org/service-profile/local-lun-ref# **set name**

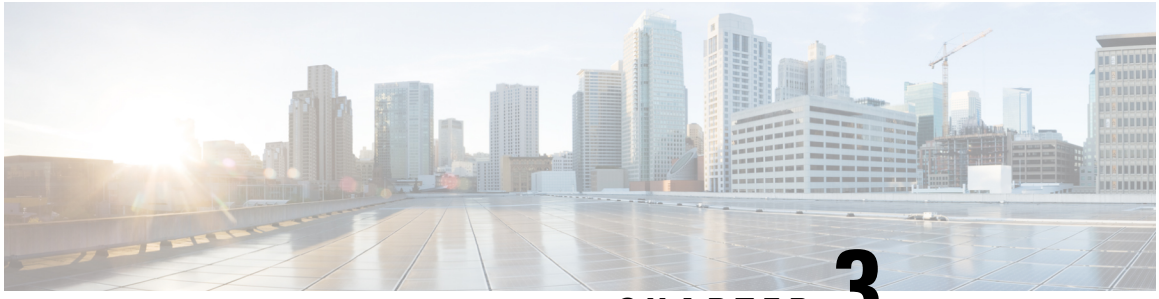
DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org# **scope service-profile** *service-profile-name*	Enters the specified service profile mode.
Step 3	UCS-A /org/service-profile# **enter local-lun-ref** *lun-name*	Enters the specified LUN.
Step 4	UCS-A /org/service-profile/local-lun-ref# **set name**	Renames the referenced LUN.

Example

This examples shows how to rename a LUN referenced by a service profile.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile* # enter local-lun-ref lun1
UCS-A /org/service-profile/local-lun-ref* # set name lun11
```



CHAPTER 3

Managing the Modular Chassis

This part contains the following chapters:

The Cisco UCSME-4308 Modular Chassis

The Cisco UCSME-4308 modular chassis consists of the following main components:

- Compute cartridges that host the modular servers.
- A shared adapter that uses system link technology to manage all the traffic that flows between hosts and the management entities in the chassis.
- A storage controller and backplane SSDs.
- PSUs, fans, and a Chassis Management Controller.

You can manage and monitor all these components through Cisco UCS Manager.

Note When Cisco UCS Manager and Cisco CMC run different versions of firmware, for example, when Cisco UCS Manager runs Release 2.5(1) and Cisco CMC runs Release 2.5(2), or when Cisco UCS Manager runs Release 2.5(2) and Cisco CMC runs Release 2.5(1), the storage controller and disks do not appear in the Cisco UCSME-4308 modular chassis. The storage controller and disks are automatically discovered when you update the Cisco CMC firmware version to match the Cisco UCS Manager firmware version.

Guidelines for Removing and Decommissioning Chassis

Consider the following guidelines when deciding whether to remove or decommission a chassis using Cisco UCS Manager:

Decommissioning a Chassis

Decommissioning is performed when a chassis is physically present and connected but you want to temporarily remove it from the Cisco UCS Manager configuration. Because it is expected that a decommissioned chassis will be eventually recommissioned, a portion of the chassis' information is retained by Cisco UCS Manager for future use.

Removing a Chassis

Removing is performed when you physically remove a chassis from the system. Once the physical removal of the chassis is completed, the configuration for that chassis can be removed in Cisco UCS Manager.

Note You cannot remove a chassis from Cisco UCS Manager if it is physically present and connected.

If you need to add a removed chassis back to the configuration, it must be reconnected and then rediscovered. During rediscovery Cisco UCS Manager will assign the chassis a new ID that may be different from ID that it held before.

Acknowledging a Chassis

Acknowledging the chassis ensures that Cisco UCS Manager is aware of the current physical configuration of the chassis.

SUMMARY STEPS

1. UCS-A# **acknowledge chassis** *chassis-num*
2. UCS-A# **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **acknowledge chassis** *chassis-num*	Acknowledges the specified chassis.
Step 2	UCS-A# **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example acknowledges chassis 2 and commits the transaction:

```
UCS-A# acknowledge chassis 2
UCS-A* # commit-buffer
UCS-A #
```

Decommissioning a Chassis

SUMMARY STEPS

1. UCS-A# **decommission chassis** *chassis-num*
2. UCS-A# **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **decommission chassis** *chassis-num*	Decommissions the specified chassis.
Step 2	UCS-A# **commit-buffer**	Commits the transaction to the system configuration.

The decommission may take several minutes to complete.

Example

The following example decommissions chassis 2 and commits the transaction in a set up with 2 chassis:

```
UCS-A# decommission chassis 2
UCS-A* # commit-buffer
UCS-A # show chassis

Chassis:
    Chassis    Overall Status           Admin State
    ---------- ------------------------ -----------
             1 Operable                 Acknowledged
             2 Decomissioning           Acknowledged

UCS-A # show chassis decommissioned
Vendor     Model      Serial (SN) Chassis ID
---------- ---------- ----------- ----------
Cisco Systems Inc UCSME-4308 FHH1816P06E 2

UCS-A # show chassis

Chassis:
    Chassis    Overall Status           Admin State
    ---------- ------------------------ -----------
             1 Operable                 Acknowledged
```

Removing a Chassis

Before you begin

Physically remove the chassis before performing the following procedure.

SUMMARY STEPS

1. UCS-A# **remove chassis** *chassis-num*
2. UCS-A# **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **remove chassis** *chassis-num*	Removes the specified chassis.
Step 2	UCS-A# **commit-buffer**	Commits the transaction to the system configuration.

The removal may take several minutes to complete.

Example

The following example removes chassis 2 and commits the transaction:

```
UCS-A# remove chassis 2
UCS-A* # commit-buffer
UCS-A #
```

Recommissioning a Chassis

Before you begin

Collect the following information about the chassis to be recommissioned by using the show chassis decommissioned or show chassis inventory commands:

- Vendor name
- Model name
- Serial number

SUMMARY STEPS

1. UCS-A# **recommission chassis** *vendor-name model-name serial-num*
2. UCS-A# **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **recommission chassis** *vendor-name model-name serial-num*	Recommissions the specified chassis.
Step 2	UCS-A# **commit-buffer**	Commits the transaction to the system configuration. **Note** After recommissioning a chassis and committing the transaction, it may take less than a minute for the state of the chassis to change.

Example

The following example recommissions a Cisco UCS 4308 chassis and commits the transaction:

```
UCS-A# show chassis decommissioned

Chassis:
    Chassis    Overall Status           Admin State
    ---------- ------------------------ -----------
             1 Accessibility Problem    Decommission

UCS-A# recommission chassis "Cisco Systems Inc" "UCSME-4308" FCH1828V370
UCS-A* # commit-buffer
UCS-A# show chassis decommissioned
UCS-A #
```

Renumbering a Chassis

Before you begin

If you are swapping IDs between chassis, you must first decommission both chassis, then wait for the chassis decommission FSM to complete before proceeding with the renumbering steps.

SUMMARY STEPS

1. UCS-A# **show chassis inventory**
2. Verify that the chassis inventory does not include the chassis that you want to renumber and a chassis with the number that you want to use.
3. UCS-A# **recommission chassis** *vendor-name model-name serial-num* [*chassis-num*]
4. UCS-A# **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **show chassis inventory**	Displays information about your chassis.

	Command or Action	Purpose
Step 2	Verify that the chassis inventory does not include the chassis that you want to renumber and a chassis with the number that you want to use.	If either of these chassis are listed in the chassis inventory, decommission those chassis. You must wait until the decommission FSM is complete and the chassis are not listed in the chassis inventory before continuing. This might take several minutes. To see which chassis have been decommissioned, issue the **show chassis decommissioned** command.
Step 3	UCS-A# **recommission chassis** *vendor-name model-name serial-num* [*chassis-num*]	Recommissions and renumbers the specified chassis.
Step 4	UCS-A# **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example decommissions two Cisco UCS chassis (chassis 2 and 3), switches their IDs, and commits the transaction:

```
UCS-A# show chassis inventory

    Chassis    PID       Vendor         Serial (SN) HW Revision
    ------- --------- ---------------   ----------- -------------
          1 UCSME-4308 Cisco Systems Inc FHH1816P06D 0
          2 UCSME-4308 Cisco Systems Inc FCH1840V04T 0
          3 UCSME-4308 Cisco Systems Inc FCH1828V2HL 0
          4 UCSME-4308 Cisco Systems Inc FCH1828V370 0

UCS-A# decommission chassis "Cisco Systems Inc" "UCSME-4308" FCH1840V04T
UCS-A*# commit-buffer
UCS-A# decommission chassis "Cisco Systems Inc" "UCSME-4308" FCH1828V2HL
UCS-A*# commit-buffer
UCS-A# show chassis inventory

    Chassis    PID       Vendor         Serial (SN) HW Revision
    ------- --------- ---------------   ----------- -------------
          1 UCSME-4308 Cisco Systems Inc FHH1816P06D 0
          4 UCSME-4308 Cisco Systems Inc FCH1828V370 0

UCS-A# show chassis decommissioned

    Chassis    PID       Vendor         Serial (SN) HW Revision
    ------- --------- ---------------   ----------- -------------
          2 UCSME-4308 Cisco Systems Inc FCH1840V04T 0
          3 UCSME-4308 Cisco Systems Inc FCH1828V2HL 0

UCS-A# recommission chassis "Cisco Systems Inc" "UCSME-4308" FCH1840V04T 3
UCS-A* # commit-buffer
UCS-A# recommission chassis "Cisco Systems Inc" "UCSME-4308" FCH1828V2HL 2
UCS-A* # commit-buffer
UCS-A # show chassis inventory

    Chassis    PID       Vendor         Serial (SN) HW Revision
    ------- --------- ---------------   ----------- -------------
          1 UCSME-4308 Cisco Systems Inc FHH1816P06D 0
          2 UCSME-4308 Cisco Systems Inc FCH1828V2HL 0
```

```
3 UCSME-4308 Cisco Systems Inc FCH1840V04T 0
4 UCSME-4308 Cisco Systems Inc FCH1828V370 0
```

Showing the Cartridges in a Chassis

You can display all the cartridges in a specified chassis from the chassis mode. You can also display the status of all the cartridges from the EXEC mode.

SUMMARY STEPS

1. UCS-A# **scope chassis** *chassis-num*
2. UCS-A /chassis # **show cartridge**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **show cartridge**	Shows all the cartridges in the specified chassis.

Example

The following example shows all the cartridges in the specified chassis from the chassis mode:

```
UCS-A # scope chassis 1
UCS-A /chassis # show cartridge

Cartridge:
    Slot ID    PID        Presence
    ---------- ---------- --------
             1 UCSME-142-M4
                          Equipped
             2 UCSME-142-M4
                          Equipped
             3 UCSME-142-M4
                          Equipped
             4 UCSME-142-M4
                          Equipped
             5 UCSME-142-M4
                          Equipped
             6 UCSME-142-M4
                          Equipped
             7 UCSME-142-M4
                          Equipped
             8 UCSME-142-M4
                          Equipped
```

The following example shows the status of all the cartridges from the EXEC mode:

```
UCS-A # show cartridge status
Cartridge  Slot Status                        PID
---------- --------------------------------- ---
```

```
1/1        Equipped                          UCSME-142-M4
1/2        Equipped                          UCSME-142-M4
1/3        Equipped                          UCSME-142-M4
1/4        Equipped                          UCSME-142-M4
1/5        Equipped                          UCSME-142-M4
1/6        Equipped                          UCSME-142-M4
1/7        Equipped                          UCSME-142-M4
1/8        Equipped                          UCSME-142-M4
2/1        Equipped                          UCSHD-B162-M4
2/2        Equipped                          UCSME-142-M4
2/3        Equipped                          UCSME-142-M4
2/4        Empty
2/5        Empty
2/6        Empty
2/7        Empty
2/8        Empty
```

Showing Chassis Information

SUMMARY STEPS

1. UCS-A# **scope chassis** *chassis-num*
2. UCS-A /chassis # **show detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **show detail**	Displays information about the specified chassis. **Note** Discovery Status, introduced in Cisco UCS Manager Release 2.5, indicates the sides from which the chassis is fully discovered. While the Connection Path attribute indicates the basic reachability of the chassis from a particular FI, the Discovery Status attribute indicates that the chassis inventory has been completed on that side.

Example

The following example shows the information about the specified chassis:

```
UCS-A # scope chassis 1
UCS-A /chassis* # show detail

Chassis:
    Chassis: 1
    User Label:
```

```
Overall Status: Operable
Oper qualifier: N/A
Operability: N/A
Conf State: Ok
Admin State: Acknowledged
Conn Path: A,B
Conn Status: A,B
Discovery Status: A,B
Presence: Unknown
Managing Instance: B
Product Name: Cisco UCS 4308
PID: UCSME-4308
VID: V00
Part Number: 68-5199-02
Vendor: Cisco Systems Inc
Model: UCSME-4308
Serial (SN): FCH1842V1QV
HW Revision: 0
Mfg Date: 2014-10-16T00:00:00.000
Power State: Ok
Thermal Status: Ok
SEEPROM operability status: N/A
Current Task:
```

Showing the Status of all the Servers in the Cisco UCS Domain

SUMMARY STEPS

1. UCS-A # **show server status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A # **show server status**	Shows the status for all servers in the Cisco UCS domain.

Example

The following example shows the status for all servers in the Cisco UCS domain:

```
UCS-A # show server status
Server  Slot Status                          Availability Overall Status        Discovery
------- --------------------------------- ------------ --------------------- ---------
1/1/1   Equipped                             Unavailable  Ok                    Complete
1/1/2   Equipped                             Unavailable  Ok                    Complete
1/2/1   Equipped                             Unavailable  Ok                    Complete
1/2/2   Equipped                             Unavailable  Ok                    Complete
1/3/1   Equipped                             Unavailable  Ok                    Complete
1/3/2   Equipped                             Unavailable  Ok                    Complete
1/4/1   Equipped                             Unavailable  Ok                    Complete
1/4/2   Equipped                             Unavailable  Ok                    Complete
1/5/1   Equipped                             Unavailable  Ok                    Complete
1/5/2   Equipped                             Unavailable  Ok                    Complete
```

```
1/6/1   Equipped                          Unavailable  Ok                   Complete
1/6/2   Equipped                          Unavailable  Ok                   Complete
1/7/1   Equipped                          Unavailable  Ok                   Complete
1/7/2   Equipped                          Unavailable  Ok                   Complete
1/8/1   Equipped                          Unavailable  Ok                   Complete
1/8/2   Equipped                          Unavailable  Ok                   Complete
2/1/1   Equipped                          Available    Unassociated         Complete
2/1/2   Equipped                          Available    Unassociated         Complete
2/2/1   Equipped                          Available    Unassociated         Complete
2/2/2   Equipped                          Available    Unassociated         Complete
2/3/1   Equipped                          Available    Unassociated         Complete
2/3/2   Equipped                          Available    Unassociated         Complete
```

Toggling the Locator LED

Turning On the Locator LED for a Chassis

SUMMARY STEPS

1. UCS-A# **scope chassis** *chassis-num*
2. UCS-A /chassis # **enable locator-led**
3. UCS-A /chassis # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **enable locator-led**	Turns on the chassis locator LED.
Step 3	UCS-A /chassis # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example turns on the locator LED for chassis 2 and commits the transaction:

```
UCS-A# scope chassis 2
UCS-A /chassis # enable locator-led
UCS-A /chassis* # commit-buffer
UCS-A /chassis #
```

Turning Off the Locator LED for a Chassis

SUMMARY STEPS

1. UCS-A# **scope chassis** *chassis-num*

2. UCS-A /chassis # **disable locator-led**
3. UCS-A /chassis # **commit-buffer**

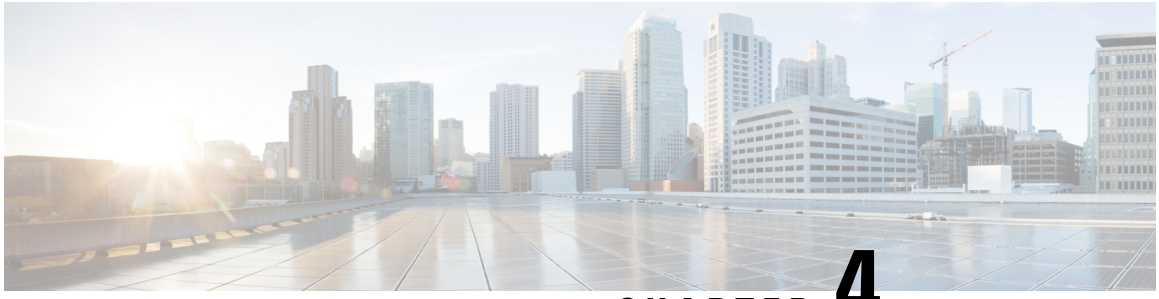
DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **disable locator-led**	Turns off the chassis locator LED.
Step 3	UCS-A /chassis # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example turns off the locator LED for chassis 2 and commits the transaction:

```
UCS-A# scope chassis 2
UCS-A /chassis # disable locator-led
UCS-A /chassis* # commit-buffer
UCS-A /chassis #
```



CHAPTER 4

Managing Compute Cartridges

This part contains the following chapters:

Compute Cartridges

Compute cartridges consist of up to two Cisco UCS servers. Each individual server is independently manageable through its own CIMC instance, and has its own memory and CPU. The cartridge does not contain any I/O adapters or local storage within it.

Acknowledging a Cartridge Slot in a Chassis

Perform the following procedure to discover the cartridge if the cartridge slot is in the mismatch state.

Note When a cartridge is replaced by another cartridge or relocated to a new slot in the same chassis or a different chassis, it goes into the mismatch state.

SUMMARY STEPS

1. UCS-A# **acknowledge cartridge-slot** *chassis-id / cartridge-id*
2. UCS-A# **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **acknowledge cartridge-slot** *chassis-id / cartridge-id*	Acknowledges the specified cartridge slot.

	Command or Action	Purpose
Step 2	UCS-A# **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example acknowledges cartridge 1 in chassis 1 and commits the transaction:

```
UCS-A# acknowledge cartridge-slot 1/1
UCS-A* # commit-buffer
UCS-A #
```

Removing a Cartridge from a Chassis

Before you begin

Physically remove the cartridge from its chassis before performing the following procedure.

SUMMARY STEPS

1. UCS-A# **remove cartridge** *chassis-num / cartridge-num*
2. UCS-A# **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **remove cartridge** *chassis-num / cartridge-num*	Removes the specified cartridge from the specified chassis.
Step 2	UCS-A# **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example removes cartridge 1 in chassis 1 and commits the transaction:

```
UCS-A# remove cartridge 1/1
UCS-A* # commit-buffer
UCS-A #
```

What to do next

If you physically re-install the cartridge, you must re-acknowledge the cartridge to have Cisco UCS Manager rediscover the cartridge.

Showing the Status of Cartridges

SUMMARY STEPS

1. UCS-A# **show cartridge status** [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **show cartridge status** [**detail**]	Shows the status of all the cartridges in the Cisco UCS domain.

Example

The following example shows the status of all the cartridges in the Cisco UCS domain:

```
UCS-A# show cartridge status
Server  Slot Status                        PID
------- ---------------------------------- ---
1/1     Equipped                           UCSME-142-M4
1/2     Empty
1/3     Empty
1/4     Empty
1/5     Empty
1/6     Empty
1/7     Empty
1/8     Empty
```

The following example shows the detailed status of all the cartridges in the Cisco UCS domain.

```
UCS-A# show cartridge status detail
Cartridge 1/1:
    Product Name: Cisco UCSME-142-M4
    Presence: Equipped
    PID: UCSME-142-M4
    Vendor: Cisco Systems Inc
    Serial (SN): FCH18037V04
    Revision: 0

Cartridge 1/2:
    Product Name:
    Presence: Empty
    PID:
    Vendor:
    Serial (SN):
    Revision: 0

Cartridge 1/3:
    Product Name:
    Presence: Empty
    PID:
    Vendor:
    Serial (SN):
    Revision: 0
```

Showing the Status of all Servers in a Cartridge

You can display the status of all servers in a cartridge from the cartridge mode.

SUMMARY STEPS

1. UCS-A# **scope chassis** *chassis-num*
2. UCS-A /chassis# **scope cartridge** *cartridge-id*
3. UCS-A /chassis/cartridge# **show server**
4. (Optional) UCS-A /chassis/cartridge# **show server detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis# **scope cartridge** *cartridge-id*	Enters cartridge mode for the specified cartridge ID.
Step 3	UCS-A /chassis/cartridge# **show server**	Shows the status for all servers in the specified cartridge.
Step 4	(Optional) UCS-A /chassis/cartridge# **show server detail**	Shows detailed information of all servers in the specified cartridge.

Example

The following example shows the status of all servers in the specified cartridge:

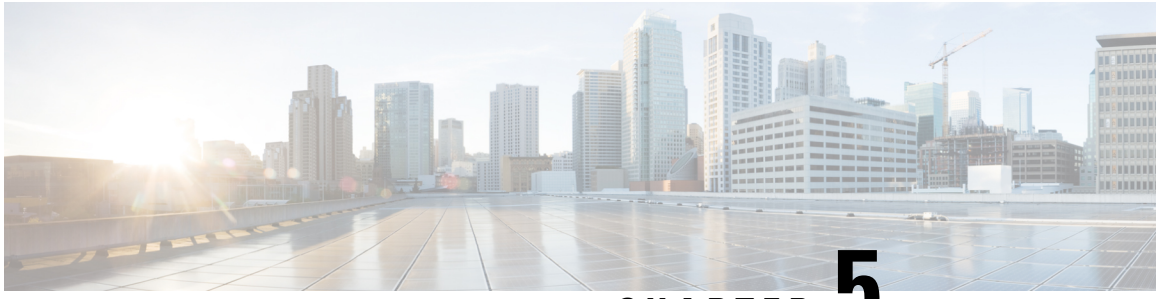
```
UCS-A# scope chassis 1
UCS-A /chassis # scope cartridge 4
UCS-A /chassis/cartridge # show server
Server:
    Instance ID     Model                     Overall Status        Availability
    --------------- ------------------------- --------------------- ------------
                  1 UCSME-142-M4              Ok                    Unavailable
```

The following example shows detailed information of all servers in the specified cartridge:

```
UCS-A /chassis/cartridge # show server detail

Server:
    Instance ID: 1
    Name:
    User Label:
    Overall Status: Ok
    Oper Qualifier: N/A
    Association: Associated
    Availability: Unavailable
    Discovery: Complete
    Conn Path: A
    Conn Status: A
    Managing Instance: A
    Admin Power: Policy
    Oper Power: On
```

```
Admin State: In Service
Product Name: Cisco UCSME-142-M4
Equipped PID: UCSME-142-M4
Equipped VID: V00
Vendor: Cisco Systems Inc
Serial (SN): JXLCH6P25F
Revision: 0
Mfg Date: 2014-10-10T01:00:00.000
Presence: Equipped
Part Number: 73-15883-05
Memory (MB): 4096
Effective Memory (MB): 4096
Operating Memory Speed (MHz): 1600
Operating Memory Voltage: Regular Voltage
Cores: 4
Num Of Cores Enabled: 4
Adapters: 0
Eth Host Interfaces: 0
FC Host Interfaces: 0
Burned-In UUID: e83a5ee7-9262-4b1a-a93b-56f34fb86f85
Dynamic UUID: 00000000-0000-0000-0000-012323400000
Current Task 1:
Current Task 2:
```



CHAPTER 5

Managing Modular Servers

This part contains the following chapters:

Modular Server Management

Modular servers, which are introduced in Cisco UCS M-Series, are contained in compute cartridges.

Note You cannot remove servers from their cartridges.

Booting a Modular Server

Before you begin

Associate a service profile with a modular server or server pool.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **scope service-profile** *profile-name*
3. UCS-A /org/service-profile # **power up**
4. UCS-A /org/service-profile # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 2	UCS-A /org # **scope service-profile** *profile-name*	Enters organization service profile mode for the specified service profile.
Step 3	UCS-A /org/service-profile # **power up**	Boots the modular server associated with the service profile.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example boots the modular server associated with the service profile named ServProf34 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile ServProf34
UCS-A /org/service-profile* # power up
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Shutting Down a Modular Server

When you use this procedure to shut down a server with an installed operating system, Cisco UCS Manager triggers the OS into a graceful shutdown sequence.

Before you begin

Associate a service profile with a modular server or server pool.

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **scope service-profile** *profile-name*
3. UCS-A /org/service-profile # **power down**
4. UCS-A /org/service-profile # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name*.
Step 2	UCS-A /org # **scope service-profile** *profile-name*	Enters organization service profile mode for the specified service profile.
Step 3	UCS-A /org/service-profile # **power down**	Shuts down the modular server associated with the service profile.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shuts down the modular server associated with the service profile named ServProf34 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile ServProf34
UCS-A /org/service-profile # power down
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Power Cycling a Modular Server

SUMMARY STEPS

1. UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*
2. UCS-A /chassis/cartridge/server # **cycle** {**cycle-immediate** | **cycle-wait**}
3. UCS-A /chassis/cartridge/server # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*	Enters cartridge server mode for the specified modular server in the specified chassis and cartridge.
Step 2	UCS-A /chassis/cartridge/server # **cycle** {**cycle-immediate** | **cycle-wait**}	Power cycles the modular server. Use the **cycle-immediate** keyword to immediately begin power cycling the modular server; use the **cycle-wait** keyword to schedule the power cycle to begin after all pending management operations have completed.
Step 3	UCS-A /chassis/cartridge/server # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example immediately power cycles modular server 2 in cartridge 2 of chassis 2 and commits the transaction:

```
UCS-A#  scope server 2/2/2
UCS-A /chassis/cartridge/server # cycle cycle-immediate
UCS-A /chassis/cartridge/server # commit-buffer
UCS-A /chassis/cartridge/server #
```

Performing a Hard Reset on a Modular Server

When you reset a server, Cisco UCS Manager sends a pulse on the reset line. You can choose to gracefully shut down the operating system. If the operating system does not support a graceful shutdown, the server is power cycled. The option to have Cisco UCS Manager complete all management operations before it resets the server does not guarantee the completion of these operations before the server is reset.

Note If you are trying to boot a server from a power-down state, you should not use **Reset**.

If you continue the power-up with this process, the desired power state of the servers become out of sync with the actual power state and the servers might unexpectedly shut down at a later time. To safely reboot the selected servers from a power-down state, click **Cancel**, then select the **Boot Server** action.

SUMMARY STEPS

1. UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*
2. UCS-A /chassis/cartridge/server # **reset** {**hard-reset-immediate** | **hard-reset-wait**}
3. UCS-A /chassis/cartridge/server # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*	Enters cartridge server mode for the specified modular server in the specified chassis and cartridge.
Step 2	UCS-A /chassis/cartridge/server # **reset** {**hard-reset-immediate** | **hard-reset-wait**}	Performs a hard reset of the modular server. Use the **hard-reset-immediate** keyword to immediately begin hard resetting the server; use the **hard-reset-wait** keyword to schedule the hard reset to begin after all pending management operations have completed.
Step 3	UCS-A /chassis/cartridge/server # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example performs an immediate hard reset of modular server 2 in cartridge 2 of chassis 2 and commits the transaction:

```
UCS-A#  scope server 2/2/2
UCS-A /chassis/cartridge/server # reset hard-reset-immediate
UCS-A /chassis/cartridge/server* # commit-buffer
UCS-A /chassis/cartridge/server #
```

Acknowledging a Modular Server

Perform the following procedure to rediscover the server and all endpoints in the server. For example, you can use this procedure if a server is stuck in an unexpected state, such as the discovery state.

SUMMARY STEPS

1. UCS-A# **acknowledge server** *chassis-id* / *cartridge-id* / *server-id*
2. UCS-A /chassis/cartridge/server # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **acknowledge server** *chassis-id* / *cartridge-id* / *server-id*	Acknowledges the specified modular server in the specified chassis and cartridge.
Step 2	UCS-A /chassis/cartridge/server # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example acknowledges modular server 2 in cartridge 2 of chassis 2 and commits the transaction:

```
UCS-A# acknowledge server 2/2/2
UCS-A /chassis/cartridge/server* # commit-buffer
UCS-A #
```

Decommissioning a Modular Server

Decommissioning of a server is performed to temporarily remove the server from the UCSM configuration.

SUMMARY STEPS

1. UCS-A# **decommission server** *chassis-num* / *cartridge-id* / *server-num*
2. UCS-A /chassis/cartridge/server # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **decommission server** *chassis-num* / *cartridge-id* / *server-num*	Decommissions the specified modular server in the specified chassis and cartridge.
Step 2	UCS-A /chassis/cartridge/server # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example decommissions modular server 2 in cartridge 2 of chassis 2 and commits the transaction:

```
UCS-A# decommission server 2/2/2
UCS-A /chassis/cartridge/server* # commit-buffer
UCS-A #
```

Showing the Status of a Modular Server

SUMMARY STEPS

1. UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*
2. UCS-A /chassis/cartridge/server # **show status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*	Enters cartridge server mode for the specified modular server in the specified chassis and cartridge.
Step 2	UCS-A /chassis/cartridge/server # **show status**	Shows the status for the specified modular server.

Example

The following example shows the status for modular server 1 in cartridge 3 of chassis 1:

```
UCS-A# scope server 1/3/1
UCS-A /chassis/cartridge/server # show status

Server  Slot Status                        Availability Overall Status        Discovery
------- ---------------------------------- ------------ --------------------- ---------
1/3/1   Equipped                           Available    Unassociated          Complete
```

Turning On the Locator LED for a Modular Server

The locator LED is shared by all servers in a cartridge. Hence, you can turn the locator LED on from any server in a cartridge.

SUMMARY STEPS

1. UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*
2. UCS-A /chassis/cartridge/server # **enable locator-led**
3. UCS-A /chassis/cartridge/server # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*	Enters chassis server mode for the specified modular server in the specified chassis and cartridge.
Step 2	UCS-A /chassis/cartridge/server # **enable locator-led**	Turns on the modular server locator LED.
Step 3	UCS-A /chassis/cartridge/server # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example turns on the locator LED for modular server 2 in cartridge 2 of chassis 2 and commits the transaction:

```
UCS-A# scope server 2/2/2
UCS-A /chassis/cartridge/server # enable  locator-led
UCS-A /chassis/cartridge/server* # commit-buffer
UCS-A /chassis/cartridge/server #
```

Turning Off the Locator LED for a Modular Server

The locator LED is shared by all servers in a cartridge. Hence, to turn off a locator LED of a cartridge, you must turn it off from all servers in the cartridge.

SUMMARY STEPS

1. UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*
2. UCS-A /chassis/cartridge/server # **disable locator-led**
3. UCS-A /chassis/cartridge/server # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*	Enters chassis mode for the specified modular server in the specified chassis and cartridge.
Step 2	UCS-A /chassis/cartridge/server # **disable locator-led**	Turns off the modular server locator LED.
Step 3	UCS-A /chassis/cartridge/server # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example turns off the locator LED for modular server 2 in cartridge 2 of chassis 2 and commits the transaction:

```
UCS-A# scope chassis 2/2/2
UCS-A /chassis/cartridge/server # disable locator-led
UCS-A /chassis/cartridge/server* # commit-buffer
UCS-A /chassis/cartridge/server #
```

Resetting the CMOS for a Modular Server

Sometimes, troubleshooting a server might require you to reset the CMOS. Resetting the CMOS is not part of the normal maintenance of a server.

SUMMARY STEPS

1. UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*
2. UCS-A /chassis/cartridge/server # **reset-cmos**
3. UCS-A /chassis/cartridge/server # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*	Enters cartridge server mode for the specified modular server in the specified chassis and cartridge.
Step 2	UCS-A /chassis/cartridge/server # **reset-cmos**	Resets the CMOS for the modular server.
Step 3	UCS-A /chassis/cartridge/server # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example resets the CMOS for modular server 2 in cartridge 2 of chassis 2 and commits the transaction:

```
UCS-A# scope server 2/2/2
UCS-A /chassis/cartridge/server # reset-cmos
```

```
UCS-A /chassis/cartridge/server* # commit-buffer
UCS-A /chassis/cartridge/server #
```

Resetting the CIMC for a Modular Server

Sometimes, with the firmware, troubleshooting a server might require you to reset the CIMC. Resetting the CIMC is not part of the normal maintenance of a server. After you reset the CIMC, the CIMC reboots the management controller of the blade server.

If the CIMC is reset, the power monitoring functions of Cisco UCS become briefly unavailable until the CIMC reboots. Typically, the reset only takes 20 seconds; however, it is possible that the peak power cap can exceed during that time. To avoid exceeding the configured power cap in a low power-capped environment, consider staggering the rebooting or activation of CIMCs.

SUMMARY STEPS

1. UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*
2. UCS-A /chassis/cartridge/server # **scope CIMC**
3. UCS-A /chassis/cartridge/server/CIMC # **reset**
4. UCS-A /chassis/cartridge/server/CIMC # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *cartridge-id* / *server-num*	Enters cartridge server mode for the specified modular server in the specified chassis and cartridge.
Step 2	UCS-A /chassis/cartridge/server # **scope CIMC**	Enters cartridge server CIMC mode
Step 3	UCS-A /chassis/cartridge/server/CIMC # **reset**	Resets the CIMC for the modular server.
Step 4	UCS-A /chassis/cartridge/server/CIMC # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example resets the CIMC for modular server 2 in cartridge 2 of chassis 2 and commits the transaction:

```
UCS-A# scope server 2/2/2
UCS-A /chassis/cartridge/server # scope CIMC
UCS-A /chassis/cartridge/server/cimc # reset
UCS-A /chassis/cartridge/server/cimc* # commit-buffer
UCS-A /chassis/cartridge/server/cimc #
```

Issuing an NMI from a Modular Server

Perform the following procedure if the system remains unresponsive and you need Cisco UCS Manager to issue a Non-Maskable Interrupt (NMI) to the BIOS or operating system from the CIMC. This action creates a core dump or stack trace, depending on the operating system installed on the server.

SUMMARY STEPS

1. UCS-A # **scope server** *chassis-id* / *cartridge-id* / *server-id*
2. UCS-A /chassis/cartridge/server # **diagnostic-interrupt**
3. UCS-A /chassis/cartridge/server # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A # **scope server** *chassis-id* / *cartridge-id* / *server-id*	Enters server mode for the specified server.
Step 2	UCS-A /chassis/cartridge/server # **diagnostic-interrupt**	
Step 3	UCS-A /chassis/cartridge/server # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example sends an NMI from server 2 in cartridge 2 of chassis 2 and commits the transaction:

```
UCS-A# scope server 2/2/2
UCS-A /chassis/cartridge/server # diagnostic-interrupt
UCS-A /chassis/cartridge/server* # commit-buffer
UCS-A /chassis/cartridge/server #
```

Health LED Alarms

The server health LED is located on the front of each Cisco UCS M-Series server. Cisco UCS Manager allows you to view the sensor faults that cause the server health LED to change color from green to amber or blinking amber.

The health LED alarms display the following information:

Name	Description
Severity column	The severity of the alarm. This can be one of the following: • Critical—The blade health LED is blinking amber. • Minor—The blade health LED is amber.
Description column	A brief description of the alarm.
Sensor ID column	The ID of the sensor the triggered the alarm.

Name	Description
Sensor Name column	The name of the sensor that triggered the alarm.

Viewing Health LED Status

SUMMARY STEPS

1. UCS-A# **scope server** *chassis-id* / *cartridge-id* / *server-num*
2. UCS-A /chassis/cartridge/server # **show health-led expand**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-id* / *cartridge-id* / *server-num*	Enters cartridge server mode for the specified modular server in the specified chassis and cartridge.
Step 2	UCS-A /chassis/cartridge/server # **show health-led expand**	Displays the health LED and sensor alarms for the selected server.

Example

The following example shows how to display the health LED status and sensor alarms for chassis 1 cartridge 2 server 1:

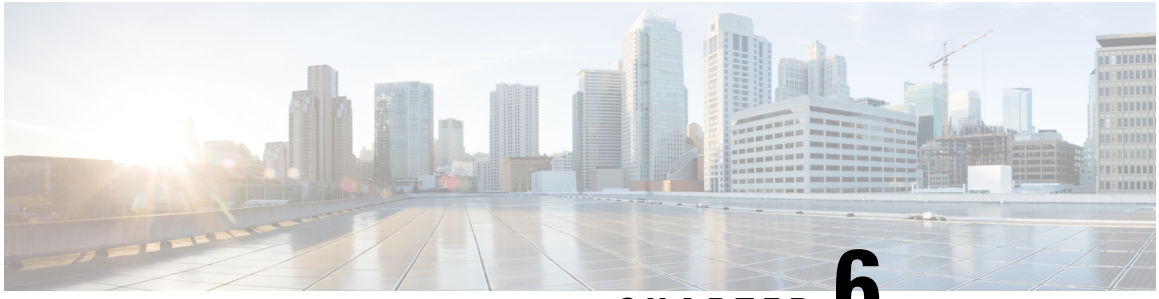
```
UCS-A# scope server 1/2/1
UCS-A /chassis/cartridge/server # show health-led
Health LED:
    Severity: Minor
    Reason:: P0V75_STBY:Voltage Threshold Crossed;TEMP_SENS_FRONT:Temperature Threshold
Crossed;
    Color: Amber
    Oper State:: On

    Sensor Alarm:
        Severity: Minor
        Sensor ID: 7
        Sensor Name: P0V75_STBY
        Alarm Desc: Voltage Threshold Crossed

        Severity: Minor
        Sensor ID: 76
        Sensor Name: TEMP_SENS_FRONT
        Alarm Desc: Temperature Threshold Crossed

        Severity: Minor
        Sensor ID: 91
        Sensor Name: DDR3_P1_D2_TMP
        Alarm Desc: Temperature Threshold Crossed

UCS-A /chassis/cartridge/server #
```



CHAPTER 6

Configuring Trusted Platform Module

Trusted Platform Module

The Trusted Platform Module (TPM) is a component that can securely store artifacts that are used to authenticate the server. These artifacts can include passwords, certificates, or encryption keys. A TPM can also be used to store platform measurements that help ensure that the platform remains trustworthy. Authentication (ensuring that the platform can prove that it is what it claims to be) and attestation (a process helping to prove that a platform is trustworthy and has not been breached) are necessary steps to ensure safer computing in all environments. It is a requirement for the Intel Trusted Execution Technology (TXT) security feature, which must be enabled in the BIOS settings for a server equipped with a TPM. Only the modular servers in Cisco UCSME-2814 compute cartridges include support for TPM. TPM is enabled by default on these servers.

Only very basic enable/activate hardware component status is provided for TPM 2.0 and later. Nearly all status indications are software status. BIOS uses “Enable/Disable “ to abstract status **Enable/Disable Platform Hierarchy**, **Enable/Disable Storage Hierarchy**, and **Enable/Disable Endorsement Hierarchy**. That is, Enable and Activate TPM will enable all three Hierarchies, and Disable and De-activate TPM will Disable these three Hierarchies. For more information on TPM flag definitions and enabling, activation, and taking ownership of these hierarchies, specific to your implementation, refer to the TCG Trusted Platform Module Specification.

Intel Trusted Execution Technology

Intel Trusted Execution Technology (TXT) provides greater protection for information that is used and stored on the business server. A key aspect of that protection is the provision of an isolated execution environment and associated sections of memory where operations can be conducted on sensitive data, invisible to the rest of the system. Intel TXT provides for a sealed portion of storage where sensitive data such as encryption keys can be kept, helping to shield them from being compromised during an attack by malicious code. Only the modular servers in Cisco UCSME-2814 compute cartridges include support for TXT. TXT is disabled by default on these servers.

TXT can be enabled only after TPM, Intel Virtualization technology (VT) and Intel Virtualization Technology for Directed I/O (VT-d) are enabled. When you only enable TXT, it also implicitly enables TPM, VT, and VT-d.

Trusted Platform

The modular servers in Cisco UCSME-2814 compute cartridges include support for TPM and TXT. Cisco UCS M4 blade and rack-mount servers include support for TPM and TXT. UCS Manager Release 2.5(2)UCS Manager Release 2.2(4) allows you to perform the following operations on TPM and TXT:

- Enabling or Disabling TPM
- Clearing TPM for a Blade Server

  or
- Clearing TPM for a Rack-Mount Server

Note For Cisco UCS M3 blade servers, press **F2** to enter the BIOS setup menu and change the settings.

Enabling or Disabling TPM

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create bios-policy** *policy-name*
3. UCS-A /org/bios-policy* # **set trusted-platform-module-config tpm-state** {**enabled** | **disabled** | **platform-default**}
4. UCS-A /org/bios-policy* # **commit-buffer**
5. UCS-A /org # **create service-profile** *sp-name*}
6. UCS-A /org/service-profile* # **set bios-policy** *policy-name*
7. UCS-A /org/service-profile* # **commit-buffer**
8. UCS-A /org/service-profile # **associate server** *chassis-id / cartridge-id / slot-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **create bios-policy** *policy-name*	Creates a BIOS policy with the specified policy name, and enters org BIOS policy mode.
Step 3	UCS-A /org/bios-policy* # **set trusted-platform-module-config tpm-state** {**enabled** | **disabled** | **platform-default**}	Specifies whether TPM is **enabled** or **disabled**. **platform-default** is TPM enabled.
Step 4	UCS-A /org/bios-policy* # **commit-buffer**	Commits the transaction to the system configuration.
Step 5	UCS-A /org # **create service-profile** *sp-name*}	Creates the service profile specified and enters service profile configuration mode.

	Command or Action	Purpose
Step 6	UCS-A /org/service-profile* # **set bios-policy** *policy-name*	Associates the specified BIOS policy with the service profile.
Step 7	UCS-A /org/service-profile* # **commit-buffer**	Commits the transaction to the system configuration.
Step 8	UCS-A /org/service-profile # **associate server** *chassis-id / cartridge-id / slot-id*	Associates the service profile with a single server.

Example

The following example shows how to enable TPM:

```
UCS-A # scope org
UCS-A /org # create bios-policy bp1
UCS-A /org/bios-policy* # set trusted-platform-module-config tpm-state enabled
UCS-A /org/bios-policy* # commit-buffer
UCS-A /org # create service-profile sp1
UCS-A /org/service-profile* # set bios-policy bp1
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile # associate server 1/3/1
```

Enabling or Disabling TXT

SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create bios-policy** *policy-name*
3. UCS-A /org/bios-policy* # **set intel-trusted-execution-technology-config txt-support** {**enabled** | **disabled** | **platform-default**}
4. UCS-A /org/bios-policy* # **commit-buffer**
5. UCS-A /org # **create service-profile** *sp-name*}
6. UCS-A /org/service-profile* # **set bios-policy** *policy-name*
7. UCS-A /org/service-profile* # **commit-buffer**
8. UCS-A /org/service-profile # **associate server** *chassis-id / cartridge-id / slot-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **create bios-policy** *policy-name*	Creates a BIOS policy with the specified policy name, and enters org BIOS policy mode.

	Command or Action	Purpose
Step 3	UCS-A /org/bios-policy* # **set intel-trusted-execution-technology-config txt-support** {**enabled** | **disabled** | **platform-default**}	Specifies whether TXT is **enabled** or **disabled**. **platform-default** is TXT disabled.
Step 4	UCS-A /org/bios-policy* # **commit-buffer**	Commits the transaction to the system configuration.
Step 5	UCS-A /org # **create service-profile** *sp-name*}	Creates the service profile specified and enters service profile configuration mode.
Step 6	UCS-A /org/service-profile* # **set bios-policy** *policy-name*	Associates the specified BIOS policy with the service profile.
Step 7	UCS-A /org/service-profile* # **commit-buffer**	Commits the transaction to the system configuration.
Step 8	UCS-A /org/service-profile # **associate server** *chassis-id / cartridge-id / slot-id*	Associates the service profile with a single server.

Example

The following example shows how to enable TXT:

```
UCS-A # scope org
UCS-A /org # create bios-policy bp1
UCS-A /org/bios-policy* # set intel-trusted-execution-technology-config txt-support enabled
UCS-A /org/bios-policy* # commit-buffer
UCS-A /org # create service-profile sp1
UCS-A /org/service-profile* # set bios-policy bp1
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile # associate server 1/3/1
```

Clearing TPM for a Modular Server

You can clear TPM only on the modular servers that include support for TPM.

Caution Clearing TPM is a potentially hazardous operation. The OS may stop booting. You may also see loss of data.

Before you begin

TPM must be enabled.

SUMMARY STEPS

1. UCS-A# **scope server** *chassis-id*/*cartridge-id*/*server-id*
2. UCS-A# /chassis/cartridge/server # **scope tpm** *tpm-ID*
3. UCS-A# /chassis/cartridge/server/tpm # **set adminaction clear-config**
4. UCS-A# /chassis/cartridge/server/tpm # **commit-buffer**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-id/cartridge-id/server-id*	Enters server mode for the specified server.
Step 2	UCS-A# /chassis/cartridge/server # **scope tpm** *tpm-ID*	Enters org TPM mode for the specified TPM.
Step 3	UCS-A# /chassis/cartridge/server/tpm # **set adminaction clear-config**	Specifies that the TPM is to be cleared.
Step 4	UCS-A# /chassis/cartridge/server/tpm # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to clear TPM for a modular server:

```
UCS-A# scope server 1/3/1
UCS-A# /chassis/cartridge/server # scope tpm 1
UCS-A# /chassis/cartridge/server/tpm # set adminaction clear-config
UCS-A#/chassis/cartridge/server/tpm* # commit-buffer
```

Viewing TPM Properties

SUMMARY STEPS

1. UCS-A# **scope server** *chassis-id/cartridge-id/server-id*
2. UCS-A /chassis/cartridge/server # **scope tpm** *tpm-id*
3. UCS-A /chassis/cartridge/server/tpm # **show**
4. UCS-A /chassis/cartridge/server/tpm # **show detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-id/cartridge-id/server-id*	Enters server mode for the specified server.
Step 2	UCS-A /chassis/cartridge/server # **scope tpm** *tpm-id*	Enters TPM mode for the specified TPM ID.
Step 3	UCS-A /chassis/cartridge/server/tpm # **show**	Displays the TPM properties.
Step 4	UCS-A /chassis/cartridge/server/tpm # **show detail**	Displays detailed TPM properties.

Example

The following example shows how to display the TPM properties a modular server:

```
UCS-A# scope server 1/3/1
UCS-A /chassis/cartridge/server # scope tpm 1
UCS-A /chassis/cartridge/server/tpm # show

Trusted Platform Module:
    Presence: Equipped
    Enabled Status: Enabled
    Active Status: Activated
    Ownership: Unowned
UCS-A /chassis/cartridge/server/tpm # show detail

Trusted Platform Module:
    Enabled Status: Enabled
    Active Status: Activated
    Ownership: Unowned
    Tpm Revision: 2
    Model: UCSX-TPM2-001
    Vendor: Cisco Systems Inc
    Serial: FCH19257E58
    Admin Action: Unspecified
    Config State: Not Applied
UCS-A /chassis/cartridge/server/tpm #
```