



Configuring Storage Profiles

This part contains the following chapters:

- [Storage Profiles, page 1](#)
- [Disk Groups and Disk Group Configuration Policies, page 2](#)
- [RAID Levels, page 3](#)
- [Automatic Disk Selection, page 4](#)
- [Supported LUN Modifications, page 5](#)
- [Unsupported LUN Modifications, page 5](#)
- [Disk Insertion Handling, page 6](#)
- [Virtual Drive Naming, page 7](#)
- [LUN Dereferencing, page 8](#)
- [Controller Constraints and Limitations, page 8](#)
- [Configuring Storage Profiles, page 8](#)

Storage Profiles

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

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. In addition, a disk can be allocated as a global hot spare, which means that it can be used by any disk group.

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 or the service profile is no longer associated with a storage profile.

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.

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.
- RAID 50 Striped Parity and Striped—Data is striped across multiple striped parity disk sets to provide high throughput and multiple disk failure tolerance.
- RAID 60 Striped Dual Parity and Striped—Data is striped across multiple striped dual parity disk sets to provide high throughput and greater disk failure tolerance.

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.

- 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:
 - 1 You specify the slot specifically by using a local disk reference
 - 2 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 6
- [Redundant Virtual Drives with No Hot Spare Drives](#), on page 6
- [Redundant Virtual Drives with Hot Spare Drives](#), on page 6
- [Replacing Hot Spare Drives](#), on page 7
- [Inserting Physical Drives into Unused Slots](#), on page 7

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, RAID 50, RAID 60) 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, RAID 50, RAID 60) 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 deleted, the LUN state is changed to **Orphaned**.

Controller Constraints and Limitations

- For Cisco UCS C240, C220, C24, and C22 servers, the storage controller allows 24 virtual drives per server. For all other servers, the storage controller allows 16 virtual drives per server.
- In Cisco UCS Manager Release 2.2(4), blade servers do not support drives with a block size of 4K, but rack-mount servers support such drives. If a drive with a block size of 4K is inserted into a blade server, discovery fails and the following error message appears: `Unable to get Scsi Device Information from the system.`

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 9](#)
- 2 [Automatically Configuring Disks in a Disk Group, on page 9](#) or [Manually Configuring Disks in a Disk Group, on page 11](#)
- 3 [Configuring Virtual Drive Properties, on page 12](#)

Setting the RAID Level

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org# create disk-group-config-policy <i>disk-group-name</i>	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 <i>raid-level</i>	Specifies the RAID level for the disk group configuration policy. The RAID levels that you can specify are: <ul style="list-style-type: none"> • raid-0-striped • raid-1-mirrored • raid-10-mirrored-and-striped • raid-5-striped-parity • raid-6-striped-dual-parity • raid-50-striped-parity-and-striped • raid-60-striped-dual-parity-and-striped
Step 4	UCS-A /org/disk-group-config-policy* # commit-buffer	Commits the transaction to the system configuration.

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.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org# enter disk-group-config-policy <i>disk-group-name</i>	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 <i>drive-type</i>	Specifies the drive type for the disk group. You can select: <ul style="list-style-type: none"> • HDD • SSD • Unspecified <p>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.</p>
Step 5	UCS-A /org/disk-group-config-policy/disk-group-qual* # set min-drive-size <i>drive-size</i>	Specifies the minimum drive size for the disk group. Only disks that match this criteria will be available for selection. <p>The range for minimum drive size is from 0 to 10240 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.</p>
Step 6	UCS-A /org/disk-group-config-policy/disk-group-qual* # set num-ded-hot-spares <i>hot-spare-num</i>	Specifies the number of dedicated hot spares for the disk group. <p>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.</p>
Step 7	UCS-A /org/disk-group-config-policy/disk-group-qual* # set num-drives <i>drive-num</i>	Specifies the number of drives for the disk group. <p>The range for drives is from 0 to 24 drives for Cisco UCS C240, C220, C24, and C22 servers. For all other servers, the limit is 16 drives per server.. You</p>

	Command or Action	Purpose
		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* # set num-glob-hot-spares <i>hot-spare-num</i>	Specifies the number of global hot spares for the disk group. The range for global hot spares is from 0 to 24 hot spares. You can also set the number of global hot spares as Unspecified . If you set the number of global hot spares as Unspecified , the global hot spares will be selected according to the disk selection process.
Step 9	UCS-A /org/disk-group-config-policy/disk-group-qual* # set use-remaining-disks {no yes}	Specifies whether the remaining disks in the disk group policy should be used or not. The default value for this command is no .
Step 10	UCS-A /org/disk-group-config-policy/disk-group-qual* # commit-buffer	Commits the transaction to the system configuration.

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 hdd
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* # set num-glob-hot-spares 2
UCS-A /org/disk-group-config-policy/disk-group-qual* # set use-remaining-disks no
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.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org# enter disk-group-config-policy <i>disk-group-name</i>	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 <i>slot-num</i>	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 <i>role</i>	Specifies the role of the local disk in the disk group. You can select: <ul style="list-style-type: none"> • ded-hot-spare: Dedicated hot spare • glob-hot-spare: Global hot spare • normal
Step 5	UCS-A /org/disk-group-config-policy/local-disk-config-ref *# set span-id <i>span-id</i>	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.
Step 6	UCS-A /org/disk-group-config-policy/local-disk-config-ref *# commit-buffer	Commits the transaction to the system configuration.

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.

If you try to associate to a server that does not support these properties, a configuration error will be generated.

Only the following storage controllers support these properties:

- LSI 6G MegaRAID SAS 9266-8i
- LSI 6G MegaRAID SAS 9271-8i
- LSI 6G MegaRAID 9265-8i
- LSI MegaRAID SAS 2208 ROMB
- LSI MegaRAID SAS 9361-8i

For the LSI MegaRAID SAS 2208 ROMB controller, these properties are supported only in the B420-M3 blade server. For the other controllers, these properties are supported in multiple rack servers.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org# scope disk-group-config-policy <i>disk-group-name</i>	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 <i>policy-type</i>	Specifies the access policy. This can be one of the following: <ul style="list-style-type: none"> • blocked • platform-default • read-only: • read-write
Step 5	UCS-A /org/disk-group-config-policy/virtual-drive-def* # set drive-cache <i>state</i>	Specifies the state of the drive cache. This can be one of the following: <ul style="list-style-type: none"> • enable • disable • no-change • platform-default
Step 6	UCS-A /org/disk-group-config-policy/virtual-drive-def* # set io-policy <i>policy-type</i>	Specifies the I/O policy. This can be one of the following: <ul style="list-style-type: none"> • cached • direct • platform-default

	Command or Action	Purpose
Step 7	UCS-A /org/disk-group-config-policy/virtual-drive-def* # set read-policy <i>policy-type</i>	Specifies the read policy. This can be one of the following: <ul style="list-style-type: none"> • normal • platform-default • read-ahead
Step 8	UCS-A /org/disk-group-config-policy/virtual-drive-def* # set strip-size <i>strip-size</i>	Specifies the strip size. This can be one of the following: <ul style="list-style-type: none"> • 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 <i>policy-type</i>	Specifies the write-cache-policy. This can be one of the following: <ul style="list-style-type: none"> • 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.

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.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org # create storage-profile <i>storage-profile-name</i>	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	UCS-A /org* # enter service-profile <i>service-profile-name</i>	(Optional) Enters the specified service profile.
Step 5	UCS-A /org/service-profile* # create storage-profile-def	(Optional) 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.

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.

Procedure

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

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.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org # enter storage-profile <i>storage-profile-name</i>	Enters storage-profile mode for the specified storage profile.
Step 3	UCS-A /org/storage-profile* # create local-lun <i>lun-name</i>	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 <i>disk-policy-name</i>	Specifies the name of the disk policy name for this LUN.
Step 6	UCS-A /org/storage-profile/local-lun* # set expand-to-avail { no yes }	Specifies whether the LUN should be expanded to the entire available disk group. For each service profile, only one LUN can be configured to use this option.
Step 7	UCS-A /org/storage-profile/local-lun* # set size <i>size</i>	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.

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 auto-deploy no-auto-deploy
UCS-A /org/storage-profile/local-lun* # set disk-policy-name dpn2
UCS-A /org/storage-profile/local-lun* # set expand-to-avail yes
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 spl
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 auto-deploy no-auto-deploy
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 expand-to-avail yes
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

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.

Procedure

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

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.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org # scope service-profile <i>service-profile-name</i>	Enters the specified service profile mode.
Step 3	UCS-A /org/service-profile # set storage-profile-name <i>storage-profile-name</i>	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.

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:

Procedure

	Command or Action	Purpose
Step 1	UCS-A /org/service-profile # show local-lun-ref	<p>Displays the following detailed information about all the local LUNs inherited by the specified service profile:</p> <ul style="list-style-type: none"> • 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. • Assigned Size (MB)—Size, in MB, assigned by UCSM. • Config State—State of LUN configuration. The states can be one of the following: <ul style="list-style-type: none"> • 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. • 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: <ul style="list-style-type: none"> • Unknown • Optimal • Degraded

	Command or Action	Purpose
		<ul style="list-style-type: none"> • Inoperable • Partially Degraded

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

Local LUN Ref:

Profile Size (MB)	LUN Name Config	Admin State	RAID Level	State Referenced Lun	RAID Level Deploy Name	ID	Provisioned Size (GB)	Assigned Drive State
1024	luna Applied	Online	RAID 0 Striped	luna-1	luna-1	1003	1	Optimal
1024	lunb Applied	Online	RAID 0 Striped	lunb-1	lunb-1	1004	1	Optimal

UCS-A /org/service-profile #

Local LUN Ref:

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

Importing Foreign Configurations for a RAID Controller on a Blade Server

Procedure

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

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

```
UCS-A# scope server 1/3
UCS-A /chassis/server # scope raid-controller 1 sas
UCS-A /chassis/server/raid-controller # set admin-state import-foreign-configuration
UCS-A /chassis/server/raid-controller* #
```

Importing Foreign Configurations for a RAID Controller on a Rack Server

Procedure

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

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

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

Configuring Local Disk Operations on a Blade Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope server [<i>chassis-num/server-num</i> <i>dynamic-uuid</i>]	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # scope raid-controller <i>raid-contr-id</i> { sas sata }	Enters RAID controller mode.
Step 3	UCS-A /chassis/server/raid-controller # scope local-disk <i>local-disk-id</i>	Enters local disk configuration mode.

	Command or Action	Purpose
Step 4	<pre>UCS-A /chassis/server/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}</pre>	<p>Configures the local disk to one of the following states:</p> <ul style="list-style-type: none"> • 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.

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

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

Configuring Local Disk Operations on a Rack Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A # scope server <i>server-id</i>	Enters server mode for the specified server.
Step 2	UCS-A /server # scope raid-controller <i>raid-contr-id</i> { sas sata }	Enters RAID controller mode.
Step 3	UCS-A /server/raid-controller # scope local-disk <i>local-disk-id</i>	Enters local disk configuration mode.
Step 4	UCS-A /server/raid-controller/local-disk # set admin-state { clear-foreign-configuration dedicated-hot-spare [<i>admin-vd-id</i>] prepare-for-removal remove-hot-spare unconfigured-good undo-prepare-for-removal }	Configures the local disk to one of the following states: <ul style="list-style-type: none"> • 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. • 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.

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

```
UCS-A /server/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 /server/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:

```
UCS-A /server/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 /server/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 /server/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:

```
UCS-A /server/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 on a Blade Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope server [<i>chassis-num/server-num</i> <i>dynamic-uuid</i>]	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # scope raid-controller <i>raid-contr-id</i> { sas sata }	Enters RAID controller chassis mode.
Step 3	UCS-A /chassis/server/raid-controller # delete virtual-drive id <i>virtual-drive-id</i>	(Optional) Deletes the orphaned virtual drive with the specified virtual drive ID.
Step 4	UCS-A /chassis/server/raid-controller # delete virtual-drive name <i>virtual-drive-name</i>	(Optional) Deletes the orphaned virtual drive with the specified virtual drive name.
Step 5	UCS-A /chassis/server/raid-controller # scope virtual-drive <i>virtual-drive-id</i>	(Optional) Enters virtual drive mode for the specified orphaned virtual drive.
Step 6	UCS-A /chassis/server/raid-controller/virtual-drive # set admin-state delete	Deletes the orphaned virtual drive.
Step 7	UCS-A /chassis/server/raid-controller/virtual-drive # commit-buffer	Commits the transaction to the system configuration.

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

```
UCS-A# scope server 1/3
UCS-A /chassis/server # scope raid-controller 1 sas
UCS-A /chassis/server/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/server/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/server/raid-controller # commit-buffer
This example shows how to delete an orphan virtual drive by specifying the virtual drive name.

```

```

UCS-A# scope server 1/3
UCS-A /chassis/server # scope raid-controller 1 sas
UCS-A /chassis/server/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/server/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/server/raid-controller # commit-buffer
```

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

```
UCS-A# scope server 1/3
```

```
UCS-A /chassis/server # scope raid-controller 1 sas
```

```
UCS-A /chassis/server/raid-controller # scope virtual-drive 1004
```

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

Deleting an Orphaned Virtual Drive on a Rack Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A # scope server <i>server-id</i>	Enters server mode for the specified server.
Step 2	UCS-A /server # scope raid-controller <i>raid-contr-id</i> {sas sata}	Enters RAID controller mode.
Step 3	UCS-A /server/raid-controller # delete virtual-drive id <i>virtual-drive-id</i>	(Optional) Deletes the orphaned virtual drive with the specified virtual drive ID.
Step 4	UCS-A /server/raid-controller # delete virtual-drive name <i>virtual-drive-name</i>	(Optional) Deletes the orphaned virtual drive with the specified virtual drive name.
Step 5	UCS-A /server/raid-controller # scope virtual-drive <i>virtual-drive-id</i>	(Optional) Enters virtual drive mode for the specified orphaned virtual drive.
Step 6	UCS-A /server/raid-controller/virtual-drive # set admin-state delete	Deletes the orphaned virtual drive.
Step 7	UCS-A /server/raid-controller/virtual-drive # commit-buffer	Commits the transaction to the system configuration.

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

```
UCS-A# scope server 1
UCS-A /server # scope raid-controller 1 sas
UCS-A /server/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 /server/raid-controller # delete virtual-drive id 1002
Warning: When committed, the virtual drive will be deleted, which may result in data loss.

```

```

UCS-A /server/raid-controller # commit-buffer

```

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

```

UCS-A# scope server 1
UCS-A /server # scope raid-controller 1 sas
UCS-A /server/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 /server/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 /server/raid-controller # commit-buffer
This example shows how to delete an orphan virtual drive by setting the admin-state.

```

```

UCS-A# scope server 1
UCS-A /server # scope raid-controller 1 sas
UCS-A /server/raid-controller # scope virtual-drive 1004
UCS-A /server/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 /server/raid-controller/virtual-drive # commit-buffer

```

Renaming an Orphaned Virtual Drive on a Blade Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope server [<i>chassis-num/server-num</i> <i>dynamic-uuid</i>]	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # scope raid-controller <i>raid-contr-id</i> { sas sata }	Enters RAID controller chassis mode.
Step 3	UCS-A /chassis/server/raid-controller # scope virtual-drive <i>virtual-drive-id</i>	Enters virtual drive mode for the specified virtual drive.
Step 4	UCS-A /chassis/server/raid-controller/virtual-drive # set name <i>virtual-drive-name</i>	Specifies a name for the orphan virtual drive.
Step 5	UCS-A /chassis/server/raid-controller/virtual-drive # commit-buffer	Commits the transaction to the system configuration.

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

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

Renaming an Orphaned Virtual Drive on a Rack Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A # scope server <i>server-id</i>	Enters server mode for the specified server.
Step 2	UCS-A /server # scope raid-controller <i>raid-contr-id</i> {sas sata}	Enters RAID controller mode.
Step 3	UCS-A /server/raid-controller # scope virtual-drive <i>virtual-drive-id</i>	Enters virtual drive mode for the specified virtual drive.
Step 4	UCS-A /server/raid-controller/virtual-drive # set name <i>virtual-drive-name</i>	Specifies a name for the orphan virtual drive.
Step 5	UCS-A /server/raid-controller/virtual-drive # commit-buffer	Commits the transaction to the system configuration.

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

```
UCS-A /server # scope raid-controller 1 sas
UCS-A /server/raid-controller # scope virtual-drive 1060
UCS-A /server/raid-controller/virtual-drive # set name vdl
UCS-A /server/raid-controller/virtual-drive # commit-buffer
```

Boot Policy for Local Storage

You can specify the primary boot device for a storage controller as a local LUN or a JBOD disk. Each storage controller can have one primary boot device. However, in a storage profile, you can set only one device as the primary boot LUN.

Configuring the Boot Policy for a Local LUN

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the

	Command or Action	Purpose
		root organization mode, type / as the <i>org-name</i> .
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.
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. Important Cisco UCS Manager Release 2.2(4) does not support secondary boot order.
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/ <i>local-storage-device</i> # commit-buffer	Commits the transaction to the system configuration.

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/or template.

Configuring the Boot Policy for a Local JBOD Disk

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # scope boot-policy <i>policy-name</i>	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-jbod	Specifies a local JBOD disk as the local storage. JBOD is supported only on the following servers: <ul style="list-style-type: none"> • Cisco UCS B200 M3 blade server • Cisco UCS B260 M4 blade server • Cisco UCS B460 M4 blade server • Cisco UCS B200 M4 blade server • Cisco UCS C220 M4 rack-mount server • Cisco UCS C240 M4 rack-mount server • Cisco UCS C460 M4 rack-mount server
Step 6	UCS-A /org/boot-policy/storage/local/local-jbod # create local-disk-image-path { primary secondary }	Specifies the boot order for the local JBOD disk. Important Cisco UCS Manager Release 2.2(4) does not support secondary boot order.

	Command or Action	Purpose
Step 7	UCS-A /org/boot-policy/storage/local/local-jbod/local-disk-image-path # set slotnumber <i>slot_number</i>	Specifies the slot number of the JBOD disk that you want to boot from.
Step 8	UCS-A /org/boot-policy/storage/local/local-jbod/local-disk-image-path # commit-buffer	Commits the transaction to the system configuration.

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 JBOD disk 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-jbod
UCS-A /org/boot-policy/storage/local/local-jbod # create local-disk-image-path primary
UCS-A /org/boot-policy/storage/local/local-jbod/local-disk-image-path* # set slotnumber 5
UCS-A /org/boot-policy/storage/local/local-jbod/local-disk-image-path # commit-buffer
UCS-A /org/boot-policy/storage/local/local-jbod/local-disk-image-path #
```

What to Do Next

Include the boot policy in a service profile and/or 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 34
- [Deploying and Undeploying a LUN](#), on page 35
- [Renaming a Service Profile Referenced LUN](#), on page 36



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**. You can also manually change the admin state of the LUN to **Undeployed** and claim an orphan LUN.

If the LUN name is empty, set a LUN name before claiming it.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# scope org <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org# scope service-profile <i>service-profile-name</i>	Enters the specified service profile mode.
Step 3	UCS-A /org/service-profile# enter local-lun-ref <i>lun-name</i>	Enters the specified LUN.
Step 4	UCS-A /org/service-profile/local-lun-ref# set ref-name <i>ref-lun-name</i>	Sets the referenced LUN name. If this LUN name exists and the LUN is orphaned, it 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.

This examples shows how to preprovision a LUN name.

```
UCS-A# scope org
UCS-A /org # scope service-profile spl
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.

Procedure

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

	Command or Action	Purpose
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.

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

Procedure

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

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
```