Cisco UCS Manager CLI Configuration Guide, Release 2.2

First Published: 2013-12-11

Last Modified: 2016-07-27

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`.

Text Type	Indication
CLI commands	CLI command keywords appear in **this font**. Variables in a CLI command appear in *this font*.
[ ]	Elements in square brackets are optional.
{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: http://www.cisco.com/go/unifiedcomputing/b-series-doc.

For a complete list of all C-Series documentation, see the *Cisco UCS C-Series Servers Documentation Roadmap* available at the following URL: http://www.cisco.com/go/unifiedcomputing/c-series-doc.

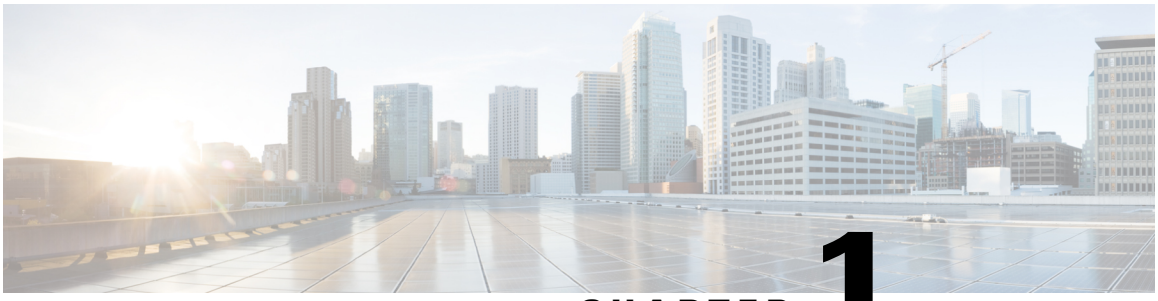
For information on supported firmware versions and supported UCS Manager versions for the rack servers that are integrated with the UCS Manager for management, refer to Release Bundle Contents for Cisco UCS Software.

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@cisco.com. We appreciate your feedback.



CHAPTER 1

Overview of Cisco Unified Computing System

This chapter includes the following sections:

About Cisco Unified Computing System

Cisco Unified Computing System (Cisco UCS) fuses access layer networking and servers. This high-performance, next-generation server system provides a data center with a high degree of workload agility and scalability.

The hardware and software components support Cisco's unified fabric, which runs multiple types of data center traffic over a single converged network adapter.

Architectural Simplification

The simplified architecture of Cisco UCS reduces the number of required devices and centralizes switching resources. By eliminating switching inside a chassis, network access-layer fragmentation is significantly reduced.

Cisco UCS implements Cisco unified fabric within racks and groups of racks, supporting Ethernet and Fibre Channel protocols over 10 Gigabit Cisco Data Center Ethernet and Fibre Channel over Ethernet (FCoE) links.

This radical simplification reduces the number of switches, cables, adapters, and management points by up to two-thirds. All devices in a Cisco UCS domain remain under a single management domain, which remains highly available through the use of redundant components.

High Availability

The management and data plane of Cisco UCS is designed for high availability and redundant access layer fabric interconnects. In addition, Cisco UCS supports existing high availability and disaster recovery solutions for the data center, such as data replication and application-level clustering technologies.

Scalability

A single Cisco UCS domain supports multiple chassis and their servers, all of which are administered through one Cisco UCS Manager. For more detailed information about the scalability, speak to your Cisco representative.

Flexibility

A Cisco UCS domain allows you to quickly align computing resources in the data center with rapidly changing business requirements. This built-in flexibility is determined by whether you choose to fully implement the stateless computing feature.

Pools of servers and other system resources can be applied as necessary to respond to workload fluctuations, support new applications, scale existing software and business services, and accommodate both scheduled and unscheduled downtime. Server identity can be abstracted into a mobile service profile that can be moved from server to server with minimal downtime and no need for additional network configuration.

With this level of flexibility, you can quickly and easily scale server capacity without having to change the server identity or reconfigure the server, LAN, or SAN. During a maintenance window, you can quickly do the following:

- Deploy new servers to meet unexpected workload demand and rebalance resources and traffic.
- Shut down an application, such as a database management system, on one server and then boot it up again on another server with increased I/O capacity and memory resources.

Optimized for Server Virtualization

Cisco UCS has been optimized to implement VM-FEX technology. This technology provides improved support for server virtualization, including better policy-based configuration and security, conformance with a company's operational model, and accommodation for VMware's VMotion.

Unified Fabric

With unified fabric, multiple types of data center traffic can run over a single Data Center Ethernet (DCE) network. Instead of having a series of different host bus adapters (HBAs) and network interface cards (NICs) present in a server, unified fabric uses a single converged network adapter. This type of adapter can carry LAN and SAN traffic on the same cable.

Cisco UCS uses Fibre Channel over Ethernet (FCoE) to carry Fibre Channel and Ethernet traffic on the same physical Ethernet connection between the fabric interconnect and the server. This connection terminates at a converged network adapter on the server, and the unified fabric terminates on the uplink ports of the fabric interconnect. On the core network, the LAN and SAN traffic remains separated. Cisco UCS does not require that you implement unified fabric across the data center.

The converged network adapter presents an Ethernet interface and Fibre Channel interface to the operating system. At the server, the operating system is not aware of the FCoE encapsulation because it sees a standard Fibre Channel HBA.

At the fabric interconnect, the server-facing Ethernet port receives the Ethernet and Fibre Channel traffic. The fabric interconnect (using Ethertype to differentiate the frames) separates the two traffic types. Ethernet frames and Fibre Channel frames are switched to their respective uplink interfaces.

Fibre Channel over Ethernet

Cisco UCS leverages Fibre Channel over Ethernet (FCoE) standard protocol to deliver Fibre Channel. The upper Fibre Channel layers are unchanged, so the Fibre Channel operational model is maintained. FCoE network management and configuration is similar to a native Fibre Channel network.

FCoE encapsulates Fibre Channel traffic over a physical Ethernet link. FCoE is encapsulated over Ethernet with the use of a dedicated Ethertype, 0x8906, so that FCoE traffic and standard Ethernet traffic can be carried on the same link. FCoE has been standardized by the ANSI T11 Standards Committee.

Fibre Channel traffic requires a lossless transport layer. Instead of the buffer-to-buffer credit system used by native Fibre Channel, FCoE depends upon the Ethernet link to implement lossless service.

Ethernet links on the fabric interconnect provide two mechanisms to ensure lossless transport for FCoE traffic:

- Link-level flow control
- Priority flow control

Link-Level Flow Control

IEEE 802.3x link-level flow control allows a congested receiver to signal the endpoint to pause data transmission for a short time. This link-level flow control pauses all traffic on the link.

The transmit and receive directions are separately configurable. By default, link-level flow control is disabled for both directions.

On each Ethernet interface, the fabric interconnect can enable either priority flow control or link-level flow control (but not both).

Priority Flow Control

The priority flow control (PFC) feature applies pause functionality to specific classes of traffic on the Ethernet link. For example, PFC can provide lossless service for the FCoE traffic, and best-effort service for the standard Ethernet traffic. PFC can provide different levels of service to specific classes of Ethernet traffic (using IEEE 802.1p traffic classes).

PFC decides whether to apply pause based on the IEEE 802.1p CoS value. When the fabric interconnect enables PFC, it configures the connected adapter to apply the pause functionality to packets with specific CoS values.

By default, the fabric interconnect negotiates to enable the PFC capability. If the negotiation succeeds, PFC is enabled and link-level flow control remains disabled (regardless of its configuration settings). If the PFC negotiation fails, you can either force PFC to be enabled on the interface or you can enable IEEE 802.x link-level flow control.

IPv6 Compliance

Cisco UCS Manager supports IPv6 addressing. This is important for the following reasons:

- IPv4 addresses have a shorter address space than IPv6 addresses.
- The number of unique IPv4 addresses is finite, and the allocation scheme used by the Internet addressing body has exacerbated the decline of available addresses.
- IPv6 addresses have a larger address space, and the pool of available IPv6 addresses is much greater than the pool of IPv4 addresses.
- Some customers require that all networking software they purchase be IPv6 standards compliant.

All features in Cisco UCS Manager that support IPv4 addressing also support IPv6.

Note Only public global unicast IPv6 addresses are supported.

IPv6 addresses can be used to configure inband access to management interfaces, the Cisco Cisco UCS Manager GUI, the KVM Console, and SSH over SoL.

Note IPv6 addresses are not supported for out-of-band access to CIMC.

Services Supported

Services that support IPv6 addresses include:

- HTTP and HTTPS
- SSH
- Telnet
- CIM XML
- SNMP
- Flash policy server

Client Support

External clients that support IPv6 addresses include:

- NTP
- DNS
- DHCP
- LDAP
- RADIUS
- TACACS+

- SSH
- Syslog
- vCenter
- Call Home
- NFS

Fabric Interconnects

Initial setup of the fabric interconnects supports the use of IPv6 addresses for the management IP address, default gateway and DNS servers.
In a cluster setup, if Fabric A is configured using IPv6 addresses and a cluster configuration is enabled, when Fabric B is subsequently configured, the setup process retrieves the address type from Fabric A, and prompts you to use IPv6 addresses. IPv4 addresses then need to be configured for both fabric interconnects for out-of-band (OOB) access after initial setup is complete.

Cisco UCS Manager and the fabric interconnects support OOB access over both IPv4 and IPv6 addresses.

Configurations that Support IPv6 Addressing

IPv6 addresses can be used to configure key ring certificate requests, SNMP traps, management IP pools and address blocks, service profiles, service profile templates, VLAN groups, backup and restore operations, the core file exporter, the Cisco UCS Manager Syslog, NTP servers, ARP targets in the Management Interface Monitoring policy, System Event Log (SEL) management, license management, firmware download, Call Home, and vCenter.

LDAP, RADIUS and TACACS+ authentication service provider configurations all support IPv6 addressing.

Servers

Cisco UCS blade and rack servers can be configured to use static IPv6 addresses. Inband access to the server Cisco Integrated Management Controller (CIMC) is possible using IPv6 addresses. Inband access is faster because management traffic flows between the fabric interconnects and the servers using the higher-bandwidth uplink port.

Note Only Cisco UCS M3 and M4 servers support IPv6 addresses. IPv6 addressing for Cisco UCS M1 and M2 servers is not supported.

Server Architecture and Connectivity

Overview of Service Profiles

Service profiles are the central concept of Cisco UCS. Each service profile serves a specific purpose: ensuring that the associated server hardware has the configuration required to support the applications it will host.

The service profile maintains configuration information about the server hardware, interfaces, fabric connectivity, and server and network identity. This information is stored in a format that you can manage

through Cisco UCS Manager. All service profiles are centrally managed and stored in a database on the fabric interconnect.

Every server must be associated with a service profile.

Important At any given time, each server can be associated with only one service profile. Similarly, each service profile can be associated with only one server at a time.

After you associate a service profile with a server, the server is ready to have an operating system and applications installed, and you can use the service profile to review the configuration of the server. If the server associated with a service profile fails, the service profile does not automatically fail over to another server.

When a service profile is disassociated from a server, the identity and connectivity information for the server is reset to factory defaults.

Network Connectivity through Service Profiles

Each service profile specifies the LAN and SAN network connections for the server through the Cisco UCS infrastructure and out to the external network. You do not need to manually configure the network connections for Cisco UCS servers and other components. All network configuration is performed through the service profile.

When you associate a service profile with a server, the Cisco UCS internal fabric is configured with the information in the service profile. If the profile was previously associated with a different server, the network infrastructure reconfigures to support identical network connectivity to the new server.

Configuration through Service Profiles

A service profile can take advantage of resource pools and policies to handle server and connectivity configuration.

Hardware Components Configured by Service Profiles

When a service profile is associated with a server, the following components are configured according to the data in the profile:

- Server, including BIOS and CIMC
- Adapters
- Fabric interconnects

You do not need to configure these hardware components directly.

Server Identity Management through Service Profiles

You can use the network and device identities burned into the server hardware at manufacture or you can use identities that you specify in the associated service profile either directly or through identity pools, such as MAC, WWN, and UUID.

The following are examples of configuration information that you can include in a service profile:

- Profile name and description

- Unique server identity (UUID)
- LAN connectivity attributes, such as the MAC address
- SAN connectivity attributes, such as the WWN

Operational Aspects configured by Service Profiles

You can configure some of the operational functions for a server in a service profile, such as the following:

- Firmware packages and versions
- Operating system boot order and configuration
- IPMI and KVM access

vNIC Configuration by Service Profiles

A vNIC is a virtualized network interface that is configured on a physical network adapter and appears to be a physical NIC to the operating system of the server. The type of adapter in the system determines how many vNICs you can create. For example, a converged network adapter has two NICs, which means you can create a maximum of two vNICs for each adapter.

A vNIC communicates over Ethernet and handles LAN traffic. At a minimum, each vNIC must be configured with a name and with fabric and network connectivity.

vHBA Configuration by Service Profiles

A vHBA is a virtualized host bus adapter that is configured on a physical network adapter and appears to be a physical HBA to the operating system of the server. The type of adapter in the system determines how many vHBAs you can create. For example, a converged network adapter has two HBAs, which means you can create a maximum of two vHBAs for each of those adapters. In contrast, a network interface card does not have any HBAs, which means you cannot create any vHBAs for those adapters.

A vHBA communicates over FCoE and handles SAN traffic. At a minimum, each vHBA must be configured with a name and fabric connectivity.

Service Profiles that Override Server Identity

This type of service profile provides the maximum amount of flexibility and control. This profile allows you to override the identity values that are on the server at the time of association and use the resource pools and policies set up in Cisco UCS Manager to automate some administration tasks.

You can disassociate this service profile from one server, then associate it with another server. This re-association can be done either manually or through an automated server pool policy. The burned-in settings, such as UUID and MAC address on the new server are overwritten with the configuration in the service profile. As a result, the change in the server is transparent to your network. You do not need to reconfigure any component or application on your network to begin using the new server.

This profile allows you to take advantage of and manage system resources through resource pools and policies, such as the following:

- Virtualized identity information, including pools of MAC addresses, WWN addresses, and UUIDs
- Ethernet and Fibre Channel adapter profile policies
- Firmware package policies

- Operating system boot order policies

Unless the service profile contains power management policies, a server pool qualification policy, or another policy that requires a specific hardware configuration, you can use the profile for any type of server in the Cisco UCS domain.

You can associate these service profiles with either a rack-mount server or a blade server. The ability to migrate the service profile depends upon whether you choose to restrict migration of the service profile.

Note If you choose not to restrict migration, Cisco UCS Manager does not perform any compatibility checks on the new server before migrating the existing service profile. If the hardware of both servers are not similar, the association might fail.

Service Profiles that Inherit Server Identity

This hardware-based service profile is the simplest to use and create. This profile uses the default values in the server and mimics the management of a rack-mounted server. It is tied to a specific server and cannot be moved or migrated to another server.

You do not need to create pools or configuration policies to use this service profile.

This service profile inherits and applies the identity and configuration information that is present at the time of association, such as the following:

- MAC addresses for the two NICs
- For a converged network adapter or a virtual interface card, the WWN addresses for the two HBAs
- BIOS versions
- Server UUID

Important The server identity and configuration information inherited through this service profile might not have the values burned into the server hardware at the manufacturer if those values were changed before this profile is associated with the server.

Initial and Existing Templates

With a service profile template, you can quickly create several service profiles with the same basic parameters, such as the number of vNICs and vHBAs, and with identity information drawn from the same pools.

Tip If you need only one service profile with similar values to an existing service profile, you can clone a service profile in the Cisco UCS Manager GUI.

For example, if you need several service profiles with similar values to configure servers to host database software, you can create a service profile template, either manually or from an existing service profile. You then use the template to create the service profiles.

Cisco UCS supports the following types of service profile templates:

Initial template

Service profiles created from an initial template inherit all the properties of the template. Service profiles created from an initial service profile template are bound to the template. However, changes to the initial template do not *automatically* propagate to the bound service profiles. If you want to propagate changes to bound service profiles, unbind and rebind the service profile to the initial template.

Updating template

Service profiles created from an updating template inherit all the properties of the template and remain connected to the template. Any changes to the template automatically update the service profiles created from the template.

Note Service profiles that are created from the initial template and normal service profiles fetch the lowest available IDs in the sequential pool when you pres**Reset**.

Service profiles created from updating template might attempt to retain the same ID when you press**Reset** even when lower IDs of sequential pool are free.

Policies

Policies determine how Cisco UCS components will act in specific circumstances. You can create multiple instances of most policies. For example, you might want different boot policies, so that some servers can PXE boot, some can SAN boot, and others can boot from local storage.

Policies allow separation of functions within the system. A subject matter expert can define policies that are used in a service profile, which is created by someone without that subject matter expertise. For example, a LAN administrator can create adapter policies and quality of service policies for the system. These policies can then be used in a service profile that is created by someone who has limited or no subject matter expertise with LAN administration.

You can create and use two types of policies in Cisco UCS Manager:

- Configuration policies that configure the servers and other components
- Operational policies that control certain management, monitoring, and access control functions

Pools

Pools are collections of identities, or physical or logical resources, that are available in the system. All pools increase the flexibility of service profiles and allow you to centrally manage your system resources.

You can use pools to segment unconfigured servers or available ranges of server identity information into groupings that make sense for the data center. For example, if you create a pool of unconfigured servers with similar characteristics and include that pool in a service profile, you can use a policy to associate that service profile with an available, unconfigured server.

If you pool identifying information, such as MAC addresses, you can preassign ranges for servers that host specific applications. For example, you can configure all database servers within the same range of MAC addresses, UUIDs, and WWNs.

Domain Pools

Domain Pools are defined locally in a Cisco UCS domain, and can only be used in that Cisco UCS domain.

Global Pools

Global Pools are defined in Cisco UCS Central, and can be shared between Cisco UCS domains. If a Cisco UCS domain is registered with Cisco UCS Central, you can assign **Global Pools** in Cisco UCS Manager.

CIMC Inband Management

A driving factor for providing inband management access to Cisco Integrated Management Controller (CIMC) is the desire to separate tenant traffic from provider traffic in multi-tenant, public or private service provider cloud deployments. Out-of-band (OOB) management traffic moves in and out of the fabric interconnects and traverses the management plane via the management port. This has the potential to cause bottlenecks and affect the CPU bandwidth in the management ports.

Inband management allows CIMC traffic to take the same path as the data traffic, entering and exiting the fabric interconnects via the uplink ports. The higher bandwidth available to the uplink ports means that inband access greatly speeds up management traffic, and reduces the risk of traffic bottlenecks and CPU stress. Both out-of-band (OOB) and inband address pools can be configured for management access in Cisco UCS Manager. Out-of-band access only supports IPv4 addresses. Inband access supports both IPv4 and IPv6 addresses, which allows for single or dual stack management.

The two OOB management interface addresses that can be configured in Cisco UCS Manager blade and rack servers are:

- An OOB IPv4 address assigned to the physical server via the global ext-mgmt pool
- An OOB IPv4 address derived from a service profile associated with the physical server

In addition, up to four inband management interface addresses can be configured:

- An inband IPv4 address assigned to the physical server
- An inband IPv4 address derived from a service profile associated with the physical server
- An inband IPv6 address assigned to the physical server
- An inband IPv6 address derived from a service profile associated with the physical server

Multiple inband management IP addresses for each server support additional CIMC sessions. When you configure both OOB and inband addresses, users can choose from a list of those addresses in the KVM Console dialog box when they launch KVM from a server, SSH to SoL, a service profile, the KVM Launch Manager, or from the Cisco UCS Manager GUI web URL.

CIMC inband access supports the following services:

- KVM Console
- SSH to CIMC for SoL

- vMedia for ISO, virtual CD/DVD, removable disk, and floppy

Note Only Cisco UCS M3 and M4 servers support inband CIMC access. Inband CIMC access for Cisco UCS M1 and M2 servers is not supported.

You can configure inband IP pools of IPv4 or IPv6 addresses and use them to assign addresses to servers. You can configure inband VLAN groups and assign them to servers using service profiles.

You need to configure an Inband Profile with an Inband VLAN group to select an Inband Network (VLAN) in Service Profiles and Service Profile templates.

You can configure the network and IP pool name in an Inband profile to assign Inband CIMC addresses to Cisco UCS M3 and M4 servers.

Inband Management Support

Inband management access is supported in Cisco UCS Manager for the following external services:

- KVM
- vMedia for ISO, virtual CD/DVD, removable disk, and floppy
- SSH to SoL

You can configure inband IP pools of IPv4 or IPv6 addresses and use them to assign addresses to servers. You can configure inband VLAN groups and assign them to servers using service profiles.

Traffic Management

Oversubscription

Oversubscription occurs when multiple network devices are connected to the same fabric interconnect port. This practice optimizes fabric interconnect use, since ports rarely run at maximum speed for any length of time. As a result, when configured correctly, oversubscription allows you to take advantage of unused bandwidth. However, incorrectly configured oversubscription can result in contention for bandwidth and a lower quality of service to all services that use the oversubscribed port.

For example, oversubscription can occur if four servers share a single uplink port, and all four servers attempt to send data at a cumulative rate higher than available bandwidth of uplink port.

Oversubscription Considerations

The following elements can impact how you configure oversubscription in a Cisco UCS domain:

Ratio of Server-Facing Ports to Uplink Ports

You need to know what how many server-facing ports and uplink ports are in the system, because that ratio can impact performance. For example, if your system has twenty ports that can communicate down to the

servers and only two ports that can communicate up to the network, your uplink ports will be oversubscribed. In this situation, the amount of traffic created by the servers can also affect performance.

Number of Uplink Ports from Fabric Interconnect to Network

You can choose to add more uplink ports between the Cisco UCS fabric interconnect and the upper layers of the LAN to increase bandwidth. In Cisco UCS, you must have at least one uplink port per fabric interconnect to ensure that all servers and NICs to have access to the LAN. The number of LAN uplinks should be determined by the aggregate bandwidth needed by all Cisco UCS servers.

For the 6100 series fabric interconnects, Fibre Channel uplink ports are available on the expansion slots only. You must add more expansion slots to increase number of available Fibre Channel uplinks. Ethernet uplink ports can exist on the fixed slot and on expansion slots.

For the 6200 series fabric interconnects running Cisco UCS Manager, version 2.0 and higher, Ethernet uplink ports and Fibre Channel uplink ports are both configurable on the base module, as well as on the expansion module.

For example, if you have two Cisco UCS 5100 series chassis that are fully populated with half width Cisco UCS B200-M1 servers, you have 16 servers. In a cluster configuration, with one LAN uplink per fabric interconnect, these 16 servers share 20GbE of LAN bandwidth. If more capacity is needed, more uplinks from the fabric interconnect should be added. We recommend that you have symmetric configuration of the uplink in cluster configurations. In the same example, if 4 uplinks are used in each fabric interconnect, the 16 servers are sharing 80 GB of bandwidth, so each has approximately 5 GB of capacity. When multiple uplinks are used on a Cisco UCS fabric interconnect the network design team should consider using a port channel to make best use of the capacity.

Number of Uplink Ports from I/O Module to Fabric Interconnect

You can choose to add more bandwidth between I/O module and fabric interconnect by using more uplink ports and increasing the number of cables. In Cisco UCS, you can have one, two, or four cables connecting a I/O module to a Cisco UCS 6100 series fabric interconnect. You can have up to eight cables if you're connecting a 2208 I/O module and a 6248 fabric interconnect. The number of cables determines the number of active uplink ports and the oversubscription ratio.

Number of Active Links from Server to Fabric Interconnect

The amount of non-oversubscribed bandwidth available to each server depends on the number of I/O modules used and the number of cables used to connect those I/O modules to the fabric interconnects. Having a second I/O module in place provides additional bandwidth and redundancy to the servers. This level of flexibility in design ensures that you can provide anywhere from 80 Gbps (two I/O modules with four links each) to 10 Gbps (one I/O module with one link) to the chassis.

With 80 Gbps to the chassis, each half-width server in the Cisco UCS domain can get up to 10 Gbps in a non-oversubscribed configuration, with an ability to use up to 20 Gbps with 2:1 oversubscription.

Guidelines for Estimating Oversubscription

When you estimate the optimal oversubscription ratio for a fabric interconnect port, consider the following guidelines:

Cost/Performance Slider

The prioritization of cost and performance is different for each data center and has a direct impact on the configuration of oversubscription. When you plan hardware usage for oversubscription, you need to know where the data center is located on this slider. For example, oversubscription can be minimized if the data center is more concerned with performance than cost. However, cost is a significant factor in most data centers, and oversubscription requires careful planning.

Bandwidth Usage

The estimated bandwidth that you expect each server to actually use is important when you determine the assignment of each server to a fabric interconnect port and, as a result, the oversubscription ratio of the ports. For oversubscription, you must consider how many GBs of traffic the server will consume on average, the ratio of configured bandwidth to used bandwidth, and the times when high bandwidth use will occur.

Network Type

The network type is only relevant to traffic on uplink ports, because FCoE does not exist outside Cisco UCS. The rest of the data center network only differentiates between LAN and SAN traffic. Therefore, you do not need to take the network type into consideration when you estimate oversubscription of a fabric interconnect port.

Pinning

Pinning in Cisco UCS is only relevant to uplink ports. You can pin Ethernet or FCoE traffic from a given server to a specific uplink Ethernet port or uplink FC port.

When you pin the NIC and HBA of both physical and virtual servers to uplink ports, you give the fabric interconnect greater control over the unified fabric. This control ensures more optimal utilization of uplink port bandwidth.

Cisco UCS uses pin groups to manage which NICs, vNICs, HBAs, and vHBAs are pinned to an uplink port. To configure pinning for a server, you can either assign a pin group directly, or include a pin group in a vNIC policy, and then add that vNIC policy to the service profile assigned to that server. All traffic from the vNIC or vHBA on the server travels through the I/O module to the same uplink port.

Guidelines for Pinning

When you determine the optimal configuration for pin groups and pinning for an uplink port, consider the estimated bandwidth usage for the servers. If you know that some servers in the system will use a lot of bandwidth, ensure that you pin these servers to different uplink ports.

Quality of Service

Cisco UCS provides the following methods to implement quality of service:

- System classes that specify the global configuration for certain types of traffic across the entire system
- QoS policies that assign system classes for individual vNICs
- Flow control policies that determine how uplink Ethernet ports handle pause frames

Global QoS changes made to the QoS system class may result in brief data-plane interruptions for all traffic. Some examples of such changes are:

- Changing the MTU size for an enabled class
- Changing packet drop for an enabled class
- Changing the CoS value for an enabled class

Guidelines and Limitations for Quality of Service on Cisco UCS 6300 Series Fabric Interconnect

- Cisco UCS 6300 Series Fabric Interconnect uses a shared buffer for all system classes.
- Multicast optimization is not supported.
- When you change the QoS parameters for any class causes traffic disruption to all classes. The following table lists the changes in the QoS system class and the conditions that trigger a system reboot.

QoS System class status	Condition	FI Reboot Status
Enabled	Change between drop and no drop	Yes
No-drop	Change between enable and disable	Yes
Enable and no-drop	Change in MTU size	Yes

- The subordinate FI reboots first as a result of the change in the QoS system class. The primary FI reboots only after you acknowledge it in **Pending Activities**.

Guidelines and Limitations for Quality of Service on Cisco UCS Mini

- Cisco UCS Mini uses a shared buffer for all system classes.
- The bronze class shares the buffer with SPAN. We recommend using either SPAN or the bronze class.
- Multicast optimization is not supported.
- Changing the QoS parameters for any class causes traffic disruption to all classes.
- When mixing Ethernet and FC or FCoE traffic, the bandwidth distribution is not equal.
- Multiple streams of traffic from the same class may not be distributed equally.
- Use the same CoS values for all no-drop policies to avoid any FC or FCoE performance issues.
- Only the platinum and gold classes support no-drop policies.

System Classes

Cisco UCS uses Data Center Ethernet (DCE) to handle all traffic inside a Cisco UCS domain. This industry standard enhancement to Ethernet divides the bandwidth of the Ethernet pipe into eight virtual lanes. Two virtual lanes are reserved for internal system and management traffic. You can configure quality of service

(QoS) for the other six virtual lanes. System classes determine how the DCE bandwidth in these six virtual lanes is allocated across the entire Cisco UCS domain.

Each system class reserves a specific segment of the bandwidth for a specific type of traffic, which provides a level of traffic management, even in an oversubscribed system. For example, you can configure the Fibre Channel Priority system class to determine the percentage of DCE bandwidth allocated to FCoE traffic.

The following table describes the system classes that you can configure.

Table 1: System Classes

System Class	Description
Platinum Gold Silver Bronze	A configurable set of system classes that you can include in the QoS policy for a service profile. Each system class manages one lane of traffic. All properties of these system classes are available for you to assign custom settings and policies. For Cisco UCS Mini, packet drop can only be disabled on the platinum and gold classes. Only one platinum and one gold class can be configured as a no drop class at a time.
Best Effort	A system class that sets the quality of service for the lane reserved for basic Ethernet traffic. Some properties of this system class are preset and cannot be modified. For example, this class has a drop policy that allows it to drop data packets if required. You cannot disable this system class.
Fibre Channel	A system class that sets the quality of service for the lane reserved for Fibre Channel over Ethernet traffic. Some properties of this system class are preset and cannot be modified. For example, this class has a no-drop policy that ensures it never drops data packets. You cannot disable this system class. **Note** FCoE traffic has a reserved QoS system class that should not be used by any other type of traffic. If any other type of traffic has a CoS value that is used by FCoE, the value is remarked to 0.

Quality of Service Policy

A quality of service (QoS) policy assigns a system class to the outgoing traffic for a vNIC or vHBA. This system class determines the quality of service for that traffic. For certain adapters, you can also specify additional controls on the outgoing traffic, such as burst and rate.

You must include a QoS policy in a vNIC policy or vHBA policy and then include that policy in a service profile to configure the vNIC or vHBA.

Flow Control Policy

Flow control policies determine whether the uplink Ethernet ports in a Cisco UCS domain send and receive IEEE 802.3x pause frames when the receive buffer for a port fills. These pause frames request that the transmitting port stop sending data for a few milliseconds until the buffer clears.

For flow control to work between a LAN port and an uplink Ethernet port, you must enable the corresponding receive and send flow control parameters for both ports. For Cisco UCS, the flow control policies configure these parameters.

When you enable the send function, the uplink Ethernet port sends a pause request to the network port if the incoming packet rate becomes too high. The pause remains in effect for a few milliseconds before traffic is reset to normal levels. If you enable the receive function, the uplink Ethernet port honors all pause requests from the network port. All traffic is halted on that uplink port until the network port cancels the pause request.

Because you assign the flow control policy to the port, changes to the policy have an immediate effect on how the port reacts to a pause frame or a full receive buffer.

Opt-In Features

Each Cisco UCS domain is licensed for all functionality. Depending upon how the system is configured, you can decide to opt in to some features or opt out of them for easier integration into existing environment. If a process change happens, you can change your system configuration and include one or both of the opt-in features.

The opt-in features are as follows:

- Stateless computing, which takes advantage of mobile service profiles with pools and policies where each component, such as a server or an adapter, is stateless.
- Multi-tenancy, which uses organizations and role-based access control to divide the system into smaller logical segments.

Stateless Computing

Stateless computing allows you to use a service profile to apply the personality of one server to a different server in the same Cisco UCS domain. The personality of the server includes the elements that identify that server and make it unique in the Cisco UCS domain. If you change any of these elements, the server could lose its ability to access, use, or even achieve booted status.

The elements that make up a server's personality include the following:

- Firmware versions
- UUID (used for server identification)
- MAC address (used for LAN connectivity)
- World Wide Names (used for SAN connectivity)
- Boot settings

Stateless computing creates a dynamic server environment with highly flexible servers. Every physical server in a Cisco UCS domain remains anonymous until you associate a service profile with it, then the server gets

the identity configured in the service profile. If you no longer need a business service on that server, you can shut it down, disassociate the service profile, and then associate another service profile to create a different identity for the same physical server. The "new" server can then host another business service.

To take full advantage of the flexibility of statelessness, the optional local disks on the servers should only be used for swap or temp space and not to store operating system or application data.

You can choose to fully implement stateless computing for all physical servers in a Cisco UCS domain, to not have any stateless servers, or to have a mix of the two types.

If You Opt In to Stateless Computing

Each physical server in the Cisco UCS domain is defined through a service profile. Any server can be used to host one set of applications, then reassigned to another set of applications or business services, if required by the needs of the data center.

You create service profiles that point to policies and pools of resources that are defined in the Cisco UCS domain. The server pools, WWN pools, and MAC pools ensure that all unassigned resources are available on an as-needed basis. For example, if a physical server fails, you can immediately assign the service profile to another server. Because the service profile provides the new server with the same identity as the original server, including WWN and MAC address, the rest of the data center infrastructure sees it as the same server and you do not need to make any configuration changes in the LAN or SAN.

If You Opt Out of Stateless Computing

Each server in the Cisco UCS domain is treated as a traditional rack mount server.

You create service profiles that inherit the identify information burned into the hardware and use these profiles to configure LAN or SAN connectivity for the server. However, if the server hardware fails, you cannot reassign the service profile to a new server.

Multitenancy

Multi-tenancy allows you to divide the large physical infrastructure of an Cisco UCS domain into logical entities known as organizations. As a result, you can achieve a logical isolation between organizations without providing a dedicated physical infrastructure for each organization.

You can assign unique resources to each tenant through the related organization in the multi-tenant environment. These resources can include different policies, pools, and quality of service definitions. You can also implement locales to assign or restrict user privileges and roles by organization, if you do not want all users to have access to all organizations.

If you set up a multi-tenant environment, all organizations are hierarchical. The top-level organization is always root. The policies and pools that you create in root are system-wide and are available to all organizations in the system. However, any policies and pools created in other organizations are only available to organizations that are above it in the same hierarchy. For example, if a system has organizations named Finance and HR that are not in the same hierarchy, Finance cannot use any policies in the HR organization, and HR cannot access any policies in the Finance organization. However, both Finance and HR can use policies and pools in the root organization.

If you create organizations in a mult-itenant environment, you can also set up one or more of the following for each organization or for a sub-organization in the same hierarchy:

- Resource pools
- Policies

- Service profiles
- Service profile templates

If You Opt In to Multitenancy

Each Cisco UCS domain is divided into several distinct organizations. The types of organizations you create in a multitenancy implementation depends upon the business needs of the company. Examples include organizations that represent the following:

- Enterprise groups or divisions within a company, such as marketing, finance, engineering, or human resources
- Different customers or name service domains, for service providers

You can create locales to ensure that users have access only to those organizations that they are authorized to administer.

If You Opt Out of Multitenancy

The Cisco UCS domain remains a single logical entity with everything in the root organization. All policies and resource pools can be assigned to any server in the Cisco UCS domain.

Virtualization in Cisco UCS

Overview of Virtualization

Virtualization allows you to create multiple Virtual Machines (VMs) to run in isolation, side by side on the same physical machine.

Each virtual machine has its own set of virtual hardware (RAM, CPU, NIC) upon which an operating system and fully configured applications are loaded. The operating system sees a consistent, normalized set of hardware regardless of the actual physical hardware components.

In a virtual machine, both hardware and software are encapsulated in a single file for rapid provisioning and moving between physical servers. You can move a virtual machine, within seconds, from one physical server to another for zero-downtime maintenance and continuous workload consolidation.

The virtual hardware makes it possible for many servers, each running in an independent virtual machine, to run on a single physical server. The advantages of virtualization include better use of computing resources, greater server density, and seamless server migration.

Overview of Cisco Virtual Machine Fabric Extender

A virtualized server implementation consists of one or more VMs that run as guests on a single physical server. The guest VMs are hosted and managed by a software layer called the hypervisor or virtual machine manager (VMM). Typically, the hypervisor presents a virtual network interface to each VM and performs Layer 2 switching of traffic from a VM to other local VMs or to another interface to the external network.

Working with a Cisco virtual interface card (VIC) adapter, the Cisco Virtual Machine Fabric Extender (VM-FEX) bypasses software-based switching of VM traffic by the hypervisor for external hardware-based

switching in the fabric interconnect. This method reduces the load on the server CPU, provides faster switching, and enables you to apply a rich set of network management features to local and remote traffic.

VM-FEX extends the IEEE 802.1Qbh port extender architecture to the VMs by providing each VM interface with a virtual Peripheral Component Interconnect Express (PCIe) device and a virtual port on a switch. This solution allows precise rate limiting and quality of service (QoS) guarantees on the VM interface.

Virtualization with Network Interface Cards and Converged Network Adapters

Network interface card (NIC) and converged network adapters support virtualized environments with the standard VMware integration with ESX installed on the server and all virtual machine management performed through the VC.

Portability of Virtual Machines

If you implement service profiles you retain the ability to easily move a server identity from one server to another. After you image the new server, the ESX treats that server as if it were the original.

Communication between Virtual Machines on the Same Server

These adapters implement the standard communications between virtual machines on the same server. If an ESX host includes multiple virtual machines, all communications must go through the virtual switch on the server.

If the system uses the native VMware drivers, the virtual switch is out of the network administrator's domain and is not subject to any network policies. As a result, for example, QoS policies on the network are not applied to any data packets traveling from VM1 to VM2 through the virtual switch.

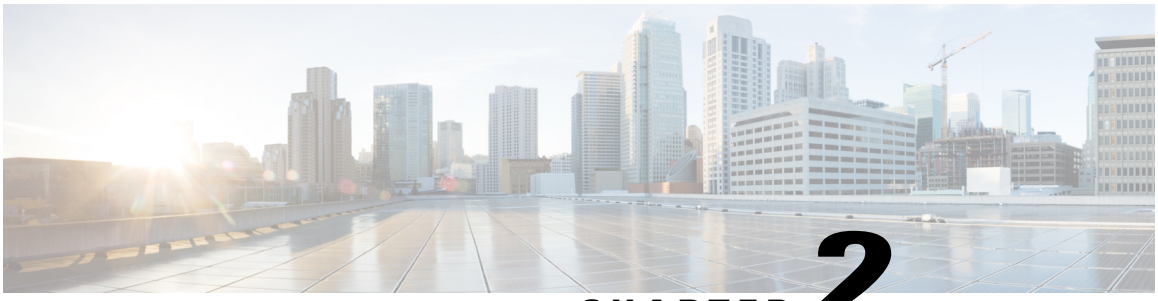
If the system includes another virtual switch, such as the Nexus 1000, that virtual switch is subject to the network policies configured on that switch by the network administrator.

Virtualization with a Virtual Interface Card Adapter

A Cisco VIC adapter is a converged network adapter (CNA) that is designed for both bare metal and VM-based deployments. The VIC adapter supports static or dynamic virtualized interfaces, which includes up to 128 virtual network interface cards (vNICs).

There are two types of vNICs used with the VIC adapter—static and dynamic. A static vNIC is a device that is visible to the OS or hypervisor. Dynamic vNICs are used for VM-FEX by which a VM is connected to a veth port on the Fabric Interconnect.

VIC adapters support VM-FEX to provide hardware-based switching of traffic to and from virtual machine interfaces.



CHAPTER 2

Overview of Cisco UCS Manager

This chapter includes the following sections:

About Cisco UCS Manager

Cisco UCS Manager is the management system for all components in a Cisco UCS domain. Cisco UCS Manager runs within the fabric interconnect. You can use any of the interfaces available with this management service to access, configure, administer, and monitor the network and server resources for all chassis connected to the fabric interconnect.

Multiple Management Interfaces

Cisco UCS Manager includes the following interfaces you can use to manage a Cisco UCS domain:

- Cisco UCS Manager GUI
- Cisco UCS Manager CLI
- XML API
- KVM
- IPMI

Almost all tasks can be performed in any of the interfaces, and the results of tasks performed in one interface are automatically displayed in another.

However, you cannot do the following:

- Use Cisco UCS Manager GUI to invoke Cisco UCS Manager CLI.
- View the results of a command invoked through Cisco UCS Manager CLI in Cisco UCS Manager GUI.

- Generate CLI output from Cisco UCS Manager GUI.

Centralized Management

Cisco UCS Manager centralizes the management of resources and devices, rather than using multiple management points. This centralized management includes management of the following devices in a Cisco UCS domain:

- Fabric interconnects.
- Software switches for virtual servers.
- Power and environmental management for chassis and servers.
- Configuration and firmware updates for server network interfaces (Ethernet NICs and converged network adapters).
- Firmware and BIOS settings for servers.

Support for Virtual and Physical Servers

Cisco UCS Manager abstracts server state information—including server identity, I/O configuration, MAC addresses and World Wide Names, firmware revision, and network profiles—into a service profile. You can apply the service profile to any server resource in the system, providing the same flexibility and support to physical servers, virtual servers, and virtual machines connected to a virtual device provided by a VIC adapter.

Role-Based Administration and Multi-Tenancy Support

Cisco UCS Manager supports flexibly defined roles so that data centers can use the same best practices with which they manage discrete servers, storage, and networks to operate a Cisco UCS domain. You can create user roles with privileges that reflect user responsibilities in the data center. For example, you can create the following:

- Server administrator roles with control over server-related configurations.
- Storage administrator roles with control over tasks related to the SAN.
- Network administrator roles with control over tasks related to the LAN.

Cisco UCS is multi-tenancy ready, exposing primitives that allow systems management software using the API to get controlled access to Cisco UCS resources. In a multi-tenancy environment, Cisco UCS Manager enables you to create locales for user roles that can limit the scope of a user to a particular organization.

Tasks You Can Perform in Cisco UCS Manager

You can use Cisco UCS Manager to perform management tasks for all physical and virtual devices within a Cisco UCS domain.

Cisco UCS Hardware Management

You can use Cisco UCS Manager to manage all hardware within a Cisco UCS domain, including the following:

- Chassis
- Servers

- Fabric interconnects
- Fans
- Ports
- Interface cards
- I/O modules

Cisco UCS Resource Management

You can use Cisco UCS Manager to create and manage all resources within a Cisco UCS domain, including the following:

- Servers
- WWN addresses
- MAC addresses
- UUIDs
- Bandwidth

Server Administration

A server administrator can use Cisco UCS Manager to perform server management tasks within a Cisco UCS domain, including the following:

- Create server pools and policies related to those pools, such as qualification policies
- Create policies for the servers, such as discovery policies, scrub policies, and IPMI policies
- Create service profiles and, if desired, service profile templates
- Apply service profiles to servers
- Monitor faults, alarms, and the status of equipment

Network Administration

A network administrator can use Cisco UCS Manager to perform tasks required to create LAN configuration for a Cisco UCS domain, including the following:

- Configure uplink ports, port channels, and LAN PIN groups
- Create VLANs
- Configure the quality of service classes and definitions
- Create the pools and policies related to network configuration, such as MAC address pools and Ethernet adapter profiles

Storage Administration

A storage administrator can use Cisco UCS Manager to perform tasks required to create SAN configuration for a Cisco UCS domain, including the following:

- Configure ports, port channels, and SAN PIN groups

- Create VSANs
- Configure the quality of service classes and definitions
- Create the pools and policies related to the network configuration, such as WWN pools and Fibre Channel adapter profiles

Tasks You Cannot Perform in Cisco UCS Manager

You cannot use Cisco UCS Manager to perform certain system management tasks that are not specifically related to device management within a Cisco UCS domain.

No Cross-System Management

You cannot use Cisco UCS Manager to manage systems or devices that are outside the Cisco UCS domain where Cisco UCS Manager is located. For example, you cannot manage heterogeneous environments, such as non-Cisco UCS x86 systems, SPARC systems, or PowerPC systems.

No Operating System or Application Provisioning or Management

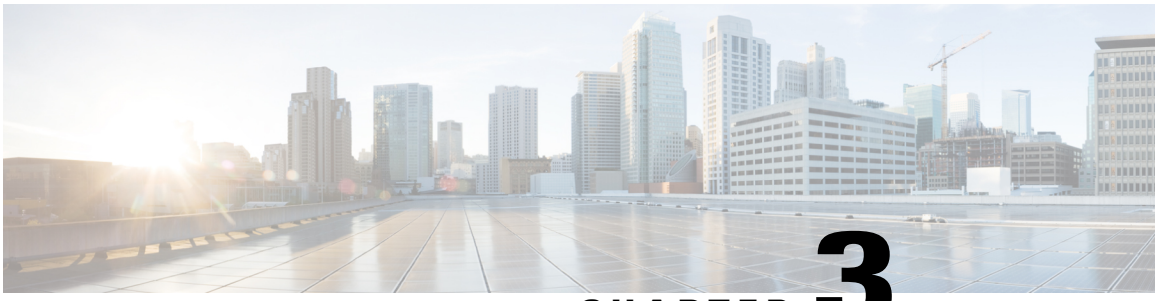
Cisco UCS Manager provisions servers and, as a result, exists below the operating system on a server. Therefore, you cannot use it to provision or manage operating systems or applications on servers. For example, you cannot do the following:

- Deploy an OS, such as Windows or Linux
- Deploy patches for software, such as an OS or an application
- Install base software components, such as anti-virus software, monitoring agents, or backup clients
- Install software applications, such as databases, application server software, or web servers
- Perform operator actions, including restarting an Oracle database, restarting printer queues, or handling non-Cisco UCS user accounts
- Configure or manage external storage on the SAN or NAS storage

Cisco UCS Manager in a High Availability Environment

In a high availability environment with two fabric interconnects, you can run a separate instance of Cisco UCS Manager on each fabric interconnect. The Cisco UCS Manager on the primary fabric interconnect acts as the primary management instance, and the Cisco UCS Manager on the other fabric interconnect is the subordinate management instance.

The two instances of Cisco UCS Manager communicate across a private network between the L1 and L2 Ethernet ports on the fabric interconnects. Configuration and status information is communicated across this private network to ensure that all management information is replicated. This ongoing communication ensures that the management information for Cisco UCS persists even if the primary fabric interconnect fails. In addition, the "floating" management IP address that runs on the primary Cisco UCS Manager ensures a smooth transition in the event of a failover to the subordinate fabric interconnect.



CHAPTER 3

Overview of Cisco UCS Manager CLI

This chapter includes the following sections:

Managed Objects

Cisco UCS uses a managed object model, where managed objects are abstract representations of physical or logical entities that can be managed. For example, servers, chassis, I/O cards, and processors are physical entities represented as managed objects, and resource pools, user roles, service profiles, and policies are logical entities represented as managed objects.

Managed objects may have one or more associated properties that can be configured.

Command Modes

The CLI is organized into a hierarchy of command modes, with the EXEC mode being the highest-level mode of the hierarchy. Higher-level modes branch into lower-level modes. You use **create**, **enter**, and **scope** commands to move from higher-level modes to modes in the next lower level , and you use the **exit** command to move up one level in the mode hierarchy.

Note Most command modes are associated with managed objects, so you must create an object before you can access the mode associated with that object. You use **create** and **enter** commands to create managed objects for the modes being accessed. The **scope** commands do not create managed objects and can only access modes for which managed objects already exist.

Each mode contains a set of commands that can be entered in that mode. Most of the commands available in each mode pertain to the associated managed object. Depending on your assigned role and locale, you may have access to only a subset of the commands available in a mode; commands to which you do not have access are hidden.

The CLI prompt for each mode shows the full path down the mode hierarchy to the current mode. This helps you to determine where you are in the command mode hierarchy, and it can be an invaluable tool when you need to navigate through the hierarchy.

The following table lists the main command modes, the commands used to access each mode, and the CLI prompt associated with each mode.

Table 2: Main Command Modes and Prompts

Mode Name	Commands Used to Access	Mode Prompt
EXEC	**top** command from any mode	#
adapter	**scope adapter** command from EXEC mode	/adapter #
chassis	**scope chassis** command from EXEC mode	/chassis #
Ethernet server	**scope eth-server** command from EXEC mode	/eth-server #
Ethernet uplink	**scope eth-uplink** command from EXEC mode	/eth-uplink #
fabric-interconnect	**scope fabric-interconnect** command from EXEC mode	/fabric-interconnect #
Fibre Channel uplink	**scope fc-uplink** command from EXEC mode	/fc-uplink #
firmware	**scope firmware** command from EXEC mode	/firmware #
Host Ethernet interface	**scope host-eth-if** command from EXEC mode	/host-eth-if #
Host Fibre Channel interface	**scope host-fc-if** command from EXEC mode	/host-fc-if #

Mode Name	Commands Used to Access	Mode Prompt
monitoring	**scope monitoring** command from EXEC mode	/monitoring #
organization	**scope org** command from EXEC mode	/org #
security	**scope security** command from EXEC mode	/security #
server	**scope server** command from EXEC mode	/server #
service-profile	**scope service-profile** command from EXEC mode	/service-profile #
system	**scope system** command from EXEC mode	/system #
virtual HBA	**scope vhba** command from EXEC mode	/vhba #
virtual NIC	**scope vnic** command from EXEC mode	/vnic #

Object Commands

Four general commands are available for object management:

- **create** *object*
- **delete** *object*
- **enter** *object*
- **scope** *object*

You can use the **scope** command with any managed object, whether a permanent object or a user-instantiated object. The other commands allow you to create and manage user-instantiated objects. For every **create** *object* command, a corresponding **delete** *object* and **enter** *object* command exists.

In the management of user-instantiated objects, the behavior of these commands depends on whether the object exists, as described in the following tables:

Table 3: Command behavior if the object does not exist

Command	Behavior
create *object*	The object is created and its configuration mode, if applicable, is entered.
delete *object*	An error message is generated.
enter *object*	The object is created and its configuration mode, if applicable, is entered.
scope *object*	An error message is generated.

Table 4: Command behavior if the object exists

Command	Behavior
create *object*	An error message is generated.
delete *object*	The object is deleted.
enter *object*	The configuration mode, if applicable, of the object is entered.
scope *object*	The configuration mode of the object is entered.

Complete a Command

You can use the Tab key in any mode to complete a command. Partially typing a command name and pressing Tab causes the command to be displayed in full or to the point where another keyword must be chosen or an argument value must be entered.

Command History

The CLI stores all commands used in the current session. You can step through the previously used commands by using the Up Arrow or Down Arrow keys. The Up Arrow key steps to the previous command in the history, and the Down Arrow key steps to the next command in the history. If you get to the end of the history, pressing the Down Arrow key does nothing.

All commands in the history can be entered again by simply stepping through the history to recall the desired command and pressing Enter. The command is entered as if you had manually typed it. You can also recall a command and change it before you press Enter.

Committing, Discarding, and Viewing Pending Commands

When you enter a configuration command in the CLI, the command is not applied until you enter the **commit-buffer** command. Until committed, a configuration command is pending and can be discarded by entering a **discard-buffer** command.

You can accumulate pending changes in multiple command modes and apply them together with a single **commit-buffer** command. You can view the pending commands by entering the **show configuration pending** command in any command mode.

Note Committing multiple commands together is not an atomic operation. If any command fails, the successful commands are applied despite the failure. Failed commands are reported in an error message.

While any commands are pending, an asterisk (*) appears before the command prompt. The asterisk disappears when you enter the **commit-buffer** command.

The following example shows how the prompts change during the command entry process:

```
switch-1# scope chassis 1
switch-1 /chassis # enable locator-led
switch-1 /chassis* # show configuration pending
 scope chassis 1
+    enable locator-led
 exit
switch-1 /chassis* # commit-buffer
switch-1 /chassis #
```

Online Help for the CLI

At any time, you can type the ? character to display the options available at the current state of the command syntax.

If you have not typed anything at the prompt, typing ? lists all available commands for the mode you are in. If you have partially typed a command, typing ? lists all available keywords and arguments available at your current position in the command syntax.

CLI Session Limits

Cisco UCS Manager limits the number of CLI sessions that can be active at one time to 32 total sessions. This value is not configurable.

Web Session Limits

Web session limits are used by Cisco UCS Manager to restrict the number of web sessions (both GUI and XML) permitted access to the system at any one time.

By default, the number of concurrent web sessions allowed by Cisco UCS Manager is set to the maximum value: 256.

Setting the Web Session Limit for Cisco UCS Manager from the CLI

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **scope web-session-limits**	Enters system services web session limits mode.
Step 4	UCS-A /system/services/web-session-limits # **set total** *num-of-logins-total*	The maximum number of concurrent HTTP and HTTPS sessions allowed for all users within the system. Enter an integer between 1 and 256.
Step 5	UCS-A /system/services/web-session-limits # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the maximum number of HTTP and HTTPS sessions allowed by the system to 200 and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # scope web-session-limits
UCS-A /system/services/web-session-limits* # set total 200
UCS-A /system/services/web-session-limits* # commit-buffer
UCS-A /system/services/web-session-limits #
```

Pre-Login Banner

With a pre-login banner, when a user logs into Cisco UCS Manager GUI, Cisco UCS Manager displays the banner text in the **Create Pre-Login Banner** dialog box and waits until the user dismisses that dialog box before it prompts for the username and password. When a user logs into Cisco UCS Manager CLI, Cisco UCS Manager displays the banner text in a dialog box and waits for the user to dismiss that dialog box before it prompts for the password. It then repeats the banner text above the copyright block that it displays to the user.

Creating the Pre-Login Banner

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.

	Command or Action	Purpose
Step 2	UCS-A /security # **scope banner**	Enters banner security mode.
Step 3	UCS-A /security/banner # **create pre-login-banner**	Creates a pre login banner.
Step 4	UCS-A /security/banner/pre-login-banner # **set message**	Specifies the message that Cisco UCS Manager displays to the user before it displays the login prompt for the Cisco UCS Manager GUI or CLI. You can enter any standard ASCII character in this field. Launches a dialog for entering the pre-login banner message text.
Step 5	At the prompt, type a pre-login banner message and press **Enter**.	On the line following your input, type ENDOFBUF to finish. Press Ctrl and C to cancel out of the set message dialog.
Step 6	UCS-A /security/banner/pre-login-banner # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates the pre-login banner:

```
UCS-A# scope security
UCS-A /security # scope banner
UCS-A /security/banner # create pre-login-banner
UCS-A /security/banner/pre-login-banner* # set message
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Enter prelogin banner:
>Welcome to UCS System 1
>ENDOFBUF
UCS-A /security/banner/pre-login-banner* # commit-buffer
UCS-A /security/banner/pre-login-banner #
```

Modifying the Pre-Login Banner

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope banner**	Enters banner security mode.
Step 3	UCS-A /security/banner # **scope pre-login-banner**	Enters pre-login-banner banner security mode.

	Command or Action	Purpose
Step 4	UCS-A /security/banner/pre-login-banner # **set message**	Specifies the message that Cisco UCS Manager displays to the user before it displays the login prompt for the Cisco UCS Manager GUI or CLI. You can enter any standard ASCII character in this field. Launches a dialog for entering the pre-login banner message text.
Step 5	At the prompt, modify the pre-login banner message and press **Enter**.	On the line following your input, type ENDOFBUF to finish. Press Ctrl and C to cancel out of the set message dialog.
Step 6	UCS-A /security/banner/pre-login-banner # **commit-buffer**	Commits the transaction to the system configuration.

The following example modifies the pre-login banner:

```
UCS-A# scope security
UCS-A /security # scope banner
UCS-A /security/banner # create pre-login-banner
UCS-A /security/banner/pre-login-banner* # set message
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Enter prelogin banner:
Welcome to UCS System 1
ENDOFBUF
UCS-A /security/banner/pre-login-banner* # commit-buffer
UCS-A /security/banner/pre-login-banner #
```

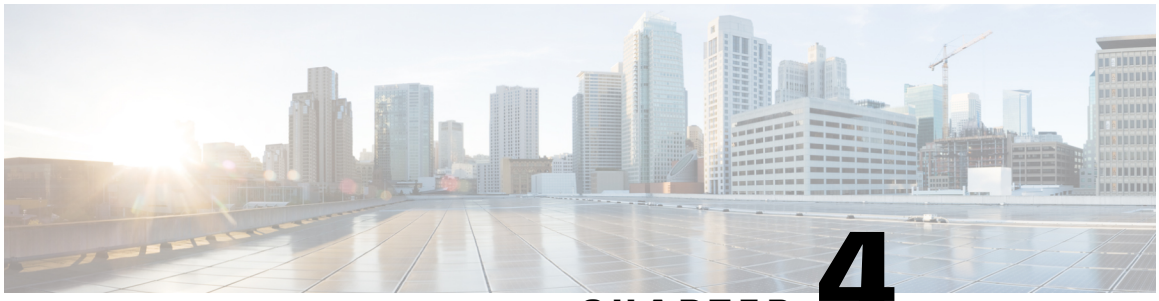
Deleting the Pre-Login Banner

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope banner**	Enters banner security mode.
Step 3	UCS-A /security/banner # **delete pre-login-banner**	Deletes the pre-login banner from the system.
Step 4	UCS-A /security/banner # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the pre-login banner:

```
UCS-A# scope security
UCS-A /security # scope banner
UCS-A /security/banner # delete pre-login-banner
UCS-A /security/banner* # commit-buffer
UCS-A /security/banner #
```



CHAPTER 4

Configuring the Fabric Interconnects

This chapter includes the following sections:

Initial System Setup

The first time that you access a fabric interconnect in a Cisco UCS domain, a setup wizard prompts you for the following information required to configure the system:

- Installation method (GUI or CLI)
- Setup mode (restore from full system backup or initial setup)
- System configuration type (standalone or cluster configuration)
- System name
- Admin password

- Management port IPv4 address and subnet mask, or IPv6 address and prefix
- Default gateway IPv4 or IPv6 address
- DNS Server IPv4 or IPv6 address
- Default domain name

Setup Mode

You can choose to either restore the system configuration from an existing backup file, or manually set up the system by going through the Setup wizard. If you choose to restore the system, the backup file must be reachable from the management network.

System Configuration Type

You can configure a Cisco UCS domain to use a single fabric interconnect in a standalone configuration or to use a redundant pair of fabric interconnects in a cluster configuration.

A cluster configuration provides high availability. If one fabric interconnect becomes unavailable, the other takes over. Only one management port (Mgmt0) connection is required to support a cluster configuration; however, both Mgmt0 ports should be connected to provide link-level redundancy.

In addition, a cluster configuration actively enhances failover recovery time for redundant virtual interface (VIF) connections. When an adapter has an active VIF connection to one fabric interconnect and a standby VIF connection to the second, the learned MAC addresses of the active VIF are replicated but not installed on the second fabric interconnect. If the active VIF fails, the second fabric interconnect installs the replicated MAC addresses and broadcasts them to the network through gratuitous ARP messages, shortening the switchover time.

Note The cluster configuration provides redundancy only for the management plane. Data redundancy is dependent on the user configuration and might require a third-party tool to support data redundancy.

To use the cluster configuration, you must directly connect the two fabric interconnects together using Ethernet cables between the L1 (L1-to-L1) and L2 (L2-to-L2) high-availability ports, with no other fabric interconnects in between. Also you can connect the fabric interconnects directly through a patch panel to allow the two fabric interconnects to continuously monitor the status of each other and quickly know when one has failed.

Both fabric interconnects in a cluster configuration must go through the initial setup process. You must enable the first fabric interconnect that you set up for a cluster configuration. When you set up the second fabric interconnect, it detects the first fabric interconnect as a peer fabric interconnect in the cluster.

For more information, see to the *Cisco UCS 6100 Series Fabric Interconnect Hardware Installation Guide*.

Management Port IP Address

In a standalone configuration, you must specify only one IPv4 address, gateway, and subnet mask, or only one IPv6 address, gateway, and network prefix for the single management port on the fabric interconnect. You can configure either an IPv4 or an IPv6 address for the management port IP address.

In a cluster configuration, you must specify the following three IPv4 addresses in the same subnet, or three IPv6 addresses with the same prefix:

- Management port IP address for fabric interconnect A
- Management port IP address for fabric interconnect B
- Cluster IP address

Note In a cluster configuration, the management port for both fabric interconnects must be configured with the same address type, either IPv4 or IPv6. If you configure the first FI with an IPv4 address then attempt to configure the second FI with an IPv6 address, the configuration will fail.

Performing an Initial System Setup for a Standalone Configuration

Before You Begin

1. Verify the following physical connections on the fabric interconnect:
   - The console port is physically connected to a computer terminal or console server
   - The management Ethernet port (mgmt0) is connected to an external hub, switch, or router

   For more information, refer to the *Cisco UCS Hardware Installation Guide* for your fabric interconnect.
2. Verify that the console port parameters on the computer terminal (or console server) attached to the console port are as follows:
   - 9600 baud
   - 8 data bits
   - No parity
   - 1 stop bit
3. Collect the following information that you will need to supply during the initial setup:
   - System name
   - Password for the admin account. Choose a strong password that meets the guidelines for Cisco UCS Manager passwords. This password cannot be blank.
   - Management port IPv4 and subnet mask, or IPv6 address and prefix.
   - Default gateway IPv4 or IPv6 address.
   - DNS server IPv4 or IPv6 address (optional).
   - Domain name for the system (optional).

Procedure

Step 1 Connect to the console port.

Step 2 Power on the fabric interconnect.
You will see the power on self-test messages as the fabric interconnect boots. The system will run a DHCP client to check for a lease.

Step 3 When the unconfigured system boots, it prompts you for the setup method to be used. Enter **console** to continue the initial setup using the console CLI.

Step 4 Enter **setup** to continue as an initial system setup.

Step 5 Enter **y** to confirm that you want to continue the initial setup.

Step 6 Enter the password for the admin account.

Step 7 To confirm, re-enter the password for the admin account.

Step 8 Enter **no** to continue the initial setup for a standalone configuration.

Step 9 Enter the system name.

Step 10 Enter the IPv4 or IPv6 address for the management port of the fabric interconnect.
If you enter an IPv4 address, you will be prompted to enter an IPv4 subnet mask. If you enter an IPv6 address, you will be prompted to enter an IPv6 network prefix.

Step 11 Enter the respective IPv4 subnet mask or IPv6 network prefix, then press Enter.
You are prompted for an IPv4 or IPv6 address for the default gateway, depending on the address type you entered for the management port of the fabric interconnect.

Step 12 Enter either the:

- IPv4 address of the default gateway
- IPv6 address of the default gateway

Step 13 Enter **yes** if you want to specify the IP address for the DNS server, or **no** if you do not.

Step 14 (Optional) Enter the IPv4 or IPv6 address for the DNS server.
The address type must be the same as the address type of the management port of the fabric interconnect.

Step 15 Enter **yes** if you want to specify the default domain name, or **no** if you do not.

Step 16 (Optional) Enter the default domain name.

Step 17 Enter **yes** if you want to join the centralized management environment (Cisco UCS Central), or **no** if you do not.

Step 18 Review the setup summary and enter **yes** to save and apply the settings, or enter **no** to go through the Setup wizard again to change some of the settings.
If you choose to go through the Setup wizard again, it provides the values you previously entered, and the values appear in brackets. To accept previously entered values, press Enter.

The following example sets up a standalone configuration using the console setup method and IPv4 management addresses:

```
Enter the installation method (console/gui)?  console
Enter the setup mode (restore from backup or initial setup) [restore/setup]? setup
You have chosen to setup a new switch.  Continue? (y/n): y
Enter the password for "admin": adminpassword%958
Confirm the password for "admin": adminpassword%958
```

```
Do you want to create a new cluster on this switch (select 'no' for standalone setup or if
 you want this switch to be added to an existing cluster)? (yes/no) [n]: no
Enter the system name: foo
Mgmt0 address: 192.168.10.10
Mgmt0 IPv4 netmask: 255.255.255.0
IPv4 address of the default gateway: 192.168.10.1
Configure the DNS Server IPv4 address? (yes/no) [n]: yes
  DNS IP address: 20.10.20.10
Configure the default domain name? (yes/no) [n]: yes
  Default domain name: domainname.com
Join centralized management environment (UCS Central)? (yes/no) [n]: no
Following configurations will be applied:
  Switch Fabric=A
  System Name=foo
  Physical Switch Mgmt0 IP Address=192.168.10.10
  Physical Switch Mgmt0 IP Netmask=255.255.255.0
  Default Gateway=192.168.10.1
  DNS Server=20.10.20.10
  Domain Name=domainname.com
Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes
```

The following example sets up a standalone configuration using the console setup method and IPv6 management addresses:

```
Enter the installation method (console/gui)?  console
Enter the setup mode (restore from backup or initial setup) [restore/setup]? setup
You have chosen to setup a new switch.  Continue? (y/n): y
Enter the password for "admin": adminpassword%652
Confirm the password for "admin": adminpassword%652
Do you want to create a new cluster on this switch (select 'no' for standalone setup or if
 you want this switch to be added to an existing cluster)? (yes/no) [n]: no
Enter the system name: foo
Mgmt0 address:  2001::107
Mgmt0 IPv6 prefix: 64
IPv6 address of the default gateway: 2001::1
Configure the DNS Server IPv6 address? (yes/no) [n]: yes
  DNS IP address: 2001::101
Configure the default domain name? (yes/no) [n]: yes
  Default domain name: domainname.com
Join centralized management environment (UCS Central)? (yes/no) [n]: no
Following configurations will be applied:
  Switch Fabric=A
  System Name=foo
  Enforced Strong Password=no
  Physical Switch Mgmt0 IPv6 Address=2001::107
  Physical Switch Mgmt0 IPv6 Prefix=64
  Default Gateway=2001::1
  Ipv6 value=1
  DNS Server=2001::101
  Domain Name=domainname.com
Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes
```

Initial System Setup for a Cluster Configuration

Performing an Initial System Setup for the First Fabric Interconnect

This procedure describes setting up the first fabric interconnect using IPv4 or IPv6 addresses for the management port, the default gateway, and the DNS server.

Before You Begin

1 Verify the following physical connections on the fabric interconnect:

- A console port on the first fabric interconnect is physically connected to a computer terminal or console server
- The management Ethernet port (mgmt0) is connected to an external hub, switch, or router
- The L1 ports on both fabric interconnects are directly connected to each other
- The L2 ports on both fabric interconnects are directly connected to each other

For more information, refer to the *Cisco UCS Hardware Installation Guide* for your fabric interconnect.

2 Verify that the console port parameters on the computer terminal (or console server) attached to the console port are as follows:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit

3 Collect the following information that you will need to supply during the initial setup:

- System name.
- Password for the admin account. Choose a strong password that meets the guidelines for Cisco UCS Manager passwords. This password cannot be blank.
- Three static IPv4 or IPv6 addresses: two for the management port on both fabric interconnects (one per fabric interconnect) and one for the cluster IP address used by Cisco UCS Manager.
- Subnet mask for the three static IPv4 addresses, or network prefix for the three static IPv6 addresses.
- Default gateway IPv4 or IPv6 address.
- DNS server IPv4 or IPv6 address (optional).
- Domain name for the system (optional).

Procedure

Step 1 Connect to the console port.

Step 2 Power on the fabric interconnect.

You will see the power on self-test messages as the fabric interconnect boots. The system will run a DHCP client to check for a lease.

Step 3 When the unconfigured system boots, it prompts you for the setup method to be used. Enter **console** to continue the initial setup using the console CLI.

Step 4 Enter **setup** to continue as an initial system setup.

Step 5 Enter **y** to confirm that you want to continue the initial setup.

Step 6 Enter the password for the admin account.

Step 7 To confirm, re-enter the password for the admin account.

Step 8 Enter **yes** to continue the initial setup for a cluster configuration.

Step 9 Enter the fabric interconnect fabric (either **A** or **B**).

Step 10 Enter the system name.

Step 11 Enter the IPv4 or IPv6 address for the management port of the fabric interconnect.
If you enter an IPv4 address, you will be prompted to enter an IPv4 subnet mask. If you enter an IPv6 address, you will be prompted to enter an IPv6 network prefix.

Step 12 Enter the respective IPv4 subnet mask or IPv6 network prefix, then press Enter.
You are prompted for an IPv4 or IPv6 address for the default gateway, depending on the address type you entered for the management port of the fabric interconnect.

Step 13 Enter either the:

- IPv4 address of the default gateway
- IPv6 address of the default gateway

Step 14 Enter **yes** if you want to specify the IP address for the DNS server, or **no** if you do not.

Step 15 (Optional) Enter the IPv4 or IPv6 address for the DNS server.
The address type must be the same as the address type of the management port of the fabric interconnect.

Step 16 Enter **yes** if you want to specify the default domain name, or **no** if you do not.

Step 17 (Optional) Enter the default domain name.

Step 18 Review the setup summary and enter **yes** to save and apply the settings, or enter **no** to go through the Setup wizard again to change some of the settings.
If you choose to go through the Setup wizard again, it provides the values you previously entered, and the values appear in brackets. To accept previously entered values, press **Enter**.

The following example sets up the first fabric interconnect for a cluster configuration using the console and IPv4 management addresses:

```
Enter the installation method (console/gui)?  console
Enter the setup mode (restore from backup or initial setup) [restore/setup]? setup
You have chosen to setup a new switch.  Continue? (y/n): y
Enter the password for "admin": adminpassword%958
Confirm the password for "admin": adminpassword%958
Do you want to create a new cluster on this switch (select 'no' for standalone setup or if
 you want this switch to be added to an existing cluster)? (yes/no) [n]: yes
Enter the switch fabric (A/B): A
Enter the system name: foo
Mgmt0 IPv4 address: 192.168.10.10
Mgmt0 IPv4 netmask: 255.255.255.0
IPv4 address of the default gateway: 192.168.10.1
Virtual IPv4 address: 192.168.10.12
```

```
Configure the DNS Server IPv4 address? (yes/no) [n]: yes
  DNS IPv4 address: 20.10.20.10
Configure the default domain name? (yes/no) [n]: yes
  Default domain name: domainname.com
Join centralized management environment (UCS Central)? (yes/no) [n]: no
Following configurations will be applied:
  Switch Fabric=A
  System Name=foo
  Management IP Address=192.168.10.10
  Management IP Netmask=255.255.255.0
  Default Gateway=192.168.10.1
  Cluster Enabled=yes
  Virtual Ip Address=192.168.10.12
  DNS Server=20.10.20.10
  Domain Name=domainname.com
Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes
```

The following example sets up the first fabric interconnect for a cluster configuration using the console and IPv6 management addresses:

```
Enter the installation method (console/gui)?  console
Enter the setup mode (restore from backup or initial setup) [restore/setup]? setup
You have chosen to setup a new switch.  Continue? (y/n): y
Enter the password for "admin": adminpassword%652
Confirm the password for "admin": adminpassword%652
Do you want to create a new cluster on this switch (select 'no' for standalone setup or if
 you want this switch to be added to an existing cluster)? (yes/no) [n]: yes
Enter the switch fabric (A/B): A
Enter the system name: foo
Mgmt0 address:  2001::107
Mgmt0 IPv6 prefix: 64
IPv6 address of the default gateway: 2001::1
Configure the DNS Server IPv6 address? (yes/no) [n]: yes
  DNS IP address: 2001::101
Configure the default domain name? (yes/no) [n]: yes
  Default domain name: domainname.com
Join centralized management environment (UCS Central)? (yes/no) [n]: no
Following configurations will be applied:
  Switch Fabric=A
  System Name=foo
  Enforced Strong Password=no
  Physical Switch Mgmt0 IPv6 Address=2001::107
  Physical Switch Mgmt0 IPv6 Prefix=64
  Default Gateway=2001::1
  Ipv6 value=1
  DNS Server=2001::101
  Domain Name=domainname.com
Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes
```

Performing an Initial System Setup for the Second Fabric Interconnect

This procedure describes setting up the second fabric interconnect using IPv4 or IPv6 addresses for the management port.

Before You Begin

1 Verify the following physical connections on the fabric interconnect:

- A console port on the second fabric interconnect is physically connected to a computer terminal or console server
- The management Ethernet port (mgmt0) is connected to an external hub, switch, or router
- The L1 ports on both fabric interconnects are directly connected to each other
- The L2 ports on both fabric interconnects are directly connected to each other

For more information, refer to the *Cisco UCS Hardware Installation Guide* for your fabric interconnect.

2 Verify that the console port parameters on the computer terminal (or console server) attached to the console port are as follows:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit

3 Collect the following information that you will need to supply during the initial setup:

- Password for the admin account of the peer fabric interconnect. Choose a strong password that meets the guidelines for Cisco UCS Manager passwords. This password cannot be blank.
- Management port IPv4 address in the same subnet, or management port IPv6 with the same network prefix as the peer fabric interconnect.

4

Procedure

Step 1 Connect to the console port.

Step 2 Power on the fabric interconnect.
You will see the power on self-test messages as the fabric interconnect boots. The system will run a DHCP client to check for a lease.

Step 3 When the unconfigured system boots, it prompts you for the setup method to be used. Enter **console** to continue the initial setup using the console CLI.

Note The fabric interconnect should detect the peer fabric interconnect in the cluster. If it does not, check the physical connections between the L1 and L2 ports, and verify that the peer fabric interconnect has been enabled for a cluster configuration.

Step 4 Enter **y** to add the subordinate fabric interconnect to the cluster.

Step 5 Enter the admin password of the peer fabric interconnect.

Step 6 Enter the IP address for the management port on the subordinate fabric interconnect.

Step 7 Review the setup summary and enter **yes** to save and apply the settings, or enter **no** to go through the Setup wizard again to change some of the settings.
If you choose to go through the Setup wizard again, it provides the values you previously entered, and the values appear in brackets. To accept previously entered values, press **Enter**.

The following example sets up the second fabric interconnect for a cluster configuration using the console and the IPv4 address of the peer:

```
Enter the installation method (console/gui)? console
Installer has detected the presence of a peer Fabric interconnect. This Fabric interconnect
 will be added to the cluster. Continue (y/n) ? y
Enter the admin password of the peer Fabric Interconnect: adminpassword%958
Peer Fabric interconnect Mgmt0 IPv4 Address: 192.168.10.11
Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes
```

The following example sets up the second fabric interconnect for a cluster configuration using the console and the IPv6 address of the peer:

```
Enter the installation method (console/gui)? console
Installer has detected the presence of a peer Fabric interconnect. This Fabric interconnect
 will be added to the cluster. Continue (y/n) ? y
Enter the admin password of the peer Fabric Interconnect: adminpassword%958
Peer Fabric interconnect Mgmt0 IPv6 Address: 2001::107
Apply and save the configuration (select 'no' if you want to re-enter)? (yes/no): yes
```

Adding Out-of-band IPv4 Addresses to a Fabric Interconnect

All fabric interconnects require an OOB IPv4 address, network mask and gateway. This procedure describes how to configure an OOB IPv4 address for a fabric interconnect that was set up with static IPv6 addresses.

Before You Begin

Collect the out-of-band (OOB) IPv4 address you want to assign to the fabric interconnect.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope fabric interconnect a**	Enters fabric configuration mode for Fabric A.
Step 2	UCS-A/fabric-interconnect # **set out-of-band ip** *ip-addr* **netmask** *ip-addr***gw** *ip-addr*	Sets the OOB IPv4 address, network mask and gateway address. The system warns that the console session change may be disconnected when the change is committed.
Step 3	UCS-A/fabric-interconnect # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows configuring an OOB IPv4 address for fabric interconnect A:

```
UCS-A# scope fabric-interconnect a
UCS-A /fabric-interconnect # set out-of-band ip 10.105.214.107 netmask 255.255.255.0 gw
10.105.214.1
Warning: When committed, this change may disconnect the current CLI session
UCS-A /fabric-interconnect* # commit-buffer
```

Enabling a Standalone Fabric Interconnect for Cluster Configuration

You can add a second fabric interconnect to an existing Cisco UCS domain that uses a single standalone fabric interconnect. To do this, you must enable the standalone fabric interconnect for cluster operation by configuring it with the virtual IP or IPv6 address of the cluster, and then add the second fabric interconnect to the cluster.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **connect local-mgmt**	Enters local management mode.
Step 2	UCS-A(local-mgmt) # **enable cluster** {*virtual-ip-addr*|*virtual-ip6-addr*}	Enables cluster operation on the standalone fabric interconnect with the specified IPv4 or IPv6 address. When you enter this command, you are prompted to confirm that you want to enable cluster operation. Type **yes** to confirm. The IP address must be the virtual IPv4 or IPv6 address for the cluster configuration, not the IP address assigned to the fabric interconnect that you are adding to the cluster.

The following example enables a standalone fabric interconnect with a virtual IPv4 address of 192.168.1.101 for cluster operation:

```
UCS-A# connect local-mgmt
UCS-A(local-mgmt)# enable cluster 192.168.1.101
This command will enable cluster mode on this setup. You cannot change it
back to stand-alone. Also, any GUI or KVM sessions may be terminated. Are you sure you want
 to continue? (yes/no): yes
UCS-A(local-mgmt)#
```

The following example enables a standalone fabric interconnect with a virtual IPv6 address of 192.168.1.101 for cluster operation:

```
UCS-A# connect local-mgmt
UCS-A(local-mgmt)# enable cluster ipv6 2001::109
This command will enable IPv6 cluster mode on this setup. You cannot change it
back to stand-alone. Also, any GUI or KVM sessions may be terminated. Are you sure you want
 to continue? (yes/no): yes
UCS-A(local-mgmt)#
```

What to Do Next

Add the second fabric interconnect to the cluster.

Changing the System Name

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope system**	Enters system mode.
Step 2	UCS-A /system # **set name** *name*	Sets the system name.
Step 3	UCS-A /system # **commit-buffer**	Commits the transaction to the system configuration.

The name is updated on both fabric interconnects within about 30 seconds after the transaction is committed.

The following example changes the system name and commits the transaction:

```
UCS-A# scope system
UCS-A /system* # set name SanJose5
UCS-A /system* # commit-buffer
UCS-A /system #
```

Changing the Management Subnet of a Cluster

When changing the IPv4 management subnet in a cluster configuration, you must change the following three IPv4 addresses simultaneously and you must configure all three in the same subnet:

- Management port IP address for fabric interconnect A
- Management port IP address for fabric interconnect B
- Cluster IP (virtual IP) address

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect a**	Enters fabric interconnect mode for fabric A.
Step 2	UCS-A /fabric-interconnect # **set out-of-band ip** *ip-address* **netmask** *netmask* **gw** *gateway-ip-address*	Sets the IP address, network mask, and gateway IP address of the fabric interconnect.
Step 3	UCS-A /fabric-interconnect # **scope fabric-interconnect b**	Enters fabric interconnect mode for fabric B.
Step 4	UCS-A /fabric-interconnect # **set out-of-band ip** *ip-address* **netmask** *netmask* **gw** *gateway-ip-address*	Sets the IP address, netmask, and gateway IP address of the fabric interconnect.
Step 5	UCS-A /fabric-interconnect # **scope system**	Enters system mode.
Step 6	UCS-A /system # **set virtual-ip** *vip-address*	Sets the virtual IP address for the cluster.
Step 7	UCS-A /system # **commit-buffer**	Commits the transaction to the system configuration.

When you commit the transaction, you are disconnected from the management session. Reconnect at the new management IP address.

This example changes both fabric-interconnect IP addresses, changes the virtual IP address, and commits the transaction, disconnecting the session:

```
UCS-A# scope fabric-interconnect a
UCS-A /fabric-interconnect # set out-of-band ip 192.0.2.111 netmask 255.255.255.0 gw 192.0.2.1
UCS-A /fabric-interconnect* # scope fabric-interconnect b
UCS-A /fabric-interconnect* # set out-of-band ip 192.0.2.112 netmask 255.255.255.0 gw
192.0.2.1
UCS-A /fabric-interconnect* # scope system
```

```
UCS-A /system* # set virtual-ip 192.0.2.113
UCS-A /system* # commit-buffer
```

Changing the Management Prefix of a Cluster

When changing the IPv6 management prefix in a cluster configuration, you must change the following three IPv6 addresses simultaneously and you must configure all three with the same network prefix:

- Management port IPv6 address for fabric interconnect A
- Management port IPv6 address for fabric interconnect B
- Cluster IPv6 (virtual IPv6) address

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect a**	Enters fabric interconnect mode for fabric A.
Step 2	UCS-A fabric-interconnect # **scope ipv6-config**	Enters IPv6 configuration mode for fabric A.
Step 3	UCS-A fabric-interconnect/ ipv6-config # **set out-of-band ipv6** *ipv6-addr* **ipv6-gw** *ipv6-gw-addr* **ipv6-prefix** *prefix*	Sets the management IPv6 address, gateway IPv6 address, and network prefix for fabric A.
Step 4	UCS-A fabric-interconnect/ipv6-config # **scope fabric-interconnect b**	Enter fabric interconnect mode for fabric B.
Step 5	UCS-A fabric-interconnect/ # **scope ipv6-config**	Enter IPv6 configuration mode for fabric B
Step 6	UCS-A/fabric-interconnect/ipv6-config # **set out-of-band ipv6** *ipv6-addr* **ipv6-gw** *ipv6-gw-addr* **ipv6-prefix** *prefix*	Sets the management IPv6 address, gateway IPv6 address, and network prefix for fabric B.
Step 7	UCS-A/fabric-interconnect/ipv6-config # **scope system**	Enters system mode.
Step 8	UCS-A/system # **set virtual-ip ipv6** *virtual-ip6-addr*	Sets the virtual IPv6 address for the cluster.
Step 9	UCS-A/system # **commit-buffer**	Commits the transaction to the system configuration.

When you commit the transaction, you are disconnected from the management session. Reconnect at the new management IPv6 address.

This example changes both management IPv6 addresses, changes the virtual IPv6 address, and commits the transaction:

```
UCS-A# scope fabric-interconnect a
UCS-A /fabric-interconnect # scope ipv6-config
UCS-A /fabric-interconnect/ipv6-config # set out-of-band ipv6 2001:10::157
```

```
UCS-A /fabric-interconnect/ipv6-config* # set out-of-band ipv6-gw 2001:10::1
UCS-A /fabric-interconnect/ipv6-config* # set out-of-band ipv6-prefix 64
UCS-A /fabric-interconnect/ipv6-config* # scope fabric-interconnect b
UCS-A /fabric-interconnect* # scope ipv6-config
UCS-A /fabric-interconnect/ipv6-config* # set out-of-band ipv6 2001:10::158
UCS-A /fabric-interconnect/ipv6-config* # set out-of-band ipv6-gw 2001:10::1
UCS-A /fabric-interconnect/ipv6-config* # set out-of-band ipv6-prefix 64
UCS-A /fabric-interconnect/ipv6-config* # scope system
UCS-A /system* # set virtual-ip ipv6 2001:10::156
UCS-A /system* # commit-buffer
UCS-A /system #
```

Configuring the Information Policy on the Fabric Interconnect

You must configure the information policy to display the uplink switches that are connected to Cisco UCS.

Important You must enable the information policy on the fabric interconnect to view the SAN, LAN, and LLDP neighbors of the fabric interconnect.

Enabling the Information Policy on the Fabric Interconnect

Note By default, the information policy is disabled on the fabric interconnect.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope system**	Enters system mode.
Step 2	UCS-A/system # **scope info-policy**	Enters the information policy state.
Step 3	UCS-A/system/info-policy # **show**	(Optional) Displays if the information policy is enabled or disabled.
Step 4	UCS-A/system/info-policy # **enable**	Determines if the information policy can enabled on the fabric interconnect.
Step 5	UCS-A/system/info-policy* # **commit-buffer**	Enables the information policy on the fabric interconnect.

The following example shows how to enable the information policy on the fabric interconnect:

```
UCS-A#  scope system
UCS-A/system # scope info-policy
UCS-A/system/info-policy # show
Info Policy:
State: Disabled
UCS-A/system/info-policy # enable
```

```
UCS-A/system/info-policy* # commit-buffer
UCS-A/system/info-policy #
```

Disabling the Information Policy on the Fabric Interconnect

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope system**	Enters system mode.
Step 2	UCS-A/system # **scope info-policy**	Enters the information policy state.
Step 3	UCS-A/system/info-policy # **show**	(Optional) Displays if the information policy is enabled or disabled.
Step 4	UCS-A/system/info-policy # **disable**	Determines if the information policy can disabled on the fabric interconnect.
Step 5	UCS-A/system/info-policy* # **commit-buffer**	Disables information policy on the fabric interconnect.

The following example shows how to disable the information policy on the fabric interconnect:

```
UCS-A#  scope system
UCS-A/system # scope info-policy
UCS-A/system/info-policy # show
Info Policy:
State: Enabled
UCS-A/system/info-policy # disable
UCS-A/system/info-policy* # commit-buffer
UCS-A/system/info-policy #
```

Viewing the Information Policy on the Fabric Interconnect

You can view the information policy state of the fabric interconnect.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope system**	Enters system mode.
Step 2	UCS-A/system # **scope info-policy**	Enters the information policy state.
Step 3	UCS-A/system/info-policy # **show**	Displays if the information policy is enabled or disabled.

The following example shows how to view the information policy state on the fabric interconnect:

```
UCS-A#  scope system
UCS-A/system # scope info-policy
UCS-A/system/info-policy # show
Info Policy:
State: Enabled
```

Viewing the LAN Neighbors of the Fabric Interconnect

You must enable the information policy on the fabric interconnect to view the LAN neighbors.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified fabric interconnect.
Step 2	UCS-A/fabric-interconnect # **show lan-neighbors**	Displays the fabric interconnect LAN neighbors.

The following example shows how to display the LAN neighbors of the fabric interconnect:

```
UCS-A #  scope fabric-interconnect a
UCS-Afabric-interconnect # show lan-neighbors
Info Policy:Enabled
Lan Neighbors:
Local Interface: Ethernet1/2
Device Id: bgl-samc02-B(SSI140305YK)
IPv4 Address: 10.105.214.105
FI Port DN: sys/switch-A/slot-1/switch-ether/port-2
```

Viewing the SAN Neighbors of the Fabric Interconnect

You must enable the information policy on the fabric interconnect to view the SAN neighbors.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified fabric interconnect.
Step 2	UCS-A/fabric-interconnect # **show san-neighbors**	Displays the fabric interconnect SAN neighbors.

The following example shows how to display the SAN neighbors of the fabric interconnect :

```
UCS-A #  scope fabric-interconnect a
UCS-A/fabric-interconnect # show san-neighbors
Info Policy: Enabled
San neighbors:
Local Interface: fc2/1
Port VSAN: 100
```

```
Fabric Mgmt Addr: 10.65.124.252
Fabric pwwn: 20:02:00:05:9b:22:ad:C0
Fabric nwwn: 20:64:00:05:9b:22:ad:C1
My pwwn: 20:41:00:0d:ec:ee:dd:00
My nwwn: 20:64:00:0d:ec:ee:dd:01
FI Port DN: sys/switch-A/slot-2/switch-fc/port-1
```

Viewing the LLDP Neighbors of the Fabric Interconnect

You must enable the information policy on the fabric interconnect to view the LLDP neighbors.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified fabric interconnect.
Step 2	UCS-A/fabric-interconnect # **show lldp-neighbors**	Displays the fabric interconnect LLDP neighbors.

The following example shows how to display the LLDP neighbors of the fabric interconnect :

```
UCS-A #  scope fabric-interconnect a
UCS-A/fabric-interconnect # show lldp-neighbors
Info Policy: Enabled

Lldp Neighbors:

Local Interface: Eth1/5
Chassis Id: 000d.ecff.5e90
Remote Interface: Eth1/9
Remote Port Description: Ethernet1/9
System Name: bgl-samc02-B
System Description: Cisco Nexus Operating System (NX-OS) Software TAC support:
http://www.cisco.com/tac Copyright (c) 2002-2011, Cisco Systems, Inc
System Capabilities: B
Enabled Capabilities: B
Native VLAN: 1
IPv4 Mgmt Address: 10.105.214.105
FI Port DN: sys/switch-A/slot-1/switch-ether/port-5
```

Fabric Evacuation

Cisco UCS Manager 2.2(4) introduces fabric evacuation, which is the ability to evacuate all traffic that flows through a Fabric Interconnect from all servers attached to it through an IOM or FEX while upgrading a system.

Upgrading the secondary Fabric Interconnect in a system disrupts the traffic that is active on the Fabric Interconnect. This traffic fails over to the primary Fabric Interconnect. You can use fabric evacuation as follows during the upgrade process:

1. Stop all the traffic that is active through a Fabric Interconnect.
2. For vNICs configured with failover, verify that the traffic has failed over by using Cisco UCS Manager or tools such as vCenter.

3 Upgrade the secondary Fabric Interconnect.

4 Restart all the stopped traffic flows.

5 Change the cluster lead to the secondary Fabric Interconnect.

6 Repeat steps 1 to 4 and upgrade the other Fabric Interconnect.

Note Fabric evacuation is supported only with the following:

- Manual install
- Cluster configuration

Stopping Traffic on a Fabric Interconnect

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified Fabric Interconnect.
Step 2	UCS-A /fabric-interconnect # **stop server traffic** [**force**]	Stops all the traffic that is active through the specified Fabric Interconnect. Use the **force** option to evacuate a Fabric Interconnect irrespective of its current evacuation state.
Step 3	UCS-A /fabric-interconnect # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to stop all traffic that is active through Fabric Interconnect B:

```
UCS-A# scope fabric-interconnect b
UCS-A /fabric-interconnect # stop server traffic
Warning: Enabling fabric evacuation will stop all traffic through this Fabric Interconnect
 from servers attached through IOM/FEX. The traffic will fail over to the Primary Fabric
Interconnect for fail over vnics.
UCS-A /fabric-interconnect # commit-buffer
```

Displaying the Status of Evacuation at a Fabric Interconnect

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified Fabric Interconnect.
Step 2	UCS-A /fabric-interconnect # **show detail**	Displays details about the specified Fabric Interconnect.

This example shows how to display the detailed status of a Fabric Interconnect.

Note Admin Evacuation and Oper Evacuation show the status of evacuation at the Fabric Interconnect.

```
 UCS-A /fabric-interconnect # show detail

Fabric Interconnect:
    ID: B
    Product Name: Cisco UCS 6248UP
    PID: UCS-FI-6248UP
    VID: V01
    Vendor: Cisco Systems, Inc.
    Serial (SN): SSI171400HG
    HW Revision: 0
    Total Memory (MB): 16165
    OOB IP Addr: 10.193.32.172
    OOB Gateway: 10.193.32.1
    OOB Netmask: 255.255.255.0
    OOB IPv6 Address: ::
    OOB IPv6 Gateway: ::
    Prefix: 64
    Operability: Operable
    Thermal Status: Ok
    Admin Evacuation: On
    Oper Evacuation: On
    Current Task 1:
    Current Task 2:
    Current Task 3:
```

Displaying the Status of Evacuation at an IOM

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.

	Command or Action	Purpose
Step 2	UCS-A /chassis # **scope iom** *iom-id*	Enters chassis IOM mode for the specified IOM.
Step 3	UCS-A /chassis/iom # **show detail**	Displays details about the specified IOM.

This example shows how to display the detailed status of an IOM.

Note **Oper Evacuation** shows the operational status of evacuation at the IOM.

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

IOM:
    ID: 1
    Side: Left
    Fabric ID: A
    User Label:
    Overall Status: Fabric Conn Problem
    Oper qualifier: Server Port Problem
    Operability: Operable
    Presence: Equipped
    Thermal Status: OK
    Discovery: Online
    Config State: Ok
    Peer Comm Status: Connected
    Product Name: Cisco UCS 2204XP
    PID: UCS-IOM-2204XP
    VID: V02
    Part Number: 73-14488-02
    Vendor: Cisco Systems Inc
    Serial (SN): FCH1718J9FT
    HW Revision: 0
    Mfg Date: 2013-05-12T00:00:00.000
    Controller Subject: Iocard
    Fabric Port Aggregation Capability: Port Channel
    Oper Evacuation: On
    Current Task 1:
    Current Task 2:
```

Verifying Fabric Evacuation

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **show service-profile circuit server** *server-id*	Shows the network circuit information for the service profile associated with the specified server.

This example shows the VIF paths before fabric evacuation.

Note

- VIF at Fabric Interconnect A shows that traffic is initially active through the Fabric interconnect.
- VIF at Fabric Interconnect B is passive before evacuation.

```
UCS-A# show service-profile circuit server 1/6
Service Profile: test1
Server: 1/6
    Fabric ID: A
        Path ID: 1
        VIF        vNIC            Link State  Oper State Prot State    Prot Role   Admin
Pin  Oper Pin   Transport
       ---------- --------------- ----------- ---------- ------------- ----------- ----------
 ---------- ---------
               692 eth0            Up          Active     Active        Primary     0/0
     1/15       Ether
    Fabric ID: B
        Path ID: 1
        VIF        vNIC            Link State  Oper State Prot State    Prot Role   Admin
Pin  Oper Pin   Transport
       ---------- --------------- ----------- ---------- ------------- ----------- ----------
 ---------- ---------
               693 eth0            Up          Active     Passive       Backup      0/0
     1/15       Ether
UCS-A#
```

This example shows the VIF paths after Fabric Interconnect A is evacuated.

Note

- After fail over, the VIF state at Fabric Interconnect A goes into error.
- VIF at Fabric Interconnect B takes over as active.

```
UCS-A# show service-profile circuit server 1/6
Service Profile: test1
Server: 1/6
    Fabric ID: A
        Path ID: 1
        VIF        vNIC            Link State  Oper State Prot State    Prot Role   Admin
Pin  Oper Pin   Transport
       ---------- --------------- ----------- ---------- ------------- ----------- ----------
 ---------- ---------
               692 eth0            Error       Error      Active        Primary     0/0
     0/0        Ether
    Fabric ID: B
        Path ID: 1
        VIF        vNIC            Link State  Oper State Prot State    Prot Role   Admin
Pin  Oper Pin   Transport
       ---------- --------------- ----------- ---------- ------------- ----------- ----------
 ---------- ---------
               693 eth0            Up          Active     Passive       Backup      0/0
     1/15       Ether
UCS-A#
```

Restarting Traffic on a Fabric Interconnect

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified Fabric Interconnect.
Step 2	UCS-A /fabric-interconnect # **start server traffic**	Restarts traffic through the specified Fabric Interconnect.
Step 3	UCS-A /fabric-interconnect # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to restart traffic through Fabric Interconnect B:

```
UCS-A# scope fabric-interconnect b
UCS-A /fabric-interconnect # start server traffic
Warning: Resetting fabric evacuation will cause server traffic that failed over to the
Primary Fabric Interconnect to fail back to this Fabric Interconnect.
UCS-A /fabric-interconnect # commit-buffer
```

Ethernet Switching Mode

The Ethernet switching mode determines how the fabric interconnect behaves as a switching device between the servers and the network. The fabric interconnect operates in either of the following Ethernet switching modes:

End-Host Mode

End-host mode allows the fabric interconnect to act as an end host to the network, representing all servers (hosts) connected to it through vNICs. This behavior is achieved by pinning (either dynamically pinned or hard pinned) vNICs to uplink ports, which provides redundancy to the network, and makes the uplink ports appear as server ports to the rest of the fabric. In end-host mode, the fabric interconnect does not run the Spanning Tree Protocol (STP) but it avoids loops by denying uplink ports from forwarding traffic to each other and by denying egress server traffic on more than one uplink port at a time. End-host mode is the default Ethernet switching mode and should be used if either of the following are used upstream:

- Layer 2 switching for Layer 2 aggregation
- Virtual Switching System (VSS) aggregation layer

Note When you enable end-host mode, if a vNIC is hard pinned to an uplink port and this uplink port goes down, the system cannot repin the vNIC, and the vNIC remains down.

Switch Mode

Switch mode is the traditional Ethernet switching mode. The fabric interconnect runs STP to avoid loops, and broadcast and multicast packets are handled in the traditional way. Switch mode is not the default Ethernet switching mode, and should be used only if the fabric interconnect is directly connected to a router, or if either of the following are used upstream:

- Layer 3 aggregation
- VLAN in a box

Note For both Ethernet switching modes, even when vNICs are hard pinned to uplink ports, all server-to-server unicast traffic in the server array is sent only through the fabric interconnect and is never sent through uplink ports. Server-to-server multicast and broadcast traffic is sent through all uplink ports in the same VLAN.

Configuring Ethernet Switching Mode

Important When you change the Ethernet switching mode, Cisco UCS Manager logs you out and restarts the fabric interconnect. For a cluster configuration, Cisco UCS Manager restarts both fabric interconnects. The subordinate fabric interconnect reboots first as a result of the change in switching mode. The primary fabric interconnect reboots only after you acknowledge it in **Pending Activities**. The primary fabric interconnect can take several minutes to complete the change in Ethernet switching mode and become system ready. The existing configuration is retained.

While the fabric interconnects are rebooting, all blade servers lose LAN and SAN connectivity, causing a complete outage of all services on the blades. This might cause the operating system to fail.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **set mode** {**end-host** | **switch**}	Sets the fabric interconnect to the specified switching mode.
Step 3	UCS-A /eth-uplink # **commit-buffer**	Commits the transaction to the system configuration. Cisco UCS Manager restarts the fabric interconnect, logs you out, and disconnects Cisco UCS Manager CLI.

The following example sets the fabric interconnect to end-host mode and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set mode end-host
Warning: When committed, this change will cause the switch to reboot
```

```
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink #
```

Fibre Channel Switching Mode

The Fibre Channel switching mode determines how the fabric interconnect behaves as a switching device between the servers and storage devices. The fabric interconnect operates in either of the following Fibre Channel switching modes:

End-Host Mode

End-host mode allows the fabric interconnect to act as an end host to the connected fibre channel networks, representing all servers (hosts) connected to it through virtual host bus adapters (vHBAs). This behavior is achieved by pinning (either dynamically pinned or hard pinned) vHBAs to Fibre Channel uplink ports, which makes the Fibre Channel ports appear as server ports (N-ports) to the rest of the fabric. When in end-host mode, the fabric interconnect avoids loops by denying uplink ports from receiving traffic from one another.

End-host mode is synonymous with N Port Virtualization (NPV) mode. This mode is the default Fibre Channel Switching mode.

Note When you enable end-host mode, if a vHBA is hard pinned to an uplink Fibre Channel port and this uplink port goes down, the system cannot repin the vHBA, and the vHBA remains down.

Switch Mode

Switch mode is the traditional Fibre Channel switching mode. Switch mode allows the fabric interconnect to connect directly to a storage device. Enabling Fibre Channel switch mode is useful in Pod models where there is no SAN (for example, a single Cisco UCS domain that is connected directly to storage), or where a SAN exists (with an upstream MDS).

Switch mode is not the default Fibre Channel switching mode.

Note In Fibre Channel switch mode, SAN pin groups are irrelevant. Any existing SAN pin groups are ignored.

Cisco UCS Fabric Interconnect in Switch Mode with Cisco MDS 9000 Family Fibre Channel Switching Modules

While creating a port channel between a Cisco MDS 9000 family FC switching module and a Cisco UCS Fabric Interconnect in switch mode, use the following order:

1. Create the port channel on the MDS side.
2. Add the port channel member ports.
3. Create the port channel on the Fabric Interconnect side.
4. Add the port channel member ports.

If you create the port channel on the Fabric Interconnect side first, the ports will go into a suspended state.

When the Cisco UCS Fabric Interconnect is in switch mode, the port channel mode can only be in **ON** mode and not **Active**. However, to get the peer wwn information for the Fabric Interconnect, the port channel must be in **Active** mode.

Configuring Fibre Channel Switching Mode

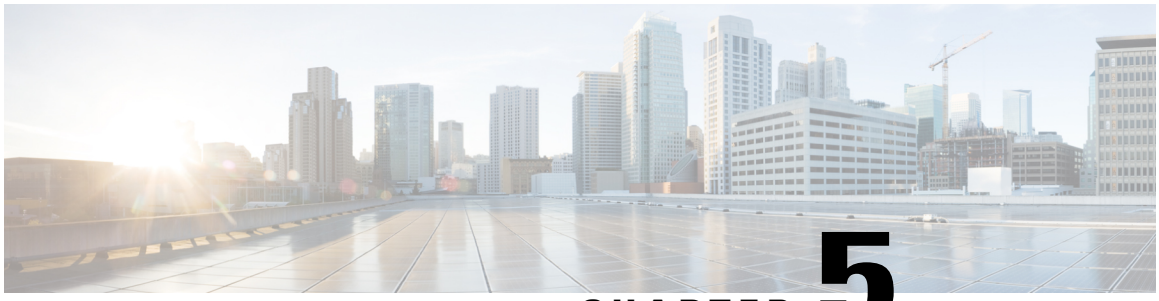
Note When the Fibre Channel switching mode is changed, both Cisco UCS fabric interconnects reload simultaneously. Reloading the fabric interconnects will cause a system-wide downtime for approximately 10 to 15 minutes.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **set mode** {**end-host** | **switch**}	Sets the fabric interconnect to the specified switching mode.
Step 3	UCS-A /fc-uplink # **commit-buffer**	Commits the transaction to the system configuration. Cisco UCS Manager restarts the fabric interconnect, logs you out, and disconnects Cisco UCS Manager CLI.

The following example shows how to set the fabric interconnect to end-host mode and commit the transaction:

```
UCS-A # scope fc-uplink
UCS-A /fc-uplink # set mode end-host
UCS-A /fc-uplink* # commit-buffer
UCS-A /fc-uplink #
```



CHAPTER 5

Configuring Ports and Port Channels

This chapter includes the following sections:

Server and Uplink Ports on the 6100 Series Fabric Interconnect

Each Cisco UCS 6100 Series Fabric Interconnect has a set of ports in a fixed port module that you can configure as either server ports or uplink Ethernet ports. These ports are not reserved. They cannot be used by a Cisco

UCS domain until you configure them. You can add expansion modules to increase the number of uplink ports on the fabric interconnect or to add uplink Fibre Channel ports to the fabric interconnect.

Note When you configure a port on a fabric interconnect, the administrative state is automatically set to enabled. If the port is connected to another device, this may cause traffic disruption. You can disable the port after configuring it.

You need to create LAN pin groups and SAN pin groups to pin traffic from servers to an uplink port.

Note Ports on the Cisco UCS 6100 Series Fabric Interconnect are not unified. For more information on Unified Ports, see Unified Ports on the Fabric Interconnect.

Each fabric interconnect can include the following port types:

Server Ports

Server ports handle data traffic between the fabric interconnect and the adapter cards on the servers.

You can only configure server ports on the fixed port module. Expansion modules do not include server ports.

Uplink Ethernet Ports

Uplink Ethernet ports handle Ethernet traffic between the fabric interconnect and the next layer of the network. All network-bound Ethernet traffic is pinned to one of these ports.

By default, Ethernet ports are unconfigured. However, you can configure them to function in the following ways:

- Uplink
- FCoE
- Appliance

You can configure uplink Ethernet ports on either the fixed module or an expansion module.

Uplink Fibre Channel Ports

Uplink Fibre Channel ports handle FCoE traffic between the fabric interconnect and the next layer of the storage area network. All network-bound FCoE traffic is pinned to one of these ports.

By default, Fibre Channel ports are uplink. However, you can configure them to function as Fibre Channel storage ports. This is useful in cases where Cisco UCS requires a connection to a Direct-Attached Storage (DAS) device.

You can only configure uplink Fibre Channel ports on an expansion module. The fixed module does not include uplink Fibre Channel ports.

Unified Ports on the Fabric Interconnect

Unified ports are ports on the fabric interconnect that can be configured to carry either Ethernet or Fibre Channel traffic. These ports are not reserved. A Cisco UCS domain cannot use these ports until you configure them.

Note When you configure a port on a fabric interconnect, the administrative state is automatically set to enabled. If the port is connected to another device, this may cause traffic disruption. You can disable the port after configuring it.

Configurable beacon LEDs indicate which unified ports are configured for the selected port mode.

Port Modes

The port mode determines whether a unified port on the fabric interconnect is configured to carry Ethernet or Fibre Channel traffic. You configure the port mode in Cisco UCS Manager. However, the fabric interconnect does not automatically discover the port mode.

Changing the port mode deletes the existing port configuration and replaces it with a new logical port. Any objects associated with that port configuration, such as VLANs and VSANS, are also removed. There is no restriction on the number of times you can change the port mode for a unified port.

Port Types

The port type defines the type of traffic carried over a unified port connection.

By default, unified ports changed to Ethernet port mode are set to the Ethernet uplink port type. Unified ports changed to Fibre Channel port mode are set to the Fibre Channel uplink port type. You cannot unconfigure Fibre Channel ports.

Changing the port type does not require a reboot.

Ethernet Port Mode

When you set the port mode to Ethernet, you can configure the following port types:

- Server ports
- Ethernet uplink ports
- Ethernet port channel members
- FCoE ports
- Appliance ports
- Appliance port channel members
- SPAN destination ports
- SPAN source ports

Note For SPAN source ports, configure one of the port types and then configure the port as SPAN source.

Fibre Channel Port Mode

When you set the port mode to Fibre Channel, you can configure the following port types:

- Fibre Channel uplink ports
- Fibre Channel port channel members
- Fibre Channel storage ports
- FCoE Uplink ports
- SPAN source ports

Note For SPAN source ports, configure one of the port types and then configure the port as SPAN source.

Beacon LEDs for Unified Ports

Each port on the 6200 series fabric interconnect has a corresponding beacon LED. When the **Beacon LED** property is configured, the beacon LEDs illuminate, showing you which ports are configured in a given port mode.

You can configure the **Beacon LED** property to show you which ports are grouped in one port mode: either Ethernet or Fibre Channel. By default, the Beacon LED property is set to Off.

Note For unified ports on the expansion module, you can reset the **Beacon LED** property to the default value of **Off** during expansion module reboot.

Guidelines for Configuring Unified Ports

Consider the following guidelines and restrictions when configuring unified ports:

Hardware and Software Requirements

Unified ports are supported on the 6200 series fabric interconnect with Cisco UCS Manager, version 2.0.

Unified ports are not supported on 6100 series fabric interconnects, even if they are running Cisco UCS Manager, version 2.0.

Port Mode Placement

Because the Cisco UCS Manager GUI interface uses a slider to configure the port mode for unified ports on a fixed or expansion module, it automatically enforces the following restrictions which limits how port modes can be assigned to unified ports. When using the Cisco UCS Manager CLI interface, these restrictions are enforced when you commit the transaction to the system configuration. If the port mode configuration violates any of the following restrictions, the Cisco UCS Manager CLI displays an error:

- Ethernet ports must be grouped together in a block. For each module (fixed or expansion), the Ethernet port block must start with the first port and end with an even numbered port.
- Fibre Channel ports must be grouped together in a block. For each module (fixed or expansion), the first port in the Fibre Channel port block must follow the last Ethernet port and extend to include the rest of the ports in the module. For configurations that include only Fibre Channel ports, the Fibre Channel block must start with the first port on the fixed or expansion module.
- Alternating Ethernet and Fibre Channel ports is not supported on a single module.

Example of a valid configuration— Might include unified ports 1–16 on the fixed module configured in Ethernet port mode and ports 17–32 in Fibre Channel port mode. On the expansion module you could configure ports 1–4 in Ethernet port mode and then configure ports 5–16 in Fibre Channel mode. The rule about alternating Ethernet and Fibre Channel port types is not violated because this port arrangement complies with the rules on each individual module.

Example of an invalid configuration— Might include a block of Fibre Channel ports starting with port 16. Because each block of ports has to start with an odd-numbered port, you would have to start the block with port 17.

Note The total number of uplink Ethernet ports and uplink Ethernet port channel members that can be configured on each fabric interconnect is limited to 31. This limitation includes uplink Ethernet ports and uplink Ethernet port channel members configured on the expansion module.

Special Considerations for UCS Manager CLI Users

Because the Cisco UCS Manager CLI does not validate port mode changes until you commit the buffer to the system configuration, it is easy to violate the grouping restrictions if you attempt to commit the buffer before creating at least two new interfaces. To prevent errors, we recommend that you wait to commit your changes to the system configuration until you have created new interfaces for all of the unified ports changing from one port mode to another.

Commiting the buffer before configuring multiple interfaces will result in an error, but you do not need to start over. You can continue to configure unified ports until the configuration satisfies the aforementioned requirements.

Cautions and Guidelines for Configuring Unified Uplink Ports and Unified Storage Ports

The following are cautions and guidelines to follow while working with unified uplink ports and unified storage ports:

- In an unified uplink port, if you enable one component as a SPAN source, the other component will automatically become a SPAN source.

Note If you create or delete a SPAN source under the Ethernet uplink port, Cisco UCS Manager automatically creates or deletes a SPAN source under the FCoE uplink port. The same happens when you create a SPAN source on the FCOE uplink port.

- You must configure a non default native VLAN on FCoE and unified uplink ports. This VLAN is not used for any traffic. Cisco UCS Manager will reuse an existing fcoe-storage-native-vlan for this purpose. This fcoe-storage-native-vlan will be used as a native VLAN on FCoE and unified uplinks.
- In an unified uplink port, if you do not specify a non default VLAN for the Ethernet uplink port the fcoe-storage-native-vlan will be assigned as the native VLAN on the unified uplink port. If the Ethernet port has a non default native VLAN specified as native VLAN, this will be assigned as the native VLAN for unified uplink port.
- When you create or delete a member port under an Ethernet port channel, Cisco UCS Manager automatically creates or deletes the member port under FCoE port channel. The same happens when you create or delete a member port in FCoE port channel.
- When you configure an Ethernet port as a standalone port, such as server port, Ethernet uplink, FCoE uplink or FCoE storage and make it as a member port for an Ethernet or FCOE port channel, Cisco UCS Manager automatically makes this port as a member of both Ethernet and FCoE port channels.
- When you remove the membership for a member port from being a member of server uplink, Ethernet uplink, FCoE uplink or FCoE storage, Cisco UCS Manager deletes the corresponding members ports from Ethernet port channel and FCoE port channel and creates a new standalone port.
- If you downgrade Cisco UCS Manager from release 2.1 to any of the prior releases, all unified uplink ports and port channels will be converted to Ethernet ports and Ethernet port channels when the downgrade is complete. Similarly, all the unified storage ports will be converted to appliance ports.
- For unified uplink ports and unified storage ports, when you create two interfaces, only one license is checked out. As long as either interface is enabled, the license remains checked out. The license will be released only if both the interfaces are disabled for a unified uplink port or a unified storage port.
- Cisco UCS 6100 series fabric interconnect switch can only support 1VF or 1VF-PO facing same downstream NPV switch.

Effect of Port Mode Changes on Data Traffic

Port mode changes can cause an interruption to the data traffic for the Cisco UCS domain. The length of the interruption and the traffic that is affected depend upon the configuration of the Cisco UCS domain and the module on which you made the port mode changes.

Tip To minimize the traffic disruption during system changes, form a Fibre Channel uplink port-channel across the fixed and expansion modules.

Impact of Port Mode Changes on an Expansion Module

After you make port mode changes on an expansion module, the module reboots. All traffic through ports on the expansion module is interrupted for approximately one minute while the module reboots.

Impact of Port Mode Changes on the Fixed Module in a Cluster Configuration

A cluster configuration has two fabric interconnects. After you make port changes to the fixed module, the fabric interconnect reboots. The impact on the data traffic depends upon whether or not you have configured the server vNICs to failover to the other fabric interconnect when one fails.

If you change the port modes on the expansion module of one fabric interconnect and then wait for that to reboot before changing the port modes on the second fabric interconnect, the following occurs:

- With server vNIC failover, traffic fails over to the other fabric interconnect and no interruption occurs.
- Without server vNIC failover, all data traffic through the fabric interconnect on which you changed the port modes is interrupted for approximately eight minutes while the fabric interconnect reboots.

If you change the port modes on the fixed modules of both fabric interconnects simultaneously, all data traffic through the fabric interconnects are interrupted for approximately eight minutes while the fabric interconnects reboot.

Impact of Port Mode Changes on the Fixed Module in a Standalone Configuration

A standalone configuration has only one fabric interconnect. After you make port changes to the fixed module, the fabric interconnect reboots. All data traffic through the fabric interconnect is interrupted for approximately eight minutes while the fabric interconnect reboots.

FC Links Rebalancing

The FC uplinks balance automatically when FC Port Channels are utilized. To create FC Port Channels, refer to Configuring a Fibre Channel Port Channel, on page 96.

For the FC uplinks that are not members of the Port Channels (Individual ISLs), load balancing is done according to the FC uplinks balancing algorithm. For a vHBA of a host or service profile to choose an available FC uplink, when FC uplink trunking is disabled, the uplink and vHBA must belong to the same VSAN

For each vHBA, the algorithm searches for an FC uplink in the following order:

1. Least used FC uplink based on the number of vHBAs currently bound to the uplink.
2. If FC uplinks are equally balanced, then round robin is used.

This process continues for all the other vHBAs. The algorithm also considers other parameters such as pre-fip/fip adapters and number of flogis. You may not see the least-used component when there are less than six flogis.

After a port configuration or any other uplink state changes, if the traffic passing through the FC uplinks is no longer balanced, you can re-balance the traffic by resetting the vHBA(s) on each adapter and allow the load balancing algorithm to evaluate for the current state of the FC uplinks.

Configuring the Port Mode

Caution Changing the port mode on either module can cause an interruption in data traffic because changes to the fixed module require a reboot of the fabric interconnect and changes on an expansion module require a reboot of that module .

If the Cisco UCS domain has a cluster configuration that is set up for high availability and servers with service profiles that are configured for failover, traffic fails over to the other fabric interconnect and data traffic is not interrupted when the port mode is changed on the fixed module.

In the Cisco UCS Manager CLI, there are no new commands to support Unified Ports. Instead, you change the port mode by scoping to the mode for the desired port type and then creating a new interface. When you create a new interface for an already configured slot ID and port ID, UCS Manager deletes the previously configured interface and creates a new one. If a port mode change is required because you configure a port that previously operated in Ethernet port mode to a port type in Fibre Channel port mode, UCS Manager notes the change.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope** *port-type-mode*	Enters the specified port type mode for one of the following port types: **eth-server** For configuring server ports. **eth-storage** For configuring Ethernet storage ports and Ethernet storage port channels. **eth-traffic-mon** For configuring Ethernet SPAN ports. **eth-uplink** For configuring Ethernet uplink ports. **fc-storage** For configuring Fibre Channel storage ports. **fc-traffic-mon** For configuring Fibre Channel SPAN ports.

	Command or Action	Purpose
		fc-uplink For configuring Fibre Channel uplink ports and Fibre Channel uplink port channels.
Step 2	UCS-A */port-type-mode* # **scope fabric** {**a** | **b**}	Enters the specified port type mode for the specified fabric.
Step 3	UCS-A */port-type-mode*/fabric # **create interface** *slot-id port-id*	Creates an interface for the specified port type. If you are changing the port type from Ethernet port mode to Fibre Channel port mode, or vice-versa, the following warning appears: `Warning: This operation will change the port mode (from Ethernet to FC or vice-versa). When committed, this change will require the module to restart.`
Step 4	Create new interfaces for other ports belonging to the Ethernet or Fibre Channel port block.	There are several restrictions that govern how Ethernet and Fibre Channel ports can be arranged on a fixed or expansion module. Among other restrictions, it is required that you change ports in groups of two. Violating any of the restrictions outlined in the Guidelines for Configuring Unified Ports section will result in an error.
Step 5	UCS-A */port-type-mode*/fabric/interface # **commit-buffer**	Commits the transaction to the system configuration.

Based on the module for which you configured the port modes, data traffic for the Cisco UCS domain is interrupted as follows:

- Fixed module—The fabric interconnect reboots. All data traffic through that fabric interconnect is interrupted. In a cluster configuration that provides high availability and includes servers with vNICs that are configured for failover, traffic fails over to the other fabric interconnect and no interruption occurs. Changing the port mode for both sides at once results in both fabric interconnects rebooting simultaneously and a complete loss of traffic until both fabric interconnects are brought back up.

  It takes about 8 minutes for the fixed module to reboot.

- Expansion module—The module reboots. All data traffic through ports in that module is interrupted.

  It takes about 1 minute for the expansion module to reboot.

The following example changes ports 3 and 4 on slot 1 from Ethernet uplink ports in Ethernet port mode to uplink Fibre Channel ports in Fibre Channel port mode:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create interface 1 3
Warning: This operation will change the port mode (from Ethernet to FC or vice-versa).
When committed, this change will require the fixed module to restart.
UCS-A /fc-uplink/fabric/interface* # up
UCS-A /fc-uplink/fabric* #create interface 1 4
Warning: This operation will change the port mode (from Ethernet to FC or vice-versa).
```

```
When committed, this change will require the fixed module to restart.
UCS-A /fc-uplink/fabric/interface* #commit-buffer
```

Configuring the Beacon LEDs for Unified Ports

Complete the following task for each module for which you want to configure beacon LEDs.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified fabric.
Step 2	UCS-A /fabric # **scope card** *slot-id*	Enters card mode for the specified fixed or expansion module.
Step 3	UCS-A /fabric/card # **scope beacon-led**	Enters beacon LED mode.
Step 4	UCS-A /fabric/card/beacon-led # **set admin-state** {**eth** | **fc** | **off**}	Specifies which port mode is represented by illuminated beacon LED lights. **eth** All of the Unified Ports configured in Ethernet mode illuminate. **fc** All of the Unified Ports configured in Fibre Channel mode illuminate. **off** Beacon LED lights for all ports on the module are turned off.
Step 5	UCS-A /fabric/card/beacon-led # **commit-buffer**	Commits the transaction to the system configuration.

The following example illuminates all of the beacon lights for Unified Ports in Ethernet port mode and commits the transaction:

```
UCS-A# scope fabric-interconnect a
UCS-A /fabric # scope card 1
UCS-A /fabric/card # scope beacon-led
UCS-A /fabric/card/beacon-led # set admin-state eth
UCS-A /fabric/card/beacon-led* # commit-buffer
UCS-A /fabric/card/beacon-led #
```

Physical and Backplane Ports

Displaying Physical Port Statistics Obtained From the ASIC

Procedure

	Command or Action	Purpose
Step 1	UCS-A /fabric-interconnect # **connect nxos** {**a** | **b**}	Enters NX-OS mode for the fabric interconnect.
Step 2	UCS-A(nxos)# **show interface ethernet** *slot/port*	Displays physical port statistics that are obtained from the ASIC.

The following example shows how to display physical port statistics that are obtained from the ASIC:

```
UCS-A /fabric-interconnect # connect nxos a
UCS-A(nxos)# show interface ethernet 1/11

Ethernet1/11 is up
 Dedicated Interface
  Hardware: 40000 Ethernet, address: a46c.2ae3.0e1a (bia a46c.2ae3.0e1a)
  Description: S: Server
  MTU 1500 bytes, BW 40000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  Port mode is fex-fabric
  full-duplex, 40 Gb/s, media type is 40G
  Beacon is turned off
  Input flow-control is off, output flow-control is off
  Rate mode is dedicated
  Switchport monitor is off
  EtherType is 0x8100
  Last link flapped 01:25:42
  Last clearing of "show interface" counters never
  2 interface resets
  30 seconds input rate 22664 bits/sec, 2833 bytes/sec, 3 packets/sec
  30 seconds output rate 9512 bits/sec, 1189 bytes/sec, 1189 bytes/sec, 4 packets/sec
  Load-Interval #2: 5 minute (300 seconds)
    input rate 33.80 Kbps, 5 pps; output rate 1.23 Mbps, 71 pps
  RX
    126057 unicast packets  1744 multicast packets  12877 broadcast packets
    140693 input packets  28702696 bytes
    3351 jumbo packets  0 storm suppression bytes
    0 runts  0 giants  0 CRC  0 no buffer
    0 input error  0 short frame  0 overrun   0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  184 input discard
    0 Rx pause
  TX
    919778 unicast packets  6991 multicast packets  29 broadcast packets
    926798 output packets  1237109219 bytes
    794275 jumbo packets
    0 output errors  0 collision  0 deferred  0 late collision
    0 lost carrier  0 no carrier  0 babble 0 output discard
    0 Tx pause

  Errors on Peer port (NIF):
  RX
    8300 toolong frames  8400 undersize frames  8500 fragment frames
```

```
  8600 crcErr_not_stomped frames  8700 crcErr_stomped frames  8800 inRangeErr frames
TX
  8200 frames_with_error
```

Displaying Physical Ports on the Fabric Interconnect That Correspond to Physical Ports on BCM

Procedure

	Command or Action	Purpose
Step 1	UCS-A /fabric-interconnect # **connect nxos** {**a** | **b**}	Enters NX-OS mode for the fabric interconnect.
Step 2	UCS-A(nxos)# **show hardware internal bcm-usd info port-info** | **grep** *interface_slot_id*	Displays physical ports on a fabric interconnect that correspond to physical ports on BCM.

The following example shows how to display physical ports on a fabric interconnect that correspond to physical ports on BCM:

```
UCS-A /fabric-interconnect # connect nxos a
UCS-A(nxos)# show hardware internal bcm-usd info port-info | grep Eth 1/11

Eth1/11      0x1a00a000  41 xe-40   57   CR4 sw 4044 0 uta  2240     0 fd dis blk dis dis
ena 40G 40G up
```

Verifying Status of Backplane Ports

Procedure

	Command or Action	Purpose
Step 1	UCS-A /fabric-interconnect # **connect nxos** {**a** | **b**}	Enters NX-OS mode for the fabric interconnect.
Step 2	UCS-A(nxos)# **show interface br**	Displays the configuration of the interface, including the speed and status of the backplane ports.

The following example shows how to verify the status of backplane ports for fabric interconnect A:

```
UCS-A /fabric-interconnect # connect nxos a
UCS-A(nxos)# show interface br
```

```
--------------------------------------------------------------------------------
Ethernet      VLAN    Type Mode   Status  Reason                   Speed     Port
Interface                                                                    Ch #
--------------------------------------------------------------------------------
Eth1/1        1       eth  access down    SFP not inserted            40G(D) --
Eth1/2        1       eth  access down    SFP not inserted            40G(D) --
Br-Eth1/3/1   1       eth  access down    Administratively down       10G(D) --
Br-Eth1/3/2   1       eth  access down    Administratively down       10G(D) --
Br-Eth1/3/3   1       eth  access down    Administratively down       10G(D) --
Br-Eth1/3/4   1       eth  access down    Administratively down       10G(D) --
Eth1/4        1       eth  access down    SFP not inserted            40G(D) --
Br-Eth1/5/1   4044    eth  trunk  down    Link not connected          10G(D) --
Br-Eth1/5/2   4044    eth  trunk  down    Link not connected          10G(D) --
Br-Eth1/5/3   4044    eth  trunk  down    Link not connected          10G(D) --
Br-Eth1/5/4   4044    eth  trunk  down    Link not connected          10G(D) --
Eth1/6        1       eth  access down    SFP not inserted            40G(D) --
Eth1/7        1       eth  access down    SFP not inserted            40G(D) --
Eth1/8        1       eth  access down    SFP not inserted            40G(D) --
Eth1/9        1       eth  access down    SFP not inserted            40G(D) --
Eth1/10       1       eth  access down    SFP not inserted            40G(D) --
Eth1/11       1       eth  fabric up      none                        40G(D) --
Eth1/12       1       eth  access down    SFP not inserted            40G(D) --
Eth1/13       1       eth  access down    SFP not inserted            40G(D) --
Eth1/14       1       eth  access down    SFP not inserted            40G(D) --
Eth1/15       1       eth  access down    SFP not inserted            40G(D) --
Eth1/16       1       eth  access down    SFP not inserted            40G(D) --
Eth1/17       1       eth  access down    SFP not inserted            40G(D) --
Eth1/18       1       eth  access down    SFP not inserted            40G(D) --
Eth1/19       1       eth  access down    SFP not inserted            40G(D) --
Eth1/20       1       eth  access down    SFP not inserted            40G(D) --
Br-Eth1/21/1  1       eth  trunk  up      none                        10G(D) --
Br-Eth1/21/2  1       eth  trunk  up      none                        10G(D) --
Br-Eth1/21/3  1       eth  trunk  down    Link not connected          10G(D) --
Br-Eth1/21/4  1       eth  trunk  up      none                        10G(D) --
Eth1/22       1       eth  access down    SFP not inserted            40G(D) --
Eth1/23       1       eth  access down    SFP not inserted            40G(D) --
Eth1/24       1       eth  access down    SFP not inserted            40G(D) --
Eth1/25       1       eth  access down    SFP not inserted            40G(D) --
Eth1/26       1       eth  access down    SFP not inserted            40G(D) --
Eth1/27       1       eth  access down    SFP not inserted            40G(D) --
Eth1/28       1       eth  access down    SFP not inserted            40G(D) --
Eth1/29       1       eth  access down    SFP not inserted            40G(D) --
Eth1/30       1       eth  access down    SFP not inserted            40G(D) --
Eth1/31       1       eth  access down    SFP not inserted            40G(D) --
Eth1/32       1       eth  access down    SFP not inserted            40G(D) --

--------------------------------------------------------------------------------
Port-channel VLAN    Type Mode   Status  Reason                   Speed   Protocol
Interface
--------------------------------------------------------------------------------
Po1285       1       eth  vntag  up      none                      a-10G(D)  none
Po1286       1       eth  vntag  up      none                      a-10G(D)  none
Po1287       1       eth  vntag  up      none                      a-10G(D)  none
Po1288       1       eth  vntag  up      none                      a-10G(D)  none
Po1289       1       eth  vntag  up      none                      a-10G(D)  none

--------------------------------------------------------------------------------
Port   VRF          Status IP Address                              Speed    MTU
--------------------------------------------------------------------------------
mgmt0  --           down   10.197.157.252                         --       1500

--------------------------------------------------------------------------------
Vethernet     VLAN    Type Mode   Status  Reason                   Speed
--------------------------------------------------------------------------------
Veth691       4047    virt trunk  down    nonParticipating         auto
Veth692       4047    virt trunk  up      none                     auto
Veth693       1       virt trunk  down    nonParticipating         auto
Veth695       1       virt trunk  up      none                     auto
Veth699       1       virt trunk  up      none                     auto

--------------------------------------------------------------------------------
```

```
Interface Secondary VLAN(Type)                    Status Reason
------------------------------------------------------------------------------
Vlan1     --                                      down   Administratively down

------------------------------------------------------------------------------
Ethernet       VLAN   Type Mode   Status  Reason                 Speed     Port
Interface                                                                Ch #
------------------------------------------------------------------------------
Eth1/1/1       1       eth  vntag  up      none                    10G(D) 1286
Eth1/1/2       1       eth  access down    Administratively down   10G(D) --
Eth1/1/3       1       eth  vntag  up      none                    10G(D) 1286
Eth1/1/4       1       eth  access down    Administratively down   10G(D) --
Eth1/1/5       1       eth  vntag  up      none                    10G(D) 1287
Eth1/1/6       1       eth  access down    Administratively down   10G(D) --
Eth1/1/7       1       eth  vntag  up      none                    10G(D) 1287
Eth1/1/8       1       eth  access down    Administratively down   10G(D) --
Eth1/1/9       1       eth  vntag  up      none                    10G(D) 1289
Eth1/1/10      1       eth  access down    Administratively down   10G(D) --
Eth1/1/11      1       eth  vntag  up      none                    10G(D) 1289
Eth1/1/12      1       eth  access down    Administratively down   10G(D) --
Eth1/1/13      1       eth  vntag  up      none                    10G(D) 1285
Eth1/1/14      1       eth  access down    Administratively down   10G(D) --
Eth1/1/15      1       eth  vntag  up      none                    10G(D) 1285
Eth1/1/16      1       eth  access down    Administratively down   10G(D) --
Eth1/1/17      1       eth  access down    Administratively down   10G(D) --
Eth1/1/18      1       eth  vntag  up      none                    10G(D) 1288
Eth1/1/19      1       eth  access down    Administratively down   10G(D) --
Eth1/1/20      1       eth  vntag  up      none                    10G(D) 1288
Eth1/1/21      1       eth  access down    Administratively down   10G(D) --
Eth1/1/22      1       eth  access down    Administratively down   10G(D) --
Eth1/1/23      1       eth  access down    Administratively down   10G(D) --
Eth1/1/24      1       eth  access down    Administratively down   10G(D) --
Eth1/1/25      1       eth  access down    Administratively down   10G(D) --
Eth1/1/26      1       eth  access down    Administratively down   10G(D) --
Eth1/1/27      1       eth  access down    Administratively down   10G(D) --
Eth1/1/28      1       eth  access down    Administratively down   10G(D) --
Eth1/1/29      1       eth  access down    Administratively down   10G(D) --
Eth1/1/30      1       eth  access down    Administratively down   10G(D) --
Eth1/1/31      1       eth  access down    Administratively down   10G(D) --
Eth1/1/32      1       eth  access down    Administratively down   10G(D) --
Eth1/1/33      4044    eth  trunk  up      none                   1000(D) --
```

Server Ports

Configuring a Server Port

All of the port types listed are configurable on both the fixed and expansion module, including server ports, which are not configurable on the 6100 series fabric interconnect expansion module, but are configurable on the 6200 series fabric interconnect expansion module.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.

	Command or Action	Purpose
Step 2	UCS-A /eth-server # **scope fabric** {**a** | **b**}	Enters Ethernet server fabric mode for the specified fabric.
Step 3	UCS-A /eth-server/fabric # **create interface** *slot-num port-num*	Creates an interface for the specified Ethernet server port.
Step 4	UCS-A /eth-server/fabric # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create an interface for Ethernet server port 4 on slot 1 of fabric B and commit the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric b
UCS-A /eth-server/fabric # create interface 1 4
UCS-A /eth-server/fabric* # commit-buffer
UCS-A /eth-server/fabric #
```

Unconfiguring a Server Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope fabric** {**a** | **b**}	Enters Ethernet server fabric mode for the specified fabric.
Step 3	UCS-A /eth-server/fabric # **delete interface** *slot-num port-num*	Deletes the interface for the specified Ethernet server port.
Step 4	UCS-A /eth-server/fabric # **commit-buffer**	Commits the transaction to the system configuration.

The following example unconfigures Ethernet server port 12 on slot 1 of fabric B and commits the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric b
UCS-A /eth-server/fabric # delete interface 1 12
UCS-A /eth-server/fabric* # commit-buffer
UCS-A /eth-server/fabric #
```

Uplink Ethernet Ports

Configuring an Uplink Ethernet Port

You can configure uplink Ethernet ports on either the fixed module or an expansion module.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **create interface** *slot-num port-num*	Creates an interface for the specified Ethernet uplink port.
Step 4	UCS-A /eth-uplink/fabric # **set speed** {**10gbps** | **1gbps**}	(Optional) Sets the speed for the specified Ethernet uplink port. **Note** For the 6100 series fabric interconnects, the admin speed is only configurable for the first eight ports on a 20-port fabric interconnect and the first 16 ports on a 40-port fabric interconnect.
Step 5	UCS-A /eth-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create an interface for Ethernet uplink port 3 on slot 2 of fabric B, set the speed to 10 gbps, and commit the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric b
UCS-A /eth-uplink/fabric # create interface 2 3
UCS-A /eth-uplink/fabric # set speed 10gbps
UCS-A /eth-uplink/fabric* # commit-buffer
UCS-A /eth-uplink/fabric #
```

Unconfiguring an Uplink Ethernet Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.

	Command or Action	Purpose
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **delete interface** *slot-num port-num*	Deletes the interface for the specified Ethernet uplink port.
Step 4	UCS-A /eth-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

The following example unconfigures Ethernet uplink port 3 on slot 2 of fabric B and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric b
UCS-A /eth-uplink/fabric # delete interface 2 3
UCS-A /eth-uplink/fabric* # commit-buffer
UCS-A /eth-uplink/fabric #
```

Appliance Ports

Appliance ports are only used to connect fabric interconnects to directly attached NFS storage.

Note When you create a new appliance VLAN, its IEEE VLAN ID is not added to the LAN Cloud. Therefore, appliance ports that are configured with the new VLAN remain down, by default, due to a pinning failure. To bring up these appliance ports, you have to configure a VLAN in the LAN Cloud with the same IEEE VLAN ID.

Cisco UCS Manager supports up to four appliance ports per fabric interconnect.

Configuring an Appliance Port

You can configure Appliance ports on either the fixed module or an expansion module.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric**{**a** | **b**}	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **create interface** *slot-num port-num*	Creates an interface for the specified appliance port.

	Command or Action	Purpose
Step 4	UCS-A /eth-storage/fabric/interface # **set portmode** {**access** | **trunk**}	(Optional) Specifies whether the port mode is access or trunk. By default, the mode is set to trunk. **Note** If traffic for the appliance port needs to traverse the uplink ports, you must also define each VLAN used by this port in the LAN cloud. For example, you need the traffic to traverse the uplink ports if the storage is also used by other servers, or if you want to ensure that traffic fails over to the secondary fabric interconnect if the storage controller for the primary fabric interconnect fails.
Step 5	UCS-A /eth-storage/fabric/interface # **set pingroupname** *pin-group name*	(Optional) Specifies the appliance pin target to the specified fabric and port, or fabric and port channel.
Step 6	UCS-A /eth-storage/fabric/interface # **set prio** *sys-class-name*	(Optional) Specifies the QoS class for the appliance port. By default, the priority is set to best-effort. The sys-class-name argument can be one of the following class keywords: • **Fc—**Use this priority for QoS policies that control vHBA traffic only. • **Platinum—**Use this priority for QoS policies that control vNIC traffic only. • **Gold—**Use this priority for QoS policies that control vNIC traffic only. • **Silver—**Use this priority for QoS policies that control vNIC traffic only. • **Bronze—**Use this priority for QoS policies that control vNIC traffic only. • **Best Effort—**Do not use this priority. It is reserved for the Basic Ethernet traffic lane. If you assign this priority to a QoS policy and configure another system class as CoS 0, Cisco UCS Manager does not default to this system class. It defaults to the priority with CoS 0 for that traffic.
Step 7	UCS-A /eth-storage/fabric/interface # **set adminspeed** {**10gbps** | **1 gbps**}	(Optional) Specifies the admin speed for the interface. By default, the admin speed is set to 10gbps.
Step 8	UCS-A /eth-storage/fabric/interface # **commit buffer**	Commits the transaction to the system configuration.

The following example creates an interface for an appliance port 2 on slot 3 of fabric B, sets the port mode to access, pins the appliance port to a pin group called pingroup1, sets the QoS class to fc, sets the admin speed to 10 gbps, and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric b
UCS-A /eth-storage/fabric # create interface 3 2
UCS-A /eth-storage/fabric* # set portmode access
UCS-A /eth-storage/fabric* # set pingroupname pingroup1
UCS-A /eth-storage/fabric* # set prio fc
UCS-A /eth-storage/fabric* # set adminspeed 10gbps
UCS-A /eth-storage/fabric* # commit-buffer
UCS-A /eth-storage/fabric #
```

What to Do Next

Assign a VLAN or target MAC address for the appliance port.

Assigning a Target MAC Address to an Appliance Port or Appliance Port Channel

The following procedure assigns a target MAC address to an appliance port. To assign a target MAC address to an appliance port channel, scope to the port channel instead of the interface.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b**}	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **scope interface** *slot-id port-id*	Enters Ethernet interface mode for the specified interface. **Note** To assign a target MAC address to an appliance port channel, use the **scope port-channel** command instead of **scope interface** .
Step 4	UCS-A /eth-storage/fabric/interface # **create eth-target** *eth-target name*	Specifies the name for the specified MAC address target.
Step 5	UCS-A /eth-storage/fabric/interface/eth-target # **set mac-address** *mac-address*	Specifies the MAC address in nn:nn:nn:nn:nn:nn format.

The following example assigns a target MAC address for an appliance device on port 3, slot 2 of fabric B and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage* # scope fabric b
```

```
UCS-A /eth-storage/fabric* # scope interface 2 3
UCS-A /eth-storage/fabric/interface* # create eth-target macname
UCS-A /eth-storage/fabric/interface* # set mac-address 01:23:45:67:89:ab
UCS-A /eth-storage/fabric/interface* # commit-buffer
UCS-A /eth-storage/fabric #
```

The following example assigns a target MAC address for appliance devices on port channel 13 of fabric B and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage* # scope fabric b
UCS-A /eth-storage/fabric* # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel* # create eth-target macname
UCS-A /eth-storage/fabric/port-channel* # set mac-address 01:23:45:67:89:ab
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric #
```

Creating an Appliance Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A/eth-storage# **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet storage VLAN mode
Step 3	UCS-A/eth-storage/vlan# **set sharing primary**	Saves the changes.
Step 4	UCS-A/eth-storage/vlan# **commit buffer**	Commits the transaction to the system configuration.
Step 5	UCS-A/eth-storage# **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet storage VLAN mode .
Step 6	UCS-A/eth-storage/vlan# **set sharing community**	Associates the primary VLAN to the secondary VLAN that you are creating.
Step 7	UCS-A/eth-storage/vlan# **set pubnwname** *primary vlan-name*	Specifies the primary VLAN to be associated with this secondary VLAN.
Step 8	UCS-A/eth-storage/vlan# **commit buffer**	Commits the transaction to the system configuration.

The following example creates an appliance port:

```
UCS-A# scope eth-storage
UCS-A/eth-storage# create vlan PRI600 600
UCS-A/eth-storage/vlan* # set sharing primary
UCS-A/eth-storage/vlan* # commit-buffer
UCS-A/eth-storage # create vlan COM602 602
UCS-A/eth-storage/vlan* # set sharing isolated
UCS-A/eth-storage/vlan* # set pubnwname PRI600
UCS-A/eth-storage/vlan* # commit-buffer
```

Mapping an Appliance Port to a Community VLAN

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A/eth-storage# **scope fabric** {*a*|*b*}	Enters Ethernet storage fabric interconnect mode for the specified fabric interconnect.
Step 3	UCS-A/eth-storage/fabric# **create interface** *slot-num port-num*	Creates an interface for the specified Ethernet server port.
Step 4	UCS-A/eth-storage/fabric/interface# **exit**	Exits from the interface. **Note** Ensure you commit the transaction after associating with the VLAN.
Step 5	UCS-A/eth-storage/fabric# **exit**	Exits from the fabric.
Step 6	UCS-A/eth-storage# **scope vlan** *vlan-name*	Enters the specified VLAN. **Note** Ensure community VLAN is created in the appliance cloud.
Step 7	UCS-A/eth-storage/vlan# **create member-port** *fabric slot-num port-num*	Creates the member port for the specified fabric, assigns the slot number, and port number and enters member port configuration.
Step 8	UCS-A/eth-storage/vlan/member-port# **commit**	Commits the transaction to the system configuration.

The following example maps an appliance port to an community VLAN:

```
UCS-A# scope eth-storage
UCS-A/eth-storage# scope fabric a
UCS-A/eth-storage/fabric# create interface 1 22
UCS-A/eth-storage/fabric/interface*# exit
UCS-A/eth-storage/fabric*# exit
UCS-A/eth-storage*# scope vlan COM602
UCS-A/eth-storage/vlan*# create member-port a 1 22
UCS-A/eth-storage/vlan/member-port* commit
```

Unconfiguring an Appliance Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope eth-storage**	Enters Ethernet storage mode.

	Command or Action	Purpose
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b**}	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **delete eth-interface** *slot-num port-num*	Deletes the interface for the specified appliance port.
Step 4	UCS-A /eth-storage/fabric # **commit-buffer**	Commits the transaction to the system configuration.

The following example unconfigures appliance port 3 on slot 2 of fabric B and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric b
UCS-A /eth-storage/fabric # delete eth-interface 2 3
UCS-A /eth-storage/fabric* # commit-buffer
UCS-A /eth-storage/fabric #
```

FCoE Uplink Ports

FCoE uplink ports are physical Ethernet interfaces between the fabric interconnects and the upstream Ethernet switch, used for carrying FCoE traffic. With this support the same physical Ethernet port can carry both Ethernet traffic and Fibre Channel traffic.

FCoE uplink ports connect to upstream Ethernet switches using the FCoE protocol for Fibre Channel traffic. This allows both the Fibre Channel traffic and Ethernet traffic to flow on the same physical Ethernet link.

Note FCoE uplinks and unified uplinks enable the multi-hop FCoE feature, by extending the unified fabric up to the distribution layer switch.

You can configure the same Ethernet port as any of the following:

- **FCoE uplink port—**As an FCoE uplink port for only Fibre Channel traffic.
- **Uplink port—**As an Ethernet port for only Ethernet traffic.
- **Unified uplink port—**As a unified uplink port to carry both Ethernet and Fibre Channel traffic.

Configuring a FCoE Uplink Port

All of the port types listed are configurable on both the fixed and expansion module, including server ports, which are not configurable on the 6100 series fabric interconnect expansion module, but are configurable on the 6200 series fabric interconnect expansion module.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters FC Uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters FC - Uplink mode for the specific fabric.
Step 3	UCS-A /fc-uplink/fabric # **create fcoeinterface** *slot-numberport-number*	Creates interface for the specified FCoE uplink port.
Step 4	UCS-A /fc-uplink/fabric/fabricinterface # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates an interface for FCoE uplink port 1 on slot 8 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create fcoeinterface 1 8
UCS-A /fc-uplink/fabric/fcoeinterface* # commit-buffer
UCS-A /fc-uplink/fabric/fcoeinterface #
```

Unconfiguring a FCoE Uplink Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters FC Uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters FC - Uplink mode for the specific fabric.
Step 3	UCS-A /fc-uplink/fabric # **delete fcoeinterface** *slot-numberport-number*	Deletes the specified interface.
Step 4	UCS-A /fc-uplink/fabric/fabricinterface # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the FCoE uplink interface on port 1 on slot 8 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # delete fcoeinterface 1 8
UCS-A /fc-uplink/fabric/fcoeinterface* # commit-buffer
UCS-A /fc-uplink/fabric/fcoeinterface #
```

Viewing FCoE Uplink Ports

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters FC Uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric**{**a** | **b**}	Enters FC - Uplink mode for the specific fabric.
Step 3	UCS-A /fc-uplink/fabric # **show fcoeinterface**	Lists the available interfaces.

The following example displays the available FCoE uplink interfaces on fabric A:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # show fcoeinterface
FCoE Interface:

Slot Id    Port Id    Admin State Operational State Operational State Reason  Li
c State            Grace Prd
---------- ---------- ----------- ----------------- ------------------------- --
------------------ ---------
         1         26 Enabled     Indeterminate                               Li
cense Ok                    0

Fcoe Member Port:

Port-channel Slot  Port  Oper State      State Reason
------------ ----- ----- --------------- ------------
1                1    10 Sfp Not Present Unknown
1                1     3 Sfp Not Present Unknown
1                1     4 Sfp Not Present Unknown
1                1     6 Sfp Not Present Unknown
1                1     8 Sfp Not Present Unknown
2                1     7 Sfp Not Present Unknown
UCS-A /fc-uplink/fabric #
```

Unified Storage Ports

Unified storage involves configuring the same physical port as both an Ethernet storage interface and an FCoE storage interface. You can configure any appliance port or FCoE storage port as a unified storage port, on either a fixed module or an expansion module. To configure a unified storage port, you must have the fabric interconnect in Fibre Channel switching mode.

In a unified storage port, you can enable or disable individual FCoE storage or appliance interfaces.

- In an unified storage port, if you do not specify a non-default VLAN for the appliance port, the FCoE-storage-native-vlan will be assigned as the native VLAN on the unified storage port. If the appliance port has a non-default native VLAN specified as native VLAN, this will be assigned as the native VLAN for the unified storage port.
- When you enable or disable the appliance interface, the corresponding physical port is enabled or disabled. So when you disable the appliance interface in unified storage, even if the FCoE storage is enabled, it goes down with the physical port.

- When you enable or disable the FCoE storage interface, the corresponding VFC is enabled or disabled. So when the FCoE storage interface is disabled in a unified storage port, the appliance interface will continue to function normally.

Configuring a Unified Storage Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric**{**a** | **b**}	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **create interface** *slot-num port-num*	Creates an interface for the specified appliance port.
Step 4	UCS-A /eth-storage/fabric/interface* # **commit buffer**	Commits the transaction to the system configuration.
Step 5	UCS-A /eth-storage/fabric/interface* # **scope fc-storage**	Enters FC storage mode.
Step 6	UCS-A /fc-storage* # **scope fabric**{**a** | **b**}	Enters Ethernet storage mode for the specific appliance port.
Step 7	UCS-A /fc-storage/fabric # **create interface fcoe** *slot-num port-num*	Adds FCoE storage port mode on the appliance port mode and creates a unified storage port..

The following example creates an interface for an appliance port 2 on slot 3 of fabric A, adds fc storage to the same port to convert it as an unified port , and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # create interface 3 2
UCS-A /eth-storage/fabric* # commit-buffer
UCS-A /eth-storage/fabric* # scope fc-storage
UCS-A /fc-storage*# scope fabric a
UCS-A /fc-storage/fabric* # create interface fcoe 3 2
UCS-A /fc-storage/fabric* # commit-buffer
UCS-A /fc-storage/fabric*
```

Unified Uplink Ports

When you configure an Ethernet uplink and an FCoE uplink on the same physical Ethernet port, it is called a unified uplink port. You can individually enable or disable either the FCoE or Ethernet interfaces independently.

- Enabling or disabling the FCoE uplink results in the corresponding VFC being enabled or disabled.

- Enabling or disabling an Ethernet uplink results in the corresponding physical port being enabled or disabled.

If you disable an Ethernet uplink, it disables the underlying physical port in a unified uplink. Therefore, even when the FCoE uplink is enabled, the FCoE uplink also goes down. But if you disable an FCoE uplink, only the VFC goes down. If the Ethernet uplink is enabled, it can still function properly in the unified uplink port.

Configuring a Unified Uplink Port

To configure a unified uplink port, you will convert an existing FCoE uplink port as a unified port.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **create interface** *15*	Converts the FCoE uplink port as a unified port.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a unified uplink port on an existing FCoE port:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric b
UCS-A /eth-uplink/fabric # create interface 1 5
UCS-A /eth-uplink/fabric/interface* # commit-buffer
UCS-A /eth-uplink/interface #
```

FCoE and Fibre Channel Storage Ports

Configuring a Fibre Channel Storage or FCoE Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-storage**	Enters Fibre Channel storage mode.
Step 2	UCS-A /fc-storage # **scope fabric** {**a** | **b**}	Enters Fibre Channel storage mode for the specified fabric.

	Command or Action	Purpose
Step 3	UCS-A /fc-storage/fabric # **create interface** {**fc** | **fcoe**} *slot-num port-num*	Creates an interface for the specified Fibre Channel storage port.
Step 4	UCS-A /fc-storage/fabric # **commit-buffer**	Commits the transaction.

The following example creates an interface for Fibre Channel storage port 10 on slot 2 of fabric A and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage # scope fabric a
UCS-A /fc-storage/fabric* # create interface fc 2 10
UCS-A /fc-storage/fabric # commit-buffer
```

What to Do Next

Assign a VSAN.

Unconfiguring a Fibre Channel Storage or FCoE Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-storage**	Enters Fibre Channel storage mode.
Step 2	UCS-A /fc-storage # **scope fabric** {**a** | **b**}	Enters Fibre Channel storage mode for the specified fabric.
Step 3	UCS-A /fc-storage/fabric # **delete interface** {**fc** | **fcoe**} *slot-num port-num*	Deletes the interface for the specified Fibre Channel or FCoE storage port.
Step 4	UCS-A /fc-storage/fabric # **commit-buffer**	Commits the transaction.

The following example unconfigures Fibre Channel storage port 10 on slot 2 of fabric A and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage # scope fabric a
UCS-A /fc-storage/fabric* # delete interface fc 2 10
UCS-A /fc-storage/fabric # commit-buffer
```

Restoring a Fibre Channel Storage Port Back to an Uplink Fibre Channel Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **create interface** *slot-num port-num*	Creates an interface for the specified Fibre Channel uplink port.
Step 4	UCS-A /fc-uplink/fabric # **commit-buffer**	Commits the transaction.

The following example creates an interface for Fibre Channel uplink port 10 on slot 2 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric* # create interface 2 10
UCS-A /fc-uplink/fabric # commit-buffer
```

Uplink Ethernet Port Channels

An uplink Ethernet port channel allows you to group several physical uplink Ethernet ports (link aggregation) to create one logical Ethernet link to provide fault-tolerance and high-speed connectivity. In Cisco UCS Manager, you create a port channel first and then add uplink Ethernet ports to the port channel. You can add up to 16 uplink Ethernet ports to a port channel.

Important The state of a configured port changes to unconfigured in the following scenarios:

- The port is deleted or removed from a port channel. The port channel can be of any type, such as, uplink or storage.
- A port channel is deleted.

Note Cisco UCS uses Link Aggregation Control Protocol (LACP), not Port Aggregation Protocol (PAgP), to group the uplink Ethernet ports into a port channel. If the ports on the upstream switch are not configured for LACP, the fabric interconnects treat all ports in an uplink Ethernet port channel as individual ports, and therefore forward packets.

Configuring an Uplink Ethernet Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b** }	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **create port-channel** *port-num*	Creates a port channel on the specified Ethernet uplink port, and enters Ethernet uplink fabric port channel mode.
Step 4	UCS-A /eth-uplink/fabric/port-channel # {**enable** | **disable**}	(Optional) Enables or disables the administrative state of the port channel. The port channel is disabled by default.
Step 5	UCS-A /eth-uplink/fabric/port-channel # **set name** *port-chan-name*	(Optional) Specifies the name for the port channel.
Step 6	UCS-A /eth-uplink/fabric/port-channel # **set flow-control-policy** *policy-name*	(Optional) Assigns the specified flow control policy to the port channel.
Step 7	UCS-A /eth-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a port channel on port 13 of fabric A, sets the name to portchan13a, enables the administrative state, assigns the flow control policy named flow-con-pol432 to the port channel, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # create port-channel 13
UCS-A /eth-uplink/fabric/port-channel* # enable
UCS-A /eth-uplink/fabric/port-channel* # set name portchan13a
UCS-A /eth-uplink/fabric/port-channel* # set flow-control-policy flow-con-pol432
UCS-A /eth-uplink/fabric/port-channel* # commit-buffer
UCS-A /eth-uplink/fabric/port-channel #
```

Unconfiguring an Uplink Ethernet Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b** }	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **delete port-channel** *port-num*	Deletes the port channel on the specified Ethernet uplink port.
Step 4	UCS-A /eth-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

The following example unconfigures the port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # delete port-channel 13
UCS-A /eth-uplink/fabric* # commit-buffer
UCS-A /eth-uplink/fabric #
```

Adding a Member Port to an Uplink Ethernet Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b** }	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **scope port-channel** *port-num*	Enters Ethernet uplink fabric port channel mode for the specified port channel.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **create member-port** *slot-num port-num*	Creates the specified member port from the port channel and enters Ethernet uplink fabric port channel member port mode.
Step 5	UCS-A /eth-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example adds the member port on slot 1, port 7 to the port channel on port 13 of fabric A and commits the transaction.

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope port-channel 13
UCS-A /eth-uplink/fabric/port-channel # create member-port 1 7
UCS-A /eth-uplink/fabric/port-channel* # commit-buffer
UCS-A /eth-uplink/fabric/port-channel #
```

Deleting a Member Port from an Uplink Ethernet Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b** }	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **scope port-channel** *port-num*	Enters Ethernet uplink fabric port channel mode for the specified port channel.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **delete member-port** *slot-num port-num*	Deletes the specified member port from the port channel.
Step 5	UCS-A /eth-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a member port from the port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope port-channel 13
UCS-A /eth-uplink/fabric/port-channel # delete member-port 1 7
UCS-A /eth-uplink/fabric/port-channel* # commit-buffer
UCS-A /eth-uplink/fabric/port-channel #
```

Appliance Port Channels

An appliance port channel allows you to group several physical appliance ports to create one logical Ethernet storage link for the purpose of providing fault-tolerance and high-speed connectivity. In Cisco UCS Manager, you create a port channel first and then add appliance ports to the port channel. You can add up to eight appliance ports to a port channel.

Configuring an Appliance Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b** }	Enters Ethernet storage fabric mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **create port-channel** *port-num*	Creates a port channel on the specified Ethernet storage port, and enters Ethernet storage fabric port channel mode.
Step 4	UCS-A /eth-storage/fabric/port-channel # {**enable** | **disable**}	(Optional) Enables or disables the administrative state of the port channel. The port channel is disabled by default.
Step 5	UCS-A /eth-storage/fabric/port-channel # **set name** *port-chan-name*	(Optional) Specifies the name for the port channel.
Step 6	UCS-A /eth-storage/fabric/port-channel # **set pingroupname** *pin-group name*	(Optional) Specifies the appliance pin target to the specified fabric and port, or fabric and port channel.
Step 7	UCS-A /eth-storage/fabric/port-channel # **set portmode** {**access** | **trunk**}	(Optional) Specifies whether the port mode is access or trunk. By default, the mode is set to trunk.
Step 8	UCS-A /eth-storage/fabric/port-channel # **set prio** *sys-class-name*	(Optional) Specifies the QoS class for the appliance port. By default, the priority is set to best-effort. The sys-class-name argument can be one of the following class keywords: • **Fc—**Use this priority for QoS policies that control vHBA traffic only. • **Platinum—**Use this priority for QoS policies that control vNIC traffic only. • **Gold—**Use this priority for QoS policies that control vNIC traffic only. • **Silver—**Use this priority for QoS policies that control vNIC traffic only. • **Bronze—**Use this priority for QoS policies that control vNIC traffic only. • **Best Effort—**Do not use this priority. It is reserved for the Basic Ethernet traffic lane. If you assign this priority to a QoS policy and configure another system

	Command or Action	Purpose
		class as CoS 0, Cisco UCS Manager does not default to this system class. It defaults to the priority with CoS 0 for that traffic.
Step 9	UCS-A /eth-storage/fabric/port-channel # **set speed** {**1gbps** | **2gbps** | **4gbps** | **8gbps** | **auto**}	(Optional) Specifies the speed for the port channel.
Step 10	UCS-A /eth-storage/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # create port-channel 13
UCS-A /eth-storage/fabric/port-channel* # enable
UCS-A /eth-storage/fabric/port-channel* # set name portchan13a
UCS-A /eth-storage/fabric/port-channel* # set pingroupname pingroup1
UCS-A /eth-storage/fabric/port-channel* # set portmode access
UCS-A /eth-storage/fabric/port-channel* # set prio fc
UCS-A /eth-storage/fabric/port-channel* # set speed 2gbps
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric/port-channel #
```

Unconfiguring an Appliance Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b** }	Enters Ethernet storage fabric mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **delete port-channel** *port-num*	Deletes the port channel from the specified Ethernet storage port.
Step 4	UCS-A /eth-storage/fabric # **commit-buffer**	Commits the transaction to the system configuration.

The following example unconfigures the port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # delete port-channel 13
UCS-A /eth-storage/fabric* # commit-buffer
```

```
UCS-A /eth-storage/fabric #
```

Enabling or Disabling an Appliance Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b** }	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **scope port-channel** *port-chan-name*	Enters Ethernet storage port channel mode.
Step 4	UCS-A /eth-storage/fabric/port-channel # {**enable** | **disable** }	Enables or disables the administrative state of the port channel. The port channel is disabled by default.
Step 5	UCS-A /eth-storage/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables port channel 13 on fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel* # enable
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric/port-channel #
```

Adding a Member Port to an Appliance Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b** }	Enters Ethernet storage fabric mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **scope port-channel** *port-num*	Enters Ethernet storage fabric port channel mode for the specified port channel.
Step 4	UCS-A /eth-storage/fabric/port-channel # **create member-port** *slot-num port-num*	Creates the specified member port from the port channel and enters Ethernet storage fabric port channel member port mode.

	Command or Action	Purpose
Step 5	UCS-A /eth-storage/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example adds the member port on slot 1, port 7 to the port channel on port 13 of fabric A and commits the transaction.

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel # create member-port 1 7
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric/port-channel #
```

Deleting a Member Port from an Appliance Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b** }	Enters Ethernet storage fabric mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **scope port-channel** *port-num*	Enters Ethernet storage fabric port channel mode for the specified port channel.
Step 4	UCS-A /eth-storage/fabric/port-channel # **delete member-port** *slot-num port-num*	Deletes the specified member port from the port channel.
Step 5	UCS-A /eth-storage/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a member port from the port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel # delete member-port 1 7
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric/port-channel #
```

Fibre Channel Port Channels

A Fibre Channel port channel allows you to group several physical Fibre Channel ports (link aggregation) to create one logical Fibre Channel link to provide fault-tolerance and high-speed connectivity. In Cisco UCS Manager, you create a port channel first and then add Fibre Channel ports to the port channel.

Note Fibre Channel port channels are not compatible with non-Cisco technology.

You can create up to four Fibre Channel port channels in each Cisco UCS domain with Cisco UCS 6200 and 6300 Series fabric interconnects. Each Fibre Channel port channel can include a maximum of 16 uplink Fibre Channel ports.

You can create up to two Fibre Channel port channels in each Cisco UCS domain with Cisco UCS 6324 fabric interconnects. Each Fibre Channel port channel can include a maximum of four uplink Fibre Channel ports.

Ensure that the Fibre Channel port channel on the upstream NPIV switch is configured with its channel mode as **active**. If both the member port(s) and peer port(s) do not have the same channel mode configured, the port channel will not come up. When the channel mode is configured as **active**, the member ports initiate port channel protocol negotiation with the peer port(s) regardless of the channel group mode of the peer port. If the peer port, while configured in a channel group, does not support the port channel protocol, or responds with a nonnegotiable status, it defaults to the On mode behavior. The **active** port channel mode allows automatic recovery without explicitly enabling and disabling the port channel member ports at either end.

This example shows how to configure channel mode as active:

```
switch(config)# int po114
switch(config-if)# channel mode active
```

Configuring a Fibre Channel Port Channel

Note If you are connecting two Fibre Channel port channels, the admin speed for both port channels must match for the link to operate. If the admin speed for one or both of the Fibre Channel port channels is set to auto, Cisco UCS adjusts the admin speed automatically.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **create port-channel** *port-num*	Creates a port channel on the specified Fibre Channel uplink port, and enters Fibre Channel uplink fabric port channel mode.

	Command or Action	Purpose
Step 4	UCS-A /fc-uplink/fabric/port-channel # {**enable** | **disable**}	(Optional) Enables or disables the administrative state of the port channel. The port channel is disabled by default.
Step 5	UCS-A /fc-uplink/fabric/port-channel # **set name** *port-chan-name*	(Optional) Specifies the name for the port channel.
Step 6	UCS-A /fc-uplink/fabric/port-channel # **set speed** {**1gbps** | **2gbps** | **4gbps** | **8gbps** | **auto**}	(Optional) Specifies the speed for the port channel.
Step 7	UCS-A /fc-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates port channel 13 on fabric A, sets the name to portchan13a, enables the administrative state, sets the speed to 2 Gbps, and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create port-channel 13
UCS-A /fc-uplink/fabric/port-channel* # enable
UCS-A /fc-uplink/fabric/port-channel* # set name portchan13a
UCS-A /fc-uplink/fabric/port-channel* # set speed 2gbps
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

Unconfiguring a Fibre Channel Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **delete port-channel** *port-num*	Deletes the port channel on the specified Fibre Channel uplink port.
Step 4	UCS-A /fc-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

The following example unconfigures port channel 13 on fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # delete port-channel 13
UCS-A /fc-uplink/fabric* # commit-buffer
```

```
UCS-A /fc-uplink/fabric #
```

Adding Channel Mode Active To The Upstream NPIV Fibre Channel Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **create port-channel** *port-num*	Creates a port channel on the specified Fibre Channel uplink port, and enters Fibre Channel uplink fabric port channel mode.
Step 4	UCS-A /fc-uplink/fabric/port-channel # {**enable** | **disable**}	(Optional) Enables or disables the administrative state of the port channel. The port channel is disabled by default.
Step 5	UCS-A /fc-uplink/fabric/port-channel # **set name** *port-chan-name*	(Optional) Specifies the name for the port channel.
Step 6	UCS-A /fc-uplink/fabric/port-channel # **scope** *port-chan-name*	(Optional) Specifies the name for the port channel.
Step 7	UCS-A /fc-uplink/fabric/port-channel # **channel mode** {**active**}	(Optional) Configures the channel-mode active on the upstream NPIV switch.
Step 8	UCS-A /fc-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables channel mode to active:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create port-channel 13
UCS-A /fc-uplink/fabric/port-channel* # enable
UCS-A /fc-uplink/fabric/port-channel* # set name portchan13a
UCS-A /fc-uplink/fabric/port-channel* # channel mode active
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel # exit
UCS-A /fc-uplink/fabric/ # show port-channel database

portchan13a
    Administrative channel mode is active
    Operational channel mode is active

UCS-A /fc-uplink/fabric/ #
```

Enabling or Disabling a Fibre Channel Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope port-channel** *port-chan-name*	Enters Fibre Channel uplink port channel mode.
Step 4	UCS-A /fc-uplink/fabric/port-channel # {**enable** | **disable** }	Enables or disables the administrative state of the port channel. The port channel is disabled by default.

The following example enables port channel 13 on fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope port-channel 13
UCS-A /fc-uplink/fabric/port-channel* # enable
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

Adding a Member Port to a Fibre Channel Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope port-channel** *port-num*	Enters Fibre Channel uplink fabric port channel mode for the specified port channel.
Step 4	UCS-A /fc-uplink/fabric/port-channel # **create member-port** *slot-num port-num*	Creates the specified member port from the port channel and enters Fibre Channel uplink fabric port channel member port mode.
Step 5	UCS-A /fc-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example adds the member port on slot 1, port 7 to port channel 13 on fabric A and commits the transaction.

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope port-channel 13
UCS-A /fc-uplink/fabric # create member-port 1 7
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

Deleting a Member Port from a Fibre Channel Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope port-channel** *port-num*	Enters Fibre Channel uplink fabric port channel mode for the specified port channel.
Step 4	UCS-A /fc-uplink/fabric/port-channel # **delete member-port** *slot-num port-num*	Deletes the specified member port from the port channel.
Step 5	UCS-A /fc-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a member port from port channel 13 on fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope port-channel 13
UCS-A /fc-uplink/fabric # delete member-port 1 7
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

FCoE Port Channels

An FCoE port channel allows you to group several physical FCoE ports to create one logical FCoE port channel. At a physical level, the FCoE port channel carries FCoE traffic over an Ethernet port channel. So an FCoE port channel with a set of members is essentially an ethernet port channel with the same members. This Ethernet port channel is used as a physical transport for FCoE traffic.

For each FCoE port channel, Cisco UCS Manager creates a VFC internally and binds it to an Ethernet port channel. FCoE traffic received from the hosts is sent over the VFC the same way as the FCoE traffic is sent over Fibre Channel uplinks.

Configuring a FCoE Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters FC Uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric**{**a** | **b**}	Enters FC - Uplink mode for the specific fabric.
Step 3	UCS-A /fc-uplink/fabric # **create fcoe-port-channel** *number*	Creates port channel for the specified FCoE uplink port.
Step 4	UCS-A /fc-uplink/fabric/fabricinterface # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates an interface for FCoE uplink port 1 on slot 4 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create fcoe-port-channel 4
UCS-A /fc-uplink/fabric/fcoe-port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/fcoe-port-channel #
```

Adding a Member Port to a FCoE Uplink Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope fcoe-port-channel** *ID*	Enters FCoE uplink port channel mode for the specified port channel.
Step 4	UCS-A /fc-uplink/fabric/fcoe-port-channel # **create member-port** *slot-num port-num*	Creates the specified member port from the port channel and enters FCoE uplink fabric port channel member port mode. **Note** If the FCoE uplink port channel is a unified uplink port channel, you will get the following message: Warning: if this is a unified port channel then member will be added to the ethernet port channel of the same id as well.

	Command or Action	Purpose
Step 5	UCS-A /fc-uplink/fabric/fcoe-port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example adds the member port on slot 1, port 7 to FCoE port channel 13 on fabric A and commits the transaction.

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope fcoe-port-channel 13
UCS-A /fc-uplink/fabric # create member-port 1 7
UCS-A /fc-uplink/fabric/fcoe-port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/fcoe-port-channel #
```

Unified Uplink Port Channel

When you create an Ethernet port channel and an FCoE port channel with the same ID, it is called a unified uplink port channel. When the unified port channel is created, a physical Ethernet port channel and a VFC are created on the fabric interconnect with the specified members. The physical Ethernet port channel is used to carry both Ethernet and FCoE traffic. The VFC binds FCoE traffic to the Ethernet port channel.

The following rules will apply to the member port sets of the unified uplink port channel:

- The Ethernet port channel and FCoE port channel on the same ID, must have the same set of member ports.
- When you add a member port channel to the Ethernet port channel, Cisco UCS Manager adds the same port channel to FCoE port channel as well. Similarly, adding a member to the FCoE port channel adds the member port to the Ethernet port channel.
- When you delete a member port from one of the port channels, Cisco UCS Manager automatically deletes the member port from the other port channel.

If you disable an Ethernet uplink port channel, it disables the underlying physical port channel in a unified uplink port channel. Therefore, even when the FCoE uplink is enabled, the FCoE uplink port channel also goes down. If you disable an FCoE uplink port channel, only the VFC goes down. If the Ethernet uplink port channel is enabled, it can still function properly in the unified uplink port channel.

Configuring a Unified Uplink Port Channel

To configure a unified uplink port channel, you will convert an existing FCoE uplink port channel as a unified port channel.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **create port-channel** *ID*	Creates a port channel for the specified Ethernet uplink port.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a unified uplink port channel on an existing FCoE port channel:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric b
UCS-A /eth-uplink/fabric # create port-channel 2
UCS-A /eth-uplink/fabric/port-channel* # commit-buffer
UCS-A /eth-uplink/fabric #
```

Event Detection and Action

Cisco UCS Manager uses the statistics collection policy to monitor and trigger an alarm when there are faults in the network interface ports connected from the I/O module (IOM) to the fabric interconnect.

The error statistics for the network interface ports is called NiErrStats and consists of the following errors:

NiErrStats Error Name	Description
frameTx	Collects the TX_FRM_ERROR counter values.
tooLong	Collects the RX_TOOLONG counter values.
tooShort	Collects the sum of RX_UNDERSIZE and RX_FRAGMENT counter values.
Crc	Collects the sum of RX_CRERR_NOT_STOMPED and RX_CRCERR_STOMPED counter values.
inRange	Collects the RX_INRANGEERR counter values.

Note The network interface port statistics is collected only from active ports and the information is sent to Cisco UCS Manager.

Policy-Based Port Error Handling

If Cisco UCS Manager detects any errors on active NI ports, and if the error-disable feature is enabled, Cisco UCS Manager automatically disables the respective FI port that is connected to the NI port that had errors. When a FI port is error disabled, it is effectively shut down and no traffic is sent or received on that port.

The error-disable function serves two purposes:

- It lets you know which FI port is error-disabled and that the connected NI Port has errors.
- It eliminates the possibility that this port can cause other ports, which are connected to the same Chassis/FEX, to fail. Such a failure can occur when the NI port has errors, which can ultimately cause serious network issues. The error-disable function helps prevent these situations.

Creating Threshold Definition

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope eth-server**	Enters Ethernet storage mode.
Step 2	UCS-A/eth-server # **scope stats-threshold-policy default**	Enters statistics threshold policy mode.
Step 3	UCSA/eth-server/stats-threshold-policy # **create class** *class-name*	Creates the specified statistics threshold policy class and enters the organization statistics threshold policy class mode. To see a list of the available class name keywords, enter the **create class ?** command in organization threshold policy mode.
Step 4	UCS-A/eth-server/stats-threshold-policy/class # **create property** *property-name*	Creates the specified statistics threshold policy class property and enters the organization statistics threshold policy class property mode. To see a list of the available property name keywords, enter the **create property ?** command in organization threshold policy class mode.
Step 5	UCS-A/eth-server/stats-threshold-policy/class/property # **set normal-value** *value*	Specifies the normal value for the class property. The *value* format can vary depending on

	Command or Action	Purpose
		the class property being configured. To see the required format, enter the **set normal-value ?** command in organization statistics threshold policy class property mode.
Step 6	UCS-A/eth-server/stats-threshold-policy/class/property # **create threshold-value** {*above-normal* | *below-normal*} {*cleared* | *condition* | *critical* | *info* | *major* | *minor* | *warning*}	Creates the specified threshold value for the class property and enters the organization statistics threshold policy class property threshold value mode.
Step 7	UCS-A/eth-server/stats-threshold-policy/class/property/threshold-value # **set** {**deescalating** | **escalating**} *value*	Specifies the deescalating and escalating class property threshold value. The *value* format can vary depending on the class property threshold value being configured. To see the required format, enter the **set deescalating ?** or **set escalating ?** command in the organization statistics threshold policy class property threshold value mode.
Step 8	UCS-A/eth-server/stats-threshold-policy/class/property/threshold-value # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a threshold definition:

```
UCS-A # scope eth-server
UCS-A /eth-server # scope stats-threshold-policy default
UCS-A /eth-server/stats-threshold-policy # create class ni-ether-error-stats
UCS-A /eth-server/stats-threshold-policy/class* # create property crc-delta
UCS-A /eth-server/stats-threshold-policy/class/property* # set normal-value 0
UCS-A /eth-server/stats-threshold-policy/class/property* # create threshold-value above-normal
 major
UCS-A /eth-server/stats-threshold-policy/class/property/threshold-value* # set escalating
5
UCS-A /eth-server/stats-threshold-policy/class/property/threshold-value* # set deescalating
 3
UCS-A /eth-server/stats-threshold-policy/class/property/threshold-value* # commit-buffer
```

Configuring Error Disable on a Fabric Interconnect Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope eth-server**	Enters Ethernet storage mode.
Step 2	UCS-A/eth-server # **scope stats-threshold-policy default**	Enters statistics threshold policy mode.
Step 3	UCSA/eth-server/stats-threshold-policy # **scope class** *class-name*	Enters the organization statistics threshold policy class mode for the specified statistics threshold policy class.
Step 4	UCS-A/eth-server/stats-threshold-policy/class # **scope property** *property-name*	Enters the organization statistics threshold policy class property mode for the specified statistics threshold policy class property.
Step 5	UCS-A/eth-server/stats-threshold-policy/class/property # **set error-disable-fi-port** {**yes** | **no**}	Specifies the error disable state for the class property. Use the **no** option to disable error disable for the class property.
Step 6	UCS-A/eth-server/stats-threshold-policy/class/property* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to enable error disable on an FI port:

```
UCS-A # scope eth-server
UCS-A /eth-server # scope stats-threshold-policy default
UCS-A /eth-server/stats-threshold-policy # scope class ni-ether-error-stats
UCS-A /eth-server/stats-threshold-policy/class # scope property crc-delta
UCS-A /eth-server/stats-threshold-policy/class/property # set error-disable-fi-port yes
UCS-A /eth-server/stats-threshold-policy/class/property* # commit-buffer
```

Configuring Auto Recovery on a Fabric Interconnect Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope eth-server**	Enters Ethernet storage mode.
Step 2	UCS-A/eth-server # **scope stats-threshold-policy default**	Enters statistics threshold policy mode.

	Command or Action	Purpose
Step 3	UCSA/eth-server/stats-threshold-policy # **scope class** *class-name*	Enters the organization statistics threshold policy class mode for the specified statistics threshold policy class.
Step 4	UCS-A/eth-server/stats-threshold-policy/class # **scope property** *property-name*	Enters the organization statistics threshold policy class property mode for the specified statistics threshold policy class property.
Step 5	UCS-A/eth-server/stats-threshold-policy/class/property # **set auto-recovery** {**enabled** | **disabled**}	Specifies the auto recovery state for the class property. Use the **disabled** option to disable auto recovery for the class property.
Step 6	UCS-A/eth-server/stats-threshold-policy/class/property* # **set auto-recovery-time** *time*	Specifies the time in minutes after which the port is automatically re-enabled. The auto recovery time can range from 0 minutes to 4294967295 minutes.
Step 7	UCS-A/eth-server/stats-threshold-policy/class/property* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to configure auto recovery on an FI port:

```
UCS-A # scope eth-server
UCS-A /eth-server # scope stats-threshold-policy default
UCS-A /eth-server/stats-threshold-policy # scope class ni-ether-error-stats
UCS-A /eth-server/stats-threshold-policy/class # scope property crc-delta
UCS-A /eth-server/stats-threshold-policy/class/property # set auto-recovery enabled
UCS-A /eth-server/stats-threshold-policy/class/property* # set auto-recovery-time 5
UCS-A /eth-server/stats-threshold-policy/class/property* # commit-buffer
```

Viewing the Network Interface Port Error Counters

Procedure

	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 iom** {**a** | **b**}	Enters chassis IOM mode for the specified IOM.
Step 3	UCS-A/chassis/iom # **scope port-group fabric**	Enters the network interface port.

	Command or Action	Purpose
Step 4	UCS-A/chassis/iom/port-group # **scope fabric-if** *fabric-if number*	Enters the specified network interface port number.
Step 5	UCS-A/chassis/iom/port-group/fabric-if # **show stats**	Displays the error counters for the network interface port.

The following example shows how to display the statistics for the network interface ports:

```
UCS-A # scope chassis 1
UCS-A/chassis # scope iom a
UCS-A/chassis/iom # scope port-group fabric
UCS-A/chassis/iom/port-group # scope faric-if 1
UCS-A/chassis/iom/port-group/fabric-if # show stats
NI Ether Error Stats:
Time Collected: 2014-08-20T15:37:24:688
Monitored Object: sys/chassis-1/slot-1/fabric/port-1/ni-err-stats
Suspect: Yes
Crc (errors): 5000
Frame Tx (errors): 0
Too Long (errors): 0
Too Short (errors): 0
In Range (errors): 0
Thresholded: 0
```

Adapter Port Channels

An adapter port channel groups into one logical link all the physical links going from a Cisco UCS Virtual Interface Card (VIC) into an I/O.

Adapter port channels are created and managed internally by Cisco UCS Manager when it detects that the correct hardware is present. Adapter port channels cannot be configured manually. Adapter port channels are viewable using the Cisco UCS Manager GUI or the Cisco UCS Manager CLI.

Viewing Adapter Port Channels

Procedure

	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 iom** {**a** **b**}	Enters chassis IOM mode for the specified IOM.
Step 3	UCS-A /chassis/iom # **scope port group**	Enters port group mode for the specified port group.
Step 4	UCS-A /chassis/iom/port group # **show host-port-channel** [**detail** | **expand**]	Displays the adapter port channels on the specified chassis.

This following example shows how to display information on host port channels within a port group mode:

```
UCS-A # scope chassis 1
UCS-A /chassis # scope iom a
UCS-A /chassis/iom # scope port group
UCS-A /chassis/iom/port group # show host-port-channel

Host Port channel:

  Port Channel Id Fabric ID Oper State       State Reason
    --------------- --------- ---------------- ------------
                1289 B         Up
                1290 B         Up
                1306 B         Up
                1307 B         Up
                1309 B         Up
                1315 B         Up

UCS-A /chassis/iom/port group #
```

Fabric Port Channels

Fabric port channels allow you to group several of the physical links from an IOM to a fabric interconnect into one logical link for redundancy and bandwidth sharing. As long as one link in the fabric port channel remains active, the fabric port channel continues to operate.

If the correct hardware is connected, fabric port channels are created by Cisco UCS Manager in the following ways:

- During chassis discovery according to the settings configured in the chassis discovery policy.
- After chassis discovery according to the settings configured in the chassis connectivity policy for a specific chassis.

For each IOM there is a single fabric port channel. Each uplink connecting an IOM to a fabric interconnect can be configured as a discrete link or included in the port channel, but an uplink cannot belong to more than one fabric port channel. For example, if a chassis with two IOMs is discovered and the chassis discovery policy is configured to create fabric port channels, Cisco UCS Manager creates two separate fabric port channels: one for the uplinks connecting IOM-1 and another for the uplinks connecting IOM-2. No other chassis can join these fabric port channels. Similarly, uplinks belonging to the fabric port channel for IOM-1 cannot join the fabric port channel for IOM-2.

Load Balancing Over Ports

Load balancing traffic among ports between IOMs and fabric interconnects uses the following criteria for hashing.

- For Ethernet traffic:

  Layer 2 source and destination address

  Layer 3 source and destination address

  Layer 4 source and destination ports

- For FCoE traffic:

Layer 2 source and destination address

Source and destination IDs (SID and DID) and Originator Exchange ID (OXID)

In this example, a 2200 Series IOM module is verified by connecting iom *X* (where *X* is the chassis number).

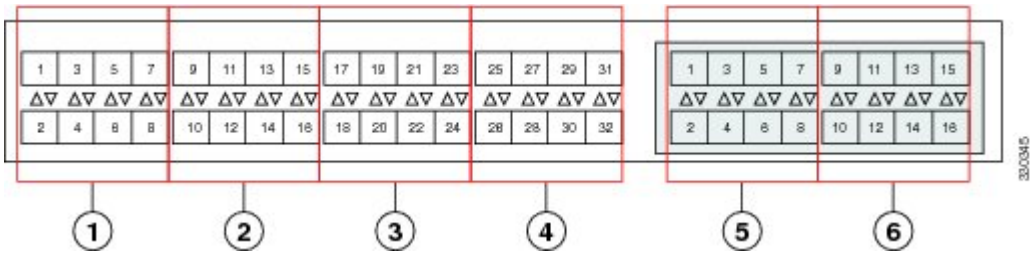
```
show platform software fwmctrl nifport
(....)
 Hash Parameters:
   l2_da: 1 l2_sa: 1 l2_vlan: 0
   l3_da: 1 l3_sa: 1
   l4_da: 1 l4_sa: 1
   FCoE l2_da: 1 l2_sa: 1 l2_vlan: 0
   FCoE l3_did: 1 l3_sid: 1 l3_oxid: 1
```

Cabling Considerations for Fabric Port Channels

When you configure the links between the Cisco UCS 2200 Series FEX and a Cisco UCS 6200 series fabric interconnect in fabric port channel mode, the available virtual interface namespace (VIF) on the adapter varies depending on where the FEX uplinks are connected to the fabric interconnect ports.

Inside the 6248 fabric interconnect there are six sets of eight contiguous ports, with each set of ports managed by a single chip. When all uplinks from an FEX are connected to a set of ports managed by a single chip, Cisco UCS Manager maximizes the number of VIFs used in service profiles deployed on the blades in the chassis. If uplink connections from an IOM are distributed across ports managed by separate chips, the VIF count is decreased.

Figure 1: Port Groups for Fabric Port Channels



Caution Adding a second link to a fabric-port-channel port group is disruptive and will automatically increase the available amount of VIF namespace from 63 to 118. Adding further links is not disruptive and the VIF namespace stays at 118.

Caution Linking a chassis to two fabric-port-channel port groups does not affect the VIF namespace unless it is manually acknowledged. The VIF namespace is then automatically set to the smaller size fabric port-channel port group usage (either 63 or 118 VIFs) of the two groups.

For high availability cluster-mode applications, we strongly recommend symmetric cabling configurations. If the cabling is asymmetric, the maximum number of VIFs available is the smaller of the two cabling configurations.

For more information on the maximum number of VIFs for your Cisco UCS environment, see the Configuration Limits document for your hardware and software configuration.

Configuring a Fabric Port Channel

Procedure

Step 1 To include all links from the IOM to the fabric interconnect in a fabric port channel during chassis discovery, set the link grouping preference in the chassis discovery policy to port channel.

Step 2 To include links from individual chassis in a fabric port channel during chassis discovery, set the link grouping preference in the chassis connectivity policy to port channel.

Step 3 After chassis discovery, enable or disable additional fabric port channel member ports.

What to Do Next

To add or remove chassis links from a fabric port channel after making a change to the chassis discovery policy or the chassis connectivity policy, reacknowledge the chassis. Chassis reacknowledgement is not required to enable or disable chassis member ports from a fabric port channel

Viewing Fabric Port Channels

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope fabric** {**a** | **b**}	Enters Ethernet server fabric mode for the specified fabric.
Step 3	UCS-A /eth-server/fabric # **show fabric-port-channel** [**detail** | **expand**]	Displays fabric port channels on the specified fabric interconnect.

The following example displays information about configured fabric port channels on fabric interconnect A:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric a
UCS-A /eth-server/fabric # show fabric-port-channel
Fabric Port Channel:
    Port Channel Id Chassis Id Admin State Oper State       State Reason
    --------------- ---------- ----------- ---------------- ------------
               1025 1          Enabled     Failed           No operational members
               1026 2          Enabled     Up

UCS-A /eth-server/fabric #
```

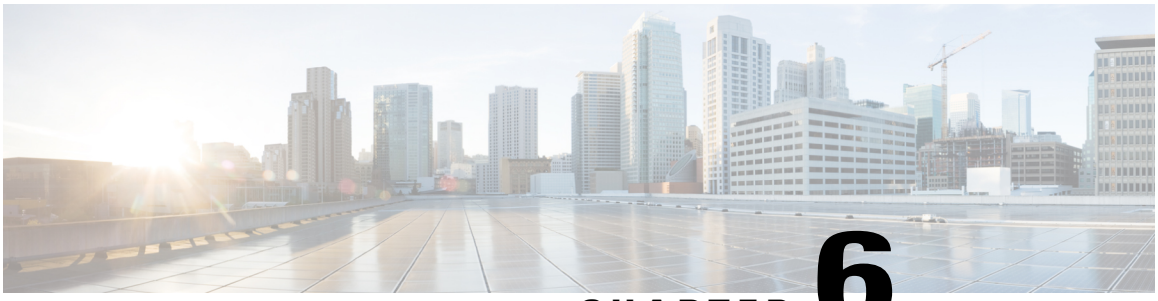
Enabling or Disabling a Fabric Port Channel Member Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope fabric** {**a** | **b**}	Enters Ethernet server fabric mode for the specified fabric.
Step 3	UCS-A /eth-server/fabric # **scope fabric-port-channel** *port-chan-id*	Enters Ethernet server fabric, fabric port channel mode for the specified fabric.
Step 4	UCS-A /eth-server/fabric/fabric-port-channel # **scope member-port** *slot-id port-id*	Enters Ethernet server fabric, fabric port channel mode for the specified member port.
Step 5	UCS-A /eth-server/fabric/fabric-port-channel # {**enable** | **disable**}	Enables or disables the specified member port.
Step 6	UCS-A /eth-server/fabric/fabric-port-channel # **commit-buffer**	Commits the transaction to the system configuration.

The following example disables fabric channel member port 1 31 on fabric port channel 1025 and commits the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric a
UCS-A /eth-server/fabric # scope fabric-port-channel 1025
UCS-A /eth-server/fabric/fabric-port-channel # scope member-port 1 31
UCS-A /eth-server/fabric/fabric-port-channel/member-port # disable
UCS-A /eth-server/fabric/fabric-port-channel/member-port* # commit-buffer
UCS-A /eth-server/fabric/fabric-port-channel/member-port #
```



CHAPTER 6

Configuring Communication Services

This chapter includes the following sections:

Communication Services

You can use the communication services defined below to interface third-party applications with Cisco UCS. Cisco UCS Manager supports IPv4 and IPv6 address access for the following services:

- CIM XML
- HTTP
- HTTPS
- SNMP
- SSH
- Telnet

Cisco UCS Manager supports out-of-band IPv4 address access to the **Cisco UCS KVM Direct** launch page from a web browser. To provide this access, you must enable the following service:

- CIMC Web Service

Communication Service	Description
CIM XML	The Common Information Model (CIM) XML) service is disabled by default and is only available in read-only mode. The default port is 5988. The CIM XML is a standards-based protocol for exchanging CIM information that the Distributed Management Task Force defines.
CIMC Web Service	This service is disabled by default. When this service is enabled, users can directly access a server CIMC using one of the out-of-band management IP addresses assigned directly to the server, or associated with the server through a service profile. **Note** CIMC Web Service can only be enabled or disabled globally. You cannot configure KVM direct access for individual CIMC IP addresses.
HTTP	By default, HTTP is enabled on port 80. You can run the Cisco UCS Manager GUI in an HTTP or HTTPS browser. If you select HTTP, all data is exchanged in clear text mode. For a secure browser session, we recommend that you enable HTTPS and disable HTTP. By default, Cisco UCS implements a browser redirects to an HTTPS equivalent and recommends that you do not change this behavior. **Note** If you are upgrading to Cisco UCS, version 1.4(1), the browser redirect to a secure browser does not occur by default. To redirect the HTTP browser to an HTTPS equivalent, enable the **Redirect HTTP to HTTPS** in Cisco UCS Manager.
HTTPS	By default, HTTPS is enabled on port. With HTTPS, all data is exchanged in encrypted mode through a secure server. For a secure browser session, We recommend that you only use HTTPS and either disable or redirect HTTP communications.
SMASH CLP	This service is enabled for read-only access and supports a limited subset of the protocols, such as the **show** command. You cannot disable it. This shell service is one of the standards that the Distributed Management Task Force defines.
SNMP	By default, this service is disabled. If enabled, the default port is 161. You must configure the community and at least one SNMP trap. Enable this service only if your system includes integration with an SNMP server.

Communication Service	Description
SSH	This service is enabled on port 22. You cannot disable it, and you cannot change the default port. This service provides access to the Cisco UCS Manager CLI.
Telnet	By default, this service is disabled. This service provides access to the Cisco UCS Manager CLI.

Configuring CIM XML

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **enable cimxml**	Enables the CIM XLM service.
Step 4	UCS-A /system/services # **set cimxml port** *port-num*	Specifies the port for the CIM XML connection.
Step 5	UCS-A /system/services # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables CIM XML, sets the port number to 5988, and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # enable cimxml
UCS-A /system/services* # set cimxml port 5988
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

Configuring HTTP

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.

	Command or Action	Purpose
Step 3	UCS-A /system/services # **enable http**	Enables the HTTP service.
Step 4	UCS-A /system/services # **set http port** *port-num*	Specifies the port to be used for the HTTP connection.
Step 5	UCS-A /system/services # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables HTTP, sets the port number to 80, and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # enable http
UCS-A /system/services* # set http port 80
Warning: When committed, this closes all the web sessions.
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

Unconfiguring HTTP

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **disable http**	Disables the HTTP service.
Step 4	UCS-A /system/services # **commit-buffer**	Commits the transaction to the system configuration.

The following example disables HTTP and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # disable http
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

Configuring HTTPS

Certificates, Key Rings, and Trusted Points

HTTPS uses components of the Public Key Infrastructure (PKI) to establish secure communications between two devices, such as a client's browser and Cisco UCS Manager.

Encryption Keys and Key Rings

Each PKI device holds a pair of asymmetric Rivest-Shamir-Adleman (RSA) encryption keys, one kept private and one made public, stored in an internal key ring. A message encrypted with either key can be decrypted with the other key. To send an encrypted message, the sender encrypts the message with the receiver's public key, and the receiver decrypts the message using its own private key. A sender can also prove its ownership of a public key by encrypting (also called 'signing') a known message with its own private key. If a receiver can successfully decrypt the message using the public key in question, the sender's possession of the corresponding private key is proven. Encryption keys can vary in length, with typical lengths from 512 bits to 2048 bits. In general, a longer key is more secure than a shorter key. Cisco UCS Manager provides a default key ring with an initial 1024-bit key pair, and allows you to create additional key rings.

The default key ring certificate must be manually regenerated if the cluster name changes or the certificate expires.

This operation is only available in the UCS Manager CLI.

Certificates

To prepare for secure communications, two devices first exchange their digital certificates. A certificate is a file containing a device's public key along with signed information about the device's identity. To merely support encrypted communications, a device can generate its own key pair and its own self-signed certificate. When a remote user connects to a device that presents a self-signed certificate, the user has no easy method to verify the identity of the device, and the user's browser will initially display an authentication warning. By default, Cisco UCS Manager contains a built-in self-signed certificate containing the public key from the default key ring.

Trusted Points

To provide stronger authentication for Cisco UCS Manager, you can obtain and install a third-party certificate from a trusted source, or trusted point, that affirms the identity of your device. The third-party certificate is signed by the issuing trusted point, which can be a root certificate authority (CA) or an intermediate CA or trust anchor that is part of a trust chain that leads to a root CA. To obtain a new certificate, you must generate a certificate request through Cisco UCS Manager and submit the request to a trusted point.

Important The certificate must be in Base64 encoded X.509 (CER) format.

Creating a Key Ring

Cisco UCS Manager supports a maximum of 8 key rings, including the default key ring.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **create keyring** *keyring-name*	Creates and names the key ring.
Step 3	UCS-A /security/keyring # **set modulus** {**mod1024** | **mod1536** | **mod2048** | **mod512**}	Sets the SSL key length in bits.

	Command or Action	Purpose
Step 4	UCS-A /security/keyring # **commit-buffer**	Commits the transaction.

The following example creates a keyring with a key size of 1024 bits:

```
UCS-A# scope security
UCS-A /security # create keyring kr220
UCS-A /security/keyring* # set modulus mod1024
UCS-A /security/keyring* # commit-buffer
UCS-A /security/keyring #
```

What to Do Next

Create a certificate request for this key ring.

Regenerating the Default Key Ring

The default key ring certificate must be manually regenerated if the cluster name changes or the certificate expires.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope keyring default**	Enters key ring security mode for the default key ring.
Step 3	UCS-A /security/keyring # **set regenerate yes**	Regenerates the default key ring.
Step 4	UCS-A /security/keyring # **commit-buffer**	Commits the transaction.

The following example regenerates the default key ring:

```
UCS-A# scope security
UCS-A /security # scope keyring default
UCS-A /security/keyring* # set regenerate yes
UCS-A /security/keyring* # commit-buffer
UCS-A /security/keyring #
```

Creating a Certificate Request for a Key Ring

Creating a Certificate Request for a Key Ring with Basic Options

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope keyring** *keyring-name*	Enters configuration mode for the key ring.
Step 3	UCS-A /security/keyring # **create certreq** {**ip** [*ipv4-addr* | *ipv6-v6*] |**subject-name** *name*}	Creates a certificate request using the IPv4 or IPv6 address specified, or the name of the fabric interconnect. You are prompted to enter a password for the certificate request.
Step 4	UCS-A /security/keyring/certreq # **commit-buffer**	Commits the transaction.
Step 5	UCS-A /security/keyring # **show certreq**	Displays the certificate request, which you can copy and send to a trust anchor or certificate authority.

The following example creates and displays a certificate request with an IPv4 address for a key ring, with basic options:

```
UCS-A# scope security
UCS-A /security # scope keyring kr220
UCS-A /security/keyring # create certreq ip 192.168.200.123 subject-name sjc04
Certificate request password:
Confirm certificate request password:
UCS-A /security/keyring* # commit-buffer
UCS-A /security/keyring # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request e-mail name:
Certificate request country name:
State, province or county (full name):
Locality (eg, city):
Organization name (eg, company):
Organization Unit name (eg, section):
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYW1jMDQwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKn1t8qMZO4UGqILKFXQQc2c8b/vW2rnRF8OPhKbhghLA1YZ1F
JqcYEG5Yl1+vgohLBTd45s0GC8m4RTLJWHo4SwccAUXQ5Zngf45YtX1WsylwUWV4
0re/zgTk/WCd56RfOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQQMA6CBnNhbWMwNIcECsEiXjAN
BgkqhkiG9w0BAQQFAAOBgQCsxN0qUHYGFoQw56RwQueLTNPnrndqUwuZHUO03Teg
nhsyu4satpyiPqVV9viKZ+spvc6x5PWIcTWgHhH8BimOb/0OKuG8kwfIGGsEDlAv
TTYvUP+BZ9OFiPbRIA718S+V8ndXr1HejiQGxlDNqoN+odCXPc5kjoXD0lZTL09H
BA==
-----END CERTIFICATE REQUEST-----

UCS-A /security/keyring #
```

Creating a Certificate Request for a Key Ring with Advanced Options

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope keyring** *keyring-name*	Enters configuration mode for the key ring.
Step 3	UCS-A /security/keyring # **create certreq**	Creates a certificate request.
Step 4	UCS-A /security/keyring/certreq* # **set country** *country name*	Specifies the country code of the country in which the company resides.
Step 5	UCS-A /security/keyring/certreq* # **set dns** *DNS Name*	Specifies the Domain Name Server (DNS) address associated with the request.
Step 6	UCS-A /security/keyring/certreq* # **set e-mail** *E-mail name*	Specifies the email address associated with the certificate request.
Step 7	UCS-A /security/keyring/certreq* # **set ip** {*certificate request ip-address*|*certificate request ip6-address* }	Specifies the IP address of the Fabric Interconnect.
Step 8	UCS-A /security/keyring/certreq* # **set locality** *locality name (eg, city)*	Specifies the city or town in which the company requesting the certificate is headquartered.
Step 9	UCS-A /security/keyring/certreq* # **set org-name** *organization name*	Specifies the organization requesting the certificate.
Step 10	UCS-A /security/keyring/certreq* # **set org-unit-name** *organizational unit name*	Specifies the organizational unit.
Step 11	UCS-A /security/keyring/certreq* # **set password** *certificate request password*	Specifies an optional password for the certificate request.
Step 12	UCS-A /security/keyring/certreq* # **set state** *state, province or county*	Specifies the state or province in which the company requesting the certificate is headquartered.
Step 13	UCS-A /security/keyring/certreq* # **set subject-name** *certificate request name*	Specifies the fully qualified domain name of the Fabric Interconnect.
Step 14	UCS-A /security/keyring/certreq* # **commit-buffer**	Commits the transaction.
Step 15	UCS-A /security/keyring # **show certreq**	Displays the certificate request, which you can copy and send to a trust anchor or certificate authority.

The following example creates and displays a certificate request with an IPv4 address for a key ring, with advanced options:

```
UCS-A# scope security
UCS-A /security # scope keyring kr220
UCS-A /security/keyring # create certreq
UCS-A /security/keyring/certreq* # set ip 192.168.200.123
UCS-A /security/keyring/certreq* #  set subject-name sjc04
UCS-A /security/keyring/certreq* # set country US
UCS-A /security/keyring/certreq* # set dns bgl-samc-15A
UCS-A /security/keyring/certreq* # set email test@cisco.com
UCS-A /security/keyring/certreq* # set locality new york city
UCS-A /security/keyring/certreq* # set org-name "Cisco Systems"
UCS-A /security/keyring/certreq* # set org-unit-name Testing
UCS-A /security/keyring/certreq* # set state new york
UCS-A /security/keyring/certreq* # commit-buffer
UCS-A /security/keyring/certreq # show certreq
Certificate request subject name: sjc04
Certificate request ip address: 192.168.200.123
Certificate request e-mail name: test@cisco.com
Certificate request country name: US
State, province or county (full name): New York
Locality name (eg, city): new york city
Organization name (eg, company): Cisco
Organization Unit name (eg, section): Testing
Request:
-----BEGIN CERTIFICATE REQUEST-----
MIIBfTCB5wIBADARMQ8wDQYDVQQDEwZzYW1jMDQwgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBALpKn1t8qMZO4UGqILKFXQQc2c8b/vW2rnRF8OPhKbhghLA1YZ1F
JqcYEG5Yll+vgohLBTd45s0GC8m4RTLJWHo4SwccAUXQ5Zngf45YtX1WsylwUWV4
0re/zgTk/WCd56RfOBvWR2Dtztu2pGA14sd761zLxt29K7R8mzj6CAUVAgMBAAGg
LTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQEB/wQQMA6CBnNhbWMwNIcECsEiXjAN
BgkqhkiG9w0BAQQFAAOBgQCsxN0qUHYGFoQw56RwQueLTNPnrndqUwuZHUO03Teg
nhsyu4satpyiPqVV9viKZ+spvc6x5PWIcTWgHhH8BimOb/0OKuG8kwfIGGsEDlAv
TTYvUP+BZ9OFiPbRIA718S+V8ndXr1HejiQGxlDNqoN+odCXPc5kjoXD0lZTL09H
BA==
-----END CERTIFICATE REQUEST-----

UCS-A /security/keyring/certreq #
```

What to Do Next

- Copy the text of the certificate request, including the BEGIN and END lines, and save it in a file. Send the file with the certificate request to a trust anchor or certificate authority to obtain a certificate for the key ring.
- Create a trusted point and set the certificate chain for the certificate of trust received from the trust anchor.

Creating a Trusted Point

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **create trustpoint** *name*	Creates and names a trusted point.
Step 3	UCS-A /security/trustpoint # **set certchain** [*certchain*]	Specifies certificate information for this trusted point.

	Command or Action	Purpose
		If you do not specify certificate information in the command, you are prompted to enter a certificate or a list of trustpoints defining a certification path to the root certificate authority (CA). On the next line following your input, type ENDOFBUF to finish. **Important** The certificate must be in Base64 encoded X.509 (CER) format.
Step 4	UCS-A /security/trustpoint # **commit-buffer**	Commits the transaction.

The following example creates a trusted point and provides a certificate for the trusted point:

```
UCS-A# scope security
UCS-A /security # create trustpoint tPoint10
UCS-A /security/trustpoint* # set certchain
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Trustpoint Certificate Chain:
> -----BEGIN CERTIFICATE-----
> MIIDMDCCApmgAwIBAgIBADANBgkqhkiG9w0BAQQFADB0MQswCQYDVQQGEwJVUzEL
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGxlIEluYy4xEzARBgNVBAsT
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWEHVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMivyCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBNKONDl
> GMbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bF5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzCl903O6Mg51zq1zXcz75+VFj2I6rH9asckCld3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtvlWvfhevskV0j6
> jtcEMyZ+f7+3yh421ido3nO4MIGeBgNVHSMEgZYwgZOAFLlNjtcEMyZ+f7+3yh42
> 1ido3nO4oXikdjB0MQswCQYDVQQGEwJVUzELMAkGA1UECBMCQ0ExFDASBgNVBAcT
> C1NhbnRhIENsYXJhMRswGQYDVQQKExJOdW92YSBTeXN0ZW1zIEluYy4xFDASBgNV
> BAsTC0VuZ2luZWVyaW5nMQ8wDQYDVQQDEwZ0ZXN0Q0GCAQAwDAYDVR0TBAUwAwEB
> /zANBgkqhkiG9w0BAQQFAAOBgQAhWaRwXNR6B4g6Lsnr+fptHv+WVhB5fKqGQqXc
> wR4pYiO4z42/j9Ijenh75tCKMhW51az8copP1EBmOcyuhf5C6vasrenn1ddkkYt4
> PR0vxGc40whuiozBolesmsmjBbedUCwQgdFDWhDIZJwK5+N3x/kfa2EHU6id1avt
> 4YL5Jg==
> -----END CERTIFICATE-----
> ENDOFBUF
UCS-A /security/trustpoint* # commit-buffer
UCS-A /security/trustpoint #
```

What to Do Next

Obtain a key ring certificate from the trust anchor or certificate authority and import it into the key ring.

Importing a Certificate into a Key Ring

Before You Begin

- Configure a trusted point that contains the certificate chain for the key ring certificate.
- Obtain a key ring certificate from a trust anchor or certificate authority.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope keyring** *keyring-name*	Enters configuration mode for the key ring that will receive the certificate.
Step 3	UCS-A /security/keyring # **set trustpoint** *name*	Specifies the trusted point for the trust anchor or certificate authority from which the key ring certificate was obtained.
Step 4	UCS-A /security/keyring # **set cert**	Launches a dialog for entering and uploading the key ring certificate. At the prompt, paste the certificate text that you received from the trust anchor or certificate authority. On the next line following the certificate, type ENDOFBUF to complete the certificate input. **Important** The certificate must be in Base64 encoded X.509 (CER) format.
Step 5	UCS-A /security/keyring # **commit-buffer**	Commits the transaction.

The following example specifies the trust point and imports a certificate into a key ring:

```
UCS-A# scope security
UCS-A /security # scope keyring kr220
UCS-A /security/keyring # set trustpoint tPoint10
UCS-A /security/keyring* # set cert
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
Keyring certificate:
> -----BEGIN CERTIFICATE-----
> MIIB/zCCAWgCAQAwgZkxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJDQTEVMBMGA1UE
> BxMMU2FuIEpvc2UsIENBMRUwEwYDVQQKEwxFeGFtcGxlIEluYy4xEzARBgNVBAsT
> ClRlc3QgR3JvdXAxGTAXBgNVBAMTEHRlc3QuZXhhbXBsZS5jb20xHzAdBgkqhkiG
> 9w0BCQEWEHVzZXJAZXhhbXBsZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
> AoGBAMZw4nTepNIDhVzb0j7Z2Je4xAG56zmSHRMQeOGHemdh66u2/XAoLx7YCcYU
> ZgAMivyCsKgb/6CjQtsofvtrmC/eAehuK3/SINv7wd6Vv2pBt6ZpXgD4VBNKOND1
> GMbkPayVlQjbG4MD2dx2+H8EH3LMtdZrgKvPxPTE+bF5wZVNAgMBAAGgJTAjBgkq
> hkiG9w0BCQcxFhMUQSBjaGFsbGVuZ2UgcGFzc3dvcmQwDQYJKoZIhvcNAQEFBQAD
> gYEAG61CaJoJaVMhzCl903O6Mg51zq1zXcz75+VFj2I6rH9asckCld3mkOVx5gJU
> Ptt5CVQpNgNLdvbDPSsXretysOhqHmp9+CLv8FDuy1CDYfuaLtvlWvfhevskV0j6
> mK3Ku+YiORnv6DhxrOoqau8r/hyI/L43l7IPN1HhOi3oha4=
> -----END CERTIFICATE-----
> ENDOFBUF
UCS-A /security/keyring* # commit-buffer
UCS-A /security/keyring #
```

What to Do Next

Configure your HTTPS service with the key ring.

Configuring HTTPS

Caution After you complete the HTTPS configuration, including changing the port and key ring for the HTTPS to use, all current HTTP and HTTPS sessions are closed without warning as soon as you save or commit the transaction.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **enable https**	Enables the HTTPS service.
Step 4	UCS-A /system/services # **set https port** *port-num*	(Optional) Specifies the port to be used for the HTTPS connection.
Step 5	UCS-A /system/services # **set https keyring** *keyring-name*	(Optional) Specifies the name of the key ring you created for HTTPS.
Step 6	UCS-A /system/services # **set https cipher-suite-mode** *cipher-suite-mode*	(Optional) The level of Cipher Suite security used by the Cisco UCS domain. *cipher-suite-mode* can be one of the following keywords: • **high-strength** • **medium-strength** • **low-strength** • **custom—**Allows you to specify a user-defined Cipher Suite specification string.
Step 7	UCS-A /system/services # **set https cipher-suite** *cipher-suite-spec-string*	(Optional) Specifies a custom level of Cipher Suite security for this Cisco UCS domain if **cipher-suite-mode** is set to **custom**. *cipher-suite-spec-string* can contain up to 256 characters and must conform to the OpenSSL Cipher Suite specifications. You cannot use any spaces or special characters except ! (exclamation point), + (plus sign), - (hyphen), and : (colon). For details, see http://httpd.apache.org/docs/2.0/mod/mod_ssl.html#sslciphersuite.. For example, the medium strength specification string Cisco UCS Manager uses as the default is: ALL:!ADH:!EXPORT56:!LOW:RC4+RSA:+HIGH:+MEDIUM:+EXP:+eNULL

	Command or Action	Purpose
		Note This option is ignored if **cipher-suite-mode** is set to anything other than **custom**.
Step 8	UCS-A /system/services # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables HTTPS, sets the port number to 443, sets the key ring name to kring7984, sets the Cipher Suite security level to high, and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # enable https
UCS-A /system/services* # set https port 443
Warning: When committed, this closes all the web sessions.
UCS-A /system/services* # set https keyring kring7984
UCS-A /system/services* # set https cipher-suite-mode high
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

Deleting a Key Ring

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **delete keyring** *name*	Deletes the named key ring.
Step 3	UCS-A /security # **commit-buffer**	Commits the transaction.

The following example deletes a key ring:

```
UCS-A# scope security
UCS-A /security # delete keyring key10
UCS-A /security* # commit-buffer
UCS-A /security #
```

Deleting a Trusted Point

Before You Begin

Ensure that the trusted point is not used by a key ring.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **delete trustpoint** *name*	Deletes the named trusted point.
Step 3	UCS-A /security # **commit-buffer**	Commits the transaction.

The following example deletes a trusted point:

```
UCS-A# scope security
UCS-A /security # delete trustpoint tPoint10
UCS-A /security* # commit-buffer
UCS-A /security #
```

Unconfiguring HTTPS

Before You Begin

Disable HTTP to HTTPS redirection.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **disable https**	Disables the HTTPS service.
Step 4	UCS-A /system/services # **commit-buffer**	Commits the transaction to the system configuration.

The following example disables HTTPS and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # disable https
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

Enabling HTTP Redirection to HTTPS

Before You Begin

Enable both HTTP and HTTPS.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **enable http-redirect**	Enables the HTTP redirect service. If enabled, all attempts to communicate via HTTP are redirected to the equivalent HTTPS address. This option effectively disables HTTP access to this Cisco UCS domain.
Step 4	UCS-A /system/services # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables HTTP to HTTPS redirection and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # enable http-redirect
Warning: When committed, this closes all the web sessions.
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

Enabling SNMP

SNMP Overview

The Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language for monitoring and managing devices in a network.

SNMP Functional Overview

The SNMP framework consists of three parts:

- An SNMP manager—The system used to control and monitor the activities of network devices using SNMP.
- An SNMP agent—The software component within Cisco UCS, the managed device that maintains the data for Cisco UCS, and reports the data as needed to the SNMP manager. Cisco UCS includes the agent and a collection of MIBs. To enable the SNMP agent and create the relationship between the manager and agent, enable and configure SNMP in Cisco UCS Manager.
- A managed information base (MIB)—The collection of managed objects on the SNMP agent. Cisco UCS release 1.4(1) and higher supports a larger number of MIBs than earlier releases.

Cisco UCS supports SNMPv1, SNMPv2c and SNMPv3. Both SNMPv1 and SNMPv2c use a community-based form of security. SNMP is defined in the following:

- RFC 3410 (http://tools.ietf.org/html/rfc3410)
- RFC 3411 (http://tools.ietf.org/html/rfc3411)
- RFC 3412 (http://tools.ietf.org/html/rfc3412)
- RFC 3413 (http://tools.ietf.org/html/rfc3413)
- RFC 3414 (http://tools.ietf.org/html/rfc3414)
- RFC 3415 (http://tools.ietf.org/html/rfc3415)
- RFC 3416 (http://tools.ietf.org/html/rfc3416)
- RFC 3417 (http://tools.ietf.org/html/rfc3417)
- RFC 3418 (http://tools.ietf.org/html/rfc3418)
- RFC 3584 (http://tools.ietf.org/html/rfc3584)

SNMP Notifications

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

Cisco UCS Manager generates SNMP notifications as either traps or informs. Traps are less reliable than informs because the SNMP manager does not send any acknowledgment when it receives a trap, and Cisco UCS Manager cannot determine if the trap was received. An SNMP manager that receives an inform request acknowledges the message with an SNMP response Protocol Data Unit (PDU). If the Cisco UCS Manager does not receive the PDU, it can send the inform request again.

SNMP Security Levels and Privileges

SNMPv1, SNMPv2c, and SNMPv3 each represent a different security model. The security model combines with the selected security level to determine the security mechanism applied when the SNMP message is processed.

The security level determines the privileges required to view the message associated with an SNMP trap. The privilege level determines whether the message requires protection from disclosure or whether the message is authenticated. The supported security level depends on which security model is implemented. SNMP security levels support one or more of the following privileges:

- noAuthNoPriv—No authentication or encryption
- authNoPriv—Authentication but no encryption
- authPriv—Authentication and encryption

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy that is set up for a user and the role in which the user resides. A security level is the permitted level of security

within a security model. A combination of a security model and a security level determines which security mechanism is employed when handling an SNMP packet.

Supported Combinations of SNMP Security Models and Levels

The following table identifies the combinations of security models and levels.

Table 5: SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	HMAC-MD5 or HMAC-SHA	No	Provides authentication based on the Hash-Based Message Authentication Code (HMAC) Message Digest 5 (MD5) algorithm or the HMAC Secure Hash Algorithm (SHA).
v3	authPriv	HMAC-MD5 or HMAC-SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides Data Encryption Standard (DES) 56-bit encryption in addition to authentication based on the Cipher Block Chaining (CBC) DES (DES-56) standard.

SNMPv3 Security Features

SNMPv3 provides secure access to devices through a combination of authenticating and encrypting frames over the network. SNMPv3 authorizes only configured users to perform management operations and encrypts SNMP messages. The SNMPv3 User-Based Security Model (USM) refers to SNMP message-level security and offers the following services:

- Message integrity—Ensures that messages are not altered or destroyed in an unauthorized manner, and that data sequences are not altered beyond what can occur non-maliciously.
- Message origin authentication—Ensures that the identity of a message originator is verifiable.
- Message confidentiality and encryption—Ensures that information is not made available or disclosed to unauthorized individuals, entities, or processes.

SNMP Support in Cisco UCS

Cisco UCS provides the following support for SNMP:

Support for MIBs

Cisco UCS supports read-only access to MIBs.

For information about the specific MIBs available for Cisco UCS and where you can obtain them, see the http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/mib/b-series/b_UCS_MIBRef.html for B-series servers, and http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/mib/c-series/b_UCS_Standalone_C-Series_MIBRef.html C-series servers.

Authentication Protocols for SNMPv3 Users

Cisco UCS supports the following authentication protocols for SNMPv3 users:

- HMAC-MD5-96 (MD5)
- HMAC-SHA-96 (SHA)

AES Privacy Protocol for SNMPv3 Users

Cisco UCS uses Advanced Encryption Standard (AES) as one of the privacy protocols for SNMPv3 message encryption and conforms with RFC 3826.

The privacy password, or priv option, offers a choice of DES or 128-bit AES encryption for SNMP security encryption. If you enable AES-128 configuration and include a privacy password for an SNMPv3 user, Cisco UCS Manager uses the privacy password to generate a 128-bit AES key. The AES privacy password can have a minimum of eight characters. If the passphrases are specified in clear text, you can specify a maximum of 64 characters.

Enabling SNMP and Configuring SNMP Properties

SNMP messages from a Cisco UCS domain display the fabric interconnect name rather than the system name.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope monitoring**	Enters monitoring mode.
Step 2	UCS-A /monitoring # **enable snmp**	Enables SNMP.
Step 3	UCS-A /monitoring # **set snmp community**	Enters snmp community mode.
Step 4	UCS-A /monitoring # **Enter a snmp community:** *community-name*	Specifies SNMP community. Use the community name as a password. The community name can be any alphanumeric string up to 32 characters.
Step 5	UCS-A /monitoring # **set snmp syscontact** *system-contact-name*	Specifies the system contact person responsible for the SNMP. The system contact name can be any alphanumeric string up to 255 characters, such as an email address or name and telephone number.
Step 6	UCS-A /monitoring # **set snmp syslocation** *system-location-name*	Specifies the location of the host on which the SNMP agent (server) runs. The system location name can be any alphanumeric string up to 512 characters.
Step 7	UCS-A /monitoring # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables SNMP, configures an SNMP community named SnmpCommSystem2, configures a system contact named contactperson, configures a contact location named systemlocation, and commits the transaction:

```
UCS-A# scope monitoring
UCS-A /monitoring # enable snmp
UCS-A /monitoring* # set snmp community
UCS-A /monitoring* # Enter a snmp community: SnmpCommSystem2
UCS-A /monitoring* # set snmp syscontact contactperson1
UCS-A /monitoring* # set snmp syslocation systemlocation
UCS-A /monitoring* # commit-buffer
UCS-A /monitoring #
```

What to Do Next

Create SNMP traps and users.

Creating an SNMP Trap

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope monitoring**	Enters monitoring mode.
Step 2	UCS-A /monitoring # **enable snmp**	Enables SNMP.

	Command or Action	Purpose
Step 3	UCS-A /monitoring # **create snmp-trap** {*hostname* | *ip-addr* | *ip6-addr*}	Creates an SNMP trap host with the specified host name, IPv4 address, or IPv6 address. The host name can be a fully qualified domain name of an IPv4 address.
Step 4	UCS-A /monitoring/snmp-trap # **set community** *community-name*	Specifies the SNMP community name to be used for the SNMP trap.
Step 5	UCS-A /monitoring/snmp-trap # **set port** *port-num*	Specifies the port to be used for the SNMP trap.
Step 6	UCS-A /monitoring/snmp-trap # **set version** {**v1** | **v2c** | **v3**}	Specifies the SNMP version and model used for the trap.
Step 7	UCS-A /monitoring/snmp-trap # **set notification type** {**traps** | **informs**}	(Optional) The type of trap to send. This can be: • **traps** if you select v2c or v3 for the version. • **informs** if you select v2c for the version. **Note** An inform notification can be send only if you select v2c for the version.
Step 8	UCS-A /monitoring/snmp-trap # **set v3 privilege** {**auth** | **noauth** | **priv**}	(Optional) If you select v3 for the version, the privilege associated with the trap. This can be: • **auth—**Authentication but no encryption • **noauth—**No authentication or encryption • **priv—**Authentication and encryption
Step 9	UCS-A /monitoring/snmp-trap # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables SNMP, creates an SNMP trap using an IPv4 address, specifies that the trap will use the SnmpCommSystem2 community on port 2, sets the version to v3, sets the notification type to traps, sets the v3 privilege to priv, and commits the transaction:

```
UCS-A# scope monitoring
UCS-A /monitoring # enable snmp
UCS-A /monitoring* # create snmp-trap 192.168.100.112
UCS-A /monitoring/snmp-trap* # set community SnmpCommSystem2
UCS-A /monitoring/snmp-trap* # set port 2
UCS-A /monitoring/snmp-trap* # set version v3
UCS-A /monitoring/snmp-trap* # set notificationtype traps
UCS-A /monitoring/snmp-trap* # set v3 privilege priv
UCS-A /monitoring/snmp-trap* # commit-buffer
UCS-A /monitoring/snmp-trap #
```

The following example enables SNMP, creates an SNMP trap using an IPv6 address, specifies that the trap will use the SnmpCommSystem3 community on port 2, sets the version to v3, sets the notification type to traps, sets the v3 privilege to priv, and commits the transaction:

```
UCS-A# scope monitoring
UCS-A /monitoring # enable snmp
UCS-A /monitoring* # create snmp-trap  2001::1
UCS-A /monitoring/snmp-trap* # set community SnmpCommSystem3
UCS-A /monitoring/snmp-trap* # set port 2
UCS-A /monitoring/snmp-trap* # set version v3
UCS-A /monitoring/snmp-trap* # set notificationtype traps
UCS-A /monitoring/snmp-trap* # set v3 privilege priv
UCS-A /monitoring/snmp-trap* # commit-buffer
UCS-A /monitoring/snmp-trap #
```

Deleting an SNMP Trap

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope monitoring**	Enters monitoring mode.
Step 2	UCS-A /monitoring # **delete snmp-trap** {*hostname* | *ip-addr*}	Deletes the specified SNMP trap host with the specified hostname or IP address.
Step 3	UCS-A /monitoring # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the SNMP trap at IP address 192.168.100.112 and commits the transaction:

```
UCS-A# scope monitoring
UCS-A /monitoring # delete snmp-trap 192.168.100.112
UCS-A /monitoring* # commit-buffer
UCS-A /monitoring #
```

Creating an SNMPv3 User

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope monitoring**	Enters monitoring mode.
Step 2	UCS-A /monitoring # **enable snmp**	Enables SNMP.
Step 3	UCS-A /monitoring # **create snmp-user** *user-name*	Creates the specified SNMPv3 user. An SNMP username cannot be the same as a local username. Choose an SNMP username that does not match a local username.

	Command or Action	Purpose
Step 4	UCS-A /monitoring/snmp-user # **set aes-128** {**no** | **yes**}	Enables or disables the use of AES-128 encryption.
Step 5	UCS-A /monitoring/snmp-user # **set auth** {**md5** | **sha**}	Specifies the use of MD5 or DHA authentication.
Step 6	UCS-A /monitoring/snmp-user # **set password**	Specifies the user password. After you enter the **set password** command, you are prompted to enter and confirm the password.
Step 7	UCS-A /monitoring/snmp-user # **set priv-password**	Specifies the user privacy password. After you enter the **set priv-password** command, you are prompted to enter and confirm the privacy password.
Step 8	UCS-A /monitoring/snmp-user # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables SNMP, creates an SNMPv3 user named snmp-user14, disables AES-128 encryption, specifies the use of MD5 authentication, sets the password and privacy password, and commits the transaction:

```
UCS-A# scope monitoring
UCS-A /monitoring # enable snmp
UCS-A /monitoring* # create snmp-user snmp-user14
UCS-A /monitoring/snmp-user* # set aes-128 no
UCS-A /monitoring/snmp-user* # set auth md5
UCS-A /monitoring/snmp-user* # set password
Enter a password:
Confirm the password:
UCS-A /monitoring/snmp-user* # set priv-password
Enter a password:
Confirm the password:
UCS-A /monitoring/snmp-user* # commit-buffer
UCS-A /monitoring/snmp-user #
```

Deleting an SNMPv3 User

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope monitoring**	Enters monitoring mode.
Step 2	UCS-A /monitoring # **delete snmp-user** *user-name*	Deletes the specified SNMPv3 user.
Step 3	UCS-A /monitoring # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the SNMPv3 user named snmp-user14 and commits the transaction:

```
UCS-A# scope monitoring
UCS-A /monitoring # delete snmp-user snmp-user14
UCS-A /monitoring* # commit-buffer
UCS-A /monitoring #
```

Enabling Telnet

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /services # **enable telnet-server**	Enables the Telnet service.
Step 4	UCS-A /services # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables Telnet and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /services # enable telnet-server
UCS-A /services* # commit-buffer
UCS-A /services #
```

Enabling the CIMC Web Service

To enable the CIMC Web Service:

- You must be logged in with admin privileges.
- The CIMC web service must be disabled, as it is enabled by default.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system** /	Enters the system mode.
Step 2	UCS-A /system #**scope services**/	Enters the services mode for the system.
Step 3	UCS-A/system/services #**enable cimcwebsvc**/	Enable the CIMC web service.
Step 4	UCS-A/system/services *# **commit-buffer**/	Commits the transaction to the system configuration.

The following example shows how to enable the CIMC web service and save the transaction:

```
UCS-A# scope system
UCS-A/system # scope services
UCS-A/system/services # enable cimcwebsvc
UCS-A/system/services *# commit-buffer
UCS-A/system/services # commit-buffer
UCS-A/system/services # show cimcwebsvc
Name: cimcwebservice
    Admin State: Enabled
```

Disabling the CIMC Web Service

To disable the CIMC Web Service:

- You must be logged in with admin privileges.
- The CIMC web service must be enabled.

Note The CIMC web service is enabled by default.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system** /	Enters the system mode.
Step 2	UCS-A /system #**scope services**/	Enters the services mode for the system.
Step 3	UCS-A/system/services #**disable cimcwebsvc**/	Disables the CIMC web service.
Step 4	UCS-A/system/services *# **commit-buffer**/	Commits the transaction to the system configuration.

The following example shows how to disable the CIMC web service and save the transaction:

```
UCS-A# scope system
UCS-A/system # scope services
UCS-A/system/services # disable cimcwebsvc
UCS-A/system/services *# commit-buffer
UCS-A/system/services # commit-buffer
UCS-A/system/services # show cimcwebsvc
Name: cimcwebservice
    Admin State: Disabled
```

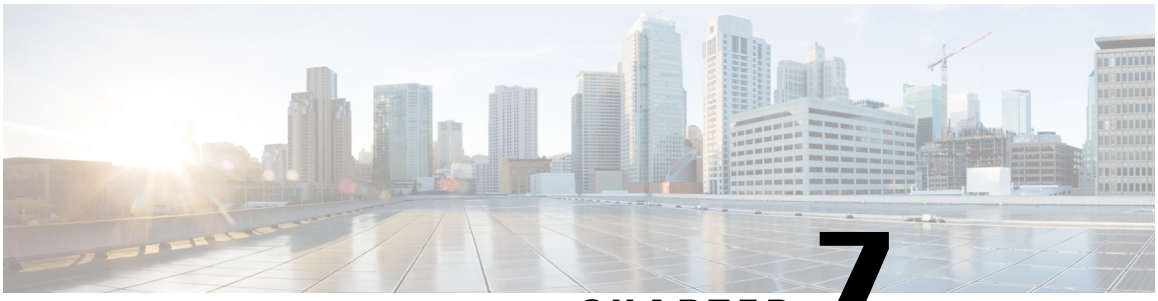
Disabling Communication Services

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **disable** *service-name*	Disables the specified service, where the *service-name* argument is one of the following keywords: • *cimxml* —Disables CIM XML service • **http** —Disables HTTP service • **https** —Disables HTTPS service • **telnet-server** —Disables Telnet service
Step 4	UCS-A /system/services # **commit-buffer**	Commits the transaction to the system configuration.

The following example disables CIM XML and commits the transaction:

```
UCS-A# scope system
UCS-A# scope services
UCS-A /system/services # disable cimxml
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```



CHAPTER 7

Configuring Authentication

This chapter includes the following sections:

Authentication Services

Cisco UCS supports the following two methods to authenticate user logins:

- Local user authentication - uses user accounts that exist locally in the Cisco UCS Manager
- Remote user authentication - uses one of the following protocols:
  - LDAP
  - RADIUS
  - TACACS+

Guidelines and Recommendations for Remote Authentication Providers

If a system is configured for one of the supported remote authentication services, you must create a provider for that service to ensure that Cisco UCS Manager can communicate with the system. The following guidelines impact user authorization:

User Accounts in Remote Authentication Services

User accounts can exist locally in Cisco UCS Manager or in the remote authentication server.

You can view the temporary sessions for users who log in through remote authentication services from the Cisco UCS Manager GUI and from the Cisco UCS Manager CLI.

User Roles in Remote Authentication Services

If you create user accounts in the remote authentication server, you must ensure that the accounts include the roles those users require for working in Cisco UCS Manager and that the names of those roles match the names used in Cisco UCS Manager. Based on the role policy, a user might not be allowed to log in, or is granted only read-only privileges.

User Attributes in Remote Authentication Providers

For RADIUS and TACACS+ configurations, you must configure a user attribute for Cisco UCS in each remote authentication provider through which users log in to Cisco UCS Manager. This user attribute holds the roles and locales assigned to each user.

Note This step is not required for LDAP configurations that use the LDAP Group Mapping to assign roles and locales.

When a user logs in, Cisco UCS Manager does the following:

1. Queries the remote authentication service.
2. Validates the user.
3. If the user is validated, checks for the roles and locales assigned to that user.

The following table contains a comparison of the user attribute requirements for the remote authentication providers supported by Cisco UCS.

Table 6: Comparison of User Attributes by Remote Authentication Provider

Authentication Provider	Custom Attribute	Schema Extension	Attribute ID Requirements
LDAP	Not required if group mapping is used Optional if group mapping is not used	Optional. You can choose to do one of the following: • Do not extend the LDAP schema and configure an existing, unused attribute that meets the requirements. • Extend the LDAP schema and create a custom attribute with a unique name, such as CiscoAVPair.	The Cisco LDAP implementation requires a unicode type attribute. If you choose to create the CiscoAVPair custom attribute, use the following attribute ID: 1.3.6.1.4.1.9.287247.1 A sample OID is provided in the following section.
RADIUS	Optional	Optional. You can choose to do one of the following: • Do not extend the RADIUS schema and use an existing unused attribute that meets the requirements. • Extend the RADIUS schema and create a custom attribute with a unique name, such as cisco-avpair.	The vendor ID for the Cisco RADIUS implementation is 009 and the vendor ID for the attribute is 001. The following syntax example shows how to specify multiples user roles and locales if you choose to create the cisco-avpair attribute: `shell:roles="admin,aaa" shell:locales="L1,abc"`. Use a comma "," as the delimiter to separate multiple values.
TACACS+	Required	Required. You must extend the schema and create a custom attribute with the name cisco-av-pair.	The cisco-av-pair name is the string that provides the attribute ID for the TACACS+ provider. The following syntax example shows how to specify multiples user roles and locales when you create the cisco-av-pair attribute: `cisco-av-pair=shell:roles="admin aaa" shell:locales*"L1 abc"`. Using an asterisk (*) in the cisco-av-pair attribute syntax flags the locale as optional, preventing authentication failures for other Cisco devices that use the same authorization profile. Use a space as the delimiter to separate multiple values.

Sample OID for LDAP User Attribute

The following is a sample OID for a custom CiscoAVPair attribute:

```
CN=CiscoAVPair,CN=Schema,
CN=Configuration,CN=X
objectClass: top
objectClass: attributeSchema
cn: CiscoAVPair
distinguishedName: CN=CiscoAVPair,CN=Schema,CN=Configuration,CN=X
instanceType: 0x4
uSNCreated: 26318654
attributeID: 1.3.6.1.4.1.9.287247.1
attributeSyntax: 2.5.5.12
isSingleValued: TRUE
showInAdvancedViewOnly: TRUE
adminDisplayName: CiscoAVPair
adminDescription: UCS User Authorization Field
oMSyntax: 64
lDAPDisplayName: CiscoAVPair
name: CiscoAVPair
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,CN=X
```

Two-Factor Authentication

Cisco UCS Manager uses two-factor authentication for remote user logins, which adds a level of security to account logins. Two-factor authentication login requires a username, a token, and a password combination in the password field. You can provide a PIN, a certificate, or a token.

Two-factor authentication uses authentication applications that maintain token servers to generate one-time tokens for users during the login process and store passwords in the AAA server. Requests are sent to the token server to retrieve a vendor-specific attribute. Cisco UCS Manager expects the token server to integrate with the AAA server, therefore it forwards the request to the AAA server. The password and token are validated at the same time by the AAA server. Users must enter the token and password sequence in the same order as it is configured in the AAA server.

Two-factor authentication is supported by associating RADIUS or TACACS+ provider groups with designated authentication domains and enabling two-factor authentication for those domains. Two-factor authentication does not support IPM and is not supported when the authentication realm is set to LDAP, local, or none.

Web Session Refresh and Web Session Timeout Period

The **Web Session Refresh Period** is the maximum amount of time allowed between refresh requests for a Cisco UCS Manager GUI web session. The **Web Session Timeout** is the maximum amount of time that can elapse after the last refresh request before a Cisco UCS Manager GUI web session becomes inactive.

You can increase the **Web Session Refresh Period** to a value greater than 60 seconds up 172800 seconds to avoid frequent session timeouts that requires regenerating and re-entering a token and password multiple times. The default value is 7200 seconds when two-factor authentication is enabled, and is 600 seconds when two-factor authentication is not enabled.

You can specify a value between 300 and 172800 for the **Web Session Timeout Period**. The default is 8000 seconds when two-factor authentication is enabled, and 7200 seconds when two-factor authentication is not enabled.

LDAP Group Rule

The LDAP group rule determines whether Cisco UCS should use LDAP groups when assigning user roles and locales to a remote user.

Nested LDAP Groups

You can add an LDAP group as a member of another group and nest groups to consolidate member accounts and to reduce the replication of traffic. Cisco UCS Manager release 2.1(2) and higher enables you to search LDAP groups that are nested within another group defined in an LDAP group map.

Note Nested LDAP search support is supported only for Microsoft Active Directory servers. The supported versions are Microsoft Windows 2003 SP3, Microsoft Windows 2008 R2, and Microsoft Windows 2012.

By default, user rights are inherited when you nest an LDAP group within another group. For example, if you make Group_1 a member of Group_2, the users in Group_1 have the same permissions as the members of Group_2. You can then search users that are members of Group_1 by choosing only Group_2 in the LDAP group map, instead of having to search Group_1 and Group_2 separately.

You do not always need to create subgroups in a group map in Cisco UCS Manager.

Configuring LDAP Providers

Configuring Properties for LDAP Providers

The properties that you configure in this task are the default settings for all provider connections of this type defined in Cisco UCS Manager. If an individual provider includes a setting for any of these properties, Cisco UCS uses that setting and ignores the default setting.

If you are using Active Directory as your LDAP server, create a user account in the Active Directory server to bind with Cisco UCS. Give this account a non-expiring password.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope ldap**	Enters security LDAP mode.
Step 3	UCS-A /security/ldap # **set attribute** *attribute*	Restricts database searches to records that contain the specified attribute.
Step 4	UCS-A /security/ldap # **set basedn** *distinguished-name*	Restricts database searches to records that contain the specified distinguished name.

	Command or Action	Purpose
Step 5	UCS-A /security/ldap # **set filter** *filter*	Restricts database searches to records that contain the specified filter.
Step 6	UCS-A /security/ldap # **set timeout** *seconds*	(Optional) Sets the time interval the system waits for a response from the LDAP server before noting the server as down.
Step 7	UCS-A /security/ldap # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the LDAP attribute to CiscoAvPair, the base distinguished name to "DC=cisco-ucsm-aaa3,DC=qalab,DC=com", the filter to sAMAccountName=$userid, and the timeout interval to 5 seconds, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope ldap
UCS-A /security/ldap # set attribute CiscoAvPair
UCS-A /security/ldap* # set basedn "DC=cisco-ucsm-aaa3,DC=qalab,DC=com"
UCS-A /security/ldap* # set filter sAMAccountName=$userid
UCS-A /security/ldap* # set timeout 5
UCS-A /security/ldap* # commit-buffer
UCS-A /security/ldap #
```

Note User login will fail if the userdn for an LDAP user exceeds 255 characters.

What to Do Next

Create an LDAP provider.

Creating an LDAP Provider

Cisco UCS Manager supports a maximum of 16 LDAP providers.

Before You Begin

If you are using Active Directory as your LDAP server, create a user account in the Active Directory server to bind with Cisco UCS. Give this account a non-expiring password.

- In the LDAP server, perform one of the following configurations:
  - Configure LDAP groups. LDAP groups contain user role and locale information.
  - Configure users with the attribute that holds the user role and locale information for Cisco UCS Manager. You can choose whether to extend the LDAP schema for this attribute. If you do not want to extend the schema, use an existing LDAP attribute to hold the Cisco UCS user roles and locales. If you prefer to extend the schema, create a custom attribute, such as the CiscoAVPair attribute.

    The Cisco LDAP implementation requires a unicode type attribute.

If you choose to create the CiscoAVPair custom attribute, use the following attribute ID: 1.3.6.1.4.1.9.287247.1

- For a cluster configuration, add the management port IPv4 or IPv6 addresses for both fabric interconnects. This configuration ensures that remote users can continue to log in if the first fabric interconnect fails and the system fails over to the second fabric interconnect. All login requests are sourced from these IP addresses, not the virtual IPv4 or IPv6 address used by Cisco UCS Manager.

- If you want to use secure communications, create a trusted point containing the certificate of the root certificate authority (CA) of the LDAP server in Cisco UCS Manager.
- If you need to change the LDAP providers or add or delete them, you need to change the authentication realm for the domain to local, make the changes to the providers, and then change the domain authentication realm back to LDAP.
- If you want to use the special characters listed in the following table for defining the attributes of an Active Directory bind distinguished name, you must replace the special character with an escape, by using a backslash (\) followed by the corresponding hexadecimal value of the character.

Special Character	Description	Hexadecimal Value
,	comma	0x2C
+	plus sign	0x2B
"	double quote	0x22
\	backslash	0x5C
<	left angle bracket	0x3C
>	right angle bracket	0x3E
;	semicolon	0x3B
LF	line feed	0x0A
CR	carriage return	0x0D
=	equals sign	0x3D
/	forwards slash	0x2F

https://msdn.microsoft.com/en-us/library/aa366101 provides more details on replacing special characters with its escape and hexadecimal equivalent.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope ldap**	Enters security LDAP mode.
Step 3	UCS-A /security/ldap # **create server** *server-name*	Creates an LDAP server instance and enters security LDAP server mode. If SSL is enabled, the *server-name* , typically an IP address or FQDN, must exactly match a Common Name (CN) in the LDAP server's security certificate. Unless an IP address is specified, a DNS server must be configured in Cisco UCS Manager.
Step 4	UCS-A /security/ldap/server # **set attribute** *attr-name*	(Optional) An LDAP attribute that stores the values for the user roles and locales. This property is always a name-value pair. The system queries the user record for the value that matches this attribute name. If you do not want to extend your LDAP schema, you can configure an existing, unused LDAP attribute with the Cisco UCS roles and locales. Alternatively, you can create an attribute named CiscoAVPair in the remote authentication service with the following attribute ID: 1.3.6.1.4.1.9.287247.1 This value is required unless a default attribute has been set on the LDAP **General** tab.
Step 5	UCS-A /security/ldap/server # **set basedn** *basedn-name*	(Optional) The specific distinguished name in the LDAP hierarchy where the server begins a search when a remote user logs in and the system attempts to obtain the user's DN based on their username. You can set the length of the base DN to a maximum of 255 characters minus the length of CN=username, where username identifies the remote user attempting to access Cisco UCS Manager using LDAP authentication. This value is required unless a default base DN has been set on the LDAP **General** tab.
Step 6	UCS-A /security/ldap/server # **set binddn** *binddn-name*	(Optional) The distinguished name (DN) for an LDAP database account that has read and search permissions for all objects under the base DN. The maximum supported string length is 255 ASCII characters.
Step 7	UCS-A /security/ldap/server # **set filter** *filter-value*	(Optional) The LDAP search is restricted to those user names that match the defined filter. This value is required unless a default filter has been set on the LDAP **General** tab.

	Command or Action	Purpose
Step 8	UCS-A /security/ldap/server # **set password**	The password for the LDAP database account specified in the **Bind DN** field. You can enter any standard ASCII characters except for space, § (section sign), ? (question mark), or = (equal sign). To set the password, press **Enter** after typing the **set password** command and enter the key value at the prompt.
Step 9	UCS-A /security/ldap/server # **set order** *order-num*	(Optional) The order that the Cisco UCS uses this provider to authenticate users.
Step 10	UCS-A /security/ldap/server # **set port** *port-num*	(Optional) The port through which Cisco UCS communicates with the LDAP database. The standard port number is 389.
Step 11	UCS-A /security/ldap/server # **set ssl** {**yes no**}	Enables or disables the use of encryption when communicating with the LDAP server. The options are as follows: • **yes —**Encryption is required. If encryption cannot be negotiated, the connection fails. • **no —**Encryption is disabled. Authentication information is sent as clear text. LDAP uses STARTTLS. This allows encrypted communication using port 389. If encryption is enabled, do not change the port to 636, leave it as 389. Cisco UCS negotiates a TLS session on port 636 for SSL, but initial connection starts unencrypted on 389.
Step 12	UCS-A /security/ldap/server # **set timeout** *timeout-num*	The length of time in seconds the system spends trying to contact the LDAP database before it times out. Enter an integer from 1 to 60 seconds, or enter 0 (zero) to use the global timeout value specified on the LDAP **General** tab. The default is 30 seconds.
Step 13	UCS-A /security/ldap/server # **set vendor** *{ms-ad | openldap}*	Enables or disables the use of the nested LDAP group search capability on the LDAP server. The options are as follows: • **ms-ad—**Nested LDAP group searches are supported with this option. If you set the vendor to *ms-ad* (Microsoft Active Directory), and enable and set the *ldap-group-rule* to recursive, Cisco UCS Manager can search through any nested LDAP groups. • **openldap—**Nested LDAP group searches are not supported with this option. If you set the vendor to *openldap*, and enable and set the *ldap-group-rule* to recursive, Cisco UCS Managerwill not search through any nested LDAP groups. If you choose this option, you must create each LDAP subgroup as an LDAP group map in Cisco UCS Manager, even if the parent group is already set up in a group map.

	Command or Action	Purpose
		Note When you upgrade Cisco UCS Manager from an earlier version to release 2.1(2), the LDAP provider's vendor attribute is set to **openldap** by default, and LDAP authentication continues to operate successfully.
Step 14	UCS-A /security/ldap/server # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates an LDAP server instance named 10.193.169.246, configures the binddn, password, order, port, SSL settings, vendor attribute, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope ldap
UCS-A /security/ldap* # create server 10.193.169.246
UCS-A /security/ldap/server* # set binddn
"cn=Administrator,cn=Users,DC=cisco-ucsm-aaa3,DC=qalab,DC=com"
UCS-A /security/ldap/server* # set password
Enter the password:
Confirm the password:
UCS-A /security/ldap/server* # set order 2
UCS-A /security/ldap/server* # set port 389
UCS-A /security/ldap/server* # set ssl yes
UCS-A /security/ldap/server* # set timeout 30
UCS-A /security/ldap/server* # set vendor ms-ad
UCS-A /security/ldap/server* # commit-buffer
UCS-A /security/ldap/server #
```

The following example creates an LDAP server instance named 12:31:71:1231:45b1:0011:011:900, configures the binddn, password, order, port, SSL settings, vendor attribute, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope ldap
UCS-A /security/ldap* # create server 12:31:71:1231:45b1:0011:011:900
UCS-A /security/ldap/server* # set binddn
"cn=Administrator,cn=Users,DC=cisco-ucsm-aaa3,DC=qalab,DC=com"
UCS-A /security/ldap/server* # set password
Enter the password:
Confirm the password:
UCS-A /security/ldap/server* # set order 1
UCS-A /security/ldap/server* # set port 389
UCS-A /security/ldap/server* # set ssl yes
UCS-A /security/ldap/server* # set timeout 45
UCS-A /security/ldap/server* # set vendor ms-ad
UCS-A /security/ldap/server* # commit-buffer
UCS-A /security/ldap/server #
```

What to Do Next

For implementations involving a single LDAP database, select LDAP as the authentication service.

For implementations involving multiple LDAP databases, configure an LDAP provider group.

Changing the LDAP Group Rule for an LDAP Provider

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope ldap**	Enters security LDAP mode.
Step 3	UCS-A /security/ldap # **scope server** *ldap-provider*	Enters security LDAP provider mode.
Step 4	UCS-A /security/ldap/server # **scope ldap-group-rule**	Enters LDAP group rule mode.
Step 5	UCS-A /security/ldap/server/ldap-group-rule # **set authorization** {**enable** | **disable**}	Specifies whether Cisco UCS searches LDAP groups when assigning user roles and locales to a remote user. • **disable—**Cisco UCS does not access any LDAP groups. • **enable—**Cisco UCS searches the LDAP provider groups mapped in this Cisco UCS domain. If the remote user is found, Cisco UCS assigns the user roles and locales defined for that LDAP group in the associated LDAP group map. **Note** Role and locale assignment is cumulative. If a user is included in multiple groups, or has a role or locale specified in the LDAP attribute, Cisco UCS assigns that user all the roles and locales mapped to any of those groups or attributes.
Step 6	UCS-A /security/ldap/server/ldap-group-rule # **set member-of-attribute** *attr-name*	The attribute Cisco UCS uses to determine group membership in the LDAP database. The supported string length is 63 characters. The default string is memberOf.
Step 7	UCS-A /security/ldap/server/ldap-group-rule # **set traversal** {**non-recursive** | **recursive**}	Specifies whether Cisco UCS takes the settings for a group member's parent group, if necessary. This can be: • **non-recursive—**Cisco UCS only searches those groups that the user belongs to. • **recursive—**Cisco UCS searches all the ancestor groups belonging to the user.
Step 8	UCS-A /security/ldap/server/ldap-group-rule # **set use-primary-group** {**yes** | **no**}	Configures the primary group as an LDAP group map in Cisco UCS domain for membership validation. You can enable Cisco UCS Manager to download and verify the user primary group membership.

	Command or Action	Purpose
Step 9	UCS-A /security/ldap/server/ldap-group-rule # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the LDAP group rule to enable authorization, sets the member of attribute to memberOf, sets the traversal to non-recursive, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope ldap
UCS-A /security/ldap # scope server ldapprovider
UCS-A /security/ldap/server # scope ldap-group-rule
UCS-A /security/ldap/server/ldap-group-rule # set authorization enable
UCS-A /security/ldap/server/ldap-group-rule* # set member-of-attribute memberOf
UCS-A /security/ldap/server/ldap-group-rule* # set traversal non-recursive
UCS-A /security/ldap/server/ldap-group-rule* # set use-primary-group yes
UCS-A /security/ldap/server/ldap-group-rule* # commit-buffer
UCS-A /security/ldap/server/ldap-group-rule #
```

Deleting an LDAP Provider

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode
Step 2	UCS-A /security # **scope ldap**	Enters security LDAP mode
Step 3	UCS-A /security/ldap # **delete server** *serv-name*	Deletes the specified server.
Step 4	UCS-A /security/ldap # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the LDAP server called ldap1 and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope ldap
UCS-A /security/ldap # delete server ldap1
UCS-A /security/ldap* # commit-buffer
UCS-A /security/ldap #
```

LDAP Group Mapping

LDAP group mapping eliminates having to define role or locale information in the LDAP user object. UCSM can use group membership information to assign a role or locale to an LDAP user during login for organizations using LDAP groups to restrict access to LDAP databases.

When a user logs in to Cisco UCS Manager, the LDAP group map pulls information about the user's role and locale. If the role and locale criteria match the information in the policy, access is granted. Cisco UCS Manager supports a maximum of 28, 128, or 160 LDAP group maps depending on the release version.

Note Cisco UCS Manager Release 3.1(1) supports a maximum of 128 LDAP group maps, and Release 3.1(2) and later releases support a maximum of 160 LDAP group maps.

The role and locale definitions that you configure locally in the Cisco UCS Manager do not update automatically based on changes to an LDAP directory. When deleting or renaming LDAP groups in an LDAP directory, you must also update the Cisco UCS Manager with the change.

You can configure an LDAP group map to include any of the following combinations of roles and locales:

- Roles only
- Locales only
- Both roles and locales

For example, consider an LDAP group representing a group of server administrators at a specific location. The LDAP group map might include user roles such as server profile and server equipment. To restrict access to server administrators at a specific location, you can set the locale to a particular site name.

Note Cisco UCS Manager includes out-of-the-box user roles, but does not include any locales. Mapping an LDAP provider group to a locale requires that you create a custom locale.

Creating an LDAP Group Map

Before You Begin

- Create an LDAP group in the LDAP server.
- Configure the distinguished name for the LDAP group in the LDAP server.
- Create locales in Cisco UCS Manager (optional).
- Create custom roles in Cisco UCS Manager (optional).

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope ldap**	Enters security LDAP mode.
Step 3	UCS-A /security/ldap # **create ldap-group** *group-dn*	Creates an LDAP group map for the specified DN. The maximum number of characters for group-dn is 240.

	Command or Action	Purpose
		Note If you plan to enter a special character for this command, you need to prefix the special character with an escape character \\ (double back slash).
Step 4	UCS-A /security/ldap/ldap-group # **create locale** *locale-name*	Maps the LDAP group to the specified locale.
Step 5	UCS-A /security/ldap/ldap-group # **create role** *role-name*	Maps the LDAP group to the specified role.
Step 6	UCS-A /security/ldap/ldap-group # **commit-buffer**	Commits the transaction to the system configuration.

The following example maps the LDAP group mapped to a DN, sets the locale to pacific, sets the role to admin, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope ldap
UCS-A /security/ldap # create ldap-group cn=security,cn=users,dc=lab,dc=com
UCS-A /security/ldap/ldap-group* # create locale pacific
UCS-A /security/ldap/ldap-group* # create role admin
UCS-A /security/ldap/ldap-group* # commit-buffer
UCS-A /security/ldap/ldap-group #
```

What to Do Next

Set the LDAP group rule.

Deleting an LDAP Group Map

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope ldap**	Enters security LDAP mode.
Step 3	UCS-A /security/ldap # **delete ldap-group** *group-dn*	Deletes the LDAP group map for the specified DN.
Step 4	UCS-A /security/ldap # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes an LDAP group map and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope ldap
UCS-A /security/ldap # delete ldap-group cn=security,cn=users,dc=lab,dc=com
UCS-A /security/ldap* # commit-buffer
UCS-A /security/ldap #
```

Configuring RADIUS Providers

Configuring Properties for RADIUS Providers

The properties that you configure in this task are the default settings for all provider connections of this type defined in Cisco UCS Manager. If an individual provider includes a setting for any of these properties, Cisco UCS uses that setting and ignores the default setting.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope radius**	Enters security RADIUS mode.
Step 3	UCS-A /security/radius # **set retries** *retry-num*	(Optional) Sets the number of times to retry communicating with the RADIUS server before noting the server as down.
Step 4	UCS-A /security/radius # **set timeout** *seconds*	(Optional) Sets the time interval that the system waits for a response from the RADIUS server before noting the server as down.
Step 5	UCS-A /security/radius # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the RADIUS retries to 4, sets the timeout interval to 30 seconds, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope radius
UCS-A /security/radius # set retries 4
UCS-A /security/radius* # set timeout 30
UCS-A /security/radius* # commit-buffer
UCS-A /security/radius #
```

What to Do Next

Create a RADIUS provider.

Creating a RADIUS Provider

Cisco UCS Manager supports a maximum of 16 RADIUS providers.

Before You Begin

Perform the following configuration in the RADIUS server:

- Configure users with the attribute that holds the user role and locale information for Cisco UCS Manager. You can choose whether to extend the RADIUS schema for this attribute. If you do not want to extend the schema, use an existing RADIUS attribute to hold the Cisco UCS user roles and locales. If you prefer to extend the schema, create a custom attribute, such as the cisco-avpair attribute.

  The vendor ID for the Cisco RADIUS implementation is 009 and the vendor ID for the attribute is 001.

  The following syntax example shows how to specify multiples user roles and locales if you choose to create the cisco-avpair attribute: `shell:roles="admin,aaa" shell:locales="L1,abc"`. Use a comma "," as the delimiter to separate multiple values.

- For a cluster configuration, add the management port IPv4 or IPv6 addresses for both fabric interconnects. This configuration ensures that remote users can continue to log in if the first fabric interconnect fails and the system fails over to the second fabric interconnect. All login requests are sourced from these IP addresses, not the virtual IP address used by Cisco UCS Manager.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope radius**	Enters security RADIUS mode.
Step 3	UCS-A /security/radius # **create server** *server-name*	Creates a RADIUS server instance and enters security RADIUS server mode
Step 4	UCS-A /security/radius/server # **set authport** *authport-num*	(Optional) Specifies the port used to communicate with the RADIUS server.
Step 5	UCS-A /security/radius/server # **set key**	Sets the RADIUS server key. To set the key value, press **Enter** after typing the **set key** command and enter the key value at the prompt.
Step 6	UCS-A /security/radius/server # **set order** *order-num*	(Optional) Specifies when in the order this server will be tried.
Step 7	UCS-A /security/radius/server # **set retries** *retry-num*	(Optional) Sets the number of times to retry communicating with the RADIUS server before noting the server as down.
Step 8	UCS-A /security/radius/server # **set timeout** *seconds*	(Optional) Sets the time interval that the system waits for a response from the RADIUS server before noting the server as down. **Tip** It is recommended that you configure a higher **Timeout** value if you select two-factor authentication for RADIUS providers.
Step 9	UCS-A /security/radius/server # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a server instance named radiusserv7, sets the authentication port to 5858, sets the key to radiuskey321, sets the order to 2, sets the retries to 4, sets the timeout to 30, enables two-factor authentication, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope radius
UCS-A /security/radius # create server radiusserv7
UCS-A /security/radius/server* # set authport 5858
UCS-A /security/radius/server* # set key
Enter the key: radiuskey321
Confirm the key: radiuskey321
UCS-A /security/radius/server* # set order 2
UCS-A /security/radius/server* # set retries 4
UCS-A /security/radius/server* # set timeout 30
UCS-A /security/radius/server* # commit-buffer
UCS-A /security/radius/server #
```

What to Do Next

For implementations involving a single RADIUS database, select RADIUS as the primary authentication service.

For implementations involving multiple RADIUS databases, configure a RADIUS provider group.

Deleting a RADIUS Provider

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope RADIUS**	Enters security RADIUS mode.
Step 3	UCS-A /security/radius # **delete server** *serv-name*	Deletes the specified server.
Step 4	UCS-A /security/radius # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the RADIUS server called radius1 and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope radius
UCS-A /security/radius # delete server radius1
UCS-A /security/radius* # commit-buffer
UCS-A /security/radius #
```

Configuring TACACS+ Providers

Configuring Properties for TACACS+ Providers

The properties that you configure in this task are the default settings for all provider connections of this type defined in Cisco UCS Manager. If an individual provider includes a setting for any of these properties, Cisco UCS uses that setting and ignores the default setting.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope tacacs**	Enters security TACACS+ mode.
Step 3	UCS-A /security/tacacs # **set timeout** *seconds*	(Optional) Sets the time interval that the system waits for a response from the TACACS+ server before noting the server as down.
Step 4	UCS-A /security/tacacs # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the TACACS+ timeout interval to 45 seconds and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope tacacs
UCS-A /security/tacacs # set timeout 45
UCS-A /security/tacacs* # commit-buffer
UCS-A /security/tacacs #
```

What to Do Next

Create a TACACS+ provider.

Creating a TACACS+ Provider

Cisco UCS Manager supports a maximum of 16 TACACS+ providers.

Before You Begin

Perform the following configuration in the TACACS+ server:

- Create the cisco-av-pair attribute. You cannot use an existing TACACS+ attribute.

  The cisco-av-pair name is the string that provides the attribute ID for the TACACS+ provider.

  The following syntax example shows how to specify multiples user roles and locales when you create the cisco-av-pair attribute: `cisco-av-pair=shell:roles="admin aaa" shell:locales*"L1 abc"`. Using an asterisk (*) in the cisco-av-pair attribute syntax flags the locale as optional, preventing

authentication failures for other Cisco devices that use the same authorization profile. Use a space as the delimiter to separate multiple values.

- For a cluster configuration, add the management port IPv4 or IPv6 addresses for both fabric interconnects. This configuration ensures that remote users can continue to log in if the first fabric interconnect fails and the system fails over to the second fabric interconnect. All login requests are sourced from these IP addresses, not the virtual IP address used by Cisco UCS Manager.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope tacacs**	Enters security TACACS+ mode.
Step 3	UCS-A /security/tacacs # **create server** *server-name*	Creates an TACACS+ server instance and enters security TACACS+ server mode
Step 4	UCS-A /security/tacacs/server # **set key**	(Optional) Sets the TACACS+ server key. To set the key value, press **Enter** after typing the **set key** command and enter the key value at the prompt.
Step 5	UCS-A /security/tacacs/server # **set order** *order-num*	(Optional) Specifies when in the order this server will be tried.
Step 6	UCS-A /security/tacacs/server # **set timeout***seconds*	(Optional) Sets the time interval that the system waits for a response from the TACACS+ server before noting the server as down. **Tip** It is recommended that you configure a higher timeout value if you select two-factor authentication for TACACS+ providers.
Step 7	UCS-A /security/tacacs/server # **set port** *port-num*	Specifies the port used to communicate with the TACACS+ server.
Step 8	UCS-A /security/tacacs/server # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a server instance named tacacsserv680, sets the key to tacacskey321 and confirms the key, sets the order to 4, sets the authentication port to 5859, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope tacacs
UCS-A /security/tacacs # create server tacacsserv680
UCS-A /security/tacacs/server* # set key
Enter the key: tacacskey321
Confirm the key: tacacskey321
UCS-A /security/tacacs/server* # set order 4
UCS-A /security/tacacs/server* # set port 5859
UCS-A /security/tacacs/server* # commit-buffer
UCS-A /security/tacacs/server #
```

What to Do Next

For implementations involving a single TACACS+ database, select TACACS+ as the primary authentication service.

For implementations involving multiple TACACS+ databases, configure a TACACS+ provider group.

Deleting a TACACS+ Provider

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope tacacs**	Enters security TACACS mode.
Step 3	UCS-A /security/tacacs # **delete server** *serv-name*	Deletes the specified server.
Step 4	UCS-A /security/tacacs # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the TACACS server called tacacs1 and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope tacacs
UCS-A /security/tacacs # delete server TACACS1
UCS-A /security/tacacs* # commit-buffer
UCS-A /security/tacacs #
```

Configuring Multiple Authentication Systems

Multiple Authentication Services

You can configure Cisco UCS to use multiple authentication services by configuring the following features:

- Provider groups
- Authentication domains

After provider groups and authentication domains are configured in Cisco UCS Manager, you can use the following syntax to log in to the system using Cisco UCS Manager CLI: **ucs:** *auth-domain* \ *user-name* .

When multiple authentication domains and native authentication are configured with a remote authentication service, use one of the following syntax examples to log in with SSH, Telnet or Putty.

Note SSH log in is case-sensitive.

From a Linux terminal using SSH:

- **ssh ucs-***auth-domain**username*@{*UCSM-ip-address*|*UCMS-ipv6-address*}

```
ssh ucs-example\\jsmith@192.0.20.11
ssh ucs-example\\jsmith@2001::1
```

- **ssh -l ucs-***auth-domain**username* {*UCSM-ip-address*| *UCSM-ipv6-address*| *UCSM-host-name*}

```
ssh -l ucs-example\\jsmith 192.0.20.11
ssh -l ucs-example\\jsmith 2001::1
```

- **ssh** {*UCSM-ip-address* | *UCSM-ipv6-address* | *UCSM-host-name*} **-l ucs-***auth-domain**username*

```
ssh 192.0.20.11 -l ucs-example\\jsmith
ssh 2001::1 -l ucs-example\\jsmith
```

- **ssh ucs-***auth-domain**username*@{*UCSM-ip-address*|*UCSM-ipv6-address*}

```
ssh ucs-ldap23\\jsmith@192.0.20.11
ssh ucs-ldap23\\jsmith@2001::1
```

From a Linux terminal using Telnet:

- **telnet ucs-***UCSM-host-name* **ucs-***auth-domain**username*

```
telnet ucs-qa-10
login: ucs-ldap23\blradmin
```

- **telnet ucs-**{*UCSM-ip-address*|*UCSM-ipv6-address*}*ucs-auth-domain**username*

```
telnet 10.106.19.12 2052
ucs-qa-10-A login: ucs-ldap23\blradmin
```

From a Putty client:

- Login as: **ucs-***auth-domain**username*

```
Login as: ucs-example\jsmith
```

Note If the default authentication is set to local, and the console authentication is set to LDAP, you can log in to the fabric interconnect from a Putty client using ucs-local\admin, where admin is the name of the local account.

Configuring Multiple Authentication Systems

Provider Groups

A provider group is a set of providers that the Cisco UCS accesses during the authentication process. All of the providers within a provider group are accessed in the order that the Cisco UCS provider uses to authenticate users. If all of the configured servers are unavailable or unreachable, Cisco UCS Manager automatically falls back to the local authentication method using the local username and password.

Cisco UCS Manager allows you to create a maximum of 16 provider groups, with a maximum of eight providers allowed per group.

Creating an LDAP Provider Group

Creating an LDAP provider group allows you to authenticate using multiple LDAP databases.

Before You Begin

Create one or more LDAP providers.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope ldap**	Enters security LDAP mode.
Step 3	UCS-A /security/ldap # **create auth-server-group** *auth-server-group-name*	Creates an LDAP provider group and enters authentication server group security LDAP mode.
Step 4	UCS-A /security/ldap/auth-server-group # **create server-ref** *ldap-provider-name*	Adds the specified LDAP provider to the LDAP provider group and enters server reference authentication server group security LDAP mode.
Step 5	UCS-A /security/ldap/auth-server-group/server-ref # **set order** *order-num*	Specifies the order in which Cisco UCS uses this provider to authenticate users. Valid values include no-value and 0-16, with the lowest value indicating the highest priority. Setting the order to no-value is equivalent to giving that server reference the highest priority.
Step 6	UCS-A /security/ldap/auth-server-group/server-ref # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates an LDAP provider group called ldapgroup, adds two previously configured providers called ldap1 and ldap2 to the provider group, sets the order, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope ldap
UCS-A /security/ldap # create auth-server-group ldapgroup
UCS-A /security/ldap/auth-server-group* # create server-ref ldap1
UCS-A /security/ldap/auth-server-group/server-ref* # set order 1
UCS-A /security/ldap/auth-server-group/server-ref* # up
UCS-A /security/ldap/auth-server-group* # create server-ref ldap2
UCS-A /security/ldap/auth-server-group/server-ref* # set order 2
UCS-A /security/ldap/auth-server-group/server-ref* # commit-buffer
UCS-A /security/ldap/auth-server-group/server-ref #
```

What to Do Next

Configure an authentication domain or select a default authentication service.

Deleting an LDAP Provider Group

Before You Begin

Remove the provider group from an authentication configuration.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope ldap**	Enters security LDAP mode.
Step 3	UCS-A /security/ldap # **delete auth-server-group** *auth-server-group-name*	Deletes the LDAP provider group.
Step 4	UCS-A /security/ldap # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes an LDAP provider group called ldapgroup and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope ldap
UCS-A /security/ldap # delete auth-server-group ldapgroup
UCS-A /security/ldap* # commit-buffer
UCS-A /security/ldap #
```

Creating a RADIUS Provider Group

Creating a RADIUS provider group allows you to authenticate using multiple RADIUS databases.

Before You Begin

Create one or more RADIUS providers.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope radius**	Enters security RADIUS mode.
Step 3	UCS-A /security/radius # **create auth-server-group** *auth-server-group-name*	Creates a RADIUS provider group and enters authentication server group security RADIUS mode.
Step 4	UCS-A /security/RADIUS/auth-server-group # **create server-ref** *radius-provider-name*	Adds the specified RADIUS provider to the RADIUS provider group and enters server reference authentication server group security RADIUS mode.

	Command or Action	Purpose
Step 5	UCS-A /security/radius/auth-server-group/server-ref # **set order** *order-num*	Specifies the order in which Cisco UCS uses this provider to authenticate users. Valid values include no-value and 0-16, with the lowest value indicating the highest priority. Setting the order to no-value is equivalent to giving that server reference the highest priority.
Step 6	UCS-A /security/radius/auth-server-group/server-ref # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a RADIUS provider group called radiusgroup, adds two previously configured providers called radius1 and radius2 to the provider group, sets the order, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope radius
UCS-A /security/radius # create auth-server-group radiusgroup
UCS-A /security/radius/auth-server-group* # create server-ref radius1
UCS-A /security/radius/auth-server-group/server-ref* # set order 1
UCS-A /security/radius/auth-server-group/server-ref* # up
UCS-A /security/radius/auth-server-group* # create server-ref radius2
UCS-A /security/radius/auth-server-group/server-ref* # set order 2
UCS-A /security/radius/auth-server-group/server-ref* # commit-buffer
UCS-A /security/radius/auth-server-group/server-ref #
```

What to Do Next

Configure an authentication domain or select a default authentication service.

Deleting a RADIUS Provider Group

Remove the provider group from an authentication configuration.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope radius**	Enters security RADIUS mode.
Step 3	UCS-A /security/radius # **delete auth-server-group** *auth-server-group-name*	Deletes the RADIUS provider group.
Step 4	UCS-A /security/radius # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a RADIUS provider group called radiusgroup and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope radius
UCS-A /security/radius # delete auth-server-group radiusgroup
UCS-A /security/radius* # commit-buffer
UCS-A /security/radius #
```

Creating a TACACS Provider Group

Creating a TACACS+ provider group allows you to authenticate using multiple TACACS+ databases.

Before You Begin

Create a TACACS provider.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope tacacs**	Enters security TACACS mode.
Step 3	UCS-A /security/tacacs # **create auth-server-group** *auth-server-group-name*	Creates a TACACS provider group and enters authentication server group security TACACS mode.
Step 4	UCS-A /security/tacacs/auth-server-group # **create server-ref** *tacacs-provider-name*	Adds the specified TACACS provider to the TACACS provider group and enters server reference authentication server group security TACACS mode.
Step 5	UCS-A /security/tacacs/auth-server-group/server-ref # **set order** *order-num*	Specifies the order in which Cisco UCS uses this provider to authenticate users. Valid values include no-value and 0-16, with the lowest value indicating the highest priority. Setting the order to no-value is equivalent to giving that server reference the highest priority.
Step 6	UCS-A /security/tacacs/auth-server-group/server-ref # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a TACACS provider group called tacacsgroup, adds two previously configured providers called tacacs1 and tacacs2 to the provider group, sets the order, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope tacacs
UCS-A /security/tacacs # create auth-server-group tacacsgroup
UCS-A /security/tacacs/auth-server-group* # create server-ref tacacs1
UCS-A /security/tacacs/auth-server-group/server-ref* # set order 1
UCS-A /security/tacacs/auth-server-group/server-ref* # up
UCS-A /security/tacacs/auth-server-group* # create server-ref tacacs2
UCS-A /security/tacacs/auth-server-group/server-ref* # set order 2
```

```
UCS-A /security/tacacs/auth-server-group/server-ref* # commit-buffer
UCS-A /security/tacacs/auth-server-group/server-ref #
```

What to Do Next

Configure an authentication domain or select a default authentication service.

Deleting a TACACS Provider Group

Remove the provider group from an authentication configuration.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope tacacs**	Enters security TACACS mode.
Step 3	UCS-A /security/tacacs # **delete auth-server-group** *auth-server-group-name*	Deletes the TACACS provider group.
Step 4	UCS-A /security/tacacs # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a TACACS provider group called tacacsgroup and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope tacacs
UCS-A /security/tacacs # delete auth-server-group tacacsgroup
UCS-A /security/tacacs* # commit-buffer
UCS-A /security/tacacs #
```

Authentication Domains

The Cisco UCS Manager uses Authentication Domains to leverage multiple authentication systems. You can specify and configure each authentication domain during login; otherwise, Cisco UCS Manager uses the default authentication service configuration.

You can create up to eight authentication domains. Each authentication domain is associated with a provider group and a realm in the Cisco UCS Manager. The Cisco UCS Manager uses all servers within the realm if you do not specify a provider group.

Creating an Authentication Domain

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **create auth-domain** *domain-name*	Creates an authentication domain and enters authentication domain mode. **Note** For systems using the remote authentication protocol, the authentication domain name is considered part of the username and counts toward the 32-character limit for locally created usernames. Because Cisco UCS inserts 5 characters for formatting, authentication fails if the domain name and username combined characters total exceeds 27.
Step 3	UCS-A /security/auth-domain # **set refresh-period** *seconds*	(Optional) When a web client connects to Cisco UCS Manager, the client must send refresh requests to Cisco UCS Manager to keep the web session active. This option specifies the maximum amount of time allowed between refresh requests for a user in this domain. If this time limit is exceeded, Cisco UCS Manager considers the web session inactive, but it does not terminate the session. Specify an integer between 60 and 172800. The default is 600 seconds when Two-Factor Authentication is not enabled and 7200 seconds when it is enabled. **Note** The number of seconds set for the **Web Session Refresh Period** must be less than the number of seconds set for the **Web Session Timeout**. Do not set the **Web Session Refresh Period** to the same value as the **Web Session Timeout**.
Step 4	UCS-A /security/auth-domain # **set session-timeout** *seconds*	(Optional) The maximum amount of time that can elapse after the last refresh request before Cisco UCS Manager considers a web session as inactive. If this time limit is exceeded, Cisco UCS Manager automatically terminates the web session. Specify an integer between 300 and 172800. The default is 7200 seconds when Two-Factor Authentication is not enabled and 8000 seconds when it is enabled. **Note** If you set two-factor authentication for a RADIUS or TACACS+ realm, consider increasing the **session-refresh** and **session-timeout** periods so that remote users will not have to re-authenticate too frequently.

	Command or Action	Purpose
Step 5	UCS-A /security/auth-domain # **create default-auth**	(Optional) Creates a default authentication for the authentication domain.
Step 6	UCS-A /security/auth-domain/default-auth # **set auth-server-group** *auth-serv-group-name*	(Optional) Sets the provider group for the authentication domain.
Step 7	UCS-A /security/auth-domain/default-auth # **set realm** {**ldap** | **local** | **radius** | **tacacs**}	Sets the realm for the authentication domain.
Step 8	UCS-A /security/auth-domain/default-auth # **set use-2-factor yes**	(Optional) Sets the authentication method to two-factor authentication for the realm. **Note** Two-factor authentication applies only to the RADIUS and TACACS+ realms.
Step 9	UCS-A /security/auth-domain/default-auth # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates an authentication domain called domain1 with a web refresh period of 3600 seconds (1 hour) and a session timeout period of 14400 seconds (4 hours). It then configures domain1 to use the providers in radius1, sets the realm type to radius, enables two-factor authentication, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # create auth-domain domain1
UCS-A /security/auth-domain* # set refresh-period 3600
UCS-A /security/auth-domain* # set session-timeout 14400
UCS-A /security/auth-domain* # create default-auth
UCS-A /security/auth-domain/auth-domain* # set auth-server-group radius1
UCS-A /security/auth-domain/auth-domain* # set realm radius
UCS-A /security/auth-domain/auth-domain* # set user-2-factor yes
UCS-A /security/auth-domain/auth-domain* # commit-buffer
UCS-A /security/auth-domain/auth-domain #
```

Selecting a Primary Authentication Service

Selecting the Console Authentication Service

Before You Begin

If the system uses a remote authentication service, create a provider for that authentication service. If the system uses only local authentication through Cisco UCS, you do not need to create a provider first.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope console-auth**	Enters console authorization security mode.
Step 3	UCS-A /security/console-auth # **set realm** *auth-type*	Specifies the console authentication, where the *auth-type* argument is one of the following keywords: • **ldap —**Specifies LDAP authentication • **local —**Specifies local authentication • **none —**Allows local users to log on without specifying a password • **radius —**Specifies RADIUS authentication • **tacacs —**Specifies TACACS+ authentication
Step 4	UCS-A /security/console-auth # **set auth-server-group** *auth-serv-group-name*	(Optional) The associated provider group, if any.
Step 5	UCS-A /security/default-auth # **set use-2-factor yes**	(Optional) Sets the authentication method to two-factor authentication for the realm. **Note** Two-factor authentication applies only to the RADIUS and TACACS+ realms.
Step 6	UCS-A /security/console-auth # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the authentication realm to TACACS+, sets the console authentication provider group to provider1, enables two-factor authentication, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope console-auth
UCS-A /security/console-auth # set realm tacacs
UCS-A /security/console-auth # set auth-server-group provider1
UCS-A /security/console-auth* # set use-2-factor yes
UCS-A /security/console-auth* # commit-buffer
UCS-A /security/console-auth #
```

Selecting the Default Authentication Service

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope default-auth**	Enters default authorization security mode.
Step 3	UCS-A /security/default-auth # **set realm** *auth-type*	Specifies the default authentication, where *auth-type* is one of the following keywords: • **ldap—**Specifies LDAP authentication • **local—**Specifies local authentication • **none—**Allows local users to log on without specifying a password • **radius—**Specifies RADIUS authentication • **tacacs—**Specifies TACACS+ authentication
Step 4	UCS-A /security/default-auth # **set auth-server-group** *auth-serv-group-name*	(Optional) The associated provider group, if any.
Step 5	UCS-A /security/default-auth # **set refresh-period** *seconds*	(Optional) When a web client connects to Cisco UCS Manager, the client must send refresh requests to Cisco UCS Manager to keep the web session active. This option specifies the maximum amount of time allowed between refresh requests for a user in this domain. If this time limit is exceeded, Cisco UCS Manager considers the web session inactive, but it does not terminate the session. Specify an integer between 60 and 172800. The default is 600 seconds when Two-Factor Authentication is not enabled and 7200 seconds when it is enabled.
Step 6	UCS-A /security/default-auth # **set session-timeout** *seconds*	(Optional) The maximum amount of time that can elapse after the last refresh request before Cisco UCS Manager considers a web session as inactive. If this time limit is exceeded, Cisco UCS Manager automatically terminates the web session. Specify an integer between 300 and 172800. The default is 7200 seconds when Two-Factor Authentication is not enabled and 8000 seconds when it is enabled. **Note** If you set two-factor authentication for a RADIUS or TACACS+ realm, consider increasing the **session-refresh** and **session-timeout** periods so that remote users will not have to re-authenticate too frequently.

	Command or Action	Purpose
Step 7	UCS-A /security/default-auth # **set use-2-factor yes**	(Optional) Sets the authentication method to two-factor authentication for the realm. **Note** Two-factor authentication applies only to the RADIUS and TACACS+ realms.
Step 8	UCS-A /security/default-auth # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the default authentication to RADIUS, the default authentication provider group to provider1, enables two-factor authentications, sets the refresh period to 7200 seconds (2 hours), the session timeout period to 28800 seconds (8 hours), and enables two-factor authentication. It then commits the transaction.

```
UCS-A# scope security
UCS-A /security # scope default-auth
UCS-A /security/default-auth # set realm radius
UCS-A /security/default-auth* # set auth-server-group provider1
UCS-A /security/default-auth* # set use-2-factor yes
UCS-A /security/default-auth* # set refresh-period 7200
UCS-A /security/default-auth* # set session-timeout 28800
UCS-A /security/default-auth* # commit-buffer
UCS-A /security/default-auth #
```

Role Policy for Remote Users

By default, if user roles are not configured in Cisco UCS Manager read-only access is granted to all users logging in to Cisco UCS Manager from a remote server using the LDAP, RADIUS, or TACACS protocols. For security reasons, it might be desirable to restrict access to those users matching an established user role in Cisco UCS Manager.

You can configure the role policy for remote users in the following ways:

assign-default-role

Does not restrict user access to Cisco UCS Manager based on user roles. Read-only access is granted to all users unless other user roles have been defined in Cisco UCS Manager.

This is the default behavior.

no-login

Restricts user access to Cisco UCS Manager based on user roles. If user roles have not been assigned for the remote authentication system, access is denied.

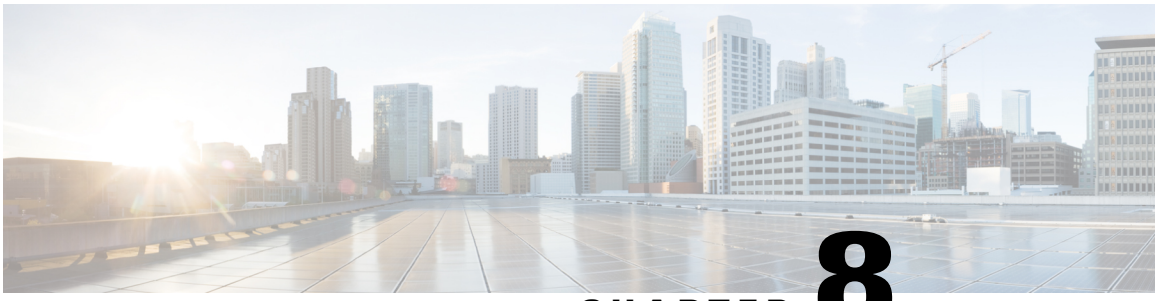
Configuring the Role Policy for Remote Users

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **set remote-user default-role** {**assign-default-role** | **no-login**}	Specifies whether user access to Cisco UCS Manager is restricted based on user roles.
Step 3	UCS-A /security # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the role policy for remote users and commits the transaction:

```
UCS-A# scope security
UCS-A /security # set remote-user default-role assign-default-role
UCS-A /security* # commit-buffer
UCS-A /security #
```



CHAPTER 8

Configuring Organizations

This chapter includes the following sections:

Organizations in a Multitenancy Environment

Multi-tenancy allows you to divide the large physical infrastructure of an Cisco UCS domain into logical entities known as organizations. As a result, you can achieve a logical isolation between organizations without providing a dedicated physical infrastructure for each organization.

You can assign unique resources to each tenant through the related organization in the multi-tenant environment. These resources can include different policies, pools, and quality of service definitions. You can also implement locales to assign or restrict user privileges and roles by organization, if you do not want all users to have access to all organizations.

If you set up a multi-tenant environment, all organizations are hierarchical. The top-level organization is always root. The policies and pools that you create in root are system-wide and are available to all organizations in the system. However, any policies and pools created in other organizations are only available to organizations that are above it in the same hierarchy. For example, if a system has organizations named Finance and HR that are not in the same hierarchy, Finance cannot use any policies in the HR organization, and HR cannot access any policies in the Finance organization. However, both Finance and HR can use policies and pools in the root organization.

If you create organizations in a mult-itenant environment, you can also set up one or more of the following for each organization or for a sub-organization in the same hierarchy:

- Resource pools
- Policies
- Service profiles

- Service profile templates

The root organization is always the top level organization.

Hierarchical Name Resolution in a Multi-Tenancy Environment

In a multi-tenant environment, Cisco UCS uses the hierarchy of an organization to resolve the names of policies and resource pools. When Cisco UCS Manager searches for details of a policy or a resource assigned to a pool, the following occurs:

1. Cisco UCS Manager checks for policies and pools with the specified name within the organization assigned to the service profile or policy.
2. If a policy is found or an available resource is inside a pool, Cisco UCS Manager uses that policy or resource. If the pool does not have any available resources at the local level, Cisco UCS Manager moves up in the hierarchy to the parent organization and searches for a pool with the same name. Cisco UCS Manager repeats this step until the search reaches the root organization.
3. If the search reaches the root organization and has not found an available resource or policy, Cisco UCS Manager returns to the local organization and begins to search for a default policy or available resource in the default pool.
4. If an applicable default policy or available resource in a default pool is found, Cisco UCS Manager uses that policy or resource. If the pool does not have any available resources, Cisco UCS Manager moves up in the hierarchy to the parent organization and searches for a default pool. Cisco UCS Manager repeats this step until the search reaches the root organization.
5. If Cisco UCS Manager cannot find an applicable policy or available resource in the hierarchy, it returns an allocation error.

Example: Server Pool Name Resolution in a Single-Level Hierarchy

In this example, all organizations are at the same level below the root organization. For example, a service provider creates separate organizations for each customer. In this configuration, organizations only have access to the policies and resource pools assigned to that organization and to the root organization.

In this example, a service profile in the XYZcustomer organization is configured to use servers from the XYZcustomer server pool. When resource pools and policies are assigned to the service profile, the following occurs:

1. Cisco UCS Manager checks for an available server in the XYZcustomer server pool.
2. If the XYZcustomer server pool has an available server, Cisco UCS Manager associates that server with the service profile and discontinues the search. If the pool does not have an available server, Cisco UCS Manager checks the root organization for a server pool with the same name.
3. If the root organization includes an XYZcustomer server pool and that pool has an available server, Cisco UCS Manager associates that server with the service profile and discontinues the search. If the pool does not have an available server, Cisco UCS Manager returns to the XYZcustomer organization to check the default server pool.
4. If the default pool in the XYZcustomer organization has an available server, Cisco UCS Manager associates that server with the service profile and discontinues the search. If the default pool does not have an available server, Cisco UCS Manager checks the default server pool in the root organization.

5 If the default server pool in the root organization has an available server, Cisco UCS Manager associates that server with the service profile and discontinues the search. If the default pool does not have an available server, Cisco UCS Manager returns an allocation error.

Example: Server Pool Name Resolution in a Multi-Level Hierarchy

In this example, each organization includes at least one suborganization. For example, a company could create organizations for each major division in the company and for subdivisions of those divisions. In this configuration, each organization has access to its local policies and resource pools and to the resource pools in the parent hierarchy.

In this example, the Finance organization includes two sub-organizations, AccountsPayable and AccountsReceivable. A service profile in the AccountsPayable organization is configured to use servers from the AP server pool. When resource pools and policies are assigned to the service profile, the following occurs:

1 Cisco UCS Manager checks for an available server in the AP server pool defined in the service profile.

2 If the AP server pool has an available server, Cisco UCS Manager associates that server with the service profile and discontinues the search. If the pool does not have an available server, Cisco UCS Manager moves one level up the hierarchy and checks the Finance organization for a pool with the same name.

3 If the Finance organization includes a pool with the same name and that pool has an available server, Cisco UCS Manager associates that server with the service profile and discontinues the search. If the pool does not have an available server, Cisco UCS Manager moves one level up in the hierarchy and checks the root organization for a pool with the same name.

4 If the root organization includes a pool with the same name and that pool has an available server, Cisco UCS Manager associates that server with the service profile and discontinues the search. If the pool does not have an available server, Cisco UCS Manager returns to the AccountsPayable organization to check the default server pool.

5 If the default pool in the AccountsPayable organization has an available server, Cisco UCS Manager associates that server with the service profile and discontinues the search. If the default pool does not have an available server, Cisco UCS Manager moves one level up in the hierarchy and checks the default server pool in the Finance organization.

6 If the default pool in the Finance organization has an available server, Cisco UCS Manager associates that server with the service profile and discontinues the search. If the default pool does not have an available server, Cisco UCS Manager moves one level up in the hierarchy and checks the default server pool in the root organization.

7 If the default server pool in the root organization has an available server, Cisco UCS Manager associates that server with the service profile and discontinues the search. If the default pool does not have an available server, Cisco UCS Manager returns an allocation error.

Configuring an Organization Under the Root Organization

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A /org # **create org** *org-name*	Creates the specified organization under the root organization and enters organization mode for the specified organization. **Note** When you move from one organization mode to another, the command prompt does not change.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates an organization named Finance under the root organization and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create org Finance
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring an Organization Under an Organization that is not Root

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A /org # **scope org** *org-name*	Enters organization mode for the specified organization. **Note** When you move from one organization mode to another, the command prompt does not change.
Step 3	UCS-A /org # **create org** *org-name*	Creates the specified organization under the previously configured non-root organization and enters organization mode for the specified organization.
Step 4	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates an organization named Finance under the NorthAmerica organization and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope org NorthAmerica
UCS-A /org # create org Finance
UCS-A /org* # commit-buffer
UCS-A /org #
```

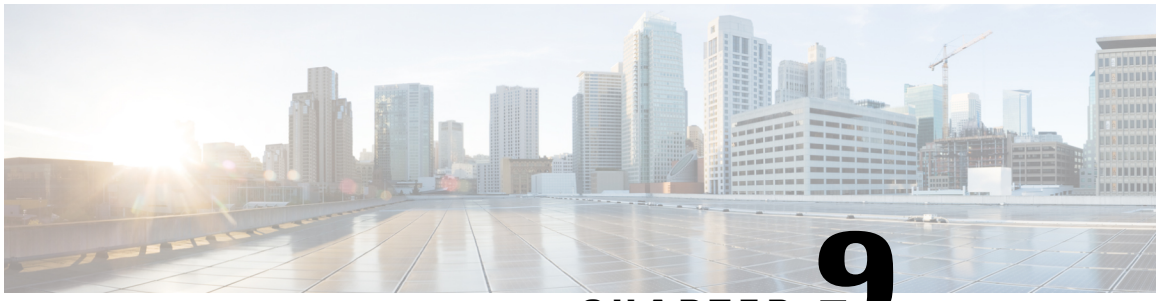
Deleting an Organization

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A /org # **delete org** *org-name*	Deletes the specified organization.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the organization under the root organization named Finance and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete org Finance
UCS-A /org* # commit-buffer
UCS-A /org #
```



CHAPTER 9

Configuring Role-Based Access Control

This chapter includes the following sections:

Role-Based Access Control Overview

Role-Based Access Control (RBAC) is a method of restricting or authorizing system access for users based on user roles and locales. A role defines the privileges of a user in the system and a locale defines the organizations (domains) that a user is allowed access. Because users are not directly assigned privileges, you can manage individual user privileges by assigning the appropriate roles and locales.

A user is granted write access to the required system resources only if the assigned role grants the access privileges and the assigned locale allows access. For example, a user with the Server Administrator role in the engineering organization can update server configurations in the Engineering organization. They cannot, however, update server configurations in the Finance organization, unless the locales assigned to the user include the Finance organization.

User Accounts for Cisco UCS

User accounts access the system. You can configure up to 48 local user accounts in each Cisco UCS Manager domain. Each user account requires a unique username and password.

You can set user accounts with an SSH public key. The public key can be set in either of the two formats: OpenSSH or SECSH.

Admin Account

An admin account comes with each Cisco UCS domain. The admin account is a default user account and cannot be modified or deleted. This account is the system administrator or superuser account s full privileges. There is no default password assigned to the admin account; you must choose the password during the initial system setup.

The admin account is always active and does not expire. You cannot configure the admin account as inactive.

Locally Authenticated User Accounts

A locally authenticated user account is authenticated directly through the fabric interconnect and can be enabled or disabled by anyone with admin or aaa privileges. After a local user account is disabled, the user cannot log in. The database does not delete the configuration details for disabled local user accounts. If you re-enable a disabled local user account, the account becomes active with the existing configuration, including the username and password.

Remotely Authenticated User Accounts

A remotely authenticated user account is any user account that is authenticated through LDAP, RADIUS, or TACACS+.

If a user maintains a local user account and a remote user account simultaneously, the roles defined in the local user account override those maintained in the remote user account.

Expiration of User Accounts

You can configure user accounts to expire at a predefined time. When the expiration time is reached, the user account is disabled.

By default, user accounts do not expire.

Note After you configure a user account with an expiration date, you cannot reconfigure the account to not expire. However, you can configure the account to use the latest expiration date available.

Guidelines for Cisco UCS Usernames

The username is also used as the login ID for Cisco UCS Manager. When you assign login IDs to Cisco UCS user accounts, consider the following guidelines and restrictions:

- The login ID can contain between 1 and 32 characters, including the following:
  - Any alphabetic character
  - Any digit
  - _ (underscore)
  - - (dash)
  - . (dot)

- The login ID must be unique within Cisco UCS Manager.
- The login ID must start with an alphabetic character. It cannot start with a number or a special character, such as an underscore.
- The login ID is case-sensitive.
- You cannot create an all-numeric login ID.
- After you create a user account, you cannot change the login ID. You must delete the user account and create a new one.

Reserved Words: Locally Authenticated User Accounts

You cannot use the following words when creating a local user account in Cisco UCS.

- root
- bin
- daemon
- adm
- lp
- sync
- shutdown
- halt
- news
- uucp
- operator
- games
- gopher
- nobody
- nscd
- mailnull
- mail
- rpcuser
- rpc
- mtsuser
- ftpuser
- ftp
- man
- sys

- samdme
- debug

Guidelines for Cisco UCS Passwords

Each locally authenticated user account requires a password. A user with admin or aaa privileges can configure Cisco UCS Manager to perform a password strength check on user passwords.

Cisco recommends using a strong password; otherwise, the password strength check for locally authenticated users, Cisco UCS Manager rejects any password that does not meet the following requirements:

- Must contain a minimum of eight characters and a maximum of 80 characters.
- If the password strength check is turned on, the minimum password length is variable and can be set from a minimum of 6 to a maximum of 80 characters.

Note The default is 8 characters.

- Must contain at least three of the following:
  - Lower case letters
  - Upper case letters
  - Digits
  - Special characters
- Must not contain a character that is repeated more than three times consecutively, such as aaabbb.
- Must not be identical to the username or the reverse of the username.
- Must pass a password dictionary check. For example, the password must not be based on a standard dictionary word.
- Must not contain the following symbols: $ (dollar sign), ? (question mark), and = (equals sign).
- Should not be blank for local user and admin accounts.

Web Session Limits for User Accounts

Cisco UCS Manager uses web session limits to restrict the number of web sessions (both GUI and XML) that a given user account is permitted to access at any one time.

Each Cisco UCS Manager domain supports a maximum of 32 concurrent web sessions per user and 256 total user sessions. By default, the number of concurrent web sessions allowed by Cisco UCS Manager is set to 32 per user, but you can configure this value up to the system maximum of 256.

User Roles

User roles contain one or more privileges that define the operations that are allowed for a user. You can assign one or more roles to each user. Users with multiple roles have the combined privileges of all assigned roles. For example, if Role1 has storage-related privileges, and Role 2 has server-related privileges, users with Role1 and Role 2 have both storage-related and server-related privileges.

A Cisco UCS domain can contain up to 48 user roles, including the default user roles. Any user roles configured after the first 48 are accepted, but they are inactive with faults raised.

All roles include read access to all configuration settings in the Cisco UCS domain. Users with read-only roles cannot modify the system state.

You can create, modify or remove existing privileges, and delete roles. When you modify a role, the new privileges apply to all users with that role. Privilege assignment is not restricted to the privileges defined for the default roles. Meaning, you can use a custom set of privileges to create a unique role. For example, the default Server Administrator and Storage Administrator roles have a different set of privileges. However, you can create a Server and Storage Administrator role that combines the privileges of both roles.

Note If you delete a role after it was assigned to users, it is also deleted from those user accounts.

Modify the user profiles on AAA servers (RADIUS or TACACS+) to add the roles corresponding to the privileges granted to that user. The attribute stores the role information. The AAA servers return this attribute with the request and parse it to obtain the roles. LDAP servers return the roles in the user profile attributes.

Note If a local and a remote user account have the same username, Cisco UCS Manager overrides any roles assigned to the remote user with those assigned to the local user.

Default User Roles

The system contains the following default user roles:

AAA Administrator

Read-and-write access to users, roles, and AAA configuration. Read access to the remaining system.

Administrator

Complete read-and-write access to the entire system. Assigns this role to the default administrator account by default. You cannot change it.

Facility Manager

Read-and-write access to power management operations through the power management privilege. Read access to the remaining system.

Network Administrator

Read-and-write access to fabric interconnect infrastructure and network security operations. Read access to the remaining system.

Operations

Read-and-write access to systems logs, including the syslog servers, and faults. Read access to the remaining system.

Read-Only

Read-only access to system configuration with no privileges to modify the system state.

Server Compute

Read and write access to most aspects of service profiles. However, the user cannot create, modify or delete vNICs or vHBAs.

Server Equipment Administrator

Read-and-write access to physical server-related operations. Read access to the remaining system.

Server Profile Administrator

Read-and-write access to logical server-related operations. Read access to the remaining system.

Server Security Administrator

Read-and-write access to server security-related operations. Read access to the remaining system.

Storage Administrator

Read-and-write access to storage operations. Read access to the remaining system.

Reserved Words: User Roles

You cannot use the following words when creating custom roles in Cisco UCS.

- network-admin
- network-operator
- vdc-admin
- vdc-operator
- server-admin

Privileges

Privileges give users, assigned to user roles, access to specific system resources and permission to perform specific tasks. The following table lists each privilege and the user role given that privilege by default.

Tip Detailed information about these privileges and the tasks that they enable users to perform is available in *Privileges in Cisco UCS* available at the following URL: http://www.cisco.com/en/US/products/ps10281/prod_technical_reference_list.html.

Table 7: User Privileges

Privilege	Description	Default Role Assignment
aaa	System security and AAA	AAA Administrator
admin	System administration	Administrator
ext-lan-config	External LAN configuration	Network Administrator
ext-lan-policy	External LAN policy	Network Administrator
ext-lan-qos	External LAN QoS	Network Administrator
ext-lan-security	External LAN security	Network Administrator
ext-san-config	External SAN configuration	Storage Administrator
ext-san-policy	External SAN policy	Storage Administrator
ext-san-qos	External SAN QoS	Storage Administrator
ext-san-security	External SAN security	Storage Administrator
fault	Alarms and alarm policies	Operations
operations	Logs and Smart Call Home	Operations
org-management	Organization management	Operations
pod-config	Pod configuration	Network Administrator
pod-policy	Pod policy	Network Administrator
pod-qos	Pod QoS	Network Administrator
pod-security	Pod security	Network Administrator
power-mgmt	Read-and-write access to power management operations	Facility Manager

Privilege	Description	Default Role Assignment
read-only	Read-only access Read-only cannot be selected as a privilege; it is assigned to every user role.	Read-Only
server-equipment	Server hardware management	Server Equipment Administrator
server-maintenance	Server maintenance	Server Equipment Administrator
server-policy	Server policy	Server Equipment Administrator
server-security	Server security	Server Security Administrator
service-profile-compute	Service profile compute	Server Compute Administrator
service-profile-config	Service profile configuration	Server Profile Administrator
service-profile-config-policy	Service profile configuration policy	Server Profile Administrator
service-profile-ext-access	Service profile endpoint access	Server Profile Administrator
service-profile-network	Service profile network	Network Administrator
service-profile-network-policy	Service profile network policy	Network Administrator
service-profile-qos	Service profile QoS	Network Administrator
service-profile-qos-policy	Service profile QoS policy	Network Administrator
service-profile-security	Service profile security	Server Security Administrator
service-profile-security-policy	Service profile security policy	Server Security Administrator
service-profile-server	Service profile server management	Server Profile Administrator
service-profile-server-oper	Service profile consumer	Server Profile Administrator
service-profile-server-policy	Service profile pool policy	Server Security Administrator
service-profile-storage	Service profile storage	Storage Administrator
service-profile-storage-policy	Service profile storage policy	Storage Administrator

User Locales

You can assign a user to one or more locales. Each locale defines one or more organizations (domains) to which a user can access. Access is usually limited to the organizations specified in the locale. An exception is a locale without any organizations. It provides unrestricted access to system resources in all organizations.

A Cisco UCS domain can contain up to 48 user locales. Any user locales configured after the first 48 are accepted, but are inactive with faults raised.

Users with admin or aaa privileges can assign organizations to the locale of other users. The assignment of organizations is restricted to only those in the locale of the user assigning the organizations. For example, if a locale contains only the Engineering organization, a user assigned to that locale can only assign the Engineering organization to other users.

Note You cannot assign a locale to users with one or more of the following privileges:

- aaa
- admin
- fault
- operations

You can hierarchically manage organizations. A user who is assigned to a top-level organization has automatic access to all organizations below it. For example, an Engineering organization can contain a Software Engineering organization and a Hardware Engineering organization. A locale containing only the Software Engineering organization has access to system resources only within that organization. However, a locale that contains the Engineering organization has access to the resources for both the Software Engineering and Hardware Engineering organizations.

Configuring User Roles

Creating a User Role

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **create role** *name*	Creates the user role and enters security role mode.
Step 3	UCS-A /security/role # **add privilege** *privilege-name*	Adds one or more privileges to the role.

	Command or Action	Purpose
		Note You can specify more than one *privilege-name* on the same command line to add multiple privileges to the role, or you can add privileges to the same role using multiple **add** commands.
Step 4	UCS-A /security/role # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates the service-profile-security-admin role, adds the service profile security and service profile security policy privileges to the role, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # create role ls-security-admin
UCS-A /security/role* # add privilege service-profile-security service-profile-security-policy
UCS-A /security/role* # commit-buffer
UCS-A /security/role #
```

Adding Privileges to a User Role

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope role** *name*	Enters security role mode for the specified role.
Step 3	UCS-A /security/role # **add privilege** *privilege-name*	Adds one or more privileges to the existing privileges of the user role. **Note** You can specify more than one *privilege-name* on the same command line to add multiple privileges to the role, or you can add privileges to the same role using multiple **add privilege** commands.
Step 4	UCS-A /security/role # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to add the server security and server policy privileges to the service-profile-security-admin role and commit the transaction:

```
UCS-A# scope security
UCS-A /security # scope role service-profile-security-admin
UCS-A /security/role # add privilege server-security server-policy
UCS-A /security/role* # commit-buffer
UCS-A /security/role #
```

Replacing Privileges for a User Role

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope role** *name*	Enters security role mode for the specified role.
Step 3	UCS-A /security/role # **set privilege** *privilege-name*	Replaces the existing privileges of the user role. **Note** You can specify more than one *privilege-name* on the same command line to replace the existing privilege with multiple privileges. After replacing the privileges, you can add privileges to the same role using the **add privilege** command.
Step 4	UCS-A /security/role # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to replace the existing privileges for the service-profile-security-admin role with the server security and server policy privileges and commit the transaction:

```
UCS-A# scope security
UCS-A /security # scope role service-profile-security-admin
UCS-A /security/role # set privilege server-security server-policy
UCS-A /security/role* # commit-buffer
UCS-A /security/role #
```

Removing Privileges from a User Role

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope role** *name*	Enters security role mode for the specified role.
Step 3	UCS-A /security/role # **remove privilege** *privilege-name*	Removes one or more privileges from the existing user role privileges. **Note** You can specify more than one *privilege-name* on the same command line to remove multiple privileges from the role, or you can remove privileges from the same role using multiple **remove privilege** commands.
Step 4	UCS-A /security/role # **commit-buffer**	Commits the transaction to the system configuration.

The following example removes the server security and server policy privileges from the service-profile-security-admin role and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope role service-profile-security-admin
UCS-A /security/role # remove privilege server-security server-policy
UCS-A /security/role* # commit-buffer
UCS-A /security/role #
```

Deleting a User Role

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **delete role** *name*	Deletes the user role.
Step 3	UCS-A /security # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the service-profile-security-admin role and commits the transaction:

```
UCS-A# scope security
UCS-A /security # delete role service-profile-security-admin
UCS-A /security* # commit-buffer
UCS-A /security #
```

Configuring Locales

Creating a Locale

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **create locale** *locale-name*	Creates a locale and enters security locale mode.
Step 3	UCS-A /security/locale # **create org-ref** *org-ref-name* **orgdn** *orgdn org-root/org-ref-name*	References (binds) an organization to the locale. The *org-ref-name* argument is the name used to identify the organization reference, and the *orgdn-name* argument is the distinguished name of the organization being referenced.

	Command or Action	Purpose
Step 4	UCS-A /security/locale # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates the western locale, references the finance organization to the locale, names the reference finance-ref, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # create locale western
UCS-A /security/locale* # create org-ref finance-ref orgdn org-root/org-finance
UCS-A /security/locale* # commit-buffer
UCS-A /security/locale #
```

Assigning an Organization to a Locale

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A# **scope locale** *locale-name*	Enters security locale mode.
Step 3	UCS-A /security/locale # **create org-ref** *org-ref-name* **orgdn** *org-root/org-ref-name*	References (binds) an organization to the locale. The *org-ref-name* argument is the name used to identify the organization reference, and the *orgdn-name* argument is the distinguished name of the organization being referenced.
Step 4	UCS-A /security/locale # **commit-buffer**	Commits the transaction to the system configuration.

The following example enters the western locale, adds (references) the marketing organization to the locale, names the reference marketing-ref, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope locale western
UCS-A /security/locale* # create org-ref marketing-ref orgdn org-root/org-marketing
UCS-A /security/locale* # commit-buffer
UCS-A /security/locale #
```

Deleting an Organization from a Locale

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.

	Command or Action	Purpose
Step 2	UCS-A /security # **scope locale** *locale-name*	Enters security locale mode.
Step 3	UCS-A /security/locale # **delete org-ref** *org-ref-name*	Deletes the organization from the locale.
Step 4	UCS-A /security/locale # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the finance organization from the western locale and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope locale western
UCS-A /security/locale # delete org-ref finance-ref
UCS-A /security/locale* # commit-buffer
UCS-A /security/locale #
```

Deleting a Locale

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **delete locale** *locale-name*	Deletes the locale.
Step 3	UCS-A /security # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the western locale and commits the transaction:

```
UCS-A# scope security
UCS-A /security # delete locale western
UCS-A /security* # commit-buffer
UCS-A /security #
```

Configuring Locally Authenticated User Accounts

Creating a User Account

At a minimum, Cisco recommends that you create the following users:

- Server administrator account
- Network administrator account
- Storage administrator

Before You Begin

Perform the following tasks, if the system includes any of the following:

- Remote authentication services—Ensures that the users exist in the remote authentication server with the appropriate roles and privileges.
- Multitenancy with organizations—Creates one or more locales. If you do not have any locales, all users are created in root and are assigned roles and privileges in all organizations.
- SSH authentication—Obtains the SSH key.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **create local-user** *local-user-name*	Creates a user account for the specified local user and enters security local user mode.
Step 3	UCS-A /security/local-user # **set account-status** {**active**| **inactive**}	Specifies whether the local user account is enabled or disabled. If the account status for a local user account is set to inactive, the user is prevented from logging into the system using their existing credentials.
Step 4	UCS-A /security/local-user # **set password** *password*	Sets the password for the user account
Step 5	UCS-A /security/local-user # **set firstname** *first-name*	(Optional) Specifies the first name of the user.
Step 6	UCS-A /security/local-user # **set lastname** *last-name*	(Optional) Specifies the last name of the user.
Step 7	UCS-A /security/local-user # **set expiration** *month day-of-month year*	(Optional) Specifies the date that the user account expires. The *month* argument is the first three letters of the month name. **Note** After you configure a user account with an expiration date, you cannot reconfigure the account to not expire. However, you can configure the account to use the latest expiration date available.
Step 8	UCS-A /security/local-user # **set email** *email-addr*	(Optional) Specifies the user e-mail address.
Step 9	UCS-A /security/local-user # **set phone** *phone-num*	(Optional) Specifies the user phone number.

	Command or Action	Purpose
Step 10	UCS-A /security/local-user # **set sshkey** *ssh-key*	(Optional) Specifies the SSH key used for passwordless access.
Step 11	UCS-A security/local-user # **commit-buffer**	Commits the transaction.

The following example creates the user account named kikipopo, enables the user account, sets the password to foo12345, and commits the transaction:

```
UCS-A# scope security
UCS-A /security # create local-user kikipopo
UCS-A /security/local-user* # set account-status active
UCS-A /security/local-user* # set password
Enter a password:
Confirm the password:
UCS-A /security/local-user* # commit-buffer
UCS-A /security/local-user #
```

The following example creates the user account named lincey, enables the user account, sets an OpenSSH key for passwordless access, and commits the transaction.

```
UCS-A# scope security
UCS-A /security # create local-user lincey
UCS-A /security/local-user* # set account-status active
UCS-A /security/local-user* #  set sshkey "ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAuo9VQ2CmWBI9/S1f30klCWjnV3lgdXMzO0WUl5iPw85lkdQqap+NFuNmHcb4K
iaQB8X/PDdmtlxQQcawclj+k8f4VcOelBxlsGk5luq5ls1ob1VOIEwcKEL/h5lrdbNlI8y3SS9I/gGiBZ9ARlop9LDpD
m8HPh2LOgyH7Ei1MI8="
UCS-A /security/local-user* # commit-buffer
UCS-A /security/local-user #
```

The following example creates the user account named jforlenz, enables the user account, sets a Secure SSH key for passwordless access, and commits the transaction.

```
UCS-A# scope security
UCS-A /security # create local-user jforlenz
UCS-A /security/local-user* # set account-status active
UCS-A /security/local-user* #  set sshkey
Enter lines one at a time. Enter ENDOFBUF to finish. Press ^C to abort.
User's SSH key:
> ---- BEGIN SSH2 PUBLIC KEY ----
>AAAAB3NzaC1yc2EAAAABIwAAAIEAuo9VQ2CmWBI9/S1f30klCWjnV3lgdXMzO0WUl5iPw8
>5lkdQqap+NFuNmHcb4KiaQB8X/PDdmtlxQQcawclj+k8f4VcOelBxlsGk5luq5ls1ob1VO
>IEwcKEL/h5lrdbNlI8y3SS9I/gGiBZ9ARlop9LDpDm8HPh2LOgyH7Ei1MI8=
> ---- END SSH2 PUBLIC KEY ----
> ENDOFBUF
UCS-A /security/local-user* # commit-buffer
UCS-A /security/local-user #
```

Enabling the Password Strength Check for Locally Authenticated Users

You must have admin or aaa privileges to enable the password strength check. If enabled, Cisco UCS Manager does not permit a user to choose a password that does not meet the guidelines for a strong password.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **enforce-strong-password** {**yes** | **no**}	Specifies whether the password strength check is enabled or disabled.

The following example enables the password strength check:

```
UCS-A# scope security
UCS-A /security # set enforce-strong-password yes
UCS-A /security #
```

Setting Web Session Limits for User Accounts

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **scope web-session-limits**	Enters system services web session limits mode.
Step 4	UCS-A /system/services/web-session-limits # **set peruser** *num-of-logins-per-user*	Sets the maximum number of concurrent HTTP and HTTPS sessions allowed for each user. Enter an integer between 1 and 256. By default, this value is set to 32.
Step 5	UCS-A /system/services/web-session-limits # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the maximum number of HTTP and HTTPS sessions allowed by each user account to 60 and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # scope web-session-limits
UCS-A /system/services/web-session-limits* # set peruser 60
UCS-A /system/services/web-session-limits* # commit-buffer
UCS-A /system/services/web-session-limits #
```

Assigning a Role to a User Account

Changes in user roles and privileges do not take effect until the next time the user logs in. If a user is logged in when you assign a new role to or remove an existing role from a user account, the active session continues with the previous roles and privileges.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope local-user** *local-user-name*	Enters security local user mode for the specified local user account.
Step 3	UCS-A /security/local-user # **create role** *role-name*	Assigns the specified role to the user account . **Note** The **create role** command can be entered multiple times to assign more than one role to a user account.
Step 4	UCS-A security/local-user # **commit-buffer**	Commits the transaction.

The following example assigns the operations role to the kikipopo local user account and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope local-user kikipopo
UCS-A /security/local-user # create role operations
UCS-A /security/local-user* # commit-buffer
UCS-A /security/local-user #
```

Assigning a Locale to a User Account

Note Do not assign locales to users with an admin or aaa role.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope local-user** *local-user-name*	Enters security local user mode for the specified local user account.
Step 3	UCS-A /security/local-user # **create locale** *locale-name*	Assigns the specified locale to the user account.

	Command or Action	Purpose
		Note The **create locale** command can be entered multiple times to assign more than one locale to a user account.
Step 4	UCS-A security/local-user # **commit-buffer**	Commits the transaction.

The following example assigns the western locale to the kikipopo local user account and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope local-user kikipopo
UCS-A /security/local-user # create locale western
UCS-A /security/local-user* # commit-buffer
UCS-A /security/local-user #
```

Removing a Role from a User Account

Changes in user roles and privileges do not take effect until the next time the user logs in. If a user is logged in when you assign a new role to or remove an existing role from a user account, the active session continues with the previous roles and privileges.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope local-user** *local-user-name*	Enters security local user mode for the specified local user account.
Step 3	UCS-A /security/local-user # **delete role** *role-name*	Removes the specified role from the user account . **Note** The **delete role** command can be entered multiple times to remove more than one role from a user account.
Step 4	UCS-A security/local-user # **commit-buffer**	Commits the transaction.

The following example removes the operations role from the kikipopo local user account and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope local-user kikipopo
UCS-A /security/local-user # delete role operations
UCS-A /security/local-user* # commit-buffer
UCS-A /security/local-user #
```

Removing a Locale from a User Account

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope local-user** *local-user-name*	Enters security local user mode for the specified local user account.
Step 3	UCS-A /security/local-user # **delete locale** *locale-name*	Removes the specified locale from the user account. **Note** The **delete locale** command can be entered multiple times to remove more than one locale from a user account.
Step 4	UCS-A security/local-user # **commit-buffer**	Commits the transaction.

The following example removes the western locale from the kikipopo local user account and commits the transaction:

```
UCS-A# scope security
UCS-A /security # scope local-user kikipopo
UCS-A /security/local-user # delete locale western
UCS-A /security/local-user* # commit-buffer
UCS-A /security/local-user #
```

Enabling or Disabling a User Account

You must have admin or aaa privileges to enable or disable a local user account.

Before You Begin

Create a local user account.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope local-user**	Enters local-user security mode.
Step 3	UCS-A /security/local-user # **set account-status** {**active** | **inactive**}	Specifies whether the local user account is enabled or disabled. The admin user account is always set to active. It cannot be modified. **Note** If you set the account status to inactive, the configuration is not deleted from the database.

	Command or Action	Purpose

The following example enables a local user account called accounting:

```
UCS-A# scope security
UCS-A /security # scope local-user accounting
UCS-A /security/local-user # set account-status active
```

Clearing the Password History for a Locally Authenticated User

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope local-user** *user-name*	Enters local user security mode for the specified user account.
Step 3	UCS-A /security/local-user # **set clear password-history yes**	Clears the password history for the specified user account.
Step 4	UCS-A /security/local-user # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures the password history count and commits the transaction:

```
UCS-A # scope security
UCS-A /security # scope local-user admin
UCS-A /security/local-user # set clear password-history yes
UCS-A /security/local-user* # commit-buffer
UCS-A /security/local-user #
```

Deleting a User Account

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **delete local-user** *local-user-name*	Deletes the local-user account.
Step 3	UCS-A /security # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the foo user account and commits the transaction:

```
UCS-A# scope security
UCS-A /security # delete local-user foo
UCS-A /security* # commit-buffer
UCS-A /security #
```

Password Profile for Locally Authenticated Users

The password profile contains the password history and the password change interval properties for all locally authenticated users of Cisco UCS Manager. You cannot specify a different password profile for locally authenticated users.

Note You must have admin or aaa privileges to change the password profile properties. Except for password history, these properties do not apply to users with admin or aaa privileges.

Password History Count

The password history count prevents locally authenticated users from reusing the same password. When you configure the password history count, Cisco UCS Manager stores up to a maximum of 15 previously used passwords. The password history count stores the passwords in reverse chronological order with the most recent password first. This ensures that the user can only reuse the oldest password when the history count reaches its threshold.

A user can create and use the number of passwords configured in the password history count before reusing a password. For example, if you set the password history count to 8, a user cannot reuse the first password until the ninth password expires.

By default, the password history is set to 0. This value disables the history count and allows users to reuse previously used passwords at any time.

You can clear the password history count for a locally authenticated user and enable reuse of previous passwords.

Password Change Interval

The password change interval restricts the number of password changes that a locally authenticated user can make within a specific number of hours. The following table describes the two interval configuration options for the password change interval.

Interval Configuration	Description	Example
No password change allowed	Does not allow changing passwords for locally authenticated user within a specified number of hours after a password change. You can specify a no change interval between 1 and 745 hours. By default, the no change interval is 24 hours.	To prevent the user from changing passwords within 48 hours after a password change: • Set **Change during interval** to disable • Set **No change interval** to 48

Interval Configuration	Description	Example
Password changes allowed within change interval	Specifies the maximum number of times that a locally authenticated user password change can occur within a pre-defined interval. You can specify a change interval between 1 and 745 hours and a maximum number of password changes between 0 and 10. By default, a locally authenticated user is permitted a maximum of two password changes within a 48-hour interval.	To allow a password change for a maximum of one time within 24 hours after a password change: • Set **Change during interval** to enable • Set **Change count** to 1 • Set **Change interval** to 24

Configuring the Maximum Number of Password Changes for a Change Interval

You must have admin or aaa privileges to change the password profile properties. Except for password history, these properties do not apply to users with admin or aaa privileges.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope password-profile**	Enters password profile security mode.
Step 3	UCS-A /security/password-profile # **set change-during-interval enable**	Restricts the number of password changes a locally authenticated user can make within a given number of hours.
Step 4	UCS-A /security/password-profile # **set change-count** *pass-change-num*	Specifies the maximum number of times a locally authenticated user can change his or her password during the Change Interval. This value can be anywhere from 0 to 10.
Step 5	UCS-A /security/password-profile # **set change-interval** *num-of-hours*	Specifies the maximum number of hours over which the number of password changes specified in the **Change Count** field are enforced. This value can be anywhere from 1 to 745 hours. For example, if this field is set to 48 and the **Change Count** field is set to 2, a locally authenticated user can make no more than 2 password changes within a 48 hour period.
Step 6	UCS-A /security/password-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables the change during interval option, sets the change count to 5, sets the change interval to 72 hours, and commits the transaction:

```
UCS-A # scope security
UCS-A /security # scope password-profile
UCS-A /security/password-profile # set change-during-interval enable
UCS-A /security/password-profile* # set change-count 5
UCS-A /security/password-profile* # set change-interval 72
UCS-A /security/password-profile* # commit-buffer
UCS-A /security/password-profile #
```

Configuring a No Change Interval for Passwords

You must have admin or aaa privileges to change the password profile properties. Except for password history, these properties do not apply to users with admin or aaa privileges.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope password-profile**	Enters password profile security mode.
Step 3	UCS-A /security/password-profile # **set change-during-interval disable**	Disables the change during interval feature.
Step 4	UCS-A /security/password-profile # **set no-change-interval** *min-num-hours*	Specifies the minimum number of hours that a locally authenticated user must wait before changing a newly created password. This value can be anywhere from 1 to 745 hours. This interval is ignored if the **Change During Interval** property is set to **Disable**.
Step 5	UCS-A /security/password-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example disables the change during interval option, sets the no change interval to 72 hours, and commits the transaction:

```
UCS-A # scope security
UCS-A /security # scope password-profile
UCS-A /security/password-profile # set change-during-interval disable
UCS-A /security/password-profile* # set no-change-interval 72
UCS-A /security/password-profile* # commit-buffer
UCS-A /security/password-profile #
```

Configuring the Password History Count

You must have admin or aaa privileges to change the password profile properties.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **scope password-profile**	Enters password profile security mode.
Step 3	UCS-A /security/password-profile # **set history-count** *num-of-passwords*	Specifies the number of unique passwords that a locally authenticated user must create before that user can reuse a previously used password This value can be anywhere from 0 to 15. By default, the **History Count** field is set to 0, which disables the history count and allows users to reuse previously used passwords at any time.
Step 4	UCS-A /security/password-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures the password history count and commits the transaction:

```
UCS-A # scope security
UCS-A /security # scope password-profile
UCS-A /security/password-profile # set history-count 5
UCS-A /security/password-profile* # commit-buffer
UCS-A /security/password-profile #
```

Monitoring User Sessions from the CLI

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope security**	Enters security mode.
Step 2	UCS-A /security # **show user-session** {**local** | **remote**} [**detail**]	Displays session information for all users logged in to the system. An asterisk (*) next to the session ID denotes the current login session.

The following example lists all local users logged in to the system. The asterisk indicates which session is the current login session.

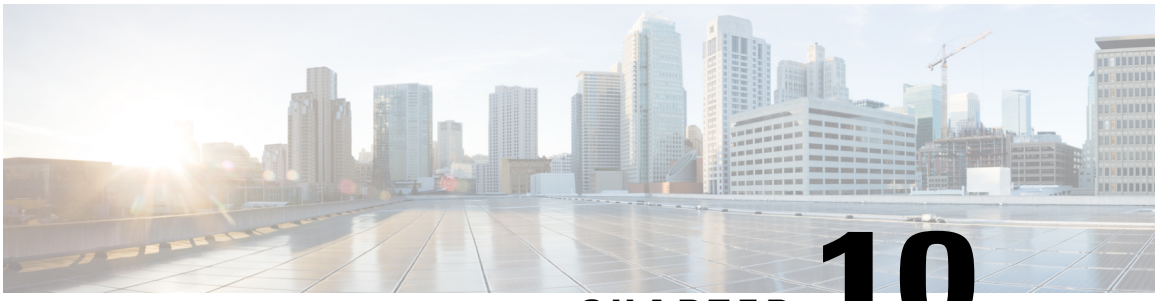
```
UCS-A# scope security
UCS-A /security # show user-session local
Session Id      User            Host                Login Time
--------------- --------------- ------------------- ----------
pts_25_1_31264*  steve           192.168.100.111      2009-05-09T14:06:59
ttyS0_1_3532    jeff            console             2009-05-02T15:11:08
web_25277_A     faye            192.168.100.112     2009-05-15T22:11:25
```

The following example displays detailed information on all local users logged in to the system:

```
UCS-A# scope security
UCS-A /security # show user-session local detail
Session Id pts_25_1_31264:
    Fabric Id: A
    Term: pts/25
    User: steve
    Host: 64.101.53.93
    Pid: 31264
    Login Time: 2009-05-09T14:06:59

Session Id ttyS0_1_3532:
    Fabric Id: A
    Term: ttyS0
    User: jeff
    Host: console
    Pid: 3532
    Login Time: 2009-05-02T15:11:08

Session Id web_25277_A:
    Fabric Id: A
    Term: web_25277
    User: faye
    Host: 192.168.100.112
    Pid: 3518
    Login Time: 2009-05-15T22:11:25
```



CHAPTER 10

Configuring DNS Servers

This chapter includes the following sections:

DNS Servers in Cisco UCS

You need to specify an external DNS server for each Cisco UCS domain to use if the system requires name resolution of hostnames. For example, you cannot use a name such as www.cisco.com when you are configuring a setting on a fabric interconnect if you do not configure a DNS server. You would need to use the IP address of the server, which can be either an IPv4 or an IPv6 address. You can configure up to four DNS servers for each Cisco UCS domain.

Note When you configure multiple DNS servers, the system searches for the servers only in any random order. If a local management command requires DNS server lookup, it can only search for three DNS servers in random order.

Configuring a DNS Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.

	Command or Action	Purpose
Step 3	UCS-A /system/services # **create dns** {*ip-addr ip6-addr*}	Configures the system to use the DNS server with the specified IPv4 or IPv6 address.
Step 4	UCS-A /system/services # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a DNS server with the IPv4 address 192.168.200.105 and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # create dns 192.168.200.105
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

The following example configures a DNS server with the IPv6 address 2001:db8::22:F376:FF3B:AB3F and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # create dns 2001:db8::22:F376:FF3B:AB3F
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

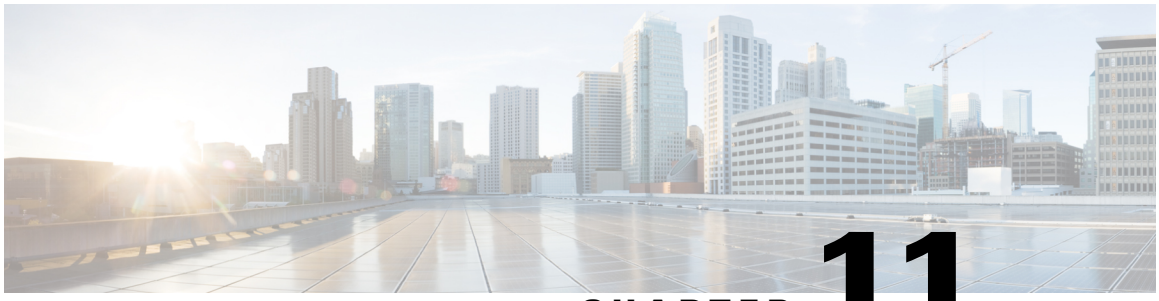
Deleting a DNS Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **delete dns** *ip-addr*	Deletes the NTP server with the specified IP address.
Step 4	UCS-A /system/services # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the DNS server with the IP address 192.168.200.105 and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # delete dns 192.168.200.105
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```



CHAPTER 11

Configuring System-Related Policies

This chapter includes the following sections:

Configuring the Chassis/FEX Discovery Policy

Chassis/FEX Discovery Policy

The chassis/FEX discovery policy determines how the system reacts when you add a new chassis or FEX. Cisco UCS Manager uses the settings in the chassis/FEX discovery policy to determine the minimum threshold for the number of links between the chassis or FEX and the fabric interconnect and whether to group links from the IOM to the fabric interconnect in a fabric port channel.

Chassis Links

If you have a Cisco UCS domain with some of the chassis' wired with one link, some with two links, some with four links, and some with eight links, Cisco recommends configuring the chassis/FEX discovery policy for the minimum number links in the domain so that Cisco UCS Manager can discover all chassis.

Tip To establish the highest available chassis connectivity in a Cisco UCS domain where Fabric Interconnect is connected to different types of IO Modules supporting different max number of uplinks, select platform max value. Setting the platform max ensures that Cisco UCS Manager discovers the chassis including the connections and servers only when the maximum supported IOM uplinks are connected per IO Module.

After the initial discovery, re-acknowledge the chassis' that are wired for a greater number of links and Cisco UCS Manager configures the chassis to use all available links.

Cisco UCS Manager cannot discover any chassis that is wired for fewer links than are configured in the chassis/FEX discovery policy. For example, if the chassis/FEX discovery policy is configured for four links,

Cisco UCS Manager cannot discover any chassis that is wired for one link or two links. Re-acknowledgement of the chassis resolves this issue.

The following table provides an overview of how the chassis/FEX discovery policy works in a multi-chassis Cisco UCS domain:

Table 8: Chassis/FEX Discovery Policy and Chassis Links

Number of Links Wired for the Chassis	1-Link Discovery Policy	2-Link Discovery Policy	4-Link Discovery Policy	8-Link Discovery Policy	Platform-Max Discovery Policy
1 link between IOM and fabric interconnects	Chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 1 link.	Chassis connections and servers cannot be discovered by Cisco UCS Manager and are not added to the Cisco UCS domain.	Chassis connections and servers cannot be discovered by Cisco UCS Manager and are not added to the Cisco UCS domain.	Chassis connections and servers cannot be discovered by Cisco UCS Manager and are not added to the Cisco UCS domain.	Chassis connections and servers cannot be discovered by Cisco UCS Manager and are not added to the Cisco UCS domain.
2 links between IOM and fabric interconnects	Chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 1 link. After initial discovery, reacknowledge the chassis and Cisco UCS Manager recognizes and uses the additional links.	Chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 2 link.	Chassis connections and servers cannot be discovered by Cisco UCS Manager and are not added to the Cisco UCS domain.	Chassis connections and servers cannot be discovered by Cisco UCS Manager and are not added to the Cisco UCS domain.	Chassis connections and servers cannot be discovered by Cisco UCS Manager and are not added to the Cisco UCS domain.

Number of Links Wired for the Chassis	1-Link Discovery Policy	2-Link Discovery Policy	4-Link Discovery Policy	8-Link Discovery Policy	Platform-Max Discovery Policy
4 links between IOM and fabric interconnects	Chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 1 link. After initial discovery, reacknowledge the chassis and Cisco UCS Manager recognizes and uses the additional links.	Chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 2 links. After initial discovery, reacknowledge the chassis and Cisco UCS Manager recognizes and uses the additional links.	Chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 4 link.	Chassis connections and servers cannot be discovered by Cisco UCS Manager and are not added to the Cisco UCS domain.	If the IOM has 4 links, the chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 4 links. If the IOM has 8 links, the chassis is not fully discovered by Cisco UCS Manager.
8 links between IOM and fabric interconnects	Chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 1 link. After initial discovery, reacknowledge the chassis and Cisco UCS Manager recognizes and uses the additional links.	Chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 2 links. After initial discovery, reacknowledge the chassis and Cisco UCS Manager recognizes and uses the additional links.	Chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 4 links. After initial discovery, reacknowledge the chassis and Cisco UCS Manager recognizes and uses the additional links.	Chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 8 links.	Chassis is discovered by Cisco UCS Manager and added to the Cisco UCS domain as a chassis wired with 8 links.

Link Grouping

For hardware configurations that support fabric port channels, link grouping determines whether all of the links from the IOM to the fabric interconnect are grouped in to a fabric port channel during chassis discovery. If the link grouping preference is set to port channel, all of the links from the IOM to the fabric interconnect

are grouped in a fabric port channel. If set to no group, links from the IOM to the fabric interconnect are not grouped in a fabric port channel.

After you create a fabric port channel, you can add or remove links by changing the link group preference and re-acknowledging the chassis, or by enabling or disabling the chassis from the port channel.

Note The link grouping preference only takes effect if both sides of the links between an IOM or FEX and the fabric interconnect support fabric port channels. If one side of the links does not support fabric port channels, this preference is ignored and the links are not grouped in a port channel.

Configuring the Chassis/FEX Discovery Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode. **Note** The chassis/FEX discovery policy can be accessed only from the root organization.
Step 2	UCS-A /org # **scope chassis-disc-policy**	Enters organization chassis/FEX discovery policy mode.
Step 3	UCS-A /org/chassis-disc-policy # **set action** {**1-link** | **2-link** | **4-link** | **8-link** | **platform-max**}	Specifies the minimum threshold for the number of links between the chassis or FEX and the fabric interconnect.
Step 4	UCS-A /org/chassis-disc-policy # **set descr** *description*	(Optional) Provides a description for the chassis/FEX discovery policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 5	UCS-A /org/chassis-disc-policy # **set link-aggregation-pref** {**none** | **port-channel**}	Specifies whether the links from the IOMs or FEXes to the fabric interconnects are grouped in a port channel. **Note** The link grouping preference only takes effect if both sides of the links between an IOM or FEX and the fabric interconnect support fabric port channels. If one side of the links does not support fabric port channels, this preference is ignored and the links are not grouped in a port channel.
Step 6	UCS-A /org/chassis-disc-policy # **set multicast-hw-hash** {**disabled** | **enabled**}	Specifies whether the all the links between the IOM and the fabric interconnect in a port channel can be used for multicast traffic. • **disabled—**Only one link between the IOM and the fabric interconnect is used for multicast traffic

	Command or Action	Purpose
		• **enabled—**All links between the IOM and the fabric interconnect can be used for multicast traffic
Step 7	UCS-A /org/chassis-disc-policy # **set qualifier** *qualifier*	(Optional) Uses the specified server pool policy qualifications to associate this policy with a server pool.
Step 8	UCS-A /org/chassis-disc-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example scopes to the default chassis/FEX discovery policy, sets it to discover chassis with four links to a fabric interconnect, provides a description for the policy, specifies the server pool policy qualifications that will be used to qualify the chassis, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope chassis-disc-policy
UCS-A /org/chassis-disc-policy* # set action 4-link
UCS-A /org/chassis-disc-policy* # set descr "This is an example chassis/FEX discovery
policy."
UCS-A /org/chassis-disc-policy* # set qualifier ExampleQual
UCS-A /org/chassis-disc-policy* # commit-buffer
UCS-A /org/chassis-disc-policy #
```

The following example scopes to the default chassis/FEX discovery policy, sets it to discover chassis with eight links to a fabric interconnect, provides a description for the policy, sets the link grouping preference to port channel, specifies the server pool policy qualifications that will be used to qualify the chassis, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope chassis-disc-policy
UCS-A /org/chassis-disc-policy* # set action 8-link
UCS-A /org/chassis-disc-policy* # set descr "This is an example chassis/FEX discovery
policy."
UCS-A /org/chassis-disc-policy* # set link-aggregation-pref port-channel
UCS-A /org/chassis-disc-policy* # set qualifier ExampleQual
UCS-A /org/chassis-disc-policy* # commit-buffer
UCS-A /org/chassis-disc-policy #
```

The following example scopes to the default chassis/FEX discovery policy, sets it to discover chassis with four links to a fabric interconnect, provides a description for the policy, sets the link grouping preference to port channel, enables multicast hardware hashing, specifies the server pool policy qualifications that will be used to qualify the chassis, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope chassis-disc-policy
UCS-A /org/chassis-disc-policy* # set action 4-link
UCS-A /org/chassis-disc-policy* # set descr "This is an example chassis/FEX discovery
policy."
UCS-A /org/chassis-disc-policy* # set link-aggregation-pref port-channel
UCS-A /org/chassis-disc-policy* # set multicast-hw-hash enabled
UCS-A /org/chassis-disc-policy* # set qualifier ExampleQual
UCS-A /org/chassis-disc-policy* # commit-buffer
UCS-A /org/chassis-disc-policy #
```

What to Do Next

To customize fabric port channel connectivity for a specific chassis, configure the chassis connectivity policy.

Configuring the Chassis Connectivity Policy

Chassis Connectivity Policy

The chassis connectivity policy determines the whether a specific chassis is included in a fabric port channel after chassis discovery. This policy is helpful for users who want to configure one or more chassis differently from what is specified in the global chassis discovery policy. The chassis connectivity policy also allows for different connectivity modes per fabric interconnect, further expanding the level of control offered with regards to chassis connectivity.

By default, the chassis connectivity policy is set to global. This means that connectivity control is configured when the chassis is newly discovered, using the settings configured in the chassis discovery policy. Once the chassis is discovered, the chassis connectivity policy controls whether the connectivity control is set to none or port channel.

Note The chassis connectivity policy is created by Cisco UCS Manager only when the hardware configuration supports fabric port channels. At this time, only the 6200 series fabric interconnects and the 2200 series IOMs support this feature. For all other hardware combinations, Cisco UCS Manager does not create a chassis connectivity policy.

Configuring a Chassis Connectivity Policy

Changing the connectivity mode for a chassis might result in decreased VIF namespace.

Caution Changing the connectivity mode for a chassis results in chassis re-acknowledgement. Traffic might be disrupted during this time.

Procedure

	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 chassis-conn-policy** *chassis-num* [**a** | **b**}	Enters chassis connection policy organization mode for the specified chassis and fabric.
Step 3	UCS-A /org/chassis-conn-policy # **set link-aggregation-pref** {**global** | **none** | **port-channel**}	Specifies whether the links from the IOMs or FEXes to the fabric interconnects are grouped in a port channel. • **None—**No links are grouped in a port channel • **Port Channel—**All links from an IOM to a fabric interconnect are grouped in a port channel.

	Command or Action	Purpose
		• **Global—**The chassis inherits this configuration from the chassis discovery policy. This is the default value.
Step 4	UCS-A /org/chassis-conn-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to change the fabric port channel connectivity for two chassis. Chassis 6, fabric A is changed to port channel and chassis 12, fabric B is changed to discrete links:

```
UCS-A# scope org /
UCS-A /org # scope chassis-conn-policy 6 a
UCS-A /org/chassis-conn-policy # set link-aggregation-pref port-channel
UCS-A /org/chassis-conn-policy* # up
UCS-A /org* # scope chassis-conn-policy 12 b
UCS-A /org/chassis-conn-policy* # set link-aggregation-pref none
UCS-A /org/chassis-conn-policy* # commit-buffer
UCS-A /org/chassis-conn-policy #
```

Configuring the Rack Server Discovery Policy

Rack Server Discovery Policy

The rack server discovery policy determines how the system reacts when you add a new rack-mount server. Cisco UCS Manager uses the settings in the rack server discovery policy to determine whether any data on the hard disks are scrubbed and whether server discovery occurs immediately or needs to wait for explicit user acknowledgement.

Cisco UCS Manager cannot discover any rack-mount server that has not been correctly cabled and connected to the fabric interconnects. For information about how to integrate a supported Cisco UCS rack-mount server with Cisco UCS Manager, see the appropriate rack-mount server integration guide.

Configuring the Rack Server Discovery Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode. **Note** The rack server discovery policy can be accessed only from the root organization.
Step 2	UCS-A /org # **scope rackserver-disc-policy**	Enters organization rack server discovery policy mode.

	Command or Action	Purpose
Step 3	UCS-A /org/rackserver-disc-policy # **set action** {**immediate** | **user-acknowledged**}	Specifies the way the system reacts when you add a new rack server.
Step 4	UCS-A /org/rackserver-disc-policy # **set descr** *description*	(Optional) Provides a description for the rack server discovery policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 5	UCS-A /org/rackserver-disc-policy # **set scrub-policy** *scrub-pol-name*	Specifies the scrub policy that should run on a newly discovered rack server.
Step 6	UCS-A /org/rackserver-disc-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example scopes to the default rack server discovery policy, sets it to immediately discover new rack servers, provides a description for the policy, specifies a scrub policy called scrubpol1, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope rackserver-disc-policy
UCS-A /org/rackserver-disc-policy* # set action immediate
UCS-A /org/rackserver-disc-policy* # set descr "This is an example rackserver discovery
policy."
UCS-A /org/rackserver-disc-policy* # set scrub-policy scrubpol1
UCS-A /org/rackserver-disc-policy* # commit-buffer
UCS-A /org/rackserver-disc-policy #
```

Configuring the Aging Time for the MAC Address Table

Aging Time for the MAC Address Table

To efficiently switch packets between ports, the fabric interconnect maintains a MAC address table. It dynamically builds the MAC address table by using the MAC source address from the packets received and the associated port on which the packets were learned. The fabric interconnect uses an aging mechanism, defined by a configurable aging timer, to determine how long an entry remains in the MAC address table. If an address remains inactive for a specified number of seconds, it is removed from the MAC address table.

You can configure the amount of time (age) that a MAC address entry (MAC address and associated port) remains in the MAC address table.

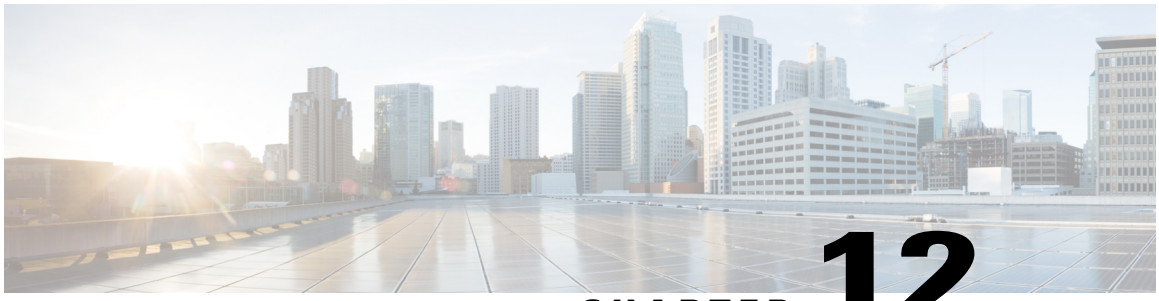
Configuring the Aging Time for the MAC Address Table

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **set mac-aging** {*dd hh mm ss* | **mode-default** | **never**}	Specifies the aging time for the MAC address table. Use the **mode-default** keyword to set the aging time to a default value dependent on the configured Ethernet switching mode. Use the **never** keyword to never remove MAC addresses from the table regardless of how long they have been idle.
Step 3	UCS-A /eth-uplink # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the aging time for the MAC address table to one day and 12 hours and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set mac-aging 01 12 00 00
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink #
```



CHAPTER 12

Managing Licenses

This chapter includes the following sections:

Licenses

Each Cisco UCS fabric interconnect comes with several port licenses that are factory installed and shipped with the hardware. You can purchase fabric interconnects fully licensed or partially licensed. You can also purchase additional licenses after delivery.

The following four new licenses are added for the 6300 Series FI and are only valid on the 6332 and 6332-16UP Fis.

- 40G_ETH_PORT_ACTIVATION_PKG – Licenses used for 40 GB Ethernet ports
- 40G_ETH_C_PORT_ACTIVATION_PKG – Licenses used for 40 GB Ethernet ports directly connected to rack servers (C-Direct)
- 10G_C_PORT_ACTIVATION_PKG – Licenses used for the first 16 10 GB unified ports on the 6332-16UP that are directly connected to rack servers (C-Direct)
- 10G_PORT_ACTIVATION_PKG – Licenses used for the first 16 10 GB unified ports on the 6332-16UP

Note The 10G_PORT_ACTIVATION_PKG and 10G_C_PORT_ACTIVATION_PKG licenses are only valid for the 6332-16UP FIs, and can only be installed on them.

At a minimum, each fabric interconnect ships with the following counted licenses pre-installed:

Fabric Interconnect	Default Base Licenses
Cisco UCS 6248 (unified ports)	For the 12 first enabled Ethernet ports and any Fibre Channel ports in the expansion module.
Cisco UCS 6296 (unified ports)	For the first 18 enabled Ethernet ports and any Fibre Channel ports in the expansion module.
Cisco UCS 6324	For 4 non-breakout ports only. The fifth port, which does not include a license, is further broken in to four 10 GB ports.
Cisco UCS 6332 16UP	For four 40 GB ports and eight 10 GB ports. **Note** The first 16 ports are 10 GB. The remaining are 40 GB.
Cisco UCS 6332	For eight 40 GB ports.

Port License Consumption

Port licenses are not bound to physical ports. When you disable a licensed port, that license is retained for use with the next enabled port. To use additional fixed ports, you must purchase and install licenses for those ports. All ports, regardless of their type (fibre, ethernet) consume licenses if they are enabled.

For breakout capable ports available in the 6332 and the 6332-16UP platforms, 40 GB licenses remain applied to the main port even if that port is a breakout port, and that port continues to consume only one 40 GB license.

Note The initial configuration of a port will enable it, and consume a license.

Important Licenses are not portable across product generations. Licenses purchased for 6200 series fabric interconnects cannot be used to enable ports on 6300 series fabric interconnects or vice-versa.

Each Cisco UCS 6324 Fabric Interconnect comes with a factory installed port license that is shipped with the hardware. This license is for the eight 40 GB unified ports, and can be used for any supported purpose. The C-direct port license is factory installed with a grace period, and can be used for Cisco UCS rack servers.

Grace Period

If you attempt to use a port that does not have an installed license, Cisco UCS initiates a 120 day grace period. The grace period is measured from the first use of the port without a license and is paused when a valid license file is installed. The amount of time used in the grace period is retained by the system.

Note Each physical port has its own grace period. Initiating the grace period on a single port does not initiate the grace period for all ports.

If a licensed port is unconfigured, that license is transferred to a port functioning within a grace period. If multiple ports are acting within grace periods, the license is moved to the port whose grace period is closest to expiring.

High Availability Configurations

To avoid inconsistencies during failover, we recommend that both fabric interconnects in the cluster have the same number of ports licensed. If symmetry is not maintained and failover occurs, Cisco UCS enables the missing licenses and initiates the grace period for each port being used on the failover node.

C-Direct Rack Licensing Support

Each Cisco UCS fabric interconnect is shipped with a default number of port licenses that are factory licensed and shipped with the hardware. C-direct support is only applicable on ports that are connected to the rack servers. The 10G_C_PORT_ACTIVATION_PKG and the 40G_ETH_C_PORT_ACTIVATION _PKG are added to the existing license package with all the same properties as the existing licensing feature. The **Subordinate Quantity** property is added to the 10G_PORT_ACTIVATION_PKG and 40G_ETH_PORT_ACTIVATION_PKG to track ports connected to rack servers.

The License Tab in the Cisco UCS Manager GUI displays the new license and the **Subordinate Quantity** for the license. You can also use the **show feature** and **show usage** commands under **scope license** to view the license feature, the vendor version type, and the grace period for each license.

Ports connected to rack servers can use existing 10G_PORT_ACTIVATION_PKG and 40G_ETH_PORT_ACTIVATION_PKG if the license is available or if the license is not in use. Otherwise, you must purchase a 10G_C_PORT_ACTIVATION_PKG and 40G_ETH_C_PORT_ACTIVATION to avoid the license grace period.

There is no change in the 10 GB ports. The 10G_PORT_ACTIVATION_PKG and 10G_C_PORT_ACTIVATION_PKG license packages include all of the same properties as the existing the ETH_PORT_ACTIVATION_PKG and the ETH_PORT_C_ACTIVATION_PKG license features.

Configuration and Restrictions

- The C-Direct rack licensing feature accounts for the rack server ports that are directly connected to the FI, but not to a CIMC port. The default quantity for the 10G_C_PORT_ACTIVATION_PKG and 40G_ETH_C_PORT_ACTIVATION_PKG is always 0.
- When a 40 GB port, or a breakout port under a 40 GB breakout port is enabled without any connections, this port is allotted a license under the 40G_ETH_PORT_ACTIVATION_PKG, if available. If this port is connected to a Direct-Connect rack server after a time lag, it triggers a complete re-allocation of licenses, then this port passes through one of the following license allocation scenarios occurs:

  When you enable a breakout port under a 40 GB breakout port, if that port is connected to a Direct-Connect rack server, and the 40G_C_PORT_ACTIVATION_PKG license files are installed on the FI, the following license allocation occurs:

- If no other ports under the breakout port are enabled, the parent 40 GB port is allotted a license under the 40G_C_PORT_ACTIVATION_PKG, and the used quantity is incremented for this instance.
- If other ports are enabled, and if at least one port is not connected to a Direct Connect rack server, even if the port is not being used, the parent 40 GB port is allotted a license under the 40G_ETH_PORT_ACTIVATION_PKG, and the used quantity is incremented for this instance.

- When you enable a breakout port under a 40 GB breakout port and that port is connected to a Direct-Connect rack server, and the 40G_C_PORT_ACTIVATION_PKG license files are not installed on the FI, the following license allocation occurs:
  - If no ports under the breakout port are enabled, the parent 40 GB port is allotted a license under the 40G_ETH_PORT_ACTIVATION_PKG. The subordinate quantity is increased if the licenses are available in the 40G_ETH_PORT_ACTIVATION_PKG. If the licenses are not available, the used quantity under this feature is increased and the entire port goes in to the grace period.
  - If other ports are enabled and at least one port is not connected to a Direct Connect rack server, even if the port is not being used, the parent 40 GB port is allotted a license under the 40G_ETH_PORT_ACTIVATION_PKG, and the used quantity is incremented for this instance.

Obtaining the Host ID for a Fabric Interconnect

The host ID is also known as the serial number.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope license**	Enters license mode.
Step 2	UCS-A /license # **show server-host-id**	Obtains the host ID or serial number for the fabric interconnect. **Tip** Use the entire host ID that displays after the equal (=) sign.

The following example obtains the host ID for a fabric interconnect:

```
UCS-A# scope license
UCS-A /license # show server-host-id
Server host id:
    Scope Host Id
    ----- -------
    A     VDH=SSI12121212
    B     VDH=SSI13131313
UCS-A /license #
```

What to Do Next

Obtain the required licenses from Cisco.

Obtaining a License

Note This process may change after the release of this document. If one or more of these steps no longer applies, contact your Cisco representative for information on how to obtain a license file.

Before You Begin

Obtain the following:

- Host ID or serial number for the fabric interconnect
- Claim certificate or other proof of purchase document for the fabric interconnect or expansion module

Procedure

Step 1 Obtain the product authorization key (PAK) from the claim certificate or other proof of purchase document.

Step 2 Locate the website URL in the claim certificate or proof of purchase document.

Step 3 Access the website URL for the fabric interconnect and enter the serial number and the PAK.
Cisco sends you the license file by email. The license file is digitally signed to authorize use on only the requested fabric interconnect. The requested features are also enabled once Cisco UCS Manager accesses the license file.

What to Do Next

Install the license on the fabric interconnect.

Installing a License

Note In a cluster setup, Cisco recommends that you download and install licenses to both fabric interconnects in matching pairs. An individual license is only downloaded to the fabric interconnect that is used to initiate the download.

Before You Begin

Obtain the required licenses from Cisco.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope license**	Enters license mode.

	Command or Action	Purpose
Step 2	UCS-A /license # **download license** *from-filesystem*	Downloads the license from its source location. For the *from-filesystem:* argument, use one of the following syntaxes: • **ftp:**// *server-ip-addr* • **scp:**// *username@server-ip-addr* • **sftp:**// *username@server-ip-addr* • **tftp:**// *server-ip-addr* **:** *port-num* You cannot have spaces anywhere in the path name or the file name. For example, `c:\Path\Folder_Name\License.lic` is a valid path, but `c:\Path\Folder Name\License.lic` is invalid due to the space in "`Folder Name`".
Step 3	UCS-A /license # **install file** *license_filename*	Installs the license.

The following example uses FTP to download and install a license:

```
UCS-A # scope license
UCS-A /license # download license ftp://192.168.10.10/license/port9.lic
UCS-A /license # install file port9.lic
UCS-A /license #
```

Viewing the Licenses Installed on a Fabric Interconnect

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope license**	Enters license mode.
Step 2	UCS-A /license # **show file** [*license_filename* | **detail**]	Displays the licenses installed on the fabric interconnect with the level of detail specified in the command.

The following example displays the full details for the licenses installed on a fabric interconnect:

```
UCS-A# scope license
UCS-A /license # show file detail

License file: UCSFEAT20100928112305377.lic
    Id: 1212121212121212
    Version: 1.0
    Scope: A
    State: Installed
    Features
```

```
    Feature Name: ETH_PORT_ACTIVATION_PKG
    Vendor: cisco
    Version: 1.0
    Quantity: 24
    Lines
        Line Id: 1
        Type: Increment
        Expiry Date: Never
        Pak:
        Quantity: 24
        Signature: B10101010101

License file: UCSFEAT20100928112332175.lic
    Id: 1313131313131313
    Version: 1.0
    Scope: B
    State: Installed
    Features
    Feature Name: ETH_PORT_ACTIVATION_PKG
    Vendor: cisco
    Version: 1.0
    Quantity: 24
    Lines
        Line Id: 1
        Type: Increment
        Expiry Date: Never
        Pak:
        Quantity: 24
        Signature: F302020202020

UCS-A /license #
```

Viewing License Usage for a Fabric Interconnect

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope license**	Enters license mode.
Step 2	UCS-A /license # **show usage**	Displays the license usage table for all license files installed on the fabric interconnect. This following are included: • **Feat Name** The name of the feature to which the license applies. • **Scope** The fabric associated with the license. • **Default** The default number of licenses provided for this Cisco UCS domain.

	Command or Action	Purpose
		• **Total Quant** The total number of licenses available. This value is the sum of the number of default licenses plus the number of purchased licenses. • **Used Quant** The number of licenses currently being used by the system. If this value exceeds the total number of licenses available, then some ports will stop functioning after their associated grace period expires. • **Subordinate Quant** C-Series Rack Servers that are currently being used by the system. • **State** The operational state of the license. • **Peer Count Comparison** The number of licenses on the peer fabric interconnect compared to this fabric interconnect. This can be one of the following: • **exceeds—**the peer fabric interconnect has more licenses installed than this fabric interconnect • **lacks—**the peer fabric interconnect has fewer licenses installed than this fabric interconnect • **matching—**the same number of licenses are installed on both fabric interconnects • **Grace Used** The amount of time (in seconds) used in the grace period. After the grace period ends, Cisco UCS sends alert messages until a new license is purchased.

The following examples display full details of the licenses installed on a fabric interconnect:

```
UCS-A# scope license
UCS-A /license # show usage
Feat Name                      Scope Default Total Quant Used Quant Subordinate Quant
State                Peer Count Comparison   Grace Used
------------------------------ ----- ------- ----------- ---------- --------------------
-------------------- ---------------------- ----------
ETH_PORT_ACTIVATION_PKG        A     20      48          12                           0
```

```
License Ok           Matching                          0
ETH_PORT_C_ACTIVATION_PKG      A     0       0          0                              0
Not Applicable       Matching                          0
ETH_PORT_ACTIVATION_PKG        B     20      48         11                             0
License Ok           Matching                          0
ETH_PORT_C_ACTIVATION_PKG      B     0       0          0                              0
Not Applicable       Matching                          0
UCS-A /license #

UCS-A# scope license
UCS-A /license # show feature

License feature:
    Name                       Vendor Version Type               Grace Period
    -------------------------- ------ ------- ------------------ ------------
    ETH_PORT_ACTIVATION_PKG    cisco  1.0     Counted                     120
    ETH_PORT_C_ACTIVATION_PKG cisco  1.0     Counted                     120
UCS-A /license #
```

Uninstalling a License

Note Permanent licenses cannot be uninstalled if they are in use. You can only uninstall a permanent license that is not in use. If you try to delete a permanent license that is being used, Cisco UCS Manager rejects the request and display an error message.

Before You Begin

Back up the Cisco UCS Manager configuration.

Procedure

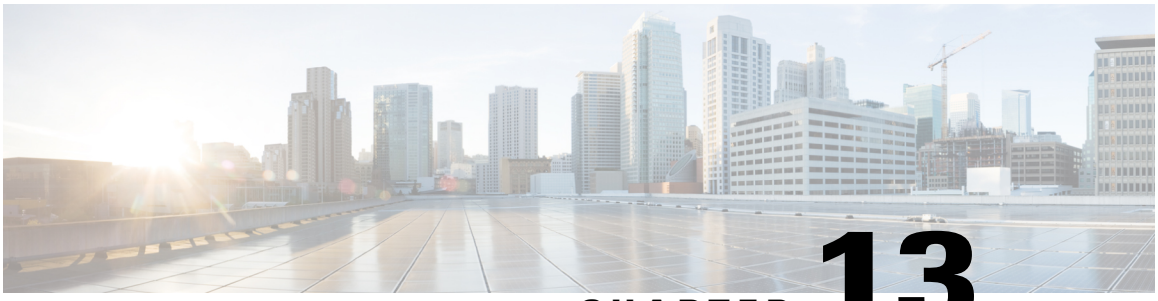
	Command or Action	Purpose
Step 1	UCS-A# **scope license**	Enters license mode.
Step 2	UCS-A /license # **clear file** *license-filename*	Uninstalls the specified license.

Cisco UCS Manager deactivates the license, removes the license from the list of licenses, and deletes the license from the fabric interconnect. The port is moved into unlicensed mode. In a cluster setup, you must uninstall the license from the other fabric interconnect.

The following example shows the uninstallation of port9.lic:

```
UCS-A # scope license
UCS-A /license # clear file port9.lic
Clearing license port9.lic:
SERVER this_host ANY
VENDOR cisco
INCREMENT ETH_PORT_ACTIVATION_PKG cisco 1.0 permanent 1 \
        VENDOR_STRING=<LIC_SOURCE>UCS_SWIFT</LIC_SOURCE><SKU>N10-L001=</SKU> \
        HOSTID=VDH=FLC12360025 \
        NOTICE="<LicFileID>20090519200954833</LicFileID><LicLineID>1</LicLineID> \
        <PAK></PAK>" SIGN=C01FAE4E87FA

Clearing license ........done
UCS-A /license #
```



CHAPTER 13

Managing Virtual Interfaces

This chapter includes the following sections:

Virtual Interfaces

In a blade server environment, the number of vNICs and vHBAs configurable for a service profile is determined by adapter capability and the amount of virtual interface (VIF) namespace available on the adapter. In Cisco UCS, portions of VIF namespace are allotted in chunks called VIFs. Depending on your hardware, the maxiumum number of VIFs are allocated on a predefined, per-port basis.
The maximum number of VIFs varies based on hardware capability and port connectivity. For each configured vNIC or vHBA, one or two VIFs are allocated. Stand-alone vNICs and vHBAs use one VIF and failover vNICs and vHBAs use two.

The following variables affect the number of VIFs available to a blade server, and therefore, how many vNICs and vHBAs you can configure for a service profile.

- Maximum number of VIFs supported on your fabric interconnect
- How the fabric interconnects are cabled
- If your fabric interconnect and IOM are configured in fabric port channel mode

For more information about the maximum number of VIFs supported by your hardware configuration, see the appropriate *Cisco UCS Configuration Limits for Cisco UCS Manager* for your software release.

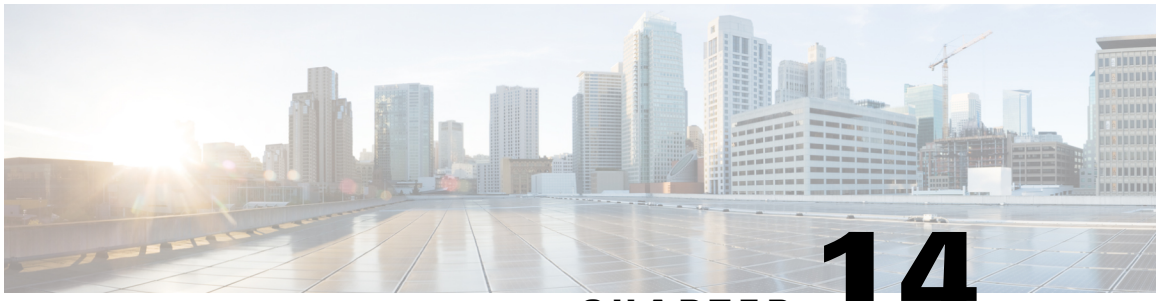
Virtual Interface Subscription Management and Error Handling

For fabric interconnects grouped in a port-channel, changes to the way you connect the fabric interconnect to the I/O module could result in a drastic change to the number of VIFs available to a blade server. To help you track the effect of these changes, Cisco UCS Manager maintains the following metrics:

- Maximum number of VIFs supported by hardware

- Connectivity type

If you change your configuration in a way that decreases the number of VIFs available to a blade, UCS Manager will display a warning and ask you if you want to proceed. This includes several scenarios, including times where adding or moving a connection decreases the number of VIFs.



CHAPTER 14

Registering Cisco UCS Domains with Cisco UCS Central

This chapter includes the following sections:

Registration of Cisco UCS Domains

You can have Cisco UCS Central manage some or all of the Cisco UCS domains in your data center.

If you want to have Cisco UCS Central manage a Cisco UCS domain, you need to register that domain. When you register, you need to choose which types of policies and other configurations, such as backups and firmware, will be managed by Cisco UCS Central and which by Cisco UCS Manager. You can have Cisco UCS Central manage the same types of policies and configurations for all registered Cisco UCS domains or you can choose to have different settings for each registered Cisco UCS domain.

Before you register a Cisco UCS domain with Cisco UCS Central, do the following:

- Configure an NTP server and the correct time zone in both Cisco UCS Manager and Cisco UCS Central to ensure that they are in sync. If the time and date in the Cisco UCS domain and Cisco UCS Central are out of sync, the registration might fail.
- Obtain the hostname or IP address of Cisco UCS Central
- Obtain the shared secret that you configured when you deployed Cisco UCS Central

Policy Resolution between Cisco UCS Manager and Cisco UCS Central

For each Cisco UCS domain that you register with Cisco UCS Central, you can choose which application will manage certain policies and configuration settings. This policy resolution does not have to be the same for every Cisco UCS domain that you register with the same Cisco UCS Central.

You have the following options for resolving these policies and configuration settings:

- **Local—**The policy or configuration is determined and managed by Cisco UCS Manager.
- **Global—**The policy or configuration is determined and managed by Cisco UCS Central.

Note The policy resolution options in Cisco UCS Central are not supported on all versions of Cisco UCS Manager. If your Cisco UCS Manager version is earlier than the earliest supported release, the policy resolution screen may display the value as global even if it is not applicable.

The following table contains a list of the policies and configuration settings that you can choose to have managed by either Cisco UCS Manager or Cisco UCS Central:

Name	Earliest Supported Release	Description
Infrastructure & Catalog Firmware	2.1(2)	Determines whether the Capability Catalog and infrastructure firmware policy are defined locally in Cisco UCS Manager or come from Cisco UCS Central.
Time Zone Management	2.1(2)	Determines whether the time zone and NTP server settings are defined locally in Cisco UCS Manager or comes from Cisco UCS Central.
Communication Services	2.1(2)	Determines whether HTTP, CIM XML, Telnet, SNMP, web session limits, and Management Interfaces Monitoring Policy settings are defined locally in Cisco UCS Manager or in Cisco UCS Central.
Global Fault Policy	2.1(2)	Determines whether the Global Fault Policy is defined locally in Cisco UCS Manager or in Cisco UCS Central.
User Management	2.1(2)	Determines whether authentication and native domains, LDAP, RADIUS, TACACS+, trusted points, locales, and user roles are defined locally in Cisco UCS Manager or in Cisco UCS Central.
DNS Management	2.1(2)	Determines whether DNS servers are defined locally in Cisco UCS Manager or in Cisco UCS Central.

Name	Earliest Supported Release	Description
Backup & Export Policies	2.1(2)	Determines whether the Full State Backup Policy and All Configuration Export Policy are defined locally in Cisco UCS Manager or in Cisco UCS Central.
Monitoring	2.1(2)	Determines whether Call Home, Syslog, and TFTP Core Exporter settings are defined locally in Cisco UCS Manager or in Cisco UCS Central.
SEL Policy	2.1(2)	Determines whether the SEL Policy is defined locally in Cisco UCS Manager or in Cisco UCS Central.
Power Allocation Policy	2.1(2)	Determines whether the Power Allocation Policy is defined locally in Cisco UCS Manager or in Cisco UCS Central.
Power Policy	2.1(2)	Determines whether the Power Policy is defined locally in Cisco UCS Manager or in Cisco UCS Central.
Equipment Policy	2.2(7)	Determines whether the Equipment Policy is defined locally in Cisco UCS Manager or in Cisco UCS Central.
Port Configuration	2.2(7)	Determines whether port configuration is defined locally in Cisco UCS Manager or in Cisco UCS Central.
Quality of Service (QoS) Configuration	2.2(7)	Determines whether QoS configuration is defined locally in Cisco UCS Manager or in Cisco UCS Central.

Registering a Cisco UCS Domain with Cisco UCS Central

Before You Begin

Configure an NTP server and the correct time zone in both Cisco UCS Manager and Cisco UCS Central to ensure that they are in sync. If the time and date in the Cisco UCS domain and Cisco UCS Central are out of sync, the registration might fail.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A/system # **create control-ep policy** *ucs-central*	Creates the policy required to register the Cisco UCS Domain with Cisco UCS Central. *ucs-central* can be the hostname or IP address of the virtual machine where Cisco UCS Central is deployed.

	Command or Action	Purpose
		Note If you use a hostname rather than an IPv4 or IPv6 address, you must configure a DNS server. If the Cisco UCS domain is not registered with Cisco UCS Central or DNS management is set to **local**, configure a DNS server in Cisco UCS Manager. If the Cisco UCS domain is registered with Cisco UCS Central and DNS management is set to **global**, configure a DNS server in Cisco UCS Central.
Step 3	Shared Secret for Registration: *shared-secret*	Enter the shared secret (or password) that was configured when Cisco UCS Central was deployed.
Step 4	UCS-A/system/control-ep # **commit-buffer**	Commits the transaction to the system configuration.

The following example registers a Cisco UCS Domain with a Cisco UCS Central system at IP address 209.165.200.233, and commits the transaction:

```
UCS-A# scope system
UCS-A /system # create control-ep policy 209.165.200.233
Shared Secret for Registration: S3cretW0rd!
UCS-A /system/control-ep* # commit-buffer
UCS-A /system/control-ep #
```

What to Do Next

Configure policy resolution between Cisco UCS Manager and Cisco UCS Central.

Configuring Policy Resolution between Cisco UCS Manager and Cisco UCS Central

Before You Begin

You must register the Cisco UCS Domain with Cisco UCS Central before you can configure policy resolution.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A/system # **scope control-ep policy**	Enters control-ep policy mode.
Step 3	UCS-A/system/control-ep # **set backup-policy-ctrl source** {**local** | **global**}	Determines whether the Full State Backup Policy and All Configuration Export Policy are defined locally or in Cisco UCS Central.

	Command or Action	Purpose
Step 4	UCS-A/system/control-ep # **set communication-policy-ctrl source** {**local** | **global**}	Determines whether HTTP, CIM XML, Telnet, SNMP, web session limits, and Management Interfaces Monitoring Policy settings are defined locally or in Cisco UCS Central.
Step 5	UCS-A/system/control-ep # **set datetime-policy-ctrl source** {**local** | **global**}	Determines whether the date and time is defined locally or comes from Cisco UCS Central.
Step 6	UCS-A/system/control-ep # **set dns-policy-ctrl source** {**local** | **global**}	Determines whether DNS servers are defined locally or in Cisco UCS Central.
Step 7	UCS-A/system/control-ep # **set fault-policy-ctrl source** {**local** | **global**}	Determines whether the Global Fault Policy is defined locally or in Cisco UCS Central.
Step 8	UCS-A/system/control-ep # **set infra-pack-ctrl source** {**local** | **global**}	Determines whether the Capability Catalog and infrastructure firmware policy are defined locally or come from Cisco UCS Central.
Step 9	UCS-A/system/control-ep # **set mep-policy-ctrl source** {**local** | **global**}	Determines whether managed endpoints are defined locally or in Cisco UCS Central.
Step 10	UCS-A/system/control-ep # **set monitoring-policy-ctrl source** {**local** | **global**}	Determines whether Call Home, Syslog, and TFTP Core Exporter settings are defined locally or in Cisco UCS Central.
Step 11	UCS-A/system/control-ep # **set powermgmt-policy-ctrl source** {**local** | **global**}	Determines whether the power management is defined locally or in Cisco UCS Central.
Step 12	UCS-A/system/control-ep # **set psu-policy-ctrl source** {**local** | **global**}	Determines whether power supply units are defined locally or in Cisco UCS Central.
Step 13	UCS-A/system/control-ep # **set security-policy-ctrl source** {**local** | **global**}	Determines whether authentication and native domains, LDAP, RADIUS, TACACS+, trusted points, locales, and user roles are defined locally or in Cisco UCS Central.
Step 14	UCS-A/system/control-ep # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures policy resolution for a Cisco UCS Domain that is registered with Cisco UCS Central and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope control-ep policy
UCS-A /system/control-ep* # set backup-policy-ctrl source global
UCS-A /system/control-ep* # set communication-policy-ctrl source local
UCS-A /system/control-ep* # set datetime-policy-ctrl source global
UCS-A /system/control-ep* # set dns-policy-ctrl source global
UCS-A /system/control-ep* # set fault-policy-ctrl source global
UCS-A /system/control-ep* # set infra-pack-ctrl source global
UCS-A /system/control-ep* # set mep-policy-ctrl source global
```

```
UCS-A /system/control-ep* # set monitoring-policy-ctrl source global
UCS-A /system/control-ep* # set powermgmt-policy-ctrl source global
UCS-A /system/control-ep* # set psu-policy-ctrl source local
UCS-A /system/control-ep* # set security-policy-ctrl source global
UCS-A /system/control-ep* # commit-buffer
UCS-A /system/control-ep #
```

Setting Cisco UCS Central Registration Properties in Cisco UCS Manager

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope control-ep policy**	Enters the registration policy.
Step 3	UCS-A /system/control-ep # **set cleanupmode** { | }	This can be one of the following: • **Localize Global—**When a Cisco UCS domain is unregistered, all global policies in the Cisco UCS domain will be localized to Cisco UCS Manager. The policies remain in the Cisco UCS domain, policy ownership is now local to Cisco UCS Manager, and Cisco UCS Manager admin users can make changes. **Note** If you reregister the Cisco UCS domain with Cisco UCS Central, there can be policy conflicts due to the policies existing both in Cisco UCS Central and in Cisco UCS Manager. Either delete the local policies, or set the local policies to global before you try to create and associate a global service profile. • **Deep Remove Global—**This option should only be used after careful consideration. When a Cisco UCS domain is unregistered, all global policies in the Cisco UCS domain are removed. If there are global service profiles, they will now refer to Cisco UCS Manager local default policies, and one of the following occurs: ◦ If there are local default policies present, the server will reboot. ◦ If there are no local default policies, the service profile association fails with a configuration error. **Note** The deep remove global cleanup mode does not remove global VSANs and VLANs when you unregister from Cisco UCS Central. Those must be removed manually if desired.

	Command or Action	Purpose
Step 4	UCS-A /system/control-ep # **set suspendstate on**	Sets the suspend state. If set automatically, the Cisco UCS domain is temporarily removed from Cisco UCS Central, and all global policies revert to their local counterparts. All service profiles maintain their current identities. However, global pools are no longer visible and cannot be accessible by new service profiles. To turn off suspend state, you need to acknowledge the situation.
Step 5	UCS-A /system/control-ep # **set ackstate acked**	Acknowledges that inconsistencies exist between Cisco UCS Manager and Cisco UCS Central and that you are still willing to reconnect the Cisco UCS domain with Cisco UCS Central. This automatically turns off suspend state.
Step 6	UCS-A /system/control-ep # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to change the Cisco UCS Central registration cleanup mode to deep-remove-global and commit the transaction:

```
UCS-A# scope system
UCS-A /system # scope control-ep policy
UCS-A /system/control-ep* # set cleanupmode deep-remove-global
UCS-A /system/control-ep* # commit-buffer
UCS-A /system/control-ep #
```

Unregistering a Cisco UCS Domain from Cisco UCS Central

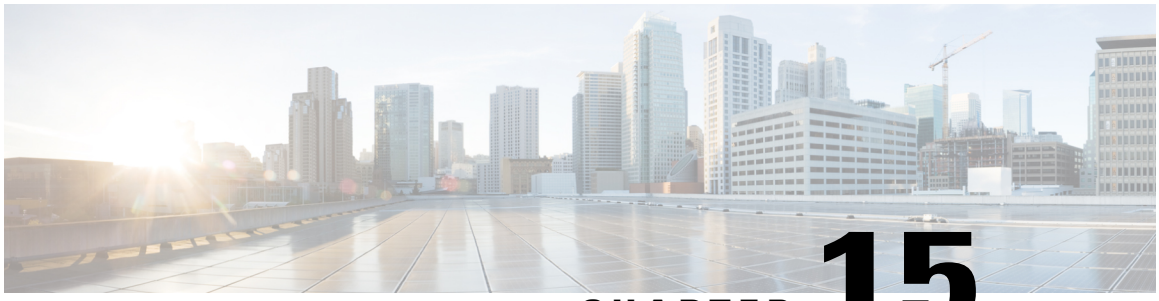
When you unregister a Cisco UCS domain from Cisco UCS Central, Cisco UCS Manager no longer receives updates to global policies.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A/system # **delete control-ep policy**	Deletes the policy and unregisters the Cisco UCS Domain from Cisco UCS Central.
Step 3	UCS-A/system # **commit-buffer**	Commits the transaction to the system configuration.

The following example unregisters a Cisco UCS Domain from Cisco UCS Central and commits the transaction:

```
UCS-A# scope system
UCS-A /system # delete control-ep policy
UCS-A /system* # commit-buffer
UCS-A /system #
```



CHAPTER 15

VLANs

Named VLANs

A named VLAN creates a connection to a specific external LAN. The VLAN isolates traffic to that external LAN, including broadcast traffic.

The name that you assign to a VLAN ID adds a layer of abstraction that allows you to globally update all servers associated with service profiles that use the named VLAN. You do not need to reconfigure the servers individually to maintain communication with the external LAN.

You can create more than one named VLAN with the same VLAN ID. For example, if servers that host business services for HR and Finance need to access the same external LAN, you can create VLANs named HR and Finance with the same VLAN ID. Then, if the network is reconfigured and Finance is assigned to a different LAN, you only have to change the VLAN ID for the named VLAN for Finance.

In a cluster configuration, you can configure a named VLAN to be accessible only to one fabric interconnect or to both fabric interconnects.

Guidelines for VLAN IDs

Important You cannot create VLANs with IDs from 4030 to 4047. This range of VLAN IDs is reserved.

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

VLAN 4048 is user configurable. However, Cisco UCS Manager uses VLAN 4048 for the following default values. If you want to assign 4048 to a VLAN, you must reconfigure these values:

- After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.
- After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049.

The VLAN name is case sensitive.

Private VLANs

A private VLAN (PVLAN) partitions the Ethernet broadcast domain of a VLAN into subdomains, and allows you to isolate some ports. Each subdomain in a PVLAN includes a primary VLAN and one or more secondary VLANs. All secondary VLANs in a PVLAN must share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another.

Isolated and Community VLANs

All secondary VLANs in a Cisco UCS domain can be Isolated or Community VLANs.

Note You cannot configure an isolated VLAN to use with a regular VLAN.

Ports on Isolated VLANs

Communications on an isolated VLAN can only use the associated port in the primary VLAN. These ports are isolated ports and are not configurable in Cisco UCS Manager. A primary VLAN can have only one isolated VLAN, but multiple isolated ports on the same isolated VLAN are allowed. These isolated ports cannot communicate with each other. The isolated ports can communicate only with a regular trunk port or promiscuous port that allows the isolated VLAN.

An isolated port is a host port that belongs to an isolated secondary VLAN. This port has complete isolation from other ports within the same private VLAN domain. PVLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports.

You can have more than one isolated port in a specified isolated VLAN. Each port is completely isolated from all other ports in the isolated VLAN.

Guidelines for Uplink Ports

When you create PVLANs, use the following guidelines:

- The uplink Ethernet port channel cannot be in promiscuous mode.
- Each primary VLAN can have only one isolated VLAN.
- VIFs on VNTAG adapters can have only one isolated VLAN.

Guidelines for VLAN IDs

Important You cannot create VLANs with IDs from 4030 to 4047. This range of VLAN IDs is reserved.

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

VLAN 4048 is user configurable. However, Cisco UCS Manager uses VLAN 4048 for the following default values. If you want to assign 4048 to a VLAN, you must reconfigure these values:

- After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.
- After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049.

The VLAN name is case sensitive.

VLAN Port Limitations

Cisco UCS Manager limits the number of VLAN port instances that you can configure under border and server domains on a fabric interconnect.

Types of Ports Included in the VLAN Port Count

The following types of ports are counted in the VLAN port calculation:

- Border uplink Ethernet ports
- Border uplink Ether-channel member ports
- FCoE ports in a SAN cloud

- Ethernet ports in a NAS cloud
- Static and dynamic vNICs created through service profiles
- VM vNICs created as part of a port profile in a hypervisor in hypervisor domain

Based on the number of VLANs configured for these ports, Cisco UCS Manager tracks the cumulative count of VLAN port instances and enforces the VLAN port limit during validation. Cisco UCS Manager reserves some pre-defined VLAN port resources for control traffic. These include management VLANs configured under HIF and NIF ports.

VLAN Port Limit Enforcement

Cisco UCS Manager validates VLAN port availability during the following operations:

- Configuring and unconfiguring border ports and border port channels
- Adding or removing VLANs from a cloud
- Configuring or unconfiguring SAN or NAS ports
- Associating or disassociating service profiles that contain configuration changes
- Configuring or unconfiguring VLANs under vNICs or vHBAs
- Receiving creation or deletion notifications from a VMWare vNIC and from an ESX hypervisor

Note This is outside the control of the Cisco UCS Manager.

- Fabric interconnect reboot
- Cisco UCS Manager upgrade or downgrade

Cisco UCS Manager strictly enforces the VLAN port limit on service profile operations. If Cisco UCS Manager detects that the VLAN port limit is exceeded, the service profile configuration fails during deployment.

Exceeding the VLAN port count in a border domain is less disruptive. When the VLAN port count is exceeded in a border domain Cisco UCS Manager changes the allocation status to Exceeded. To change the status back to **Available**, complete one of the following actions:

- Unconfigure one or more border ports
- Remove VLANs from the LAN cloud
- Unconfigure one or more vNICs or vHBAs

Configuring Named VLANs

Creating a Named VLAN Accessible to Both Fabric Interconnects (Uplink Ethernet Mode)

Important You cannot create VLANs with IDs from 4030 to 4047. This range of VLAN IDs is reserved.

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink VLAN mode. The VLAN name is case sensitive.
Step 3	UCS-A /eth-uplink/fabric/vlan # **set sharing** {**isolated** | **none** | **primary**}	Sets the sharing for the specified VLAN. This can be one of the following: • **isolated —**This is a secondary VLAN associated with a primary VLAN. This VLAN is private. • **none —**This VLAN does not have any secondary or private VLANs. • **primary —**This VLAN can have one or more secondary VLANs.
Step 4	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VLAN for both fabric interconnects, names the VLAN accounting, assigns the VLAN ID 2112, sets the sharing to none, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan accounting 2112
UCS-A /eth-uplink/vlan* # set sharing none
```

```
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan #
```

Creating a Named VLAN Accessible to Both Fabric Interconnects (Ethernet Storage Mode)

Important You cannot create VLANs with IDs from 4030 to 4047. This range of VLAN IDs is reserved.

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet storage VLAN mode. The VLAN name is case sensitive.
Step 3	UCS-A /eth-storage/vlan # **create member-port** {**a** | **b**} *slot-id port-id*	Creates a member port for the specified VLAN on the specified fabric.
Step 4	UCS-A /eth-storage/vlan/member-port # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VLAN for both fabric interconnects, names the VLAN accounting, assigns the VLAN ID 2112, creates a member port on slot 2, port 20, and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # create vlan accounting 2112
UCS-A /eth-storage/vlan* # create member-port a 2 20
UCS-A /eth-storage/vlan/member-port* # commit-buffer
UCS-A /eth-storage/vlan/member-port #
```

Creating a Named VLAN Accessible to One Fabric Interconnect (Uplink Ethernet Mode)

Important You cannot create VLANs with IDs from 4030 to 4047. This range of VLAN IDs is reserved.

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A /eth-uplink/fabric # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink fabric interconnect VLAN mode. The VLAN name is case sensitive.
Step 4	UCS-A /eth-uplink/fabric/vlan # **set sharing** {**isolated** | **none** | **primary**}	Sets the sharing for the specified VLAN. This can be one of the following: • **isolated —**This is a secondary VLAN associated with a primary VLAN. This VLAN is private. • **none —**This VLAN does not have any secondary or private VLANs. • **primary —**This VLAN can have one or more secondary VLANs.
Step 5	UCS-A /eth-uplink/fabric/vlan # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VLAN for fabric interconnect A, names the VLAN finance, assigns the VLAN ID 3955, sets the sharing to none, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # create vlan finance 3955
```

```
UCS-A /eth-uplink/fabric/vlan* # set sharing none
UCS-A /eth-uplink/fabric/vlan* # commit-buffer
UCS-A /eth-uplink/fabric/vlan #
```

Creating a Named VLAN Accessible to One Fabric Interconnect (Ethernet Storage Mode)

Important You cannot create VLANs with IDs from 4030 to 4047. This range of VLAN IDs is reserved.

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b**}	Enters Ethernet storage fabric interconnect mode for the specified fabric interconnect.
Step 3	UCS-A /eth-storage/fabric # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet storage fabric interconnect VLAN mode. The VLAN name is case sensitive.
Step 4	UCS-A /eth-storage/vlan # **create member-port** {**a** | **b**} *slot-id port-id*	Creates a member port for the specified VLAN on the specified fabric.
Step 5	UCS-A /eth-storage/fabric/vlan/member-port # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VLAN for fabric interconnect A, names the VLAN finance, assigns the VLAN ID 3955, creates a member port on slot 2, port 20, and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # create vlan finance 3955
UCS-A /eth-storage/fabric/vlan* # create member-port a 2 20
UCS-A /eth-storage/fabric/vlan/member-port* # commit-buffer
UCS-A /eth-storage/fabric/vlan/member-port #
```

Deleting a Named VLAN

If Cisco UCS Manager includes a named VLAN with the same VLAN ID as the one you delete, the VLAN is not removed from the fabric interconnect configuration until all named VLANs with that ID are deleted.

If you are deleting a private primary VLAN, ensure that you reassign the secondary VLANs to another working primary VLAN.

Before You Begin

Before you delete a VLAN from a fabric interconnect, ensure that the VLAN was removed from all vNICs and vNIC templates.

Note If you delete a VLAN that is assigned to a vNIC or vNIC template, the vNIC might allow that VLAN to flap.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	(Optional) Enters Ethernet uplink fabric mode. Use this command when you want to delete a named VLAN only from the specified fabric (a or b).
Step 3	UCS-A /eth-uplink # **delete vlan** *vlan-name*	Deletes the specified named VLAN.
Step 4	UCS-A /eth-uplink # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a named VLAN accessible to both fabric interconnects and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # delete vlan accounting
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink #
```

The following example deletes a named VLAN accessible to one fabric interconnect and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # delete vlan finance
UCS-A /eth-uplink/fabric* # commit-buffer
UCS-A /eth-uplink/fabric #
```

Configuring Private VLANs

Creating a Primary VLAN for a Private VLAN (Accessible to Both Fabric Interconnects)

Important You cannot create VLANs with IDs from 4030 to 4047. This range of VLAN IDs is reserved.

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink VLAN mode. The VLAN name is case sensitive.
Step 3	UCS-A /eth-uplink/vlan # **set sharing primary**	Sets the VLAN as the primary VLAN.
Step 4	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VLAN for both fabric interconnects, names the VLAN accounting, assigns the VLAN ID 2112, makes this VLAN the primary VLAN, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan accounting 2112
UCS-A /eth-uplink/vlan* # set sharing primary
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan #
```

Creating a Primary VLAN for a Private VLAN (Accessible to One Fabric Interconnect)

Important You cannot create VLANs with IDs from 4030 to 4047. This range of VLAN IDs is reserved.

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric interconnect mode for the specified fabric interconnect.
Step 3	UCS-A /eth-uplink/fabric # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink fabric interconnect VLAN mode. The VLAN name is case sensitive.
Step 4	UCS-A /eth-uplink/fabric/vlan # **set sharing primary**	Sets the VLAN as the primary VLAN.
Step 5	UCS-A /eth-uplink/fabric/vlan # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VLAN for fabric interconnect A, names the VLAN finance, assigns the VLAN ID 3955, makes this VLAN the primary VLAN, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # create vlan finance 3955
UCS-A /eth-uplink/fabric/vlan* # set sharing primary
UCS-A /eth-uplink/fabric/vlan* # commit-buffer
UCS-A /eth-uplink/fabric/vlan #
```

Creating a Secondary VLAN for a Private VLAN (Accessible to Both Fabric Interconnects)

Important You cannot create VLANs with IDs from 4030 to 4047. This range of VLAN IDs is reserved.

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink VLAN mode. The VLAN name is case sensitive.
Step 3	UCS-A /eth-uplink/vlan # **set sharing isolated**	Sets the VLAN as the secondary VLAN.
Step 4	UCS-A /eth-uplink/vlan # **set pubnwname** *primary-vlan-name*	Specifies the primary VLAN to be associated with this secondary VLAN.
Step 5	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VLAN for both fabric interconnects, names the VLAN accounting, assigns the VLAN ID 2112, makes this VLAN the secondary VLAN, associates the secondary VLAN with the primary VLAN, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan accounting 2112
UCS-A /eth-uplink/vlan* # set sharing isolated
UCS-A /eth-uplink/vlan* # set pubnwname pvlan1000
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan #
```

Creating a Secondary VLAN for a Private VLAN (Accessible to One Fabric Interconnect)

Important You cannot create VLANs with IDs from 4030 to 4047. This range of VLAN IDs is reserved.

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A /eth-uplink/fabric # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink fabric interconnect VLAN mode. The VLAN name is case sensitive.
Step 4	UCS-A /eth-uplink/vlan # **set sharing isolated**	Sets the VLAN as the secondary VLAN.
Step 5	UCS-A /eth-uplink/vlan # **set pubnwname** *primary-vlan-name*	Specifies the primary VLAN to be associated with this secondary VLAN.
Step 6	UCS-A /eth-uplink/fabric/vlan/member-port # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VLAN for fabric interconnect A, names the VLAN finance, assigns the VLAN ID 3955, makes this VLAN the secondary VLAN, associates the secondary VLAN with the primary VLAN, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # create vlan finance 3955
UCS-A /eth-uplink/fabric/vlan* # set sharing isolated
UCS-A /eth-uplink/fabric/vlan* # set pubnwname pvlan1000
UCS-A /eth-uplink/fabric/vlan* # commit-buffer
UCS-A /eth-uplink/fabric/vlan #
```

Community VLANs

Cisco UCS Manager supports Community VLANs in UCS Fabric Interconnects. Community ports communicate with each other and with promiscuous ports. Community ports have Layer 2 isolation from all other ports in other communities, or isolated ports within the PVLAN. Broadcasts are transmitted between the community ports associated with the PVLAN only and the other promiscuous ports. A promiscuous port can communicate with all interfaces, including the isolated and community ports within a PVLAN.

Creating a Community VLAN

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**.	Enters Ethernet uplink mode.
Step 2	UCS-A# /eth-uplink/ # **create vlan** *ID* .	Create a VLAN with the specified VLAN ID.
Step 3	UCS-A# /eth-uplink/ vlan # **set sharing** *Type* .	Specifies the vlan type.
Step 4	UCS-A# /eth-uplink/ vlan # **set pubnwname** *Name* .	Specifies the primary vlan association.
Step 5	UCS-A# /eth-uplink/ vlan # **commit-buffer**.	Commits the transaction to the system configuration.

The following example shows how to create a Community VLAN:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan vlan203 203
UCS-A /eth-uplink/vlan* # set sharing community
UCS-A /eth-uplink/vlan* # set pubname vlan200
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan* # exit
UCS-A /vlan-group #
```

Allowing Community VLANs on vNICs

Procedure

	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 # **scope service-profile** *profile-name*	Commits the transaction to the system configuration.
Step 3	UCS-A /org/service-profile # **scope vnic** *vnic-name*	Enters command mode for the specified vNIC.
Step 4	UCS-A /org/service-profile/vnic # **create eth-if** *community-vlan-name*	Allows the community VLAN to access the specified vNIC.
Step 5	UCS-A /org/service-profile/vnic # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to assign the community VLAN cVLAN101 to the vNIC vnic_1 and commits the transaction.

```
UCS-A# scope org /
UCS-A /org # scope service-profile GSP1
UCS-A /org/service-profile # scope vnic vnic_1
UCS-A /org/service-profile/vnic # create eth-if cVLAN101
UCS-A /org/service-profile/vnic* # commit-buffer
```

Allowing PVLAN on Promiscuous Access or Trunk Port

For a promiscuous access port, the isolated and community VLANs must be associated to the same primary VLAN.

For a promiscuous trunk port, isolated and community VLANs belonging to different primary VLANs are allowed, as well as regular VLANs.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope vlan** *iso-vlan-name*	Enters the specified isolated VLAN.
Step 3	UCS-A /eth-storage/vlan # **create member-port** *fabric slot- num port- num*	Creates the member port for the specified fabric, assigns the slot number and port number, and enters member port configuration scope.
Step 4	UCS-A /eth-storage/vlan/member-port # **exit**	Returns to VLAN mode.
Step 5	UCS-A /eth-storage/vlan # **exit**	Returns to Ethernet storage mode.
Step 6	UCS-A /eth-storage # **scope vlan** *comm-vlan-name*	Enters the specified community VLAN.

	Command or Action	Purpose
Step 7	UCS-A /eth-storage/vlan # **create member-port** *fabric slot- num port- num*	Creates the member port for the specified fabric, assigns the slot number and port number, and enters member port configuration scope.
Step 8	UCS-A /eth-storage/vlan/member-port # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to assign the isolated and community associated with the same primary VLAN to the same appliance port and commits the transaction.

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope vlan isovlan501
UCS-A /eth-storage/vlan # create member-port a 1 2
UCS-A /eth-storage/vlan/member-port* # exit
UCS-A /eth-storage/vlan* # exit
UCS-A /eth-storage* # scope vlan cvlan502
UCS-A /eth-storage/vlan* # create member-port a 1 2
UCS-A /eth-storage/vlan/member-port* # commit-buffer
UCS-A /eth-storage/vlan/member-port #
```

Deleting a Community VLAN

If Cisco UCS Manager includes a named VLAN with the same VLAN ID as the one you delete, the VLAN is not removed from the fabric interconnect configuration until all named VLANs with that ID are deleted.

If you are deleting a private primary VLAN, ensure that you reassign the secondary VLANs to another working primary VLAN.

Before You Begin

Before you delete a VLAN from a fabric interconnect, ensure that the VLAN was removed from all vNICs and vNIC templates.

Note If you delete a VLAN that is assigned to a vNIC or vNIC template, the vNIC might allow that VLAN to flap.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric**{**a** | **b**}	(Optional) Enters Ethernet uplink fabric mode. Use this command when you want to delete a named VLAN only from the specified fabric (a or b).

	Command or Action	Purpose
Step 3	UCS-A /eth-uplink # **delete community vlan** *vlan-name*	Deletes the specified community VLAN.
Step 4	UCS-A /eth-uplink # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a Community VLAN and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # delete commnity vlan vlan203
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink #
```

Viewing the VLAN Port Count

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified fabric interconnect.
Step 2	UCS-A /fabric-interconnect # **show vlan-port-count**	Displays the VLAN port count.

The following example displays the VLAN port count for fabric interconnect A:

```
UCS-A# scope fabric-interconnect a
UCS-A /fabric-interconnect # show vlan-port-count

VLAN-Port Count:
VLAN-Port Limit     Access VLAN-Port Count     Border VLAN-Port Count     Alloc Status
----------              ---------------           ----------------          ----------
6000                         3                         0                     Available
```

VLAN Port Count Optimization

VLAN port count optimization enables mapping the state of multiple VLANs into a single internal state. When you enable the VLAN port count optimization, Cisco UCS Manager logically groups VLANs based on the port VLAN membership. This grouping increases the port VLAN count limit. VLAN port count optimization also compresses the VLAN state and reduces the CPU load on the fabric interconnect. This reduction in the CPU load enables you to deploy more VLANs over more vNICs. Optimizing VLAN port count does not change any of the existing VLAN configuration on the vNICs.

VLAN port count optimization is disabled by default. You can enable or disable the option based on your requirements.

Important

- Enabling VLAN port count optimization increases the number of available VLAN ports for use. If the port VLAN count exceeds the maximum number of VLANs in a non-optimized state, you cannot disable the VLAN port count optimization.
- VLAN port count optimization is not supported in Cisco UCS 6100 Series fabric interconnect.

Enabling Port VLAN Count Optimization

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink# **set vlan-port-count-optimization enable**	Enables the vlan for port VLAN count optimization.
Step 3	UCS-A /eth-uplink* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to enable VLAN port count optimization:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set vlan-port-count-optimization enable
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink#
```

Disabling Port VLAN Count Optimization

If you have more Port VLAN count than that is allowed in the non port VLAN port count optimization state, you cannot disable the optimization.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink# **set vlan-port-count-optimization disable**	Disables the port VLAN count optimization.
Step 3	UCS-A /eth-uplink # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to disable VLAN port count optimization:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set vlan-port-count-optimization disable
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink#
```

Viewing the Port VLAN Count Optimization Groups

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink# **show vlan-port-count-optimization group**	Displays the vlan for port VLAN count optimization groups.

The following example shows port VLAN count optimization group in fabric a and b:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # show vlan-port-count-optimization group
VLAN Port Count Optimization Group:
    Fabric ID  Group ID   VLAN ID
    --------   -------    -------
    A          5          6
    A          5          7
    A          5          8
    B          10         100
    B          10         101
```

VLAN Groups

VLAN groups allow you to group VLANs on Ethernet uplink ports, by function or by VLANs that belong to a specific network. You can define VLAN membership and apply the membership to multiple Ethernet uplink ports on the fabric interconnect.

Note Cisco UCS Manager supports a maximum of 200 VLAN Groups. If Cisco UCS Manager determines that you create more than 200 VLAN groups, the system disables VLAN compression.

You can configure inband and out-of-band (OOB) VLAN groups to use to access the Cisco Integrated Management Interface (CIMC) on blade and rack servers. Cisco UCS Manager supports OOB IPv4 and inband IPv4 and IPv6 VLAN groups for use with the uplink interfaces or uplink port channels.

After you assign a VLAN to a VLAN group, any changes to the VLAN group are applied to all Ethernet uplink ports that are configured with the VLAN group. The VLAN group also enables you to identify VLAN overlaps between disjoint VLANs.

You can configure uplink ports under a VLAN group. When you configure an uplink port for a VLAN group, that uplink port will support all the VLANs that are part of the associated VLAN groups and individual VLANs that are associated with the uplink using LAN Uplinks Manager, if any. Further, any uplink that is not selected for association with that VLAN group will stop supporting the VLANs that are part of that VLAN group.

You can create VLAN groups from the **LAN Cloud** or from the **LAN Uplinks Manager**.

Creating a VLAN Group

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**.	Enters Ethernet uplink mode. The VLAN Group name is case sensitive.
Step 2	UCS-A# /eth-uplink/ #**create vlan-group***Name* .	Create a VLAN group with the specified name. This name can be between 1 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	UCS-A# /eth-uplink/ vlan-group#**create member-vlan***ID* .	Adds the specified VLANs to the created VLAN group.
Step 4	UCS-A# /eth-uplink/vlan-group #**create member-port** [member-port-channel] .	Assigns the uplink Ethernet ports to the VLAN group.
Step 5	UCS-A#/vlan-group* # **commit-buffer**.	Commits the transaction to the system configuration.

The following example shows how to create a VLAN group:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan-group eng
UCS-A /eth-uplink/vlan-group* # create member-vlan 3
UCS-A /eth-uplink/vlan-group* # commit-buffer
UCS-A /vlan-group #
```

Creating an Inband VLAN Group

Configure inband VLAN groups to provide access to remote users via an inband service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth uplink**	Enters Ethernet uplink configuration mode.
Step 2	UCS-A /eth-uplink # **create vlan-group** *inband-vlan-name*	Creates a VLAN group with the specified name and enters VLAN group configuration mode.

	Command or Action	Purpose
Step 3	UCS-A /eth-uplink/vlan-group # **create member-vlan***inband-vlan-nameinband-vlan-id*	Adds the specified VLAN to the VLAN group and enters VLAN group member configuration mode.
Step 4	UCS-A /eth-uplink/vlan-group/member-vlan # **exit**	Exits VLAN group member configuration mode.
Step 5	UCS-A /eth-uplink/vlan-group # **create member-port***fabricslot-numport-num*	Creates the member port for the specified fabric, assigns the slot number, and port number and enters member port configuration.
Step 6	UCS-A /eth-uplink/vlan-group/member-port # **commit-buffer**	Commits the transaction.

The example below creates a VLAN group named inband-vlan-group, creates a member of the group named Inband_VLAN and assigns VLAN ID 888, creates member ports for Fabric A and Fabric B, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan-group inband-vlan-group
UCS-A /eth-uplink/vlan-group* # create member-vlan Inband_VLAN 888
UCS-A /eth-uplink/vlan-group/member-vlan* # exit
UCS-A /eth-uplink/vlan-group* # create member-port a 1 23
UCS-A /eth-uplink/vlan-group/member-port* # exit
UCS-A /eth-uplink/vlan-group* # create member-port b 1 23
UCS-A /eth-uplink/vlan-group/member-port* # commit-buffer
UCS-A /eth-uplink/vlan-group/member-port # exit
UCS-A /eth-uplink/vlan-group # exit
```

What to Do Next

Assign the inband VLAN group to an inband service profile.

Deleting a VLAN Group

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**.	Enters Ethernet uplink mode.
Step 2	UCS-A# /eth-uplink/ #**delete vlan-group***Name* .	Deletes the specified VLAN group.
Step 3	UCS-A#/eth-uplink* # **commit-buffer**.	Commits the transaction to the system configuration.

The following example shows how to delete a VLAN group:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # delete vlan-group eng
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink #
```

Viewing VLAN Groups

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters Cisco UCS Manager organization.
Step 2	UCS-A /org # **show vlan-group**	Displays the available groups in the organization.

The following example shows the available VLAN groups in the root org:

```
UCS-A# scope org
UCS-A# /org/# show vlan-group
VLAN Group:
    Name
    ----
    eng
    hr
    finance
```

VLAN Permissions

VLAN permissions restrict access to VLANs based on specified organizations and on the service profile organizations to which the VLANs belong. VLAN permissions also restrict the set of VLANs that you can assign to service profile vNICs. VLAN permissions is an optional feature and is disabled by default. You can enable or disable the feature based on your requirements. If you disable the feature, all of the VLANs are globally accessible to all organizations.

Note If you enable the org permission in **LAN** > **LAN Cloud** > **Global Policies** > **Org Permissions**, when you create a VLAN, the **Permitted Orgs for VLAN(s)** option displays in the **Create VLANs** dialog box. If you do not enable the **Org Permissions**, the **Permitted Orgs for VLAN(s)** option does not display.

Enabling the org permission allows you to specify the organizations for the VLAN. When you specify the organizations, the VLAN becomes available to that specific organization and all of the sub organizations below the structure. Users from other organizations cannot access this VLAN. You can also modify the VLAN permission anytime based on changes to your VLAN access requirements.

Caution When you assign the VLAN org permission to an organization at the root level, all sub organizations can access the VLANs. After assigning the org permission at the root level, and you change the permission for a VLAN that belongs to a sub organization, that VLAN becomes unavailable to the root level organization.

Creating VLAN Permissions

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**.	Enters the Cisco UCS Manager VLAN organization.
Step 2	UCS-A# /org/ #**create vlan-permit***VLAN permission name*.	Creates the specified VLAN permission and assigns VLAN access permission to the organization.
Step 3	UCS-A#/org* # **commit-buffer**.	Commits the transaction to the system configuration.

The following example shows how to create a VLAN permission for an organization:

```
UCS-A# scope org
UCS-A /org # create vlan-permit dev
UCS-A /org* # commit-buffer
UCS-A /org #
```

Deleting a VLAN Permission

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**.	Enters the Cisco UCS Manager VLAN organization.
Step 2	UCS-A# /org/ #**delete vlan-permit***VLAN permission name*.	Deletes the access permission to the VLAN.
Step 3	UCS-A#/org* # **commit-buffer**.	Commits the transaction to the system configuration.

The following example shows how to delete a VLAN permission from an organization:

```
UCS-A# scope org
UCS-A /org # delete vlan-permit dev
UCS-A /org* # commit-buffer
UCS-A /org #
```

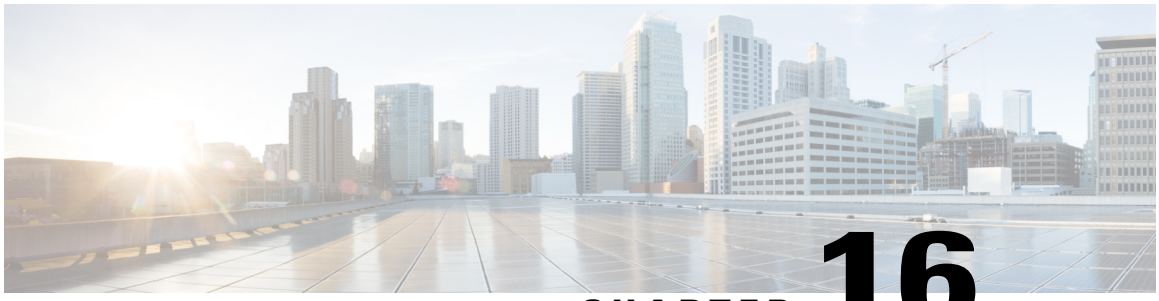
Viewing VLAN Permissions

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters Cisco UCS Manager organization.
Step 2	UCS-A /org # **show vlan-permit**	Displays the available permissions in the organization.

The following example shows the VLAN groups that have permission to access this VLAN:

```
UCS-A# scope org
UCS-A# /org/# show vlan-permit
VLAN Group:
    Name
    ----
    eng
    hr
    finance
```



CHAPTER 16

Configuring LAN Pin Groups

This chapter includes the following sections:

LAN Pin Groups

Cisco UCS uses LAN pin groups to pin Ethernet traffic from a vNIC on a server to an uplink Ethernet port or port channel on the fabric interconnect. You can use this pinning to manage the distribution of traffic from the servers.

To configure pinning for a server, you must include the LAN pin group in a vNIC policy. The vNIC policy is then included in the service profile assigned to that server. All traffic from the vNIC travels through the I/O module to the specified uplink Ethernet port.

Note If you do not assign a pin group to a server interface through a vNIC policy, Cisco UCS Manager chooses an uplink Ethernet port or port channel for traffic from that server interface dynamically. This choice is not permanent. A different uplink Ethernet port or port channel may be used for traffic from that server interface after an interface flap or a server reboot.

If an uplink is part of a LAN pin group, the uplink is not necessarily reserved for only that LAN pin group. Other vNIC's policies that do not specify a LAN pin group can use the uplink as a dynamic uplink.

Configuring a LAN Pin Group

In a system with two fabric interconnects, you can associate the pin group with only one fabric interconnect or with both fabric interconnects.

Before You Begin

Configure the ports and port channels with which you want to configure the pin group. You can only include ports and port channels configured as uplink ports in a LAN pin group.

Procedure

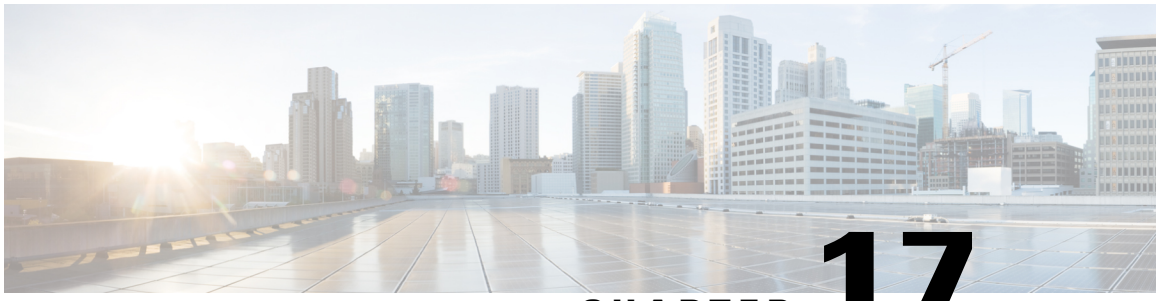
	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **create pin-group** *pin-group-name*	Creates an Ethernet (LAN) pin group with the specified name, and enters Ethernet uplink pin group mode.
Step 3	UCS-A /eth-uplink/pin-group # **set descr** *description*	(Optional) Provides a description for the pin group. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /eth-uplink/pin-group # **set target** {**a** | **b** | **dual**} {**port** *slot-num* / *port-num* | **port-channel** *port-num*}	(Optional) Sets the Ethernet pin target to the specified fabric and port, or fabric and port channel.
Step 5	UCS-A /eth-uplink/pin-group # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a LAN pin group named pingroup54 on fabric A, provides a description for the pin group, sets the pin group target to port channel 28, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create pin-group pingroup54
UCS-A /eth-uplink/pin-group* # set descr "This is my pin group #54"
UCS-A /eth-uplink/pin-group* # set target a port-channel 28
UCS-A /eth-uplink/pin-group* # commit-buffer
UCS-A /eth-uplink/pin-group #
```

What to Do Next

Include the pin group in a vNIC template.



CHAPTER 17

Configuring MAC Pools

This chapter includes the following sections:

MAC Pools

A MAC pool is a collection of network identities, or MAC addresses, that are unique in their Layer 2 environment and are available to be assigned to vNICs on a server. If you use MAC pools in service profiles, you do not have to manually configure the MAC addresses to be used by the server associated with the service profile.

In a system that implements multitenancy, you can use the organizational hierarchy to ensure that MAC pools can be used only by specific applications or business services. Cisco UCS uses the name resolution policy to assign MAC addresses from the pool.

To assign a MAC address to a server, you must include the MAC pool in a vNIC policy. The vNIC policy is then included in the service profile assigned to that server.

You can specify your own MAC addresses or use a group of MAC addresses provided by Cisco.

Creating a MAC Pool

Procedure

	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 mac-pool** *mac-pool-name*	Creates a MAC pool with the specified name, and enters organization MAC pool mode.

	Command or Action	Purpose
		This name can be between 1 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	UCS-A /org/mac-pool # **set descr** *description*	(Optional) Provides a description for the MAC pool. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/mac-pool # **set assignmentorder** {**default** | **sequential**}	This can be one of the following: • **default—**Cisco UCS Manager selects a random identity from the pool. • **sequential—**Cisco UCS Manager selects the lowest available identity from the pool.
Step 5	UCS-A /org/mac-pool # **create block** *first-mac-addr last-mac-addr*	Creates a block (range) of MAC addresses, and enters organization MAC pool block mode. You must specify the first and last MAC addresses in the address range using the form *nn***:***nn***:***nn***:***nn***:***nn***:***nn*, with the addresses separated by a space. **Note** A MAC pool can contain more than one MAC address block. To create multiple MAC address blocks, you must enter multiple **create block** commands from organization MAC pool mode.
Step 6	UCS-A /org/mac-pool # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a MAC pool named pool37, provide a description for the pool, define a MAC address block by specifying the first and last MAC addresses in the block, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create mac-pool pool37
UCS-A /org/mac-pool* # set descr "This is my MAC pool"
UCS-A /org/mac-pool* # create block 00:A0:D7:42:00:01 00:A0:D7:42:01:00
UCS-A /org/mac-pool/block* # commit-buffer
UCS-A /org/mac-pool/block #
```

What to Do Next

Include the MAC pool in a vNIC template.

Deleting a MAC Pool

If you delete a pool, Cisco UCS Manager does not reallocate any addresses from that pool that were assigned to vNICs or vHBAs. All assigned addresses from a deleted pool remain with the vNIC or vHBA to which they are assigned until one of the following occurs:

- The associated service profiles are deleted.
- The vNIC or vHBA to which the address is assigned is deleted.
- The vNIC or vHBA is assigned to a different pool.

Procedure

	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 mac-pool** *pool-name*	Deletes the specified MAC pool.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the MAC pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete mac-pool pool4
UCS-A /org* # commit-buffer
UCS-A /org #
```

CHAPTER 18

Configuring Quality of Service

This chapter includes the following sections:

Quality of Service

Cisco UCS provides the following methods to implement quality of service:

- System classes that specify the global configuration for certain types of traffic across the entire system
- QoS policies that assign system classes for individual vNICs
- Flow control policies that determine how uplink Ethernet ports handle pause frames

Global QoS changes made to the QoS system class may result in brief data-plane interruptions for all traffic. Some examples of such changes are:

- Changing the MTU size for an enabled class
- Changing packet drop for an enabled class
- Changing the CoS value for an enabled class

Guidelines and Limitations for Quality of Service on Cisco UCS 6300 Series Fabric Interconnect

- Cisco UCS 6300 Series Fabric Interconnect uses a shared buffer for all system classes.
- Multicast optimization is not supported.
- When you change the QoS parameters for any class causes traffic disruption to all classes. The following table lists the changes in the QoS system class and the conditions that trigger a system reboot.

QoS System class status	Condition	FI Reboot Status
Enabled	Change between drop and no drop	Yes
No-drop	Change between enable and disable	Yes
Enable and no-drop	Change in MTU size	Yes

- The subordinate FI reboots first as a result of the change in the QoS system class. The primary FI reboots only after you acknowledge it in **Pending Activities**.

Guidelines and Limitations for Quality of Service on Cisco UCS Mini

- Cisco UCS Mini uses a shared buffer for all system classes.
- The bronze class shares the buffer with SPAN. We recommend using either SPAN or the bronze class.
- Multicast optimization is not supported.
- Changing the QoS parameters for any class causes traffic disruption to all classes.
- When mixing Ethernet and FC or FCoE traffic, the bandwidth distribution is not equal.
- Multiple streams of traffic from the same class may not be distributed equally.
- Use the same CoS values for all no-drop policies to avoid any FC or FCoE performance issues.
- Only the platinum and gold classes support no-drop policies.

Configuring System Classes

System Classes

Cisco UCS uses Data Center Ethernet (DCE) to handle all traffic inside a Cisco UCS domain. This industry standard enhancement to Ethernet divides the bandwidth of the Ethernet pipe into eight virtual lanes. Two virtual lanes are reserved for internal system and management traffic. You can configure quality of service (QoS) for the other six virtual lanes. System classes determine how the DCE bandwidth in these six virtual lanes is allocated across the entire Cisco UCS domain.

Each system class reserves a specific segment of the bandwidth for a specific type of traffic, which provides a level of traffic management, even in an oversubscribed system. For example, you can configure the Fibre Channel Priority system class to determine the percentage of DCE bandwidth allocated to FCoE traffic.

The following table describes the system classes that you can configure.

Table 9: System Classes

System Class	Description
Platinum Gold Silver Bronze	A configurable set of system classes that you can include in the QoS policy for a service profile. Each system class manages one lane of traffic. All properties of these system classes are available for you to assign custom settings and policies. For Cisco UCS Mini, packet drop can only be disabled on the platinum and gold classes. Only one platinum and one gold class can be configured as a no drop class at a time.
Best Effort	A system class that sets the quality of service for the lane reserved for basic Ethernet traffic. Some properties of this system class are preset and cannot be modified. For example, this class has a drop policy that allows it to drop data packets if required. You cannot disable this system class.
Fibre Channel	A system class that sets the quality of service for the lane reserved for Fibre Channel over Ethernet traffic. Some properties of this system class are preset and cannot be modified. For example, this class has a no-drop policy that ensures it never drops data packets. You cannot disable this system class. **Note** FCoE traffic has a reserved QoS system class that should not be used by any other type of traffic. If any other type of traffic has a CoS value that is used by FCoE, the value is remarked to 0.

Configuring a System Class

The type of adapter in a server might limit the maximum MTU supported. For example, network MTU above the maximums might cause the packet to be dropped for the following adapters:

- The Cisco UCS M71KR CNA adapter, which supports a maximum MTU of 9216.
- The Cisco UCS 82598KR-CI adapter, which supports a maximum MTU of 14000.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope qos**	Enters Ethernet server QoS mode.
Step 3	UCS-A /eth-server/qos # **scope eth-classified** {**bronze** | **gold** | **platinum** | **silver**}	Enters Ethernet server QoS Ethernet classified mode for the specified system class.

	Command or Action	Purpose
Step 4	UCS-A /eth-server/qos/eth-classified # **enable**	Enables the specified system class.
Step 5	UCS-A /eth-server/qos/eth-classified # **set cos** *cos-value*	Specifies the class of service for the specified system class. Valid class of service values are 0 to 6. **Important** Use the same CoS values on UCS and N5K for all the no-drop policies. To insure that end-to-end PFC works correctly, have the same QoS policy configured on all intermediate switches.
Step 6	UCS-A /eth-server/qos/eth-classified # **set drop** {**drop** | **no-drop**}	Specifies whether the channel can drop packets or not. **Note** Changes saved to the drop displays the following warning message: Warning: The operation will cause momentary disruption to traffic forwarding.
Step 7	UCS-A /eth-server/qos/eth-classified # **set mtu** {*mtu-value* | **fc** | **normal**}	The maximum transmission unit, or packet size to be used. The maximum value for MTU is 9216. **Note** If the vNIC has an associated QoS policy, the MTU specified here must be equal to or less than the MTU specified in the associated QoS system class. If this MTU value exceeds the MTU value in the QoS system class, packets might get dropped during data transmission. Changes saved to the MTU displays the following warning message: Warning: The operation will cause momentary disruption to traffic forwarding.
Step 8	UCS-A /eth-server/qos/eth-classified # **set multicast-optimize** {**no** | **yes**}	Specifies whether the class is optimized to for sending multicast packets.
Step 9	UCS-A /eth-server/qos/eth-classified # **set weight** {*weight-value* | **best-effort** | **none**}	Specifies the relative weight for the specified system class. Valid weight values are 0 to 10.
Step 10	UCS-A /eth-server/qos/eth-classified # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to enable the platinum system class, allow the channel to drop packets, set the class of service to 6, set the MTU to normal, set the relative weight to 5, and commit the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope qos
UCS-A /eth-server/qos # scope eth-classified platinum
UCS-A /eth-server/qos/eth-classified # enable
UCS-A /eth-server/qos/eth-classified* # set drop drop
Warning: The operation will cause momentary disruption to traffic forwarding
UCS-A /eth-server/qos/eth-classified* # set cos 6
UCS-A /eth-server/qos/eth-classified* # set mtu normal
Warning: The operation will cause momentary disruption to traffic forwarding
UCS-A /eth-server/qos/eth-classified* # set weight 5
UCS-A /eth-server/qos/eth-classified* # commit-buffer
UCS-A /eth-server/qos/eth-classified #
```

Disabling a System Class

If you disable a system class that is used in a QoS policy, Cisco UCS Manager uses the system class configured with CoS 0 for traffic on servers that are configured with the QoS policy. If no system class is configured as CoS 0, the Best Effort system class is used. You cannot disable the Best Effort or Fibre Channel system classes.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope qos**	Enters Ethernet server QoS mode.
Step 3	UCS-A /eth-server/qos # **scope eth-classified** {**bronze** | **gold** | **platinum** | **silver**}	Enters Ethernet server QoS Ethernet classified mode for the specified system class.
Step 4	UCS-A /eth-server/qos/eth-classified # **disable**	Disables the specified system class.
Step 5	UCS-A /eth-server/qos/eth-classified # **commit-buffer**	Commits the transaction to the system configuration.

The following example disables the platinum system class and commits the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope qos
UCS-A /eth-server/qos # scope eth-classified platinum
UCS-A /eth-server/qos/eth-classified # disable
UCS-A /eth-server/qos/eth-classified* # commit-buffer
UCS-A /eth-server/qos/eth-classified #
```

Configuring Quality of Service Policies

Quality of Service Policy

A quality of service (QoS) policy assigns a system class to the outgoing traffic for a vNIC or vHBA. This system class determines the quality of service for that traffic. For certain adapters, you can also specify additional controls on the outgoing traffic, such as burst and rate.

You must include a QoS policy in a vNIC policy or vHBA policy and then include that policy in a service profile to configure the vNIC or vHBA.

Configuring a QoS Policy

Procedure

	Command or Action	Purpose
Step 1	Switch-A# **scope org** *org-name*	Enters org mode for the specified organization. To enter the default org mode, type **/** as the *org-name* .
Step 2	Switch-A /org # **create qos-policy** *policy-name*	Creates the specified QoS policy, and enters org QoS policy mode.
Step 3	Switch-A /org/qos-policy # **create egress-policy**	Creates the egress policy (for both vNICs and vHBAs) to be used by the QoS policy, and enters org QoS policy egress policy mode.
Step 4	Switch-A /org/qos-policy/egress-policy # **set host-cos-control** {**full** | **none**}	(Optional) Specifies whether the host or Cisco UCS Manager controls the class of service (CoS) for a vNIC. This setting has no effect on a vHBA. Use the **full** keyword to have the host control the CoS. If the packet has a valid CoS value, the host uses that value. Otherwise, it uses the CoS value associated with the specified class priority. Use the **none** keyword to have Cisco UCS Manager use the CoS value associated with the specified priority.
Step 5	Switch-A /org/qos-policy/egress-policy # **set prio** *sys-class-name*	Specifies the system class to be used for the egress policy. The *sys-class-name* argument can be one of the following class keywords: • **Fc—**Use this priority for QoS policies that control vHBA traffic only. • **Platinum—**Use this priority for QoS policies that control vNIC traffic only. • **Gold—**Use this priority for QoS policies that control vNIC traffic only. • **Silver—**Use this priority for QoS policies that control vNIC traffic only. • **Bronze—**Use this priority for QoS policies that control vNIC traffic only. • **Best Effort—**Do not use this priority. It is reserved for the Basic Ethernet traffic lane. If you assign this priority to a QoS policy and configure another system class as CoS 0, Cisco UCS Manager does not default to this system class. It defaults to the priority with CoS 0 for that traffic.

	Command or Action	Purpose
Step 6	Switch-A /org/qos-policy/egress-policy # **set rate** {**line-rate** | *kbps*} **burst** *bytes*	Specifies the expected average rate of traffic. Traffic that falls under this rate will always conform. The default is **line-rate**, which equals a value of 10,000,000. The minimum value is 8, and the maximum value is 40,000,000. Rate limiting is supported only on vNICs on the Cisco UCS VIC-1240 Virtual Interface Card and Cisco UCS VIC-1280 Virtual Interface Card. The Cisco UCS M81KR Virtual Interface Card supports rate limiting on both vNICs and vHBAs.
Step 7	Switch-A /org/qos-policy/egress-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a QoS policy for vNIC traffic, assigns the platinum system class and sets the rate limit (traffic rate and burst size) for the egress policy, and commits the transaction:

```
Switch-A# scope org /
Switch-A /org # create qos-policy VnicPolicy34
Switch-A /org/qos-policy* # create egress-policy
Switch-A /org/qos-policy/egress-policy* # set prio platinum
Switch-A /org/qos-policy/egress-policy* # set rate 5000000 burst 65000
Switch-A /org/qos-policy/egress-policy* # commit-buffer
Switch-A /org/qos-policy/egress-policy #
```

The following example creates a QoS policy for vHBA traffic, assigns the fc (Fibre Channel) system class and sets the rate limit (traffic rate and burst size) for the egress policy, and commits the transaction:

```
Switch-A# scope org /
Switch-A /org # create qos-policy VhbaPolicy12
Switch-A /org/qos-policy* # create egress-policy
Switch-A /org/qos-policy/egress-policy* # set prio fc
Switch-A /org/qos-policy/egress-policy* # set rate 5000000 burst 65000
Switch-A /org/qos-policy/egress-policy* # commit-buffer
Switch-A /org/qos-policy/egress-policy #
```

What to Do Next

Include the QoS policy in a vNIC or vHBA template.

Deleting a QoS Policy

If you delete a QoS policy that is in use or you disable a system class that is used in a QoS policy, any vNIC or vHBA that uses that QoS policy is assigned to the Best Effort system class or to the system class with a CoS of 0. In a system that implements multitenancy, Cisco UCS Manager first attempts to find a matching QoS policy in the organization hierarchy.

Procedure

	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 # **delete qos-policy** *policy-name*	Deletes the specified QoS policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following deletes the QoS policy named QosPolicy34 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete qos-policy QosPolicy34
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Flow Control Policies

Flow Control Policy

Flow control policies determine whether the uplink Ethernet ports in a Cisco UCS domain send and receive IEEE 802.3x pause frames when the receive buffer for a port fills. These pause frames request that the transmitting port stop sending data for a few milliseconds until the buffer clears.

For flow control to work between a LAN port and an uplink Ethernet port, you must enable the corresponding receive and send flow control parameters for both ports. For Cisco UCS, the flow control policies configure these parameters.

When you enable the send function, the uplink Ethernet port sends a pause request to the network port if the incoming packet rate becomes too high. The pause remains in effect for a few milliseconds before traffic is reset to normal levels. If you enable the receive function, the uplink Ethernet port honors all pause requests from the network port. All traffic is halted on that uplink port until the network port cancels the pause request.

Because you assign the flow control policy to the port, changes to the policy have an immediate effect on how the port reacts to a pause frame or a full receive buffer.

Configuring a Flow Control Policy

Before You Begin

Configure the network port with the corresponding setting for the flow control that you need. For example, if you enable the send setting for flow-control pause frames in the policy, ensure that the receive parameter in the network port is set to on or to desired. If you want the Cisco UCS port to receive flow-control frames, ensure that the send parameter is set to on or to desire on the network port. If you do not want to use flow control, you can set the send and receive parameters on the network port to off.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope flow-control**	Enters Ethernet uplink flow control mode.
Step 3	UCS-A /eth-uplink/flow-control # **create policy** *policy-name*	Creates the specified flow control policy.
Step 4	UCS-A /eth-uplink/flow-control/policy # **set prio** *prio-option*	Specifies one of the following flow control priority options: • **auto —**The Cisco UCS system and the network negotiate whether PPP will be used on this fabric interconnect. • **on —**PPP is enabled on this fabric interconnect.
Step 5	UCS-A /eth-uplink/flow-control/policy # **set receive** *receive-option*	Specifies one of the following flow control receive options: • **off —**Pause requests from the network are ignored and traffic flow continues as normal. • **on —**Pause requests are honored and all traffic is halted on that uplink port until the network cancels the pause request.
Step 6	UCS-A /eth-uplink/flow-control/policy # **set send** *send-option*	Specifies one of the following flow control send options: • **off —**Traffic on the port flows normally regardless of the packet load. • **on —**The Cisco UCS system sends a pause request to the network if the incoming packet rate becomes too high. The pause remains in effect for a few milliseconds before traffic is reset to normal levels.
Step 7	UCS-A /eth-uplink/flow-control/policy # **commit-buffer**	Commits the transaction to the system configuration.

The following configures a flow control policy and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope flow-control
UCS-A /eth-uplink/flow-control # create policy FlowControlPolicy23
UCS-A /eth-uplink/flow-control/policy* # set prio auto
UCS-A /eth-uplink/flow-control/policy* # set receive on
UCS-A /eth-uplink/flow-control/policy* # set send on
UCS-A /eth-uplink/flow-control/policy* # commit-buffer
UCS-A /eth-uplink/flow-control/policy #
```

What to Do Next

Associate the flow control policy with an uplink Ethernet port or port channel.

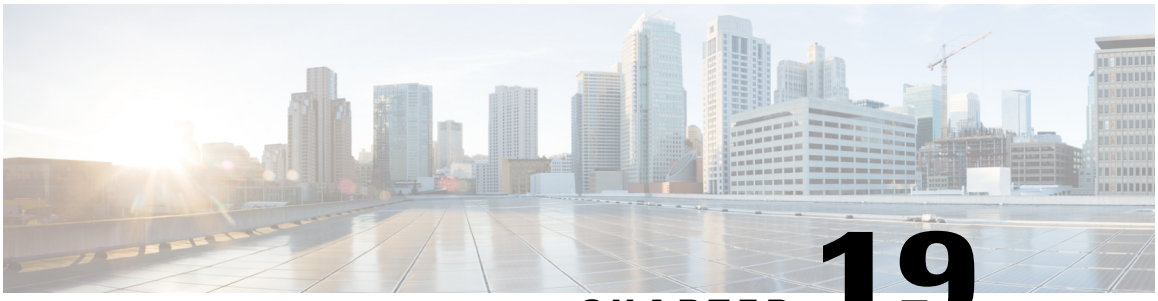
Deleting a Flow Control Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope flow-control**	Enters Ethernet uplink flow control mode.
Step 3	UCS-A /eth-uplink/flow-control # **delete policy** *policy-name*	Deletes the specified flow control policy.
Step 4	UCS-A /eth-uplink/flow-control # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the flow control policy named FlowControlPolicy23 and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope flow-control
UCS-A /eth-uplink/flow-control # delete policy FlowControlPolicy23
UCS-A /eth-uplink/flow-control* # commit-buffer
UCS-A /eth-uplink/flow-control #
```



CHAPTER 19

Configuring Network-Related Policies

This chapter includes the following sections:

Configuring vNIC Templates

vNIC Template

The vNIC LAN connectivity policy defines how a vNIC on a server connects to the LAN.

Cisco UCS Manager does not automatically create a VM-FEX port profile with the correct settings when you create a vNIC template. If you want to create a VM-FEX port profile, you must configure the target of the vNIC template as a VM. You must include this policy in a service profile for it to take effect.

You can select VLAN groups in addition to any individual VLAN while creating a vNIC template.

Note If your server has two Emulex or QLogic NICs (Cisco UCS CNA M71KR-E or Cisco UCS CNA M71KR-Q), you must configure vNIC policies for both adapters in your service profile to get a user-defined MAC address for both NICs. If you do not configure policies for both NICs, Windows still detects both of them in the PCI bus. Then because the second eth is not part of your service profile, Windows assigns it a hardware MAC address. If you then move the service profile to a different server, Windows sees additional NICs because one NIC did not have a user-defined MAC address.

Configuring a vNIC Template

Procedure

	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 vnic-templ** *vnic-templ-name* [**eth-if** *vlan-name*] [**fabric** {**a** | **b**}] [**target** [**adapter** | **vm**]]	Creates a vNIC template and enters organization vNIC template mode. The target you choose determines whether or not Cisco UCS Manager automatically creates a VM-FEX port profile with the appropriate settings for the vNIC template. This can be one of the following: • **Adapter—**The vNICs apply to all adapters. No VM-FEX port profile is created if you choose this option. • **VM—**The vNICs apply to all virtual machines. A VM-FEX port profile is created if you choose this option.
Step 3	UCS-A /org/vnic-templ # **set descr** *description*	(Optional) Provides a description for the vNIC template.
Step 4	UCS-A /org/vnic-templ # **set fabric** {**a** | **a-b** | **b** | **b-a**}	(Optional) Specifies the fabric to use for the vNIC. If you did not specify the fabric when creating the vNIC template in Step 2, you have the option to specify it with this command. If you want this vNIC to be able to access the second fabric interconnect if the default one is unavailable, choose **a-b** (A is the primary) or **b-a** (B is the primary) .

	Command or Action	Purpose
		Note Do not enable fabric failover for the vNIC under the following circumstances: • If the Cisco UCS domain is running in Ethernet Switch Mode. vNIC fabric failover is not supported in that mode. If all Ethernet uplinks on one fabric interconnect fail, the vNICs do not fail over to the other. • If you plan to associate this vNIC to a server with an adapter that does not support fabric failover, such as the Cisco UCS 82598KR-CI 10-Gigabit Ethernet Adapter. If you do so, Cisco UCS Manager generates a configuration fault when you associate the service profile with the server.
Step 5	UCS-A /org/vnic-templ # **set mac-pool** *mac-pool-name*	The MAC address pool that vNICs created from this vNIC template should use.
Step 6	UCS-A /org/vnic-templ # **set mtu** *mtu-value*	The maximum transmission unit, or packet size, that vNICs created from this vNIC template should use. Enter an integer between 1500 and 9000. **Note** If the vNIC template has an associated QoS policy, the MTU specified here must be equal to or less than the MTU specified in the associated QoS system class. If this MTU value exceeds the MTU value in the QoS system class, packets may be dropped during data transmission.
Step 7	UCS-A /org/vnic-templ # **set nw-control-policy** *policy-name*	The network control policy that vNICs created from this vNIC template should use.
Step 8	UCS-A /org/vnic-templ # **set pin-group** *group-name*	The LAN pin group that vNICs created from this vNIC template should use.
Step 9	UCS-A /org/vnic-templ # **set qos-policy** *policy-name*	The quality of service policy that vNICs created from this vNIC template should use.
Step 10	UCS-A /org/vnic-templ # **set stats-policy** *policy-name*	The statistics collection policy that vNICs created from this vNIC template should use.
Step 11	UCS-A /org/vnic-templ # **set type** {**initial-template** | **updating-template**}	Specifies the vNIC template update type. If you do not want vNIC instances created from this template to be automatically updated when the template is updated, use the **initial-template** keyword; otherwise, use the **updating-template** keyword to ensure that all vNIC instances are updated when the vNIC template is updated.
Step 12	UCS-A /org/vnic-templ # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a vNIC template and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create vnic template VnicTempFoo
UCS-A /org/vnic-templ* # set descr "This is a vNIC template example."
UCS-A /org/vnic-templ* # set fabric a
UCS-A /org/vnic-templ* # set mac-pool pool137
UCS-A /org/vnic-templ* # set mtu 8900
UCS-A /org/vnic-templ* # set nw-control-policy ncp5
UCS-A /org/vnic-templ* # set pin-group PinGroup54
UCS-A /org/vnic-templ* # set qos-policy QosPol5
UCS-A /org/vnic-templ* # set stats-policy ServStatsPolicy
UCS-A /org/vnic-templ* # set type updating-template
UCS-A /org/vnic-templ* # commit-buffer
UCS-A /org/vnic-templ #
```

Redundancy Template Pairs

Creating vNIC and vHBA template pairs enables you to group vNICs or vHBAs that belong to a specific server. For example, you can create a vNIC or a vHBA template and specify it as the Primary template, then create a different vNIC or vHBA template and specify it as the Secondary template. You can link the two templates to create a pair that share attributes that you define in the Primary template. The Secondary template inherits the attributes from the Primary template and any changes made to the Primary template are propagated to the Secondary template in the template pair. You can also modify any non-shared configurations on each individual template in the pair.

When creating the pair, you can assign one template, for example the Primary template to Fabric A and the other template, for example the Secondary template to Fabric B or vice versa. This feature eliminates the need to configure vNIC or vHBA pairs independently using one or more templates.

The number of vNIC and vHBA pairs that can be created using a template pair is only limited by the adapter's maximum capabilities.

Use the **Initial Template** type for one time provisioning.

Use the **Updating Template** type to have the Primary template drive the changes in the redundancy pair for shared configurations. See the shared configurations listed below.

Creating vNIC Template Pairs

Procedure

	Command or Action	Purpose
Step 1	UCS-A/ org # **create vnic-templ** *vnic-primary* .	Creates a Primary vNIC template.
Step 2	UCS-A/ # org vnic-templ **set type updating-template** .	Set the template type to updating, which drives the configurations in the Primary vNIC template for shared configurations to the peer vNIC template. See the shared configurations listed below.

	Command or Action	Purpose
Step 3	UCS-A/ # org vnic-templ [**set fabric** {**a** | **b**}] .	Specifies the fabric for the Primary vNIC template. If you specify Fabric A for the Primary vNIC template, the Secondary vNIC template must be Fabric B or vice versa.
Step 4	UCS-A/ # org vnic-templ **set descr primaryinredundancypair** .	Sets the template as the Primary vNIC template.
Step 5	UCS-A/ # org vnic-templ **set redundancy-type** *primary*.	Sets the redundancy template type as the Primary vNIC template. Following are descriptions of the **Redundancy Types**: **Primary—**Creates configurations that can be shared with the Secondary vNIC template. Any shared changes on the Primary vNIC template are automatically synchronized to the Secondary vNIC template. **Secondary —** All shared configurations are inherited from the Primary template. **No Redundancy—** Legacy vNIC template behavior. Following is a list of shared configurations: • **Network Control Policy** • **QoS Policy** • **Stats Threshold Policy** • **Template Type** • **Connection Policies** • **VLANS** • **MTU** Following is a list of non-shared configurations: • **Fabric ID** • **CDN Source** • **MAC Pool** • **Description** • **Pin Group Policy**
Step 6	UCS-A/ # org vnic-templ **exit** .	Exits creating the redundancy template pairing. **Note** Ensure to commit the transaction after linking the Primary vNIC template to a peer Secondary vNIC template to create the redundancy pair.
Step 7	UCS-A/ # org vnic-templ **create vNIC-templ** *vNICsecondary* .	Creates the Secondary vNIC template.

	Command or Action	Purpose
Step 8	UCS-A/ # org vnic-templ **set type updating-template** .	Sets the template type to updating, which automatically inherits the configurations from the Primary vNIC template.
Step 9	UCS-A/ org # vnic-templ [**set fabric** {**a** | **b**}] .	Specifies the fabric for the Secondary vNIC template. If you specify Fabric A for the Primary vNIC template, the Secondary vNIC template must be Fabric B or vice versa.
Step 10	UCS-A/ # org vnic-templ **set descr secondaryredundancypair**.	Sets the secondary vNIC template as a redundancy pair template.
Step 11	UCS-A/ # org vnic-templ **set redundancy-type** *secondary*.	Sets the vNIC template type as Secondary.
Step 12	UCS-A/ # org vnic-templ **set peer-template-name** *vNIC-primary*.	Sets the Primary vNIC template as the peer to the Secondary vNIC template.
Step 13	UCS-A/ # org vnic-templ **commit-buffer** .	Commits the transaction to the system configuration.

The following example configures a vNIC redundancy template pair and commits the transaction:

```
UCS-A /org* # create vnic-template vnic-primary
UCS-A /org/vnic-templ* # set type updating-template
UCS-A /org/vnic-templ* # set fabric a
UCS-A /org/vnic-templ* # set descr primaryinredundancypair
UCS-A /org/vnic-templ* # set redundancy-type primary
UCS-A /org/vnic-templ* # exit
UCS-A /org* # create vnic-templ vnicsecondary
UCS-A /org/vnic-templ* # set fabric b
UCS-A /org/vnic-templ* # set descr secondaryinredundancypair
UCS-A /org/vnic-templ* # set redundancy-type secondary
UCS-A /org/vnic-templ* # set peer-template-name vnic-primary
UCS-A /org/vnic-templ* # commit-buffer
UCS-A /org/vnic-templ #
```

What to Do Next

After you create the vNIC redundancy template pair, you can use the redundancy template pair to create redundancy vNIC pairs for any service profile in the same organization or sub- organization.

Undo vNIC Template Pairs

You can undo the vNIC template pair by changing the Peer Redundancy Template so that there is no peer template for the Primary or the Secondary template. When you undo a vNIC template pair, the corresponding vNIC pairs also becomes undone.

Procedure

	Command or Action	Purpose
Step 1	UCS-A /org # **scope vnic-templ** *template1*.	Specifies the name of the vNIc template that you want to undo from the template pair.
Step 2	UCS-A /org/ vnic-templ # **set redundancy-type** *no redundancy*.	Removes the paring between the peer Primary or Secondary redundancy template used to perform the template pairing.
Step 3	UCS-A /org/vnic-templ* # **commit-buffer** .	Commits the transaction to the system configuration.

The following example shows how to undo a template pairing:

```
UCS-A /org # scope vnic-templ template1
UCS-A /org/vnic-templ # set redundancy-type no-redundancy
UCS-A /org/vnic-templ* # commit buffer
```

Deleting a vNIC Template

Procedure

	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 vnic-templ** *vnic-templ-name*	Deletes the specified vNIC template.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the vNIC template named VnicTemp42 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete vnic template VnicTemp42
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Ethernet Adapter Policies

Ethernet and Fibre Channel Adapter Policies

These policies govern the host-side behavior of the adapter, including how the adapter handles traffic. For example, you can use these policies to change default settings for the following:

- Queues
- Interrupt handling
- Performance enhancement
- RSS hash
- Failover in a cluster configuration with two fabric interconnects

Note For Fibre Channel adapter policies, the values displayed by Cisco UCS Manager may not match those displayed by applications such as QLogic SANsurfer. For example, the following values may result in an apparent mismatch between SANsurfer and Cisco UCS Manager:

- Max LUNs Per Target—SANsurfer has a maximum of 256 LUNs and does not display more than that number. Cisco UCS Manager supports a higher maximum number of LUNs.
- Link Down Timeout—In SANsurfer, you configure the timeout threshold for link down in seconds. In Cisco UCS Manager, you configure this value in milliseconds. Therefore, a value of 5500 ms in Cisco UCS Manager displays as 5s in SANsurfer.
- Max Data Field Size—SANsurfer has allowed values of 512, 1024, and 2048. Cisco UCS Manager allows you to set values of any size. Therefore, a value of 900 in Cisco UCS Manager displays as 512 in SANsurfer.
- LUN Queue Depth—The LUN queue depth setting is available for Windows system FC adapter policies. Queue depth is the number of commands that the HBA can send and receive in a single transmission per LUN. Windows Storport driver sets this to a default value of 20 for physical miniports and to 250 for virtual miniports. This setting adjusts the initial queue depth for all LUNs on the adapter. Valid range for this value is 1 to 254. The default LUN queue depth is 20. This feature only works with Cisco UCS Manager version 3.1(2) and higher.
- IO TimeOut Retry—When the target device is not responding to an IO request within the specified timeout, the FC adapter will abort the pending command then resend the same IO after the timer expires. The FC adapter valid range for this value is 1 to 59 seconds. The default IO retry timeout is 5 seconds. This feature only works with Cisco UCS Manager version 3.1(2) and higher.

Operating System Specific Adapter Policies

By default, Cisco UCS provides a set of Ethernet adapter policies and Fibre Channel adapter policies. These policies include the recommended settings for each supported server operating system. Operating systems are sensitive to the settings in these policies. Storage vendors typically require non-default adapter settings. You can find the details of these required settings on the support list provided by those vendors.

Important We recommend that you use the values in these policies for the applicable operating system. Do not modify any of the values in the default policies unless directed to do so by Cisco Technical Support.

However, if you are creating an Ethernet adapter policy for a Windows OS (instead of using the default Windows adapter policy), you must use the following formulas to calculate values that work with Windows:

Completion Queues = Transmit Queues + Receive Queues
Interrupt Count = (Completion Queues + 2) rounded up to nearest power of 2

For example, if Transmit Queues = 1 and Receive Queues = 8 then:

Completion Queues = 1 + 8 = 9
Interrupt Count = (9 + 2) rounded up to the nearest power of 2 = 16

Accelerated Receive Flow Steering

Accelerated Receive Flow Steering (ARFS) is hardware-assisted receive flow steering that can increase CPU data cache hit rate by steering kernel level processing of packets to the CPU where the application thread consuming the packet is running.

Using ARFS can improve CPU efficiency and reduce traffic latency. Each receive queue of a CPU has an interrupt associated with it. You can configure the Interrupt Service Routine (ISR) to run on a CPU. The ISR moves the packet from the receive queue to the backlog of one of the current CPUs, which processes the packet later. If the application is not running on this CPU, the CPU must copy the packet to non-local memory, which adds to latency. ARFS can reduce this latency by moving that particular stream to the receive queue of the CPU on which the application is running.

ARFS is disabled by default and can be enabled through Cisco UCS Manager. To configure ARFS, do the following:

1. Create an adapter policy with ARFS enabled.
2. Associate the adapter policy with a service profile.
3. Enable ARFS on a host.
    1. Turn off Interrupt Request Queue (IRQ) balance.
    2. Associate IRQ with different CPUs.
    3. Enable ntuple by using ethtool.

Guidelines and Limitations for Accelerated Receive Flow Steering

- ARFS supports 64 filters per vNIC
- ARFS is supported on the following adapters:
    - Cisco UCS VIC 1280, 1240, 1340, and 1380
    - Cisco UCS VIC 1225, 1225T, 1285, 1223, 1227, 1227T, 1385, 1387

- ARFS is supported on the following Operating Systems:
  - Red Hat Enterprise Linux 6.5, and 6.6
  - Red Hat Enterprise Linux 7.0 and higher versions
  - SUSE Linux Enterprise Server 11 SP2 and SP3
  - SUSE Linux Enterprise Server 12 and higher versions
  - Ubuntu 14.04.2

Interrupt Coalescing

Adapters typically generate a large number of interrupts that a host CPU must service. Interrupt coalescing reduces the number of interrupts serviced by the host CPU. This is done by interrupting the host only once for multiple occurrences of the same event over a configurable coalescing interval.

When interrupt coalescing is enabled for receive operations, the adapter continues to receive packets, but the host CPU does not immediately receive an interrupt for each packet. A coalescing timer starts when the first packet is received by the adapter. When the configured coalescing interval times out, the adapter generates one interrupt with the packets received during that interval. The NIC driver on the host then services the multiple packets that are received. Reduction in the number of interrupts generated reduces the time spent by the host CPU on context switches. This means that the CPU has more time to process packets, which results in better throughput and latency.

Adaptive Interrupt Coalescing

Due to the coalescing interval, the handling of received packets adds to latency. For small packets with a low packet rate, this latency increases. To avoid this increase in latency, the driver can adapt to the pattern of traffic flowing through it and adjust the interrupt coalescing interval for a better response from the server.

Adaptive interrupt coalescing (AIC) is most effective in connection-oriented low link utilization scenarios including email server, databases server, and LDAP server. It is not suited for line-rate traffic.

Guidelines and Limitations for Adaptive Interrupt Coalescing

- Adaptive Interrupt Coalescing (AIC) does not provide any reduction in latency when the link utilization is more than 80 percent.
- Enabling AIC disables static coalescing.
- AIC is supported on the following Operating Systems:
  - Red Hat Enterprise Linux 6.4 and higher versions
  - Red Hat Enterprise Linux 7.0 and higher versions
  - SUSE Linux Enterprise Server 11 SP2 and SP3
  - SUSE Linux Enterprise Server 12
  - XenServer 6.5
  - Ubuntu 14.04.2

RDMA Over Converged Ethernet for SMB Direct

RDMA over Converged Ethernet (RoCE) allows direct memory access over an Ethernet network. RoCE is a link layer protocol, and hence, it allows communication between any two hosts in the same Ethernet broadcast domain. RoCE delivers superior performance compared to traditional network socket implementations because of lower latency, lower CPU utilization and higher utilization of network bandwidth. Windows 2012 R2 and later versions use RDMA for accelerating and improving the performance of SMB file sharing and Live Migration.

Cisco UCS Manager Release 2.2(4) supports RoCE for Microsoft SMB Direct. It sends additional configuration information to the adapter while creating or modifying an Ethernet adapter policy.

Guidelines and Limitations for SMB Direct with RoCE

- Microsoft SMB Direct with RoCE is supported:
  - on Windows 2012 R2 for Cisco UCS Manager release 2.2(4) and later releases.

    .
  - on Windows 2016 for Cisco UCS Manager release 2.2(8) and later releases.

    .
- Microsoft SMB Direct with RoCE is supported only with third generation Cisco UCS VIC 1340, 1380, 1385, 1387 adapters. Second generation UCS VIC 1225 and 1227 adapters are not supported.
- RoCE configuration is supported between Cisco adapters. Interoperability between Cisco adapters and third party adapters is not supported.
- Cisco UCS Manager does not support more than 4 RoCE-enabled vNICs per adapter.
- Cisco UCS Manager does not support RoCE with NVGRE, VXLAN, NetFlow, VMQ, or usNIC.
- Maximum number of queue pairs per adapter is 8192.
- Maximum number of memory regions per adapter is 524288.
- If you do not disable RoCE before downgrading Cisco UCS Manager from Release 2.2(4), downgrade will fail.
- Cisco UCS Manager does not support fabric failover for vNICs with RoCE enabled.

Configuring an Ethernet Adapter Policy

Procedure

	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* .

	Command or Action	Purpose
Step 2	UCS-A /org # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 3	UCS-A /org/eth-policy # **set arfs accelaratedrfs** {**enabled** | **disabled**}	(Optional) Configures Accelerated RFS.
Step 4	UCS-A /org/eth-policy # **set comp-queue count** *count*	(Optional) Configures the Ethernet completion queue.
Step 5	UCS-A /org/eth-policy # **set descr** *description*	(Optional) Provides a description for the policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 6	UCS-A /org/eth-policy # **set failover timeout** *timeout-sec*	(Optional) Configures the Ethernet failover.
Step 7	UCS-A /org/eth-policy # **set interrupt** {**coalescing-time** *sec* | **coalescing-type** {**idle** | **min**} | **count** *count* | **mode** {**intx** | **msi** | **msi-x**}}	(Optional) Configures the Ethernet interrupt.
Step 8	UCS-A /org/eth-policy # **set nvgre adminstate** {**disabled** | **enabled**}	(Optional) Configures NVGRE.
Step 9	UCS-A /org/eth-policy # **set offload** {**large-receive** | **tcp-rx-checksum** | **tcp-segment** | **tcp-tx-checksum**} {**disabled** | **enabled**}	(Optional) Configures the Ethernet offload.
Step 10	UCS-A /org/eth-policy # **set policy-owner** {**local** | **pending**}	(Optional) Specifies the owner for the Ethernet adapter policy.
Step 11	UCS-A /org/eth-policy # **set recv-queue** {**count** *count* | **ring-size** *size-num*}	(Optional) Configures the Ethernet receive queue.
Step 12	UCS-A /org/eth-policy # **set roce adminstate** {**disabled** | **enabled**} | **memoryregions** *number-of-memory-regions* | **queuepairs** *number-of-queue-pairs* | **resourcegroups** *number-of-resource-groups*	(Optional) Configures RDMA over converged Ethernet (RoCE) by using the following options: • **adminstate—**Enables or disables RoCE. • **memoryregion**s**—**Configures the number of memory regions to be used per adapter. The values range from 1-524288 memory regions, and should be an integer rounded up to the nearest power of 2.

	Command or Action	Purpose
		• **queuepairs—**Configures the number of queue pairs to be used per adapter. The values range from 1-8192 queue pairs, and should be an integer rounded up to the nearest power of 2. • **resourcegroups—**Configures the number of resource groups to be used. The values range from 1-128 resource groups. The value should be an integer rounded up to the nearest power of 2 and greater than or equal to the number of CPU cores on the system for optimum performance.
Step 13	UCS-A /org/eth-policy # **set rss receivesidescaling** {**disabled** | **enabled**}	(Optional) Configures the RSS.
Step 14	UCS-A /org/eth-policy # **set trans-queue** {**count** *count* | **ring-size** *size-num*}	(Optional) Configures the Ethernet transmit queue.
Step 15	UCS-A /org/eth-policy # **set vxlan adminstate** {**disabled** | **enabled**}	(Optional) Configures VXLAN.
Step 16	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures an Ethernet adapter policy, and commits the transaction:

```
UCS-A# scope org
UCS-A /org* # create eth-policy EthPolicy19
UCS-A /org/eth-policy* # set comp-queue count 16
UCS-A /org/eth-policy* # set descr "This is an Ethernet adapter policy example."
UCS-A /org/eth-policy* # set failover timeout 300
UCS-A /org/eth-policy* # set interrupt count 64
UCS-A /org/eth-policy* # set offload large-receive disabled
UCS-A /org/eth-policy* # set recv-queue count 32
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set trans-queue
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

The following example configures an Ethernet adapter policy with RoCE, and commits the transaction:

```
UCS-A# scope org
UCS-A /org* # create eth-policy EthPolicy20
UCS-A /org/eth-policy* # set roce adminstate enable
UCS-A /org/eth-policy* # set roce memoryregions 131072
UCS-A /org/eth-policy* # set roce queuepairs 256
UCS-A /org/eth-policy* # set roce resourcegroups 32
UCS-A /org/eth-policy # commit buffer
UCS-A /org # show eth-policy EthPolicy20 detail expand

Eth Adapter Policy:
    Name: EthPolicy20
    Description:
    Policy Owner: Local

    ARFS:
        Accelarated Receive Flow Steering: Disabled

    Ethernet Completion Queue:
```

```
    Count: 2

Ethernet Failback:
    Timeout (sec): 5

Ethernet Interrupt:
    Coalescing Time (us): 125
    Coalescing Type: Min
    Count: 4
    Driver Interrupt Mode: MSI-X

NVGRE:
    NVGRE: Disabled

Ethernet Offload:
    Large Receive: Enabled
    TCP Segment: Enabled
    TCP Rx Checksum: Enabled
    TCP Tx Checksum: Enabled

Ethernet Receive Queue:
    Count: 1
    Ring Size: 512

ROCE:
    RoCE: Enabled
    Resource Groups: 32
    Memory Regions: 131072
    Queue Pairs: 256

VXLAN:
    VXLAN: Disabled

Ethernet Transmit Queue:
    Count: 1
    Ring Size: 256

RSS:
    Receive Side Scaling: Disabled
```

Configuring an Ethernet Adapter Policy to Enable eNIC Support for MRQS on Linux Operating Systems

Cisco UCS Manager includes eNIC support for the Multiple Receive Queue Support (MRQS) feature on Red Hat Enterprise Linux Version 6.x and SUSE Linux Enterprise Server Version 11.x.

Procedure

Step 1 Create an Ethernet adapter policy.
Use the following parameters when creating the Ethernet adapter policy:

- Transmit Queues = 1
- Receive Queues = n (up to 8)
- Completion Queues = # of Transmit Queues + # of Receive Queues
- Interrupts = # Completion Queues + 2
- Receive Side Scaling (RSS) = Enabled

- Interrupt Mode = Msi-X

See Configuring an Ethernet Adapter Policy, on page 285.

Step 2 Install an eNIC driver Version 2.1.1.35 or later.
See Cisco UCS Virtual Interface Card Drivers for Linux Installation Guide.

Step 3 Reboot the server

Configuring an Ethernet Adapter Policy to Enable Stateless Offloads with NVGRE

Cisco UCS Manager supports stateless offloads with NVGRE only with Cisco UCS VIC 1340 and/or Cisco UCS VIC 1380 adapters that are installed on servers running Windows Server 2012 R2 operating systems. Stateless offloads with NVGRE cannot be used with NetFlow, usNIC, or VM-FEX.

Procedure

	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 # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 3	To enable stateless offloads with NVGRE, set the following options:	• Transmit Queues = 1 • Receive Queues = n (up to 8) • Completion Queues = # of Transmit Queues + # of Receive Queues • Interrupts = # Completion Queues + 2 • Network Virtualization using Generic Routing Encapsulation = Enabled • Interrupt Mode = Msi-X For more information on creating an Ethernet adapter policy, see Configuring an Ethernet Adapter Policy, on page 285.
Step 4	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.
Step 5	Install an eNIC driver Version 3.0.0.8 or later.	For more information, see http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/sw/vic_drivers/install/Windows/b_Cisco_VIC_Drivers_for_Windows_Installation_Guide.html.
Step 6	Reboot the server.	

The following example shows how to configure an Ethernet adapter policy to enable stateless offloads with NVGRE and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create eth-policy NVGRE
UCS-A /org/eth-policy* # set descr "Ethernet adapter policy with stateless offloads"
UCS-A /org/eth-policy* # set nvgre adminstate enabled
UCS-A /org/eth-policy* # set comp-queue count 16
UCS-A /org/eth-policy* # set interrupt count 64
UCS-A /org/eth-policy* # set recv-queue count 32
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set trans-queue 1
UCS-A /org/eth-policy* # set interrupt mode mxi-x
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

Configuring an Ethernet Adapter Policy to Enable Stateless Offloads with VXLAN

Cisco UCS Manager supports stateless offloads with VXLAN only with Cisco UCS VIC 1340 and/or Cisco UCS VIC 1380 adapters that are installed on servers running VMWare ESXi Release 5.5 and later releases of the operating system. Stateless offloads with VXLAN cannot be used with NetFlow, usNIC, or VM-FEX.

Procedure

	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 # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 3	To enable stateless offloads with VXLAN, set the following options:	• Transmit Queues = 1 • Receive Queues = n (up to 8) • Completion Queues = # of Transmit Queues + # of Receive Queues • Interrupts = # Completion Queues + 2 • Virtual Extensible LAN = Enabled • Interrupt Mode = Msi-X For more information on creating an Ethernet adapter policy, see Configuring an Ethernet Adapter Policy, on page 285.
Step 4	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.

	Command or Action	Purpose
Step 5	Install an eNIC driver Version 2.1.2.59 or later.	For more information, see http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/sw/vic_drivers/install/ESX/2-0/b_Cisco_VIC_Drivers_for_ESX_Installation_Guide.html.
Step 6	Reboot the server.	

The following example shows how to configure an Ethernet adapter policy to enable stateless offloads with VXLAN and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create eth-policy VXLAN
UCS-A /org/eth-policy* # set descr "Ethernet adapter policy with stateless offloads"
UCS-A /org/eth-policy* # set vxlan adminstate enabled
UCS-A /org/eth-policy* # set comp-queue count 16
UCS-A /org/eth-policy* # set interrupt count 64
UCS-A /org/eth-policy* # set recv-queue count 32
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set trans-queue 1
UCS-A /org/eth-policy* # set interrupt mode mxi-x
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

Deleting an Ethernet Adapter Policy

Procedure

	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 # **delete eth-policy** *policy-name*	Deletes the specified Ethernet adapter policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the Ethernet adapter policy named EthPolicy19 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete eth-policy EthPolicy19
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring the Default vNIC Behavior Policy

Default vNIC Behavior Policy

Default vNIC behavior policy allows you to configure how vNICs are created for a service profile. You can choose to create vNICS manually, or you can create them automatically.

You can configure the default vNIC behavior policy to define how vNICs are created. This can be one of the following:

- **None—**Cisco UCS Manager does not create default vNICs for a service profile. All vNICs must be explicitly created.
- **HW Inherit—**If a service profile requires vNICs and none have been explicitly defined, Cisco UCS Manager creates the required vNICs based on the adapter installed in the server associated with the service profile.

Note If you do not specify a default behavior policy for vNICs, **HW Inherit** is used by default.

Configuring a Default vNIC Behavior Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A/org # **scope vnic-beh-policy**	Enters default vNIC behavior policy mode.
Step 3	UCS-A/org/vnic-beh-policy # **set action** {**hw-inherit** [**template_name** *name*] | **none**}	Specifies the default vNIC behavior policy. This can be one of the following: • **hw-inherit—**If a service profile requires vNICs and none have been explicitly defined, Cisco UCS Manager creates the required vNICs based on the adapter installed in the server associated with the service profile. If you specify **hw-inherit**, you can also specify a vNIC template to create the vNICs. • **none—**Cisco UCS Manager does not create default vNICs for a service profile. All vNICs must be explicitly created.
Step 4	UCS-A/org/vnic-beh-policy # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to set the default vNIC behavior policy to **hw-inherit**:

```
UCS-A # scope org /
UCS-A/org # scope vnic-beh-policy
UCS-A/org/vnic-beh-policy # set action hw-inherit
UCS-A/org/vnic-beh-policy* # commit-buffer
UCS-A/org/vnic-beh-policy #
```

Configuring LAN Connectivity Policies

About the LAN and SAN Connectivity Policies

Connectivity policies determine the connections and the network communication resources between the server and the LAN or SAN on the network. These policies use pools to assign MAC addresses, WWNs, and WWPNs to servers and to identify the vNICs and vHBAs that the servers use to communicate with the network.

Note We do not recommend that you use static IDs in connectivity policies, because these policies are included in service profiles and service profile templates and can be used to configure multiple servers.

Privileges Required for LAN and SAN Connectivity Policies

Connectivity policies enable users without network or storage privileges to create and modify service profiles and service profile templates with network and storage connections. However, users must have the appropriate network and storage privileges to create connectivity policies.

Privileges Required to Create Connectivity Policies

Connectivity policies require the same privileges as other network and storage configurations. For example, you must have at least one of the following privileges to create connectivity policies:

- admin—Can create LAN and SAN connectivity policies
- ls-server—Can create LAN and SAN connectivity policies
- ls-network—Can create LAN connectivity policies
- ls-storage—Can create SAN connectivity policies

Privileges Required to Add Connectivity Policies to Service Profiles

After the connectivity policies have been created, a user with ls-compute privileges can include them in a service profile or service profile template. However, a user with only ls-compute privileges cannot create connectivity policies.

Interactions between Service Profiles and Connectivity Policies

You can configure the LAN and SAN connectivity for a service profile through either of the following methods:

- LAN and SAN connectivity policies that are referenced in the service profile
- Local vNICs and vHBAs that are created in the service profile
- Local vNICs and a SAN connectivity policy
- Local vHBAs and a LAN connectivity policy

Cisco UCS maintains mutual exclusivity between connectivity policies and local vNIC and vHBA configuration in the service profile. You cannot have a combination of connectivity policies and locally created vNICs or vHBAs. When you include a LAN connectivity policy in a service profile, all existing vNIC configuration is erased, and when you include a SAN connectivity policy, all existing vHBA configuration in that service profile is erased.

Creating a LAN Connectivity Policy

Procedure

	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 lan-connectivity-policy** *policy-name*	Creates the specified LAN connectivity policy, and enters organization LAN connectivity policy mode. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	UCS-A /org/lan-connectivity-policy # **set descr** *policy-name*	(Optional) Adds a description to the policy. We recommend that you include information about where and how the policy should be used. Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote).
Step 4	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create lan-connectivity-policy LanConnect42
```

```
UCS-A /org/lan-connectivity-policy* # set descr "LAN connectivity policy"
UCS-A /org/lan-connectivity-policy* # commit-buffer
UCS-A /org/lan-connectivity-policy #
```

What to Do Next

Add one or more vNICs and/or iSCSI vNICs to this LAN connectivity policy.

Creating a vNIC for a LAN Connectivity Policy

If you are continuing from Creating a LAN Connectivity Policy, on page 294, begin this procedure at Step 3.

Procedure

	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 lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **create vnic** *vnic-name* [**eth-if** *eth-if-name*] [**fabric** {**a** | **b**}]	Creates a vNIC for the specified LAN connectivity policy. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 4	UCS-A /org/lan-connectivity-policy/vnic # **set fabric** {**a** | **a-b** | **b** | **b-a**}	Specifies the fabric to use for the vNIC. If you did not specify the fabric when you created the vNIC in Step 3, you have the option to specify it with this command. If you want this vNIC to be able to access the second fabric interconnect if the default one is unavailable, choose **a-b** (A is the primary) or **b-a** (B is the primary) . **Note** Do not enable fabric failover for the vNIC under the following circumstances: • If the Cisco UCS domain is running in Ethernet Switch Mode. vNIC fabric failover is not supported in that mode. If all Ethernet uplinks on one fabric interconnect fail, the vNICs do not fail over to the other. • If you plan to associate this vNIC to a server with an adapter that does not support fabric failover, such as the Cisco UCS 82598KR-CI 10-Gigabit Ethernet Adapter. If you do so, Cisco UCS Manager generates a configuration fault when you associate the service profile with the server.

	Command or Action	Purpose
Step 5	UCS-A /org/lan-connectivity-policy/vnic # **set adapter-policy** *policy-name*	Specifies the adapter policy to use for the vNIC.
Step 6	UCS-A /org/lan-connectivity-policy/vnic # **set identity** {**dynamic-mac** {*mac-addr* | **derived**} | **mac-pool** *mac-pool-name*}	Specifies the identity (MAC address) for the vNIC. You can set the identity using one of the following options: • Create a unique MAC address in the form *nn*:*nn*:*nn*:*nn*:*nn*:*nn*. • Derive the MAC address from one burned into the hardware at manufacture. • Assign a MAC address from a MAC pool.
Step 7	UCS-A /org/lan-connectivity-policy/vnic # **set mtu** *size-num*	Specifies the maximum transmission unit, or packet size, that this vNIC accepts. Enter an integer between 1500 and 9216. **Note** If the vNIC has an associated QoS policy, the MTU specified here must be equal to or less than the MTU specified in the associated QoS system class. If this MTU value exceeds the MTU value in the QoS system class, packets might get dropped during data transmission.
Step 8	UCS-A /org/lan-connectivity-policy/vnic # **set nw-control-policy** *policy-name*	Specifies the network control policy that the vNIC should use.
Step 9	UCS-A /org/lan-connectivity-policy/vnic # **set order** {*order-num* | **unspecified**}	Specifies the relative order for the vNIC.
Step 10	UCS-A /org/lan-connectivity-policy/vnic # **set pin-group** *group-name*	Specifies the LAN pin group that the vNIC should use.
Step 11	UCS-A /org/lan-connectivity-policy/vnic # **set qos-policy** *policy-name*	Specifies the quality of service policy that the vNIC should use.
Step 12	UCS-A /org/lan-connectivity-policy/vnic # **set stats-policy** *policy-name*	Specifies the statistics collection policy that the vNIC should use.
Step 13	UCS-A /org/lan-connectivity-policy/vnic # **set template-name** *policy-name*	Specifies the dynamic vNIC connectivity policy to use for the vNIC.
Step 14	UCS-A /org/lan-connectivity-policy/vnic # **set vcon** {**1** | **2** | **3** | **4** | **any**}	Assigns the vNIC to the specified vCon. Use the **any** keyword to have Cisco UCS Manager automatically assign the vNIC.

	Command or Action	Purpose
Step 15	UCS-A /org/lan-connectivity-policy/vnic # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to configure a vNIC for a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy* # create vnic vnic3 fabric a
UCS-A /org/lan-connectivity-policy/vnic* # set fabric a-b
UCS-A /org/lan-connectivity-policy/vnic* # set adapter-policy AdaptPol2
UCS-A /org/lan-connectivity-policy/vnic* # set identity mac-pool MacPool3
UCS-A /org/lan-connectivity-policy/vnic* # set mtu 8900
UCS-A /org/lan-connectivity-policy/vnic* # set nw-control-policy ncp5
UCS-A /org/lan-connectivity-policy/vnic* # set order 0
UCS-A /org/lan-connectivity-policy/vnic* # set pin-group EthPinGroup12
UCS-A /org/lan-connectivity-policy/vnic* # set qos-policy QosPol5
UCS-A /org/lan-connectivity-policy/vnic* # set stats-policy StatsPol2
UCS-A /org/lan-connectivity-policy/vnic* # set template-name VnicConnPol3
UCS-A /org/lan-connectivity-policy/vnic* # set vcon any
UCS-A /org/lan-connectivity-policy/vnic* # commit-buffer
UCS-A /org/lan-connectivity-policy/vnic #
```

What to Do Next

If desired, add another vNIC or an iSCSI vNIC to the LAN connectivity policy. If not, include the policy in a service profile or service profile template.

Deleting a vNIC from a LAN Connectivity Policy

Procedure

	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 lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **delete vnic** *vnic-name*	Deletes the specified vNIC from the LAN connectivity policy.
Step 4	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete a vNIC named vnic3 from a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy # delete vnic vnic3
UCS-A /org/lan-connectivity-policy* # commit-buffer
UCS-A /org/lan-connectivity-policy #
```

Creating an iSCSI vNIC for a LAN Connectivity Policy

If you are continuing from Creating a LAN Connectivity Policy, on page 294, begin this procedure at Step 3.

Before You Begin

The LAN connectivity policy must include an Ethernet vNIC that can be used as the overlay vNIC for the iSCSI device.

Procedure

	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 lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **create vnic-iscsi** *iscsi-vnic-name* .	Creates an iSCSI vNIC for the specified LAN connectivity policy. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 4	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set iscsi-adaptor-policy** *iscsi-adaptor-name*	(Optional) Specifies the iSCSI adapter policy that you have created for this iSCSI vNIC.
Step 5	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set auth-name** *authentication-profile-name*	(Optional) Sets the authentication profile to be used by the iSCSI vNIC. The authentication profile must already exist for it to be set. For more information, see Creating an Authentication Profile, on page 537.
Step 6	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set identity** { **dynamic-mac** {*dynamic-mac-address* | **derived** } | **mac-pool** *mac-pool-name* }	Specifies the MAC address for the iSCSI vNIC. **Note** The MAC address is set only for the Cisco UCS NIC M51KR-B Adapters.

	Command or Action	Purpose
Step 7	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set iscsi-identity** {**initiator-name** *initiator-name* | **initiator-pool-name** *iqn-pool-name*}	Specifies the name of the iSCSI initiator or the name of an IQN pool from which the iSCSI initiator name will be provided. The iSCSI initiator name can be up to 223 characters.
Step 8	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set overlay-vnic-name** *overlay-vnic-name*	Specifies the Ethernet vNIC that is used by the iSCSI device as the overlay vNIC. For more information, see Configuring a vNIC for a Service Profile, on page 610.
Step 9	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **create eth-if**	Creates an Ethernet interface for a VLAN assigned to the iSCSI vNIC.
Step 10	UCS-A /org/ex/vnic-iscsi/eth-if # **set vlanname** *vlan-name*	Specifies the VLAN name. The default VLAN is default. For the Cisco UCS M81KR Virtual Interface Card and the Cisco UCS VIC-1240 Virtual Interface Card, the VLAN that you specify must be the same as the native VLAN on the overlay vNIC. For the Cisco UCS M51KR-B Broadcom BCM57711 Adapter, the VLAN that you specify can be any VLAN assigned to the overlay vNIC.
Step 11	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to configure an iSCSI vNIC for a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy # create vnic-iscsi iSCSI1
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set iscsi-adaptor-policy iscsiboot
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set auth-name initauth
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set identity dynamic-mac derived
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set iscsi-identity initiator-name iSCSI1
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set overlay-vnic-name eth1
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # create eth-if
UCS-A /org/lan-connectivity-policy/vnic-iscsi/eth-if* # set vlanname default
UCS-A /org/lan-connectivity-policy/vnic-iscsi/eth-if* # commit buffer
UCS-A /org/lan-connectivity-policy/vnic-iscsi/eth-if
```

What to Do Next

If desired, add another iSCI vNIC or a vNIC to the LAN connectivity policy. If not, include the policy in a service profile or service profile template.

Deleting an iSCSI vNIC from a LAN Connectivity Policy

Procedure

	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 lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **delete vnic-iscsi** *iscsi-vnic-name*	Deletes the specified iSCSI vNIC from the LAN connectivity policy.
Step 4	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI vNIC named iscsivnic3 from a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy # delete vnic-iscsi iscsivnic3
UCS-A /org/lan-connectivity-policy* # commit-buffer
UCS-A /org/lan-connectivity-policy #
```

Deleting a LAN Connectivity Policy

If you delete a LAN connectivity policy that is included in a service profile, it also deletes all vNICs and iSCSI vNICs from that service profile, and disrupt LAN data traffic for the server associated with the service profile.

Procedure

	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 lan-connectivity-policy** *policy-name*	Deletes the specified LAN connectivity policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the LAN connectivity policy named LanConnectiSCSI42 from the root organization and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete lan-connectivity-policy LanConnectiSCSI42
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Network Control Policies

Network Control Policy

This policy configures the network control settings for the Cisco UCS domain, including the following:

- Whether the Cisco Discovery Protocol (CDP) is enabled or disabled
- How the virtual interface (VIF) behaves if no uplink port is available in end-host mode
- The action that Cisco UCS Manager takes on the remote Ethernet interface, vEthernet interface , or vFibre Channel interface when the associated border port fails
- Whether the server can use different MAC addresses when sending packets to the fabric interconnect
- Whether MAC registration occurs on a per-VNIC basis or for all VLANs

Action on Uplink Fail

By default, the **Action on Uplink Fail** property in the network control policy is configured with a value of link-down. For adapters such as the Cisco UCS M81KR Virtual Interface Card, this default behavior directs Cisco UCS Manager to bring the vEthernet or vFibre Channel interface down if the associated border port fails. For Cisco UCS systems using a non-VM-FEX capable converged network adapter that supports both Ethernet and FCoE traffic, such as Cisco UCS CNA M72KR-Q and the Cisco UCS CNA M72KR-E, this default behavior directs Cisco UCS Manager to bring the remote Ethernet interface down if the associated border port fails. In this scenario, any vFibre Channel interfaces that are bound to the remote Ethernet interface are brought down as well.

Note If your implementation includes those types of non-VM-FEX capable converged network adapters mentioned in this section and the adapter is expected to handle both Ethernet and FCoE traffic, we recommend that you configure the **Action on Uplink Fail** property with a value of warning. Note that this configuration might result in an Ethernet teaming driver not being able to detect a link failure when the border port goes down.

MAC Registration Mode

MAC addresses are installed only on the native VLAN by default, which maximizes the VLAN port count in most implementations.

Note If a trunking driver is being run on the host and the interface is in promiscuous mode, we recommend that you set the MAC Registration Mode to All VLANs.

NIC Teaming and Port Security

NIC teaming is a grouping together of network adapters to build in redundancy, and is enabled on the host. This teaming or bonding facilitates various functionalities, including load balancing across links and failover. When NIC teaming is enabled and events such as failover or reconfiguration take place, MAC address conflicts and movement may happen.

Port security, which is enabled on the fabric interconnect side, prevents MAC address movement and deletion. Therefore, you must not enable port security and NIC teaming together.

Configuring Link Layer Discovery Protocol for Fabric Interconnect vEthernet Interfaces

Cisco UCS Manager Release 2.2.4 allows you to enable and disable LLDP on a vEthernet interface. You can also retrieve information about these LAN uplink neighbors. This information is useful while learning the topology of the LAN connected to the UCS system and while diagnosing any network connectivity issues from the Fabric Interconnect (FI). The FI of a UCS system is connected to LAN uplink switches for LAN connectivity and to SAN uplink switches for storage connectivity. When using Cisco UCS with Cisco Application Centric Infrastructure (ACI), LAN uplinks of the FI are connected to ACI leaf nodes. Enabling LLDP on a vEthernet interface will help the Application Policy Infrastructure Controller (APIC) to identify the servers connected to the FI by using vCenter.

To permit the discovery of devices in a network, support for Link Layer Discovery Protocol (LLDP), a vendor-neutral device discovery protocol that is defined in the IEEE 802.1ab standard, is introduced. LLDP is a one-way protocol that allows network devices to advertise information about themselves to other devices on the network. LLDP transmits information about the capabilities and current status of a device and its interfaces. LLDP devices use the protocol to solicit information only from other LLDP devices.

You can enable or disable LLDP on a vEthernet interface based on the Network Control Policy (NCP) that is applied on the vNIC in the service profile.

Configuring a Network Control Policy

MAC address-based port security for Emulex converged Network Adapters (N20-AE0102) is not supported. When MAC address-based port security is enabled, the fabric interconnect restricts traffic to packets that contain the MAC address that it first learns. This is either the source MAC address used in the FCoE Initialization Protocol packet, or the MAC address in an ethernet packet, whichever is sent first by the adaptor. This configuration can result in either FCoE or Ethernet packets being dropped.

Procedure

	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 nw-ctrl-policy** *policy-name*	Creates the specified network control policy, and enters organization network control policy mode.
Step 3	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **cdp**	Disables or enables Cisco Discovery Protocol (CDP).
Step 4	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **lldp transmit**	Disables or enables the transmission of LLDP packets on an interface.
Step 5	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **lldp receive**	Disables or enables the reception of LLDP packets on an interface.
Step 6	UCS-A /org/nw-ctrl-policy # **set uplink-fail-action** {**link-down** | **warning**}	Specifies the action to be taken when no uplink port is available in end-host mode. Use the **link-down** keyword to change the operational state of a vNIC to down when uplink connectivity is lost on the fabric interconnect, and facilitate fabric failover for vNICs. Use the **warning** keyword to maintain server-to-server connectivity even when no uplink port is available, and disable fabric failover when uplink connectivity is lost on the fabric interconnect. The default uplink failure action is link-down.
Step 7	UCS-A /org/nw-ctrl-policy # **set mac-registration-mode**{**all-host-vlans** | **only-native-vlan**	Whether adapter-registered MAC addresses are added only to the native VLAN associated with the interface or added to all VLANs associated with the interface. This can be one of the following: • **Only Native Vlan—**MAC addresses are only added to the native VLAN. This option is the default, and it maximizes the port+VLAN count. • **All Host Vlans—**MAC addresses are added to all VLANs with which they are associated. Select this option if your VLANs are configured to use trunking but are *not* running in Promiscuous mode.
Step 8	UCS-A /org/nw-ctrl-policy # **create mac-security**	Enters organization network control policy MAC security mode
Step 9	UCS-A /org/nw-ctrl-policy/mac-security # **set forged-transmit** {**allow** | **deny**}	Allows or denies the forging of MAC addresses when sending traffic. MAC security is disabled when forged MAC addresses are allowed, and MAC security is enabled when forged MAC addresses are denied. By default,

	Command or Action	Purpose
		forged MAC addresses are allowed (MAC security is disabled).
Step 10	UCS-A /org/nw-ctrl-policy/mac-security # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a network control policy named ncp5, enable CDP, enable LLDP transmit and LLDP recive, set the uplink fail action to link-down, deny forged MAC addresses (enable MAC security), and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create nw-ctrl-policy ncp5
UCS-A /org/nw-ctrl-policy* # enable cdp
UCS-A /org/nw-ctrl-policy* # enable lldp transmit
UCS-A /org/nw-ctrl-policy* # enable lldp receive
UCS-A /org/nw-ctrl-policy* # set uplink-fail-action link-down
UCS-A /org/nw-ctrl-policy* # create mac-security
UCS-A /org/nw-ctrl-policy/mac-security* # set forged-transmit deny
UCS-A /org/nw-ctrl-policy/mac-security* # commit-buffer
UCS-A /org/nw-ctrl-policy/mac-security #
```

Displaying Network Control Policy Details

Procedure

	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 nw-ctrl-policy** {**default** | *policy-name*}	Enters organization network control policy mode for the specified network control policy.
Step 3	UCS-A /org/nw-ctrl-policy # **show detail**	Displays details about the specified network control policy.

The following example shows how to display the details of a network control policy named ncp5:

```
UCS-A# scope org /
UCS-A /org # scope nw-ctrl-policy ncp5
UCS-A /org/nw-ctrl-policy* # show detail

Network Control Policy:
     Name: ncp5
     CDP: Enabled
     LLDP Transmit: Enabled
     LLDP Receive: Enabled
     Uplink fail action: Link Down
     Adapter MAC Address Registration: Only Native Vlan
     Policy Owner: Local
     Description:
```

```
UCS-A /org/nw-ctrl-policy #
```

Deleting a Network Control Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A /org # **delete nwctrl-policy** *policy-name*	Deletes the specified network control policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the network control policy named ncp5 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete nwctrl-policy ncp5
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Multicast Policies

Multicast Policy

This policy is used to configure Internet Group Management Protocol (IGMP) snooping and IGMP querier. IGMP Snooping dynamically determines hosts in a VLAN that should be included in particular multicast transmissions. You can create, modify, and delete a multicast policy that can be associated to one or more VLANs. When a multicast policy is modified, all VLANs associated with that multicast policy are re-processed to apply the changes. For private VLANs, you can set a multicast policy for primary VLANs but not for their associated isolated VLANs due to a Cisco NX-OS forwarding implementation.

By default, IGMP snooping is enabled and IGMP querier is disabled. When IGMP snooping is enabled, the fabric interconnects send the IGMP queries only to the hosts. They do not send IGMP queries to the upstream network. To send IGMP queries to the upstream, do one of the following:

- Configure IGMP querier on the upstream fabric interconnect with IGMP snooping enabled
- Disable IGMP snooping on the upstream fabric interconnect
- Change the fabric interconnects to switch mode

The following limitations and guidelines apply to multicast policies:

- On a 6200 series fabric interconnect, user-defined multicast policies can also be assigned along with the default multicast policy.
- Only the default multicast policy is allowed for a global VLAN.

- If a Cisco UCS domain includes 6300 and 6200 series fabric interconnects, any multicast policy can be assigned.
- We highly recommend you use the same IGMP snooping state on the fabric interconnects and the associated LAN switches. For example, if IGMP snooping is disabled on the fabric interconnects, it should be disabled on any associated LAN switches as well.

Creating a Multicast Policy

A multicast policy can be created only in the root organization and not in a sub-organization.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **create mcast-policy** *policy-name*	Creates a multicast policy with the specified policy name, and enters organization multicast policy mode.
Step 3	UCS-A /org/mcast-policy* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # create mcast-policy policy1
UCS-A /org/mcast-policy* # commit-buffer
UCS-A /org/mcast-policy #
```

Configuring IGMP Snooping Parameters

You can enable or disable IGMP snooping for a multicast policy. By default, the IGMP snooping state is enabled for a multicast policy. You can also set the IGMP snooping querier state and IPv4 address for the multicast policy.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **create mcast-policy** *policy-name*	Creates a new multicast policy with the specified policy name, and enters organization multicast policy mode.

	Command or Action	Purpose
Step 3	UCS-A /org/mcast-policy* # **set querier**{**enabled** | **disabled**}	Enables or disables IGMP snooping querier. By default, IGMP snooping querier is disabled for a multicast policy.
Step 4	UCS-A /org/mcast-policy* # **set querierip** *IGMP snooping querier IPv4 address*	Specifies the IPv4 address for the IGMP snooping querier.
Step 5	UCS-A /org/mcast-policy* # **set snooping**{**enabled** | **disabled**}	Enables or disables IGMP snooping. By default, IGMP snooping is enabled for a multicast policy.
Step 6	UCS-A /org/mcast-policy* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create and enter a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # create mcast-policy policy1
UCS-A /org/mcast-policy* # set querier enabled
UCS-A /org/mcast-policy* # set querierip 1.2.3.4
UCS-A /org/mcast-policy* # set snooping enabled
UCS-A /org/mcast-policy* # commit-buffer
UCS-A /org/mcast-policy #
```

Modifying Multicast Policy Parameters

You can modify an existing multicast policy to change the state of IGMP snooping or IGMP snooping querier. When a multicast policy is modified, all VLANs associated with that multicast policy are re-processed to apply the changes.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **scope mcast-policy** *policy-name*	Enters organization multicast policy mode.
Step 3	UCS-A /org/mcast-policy* # **set querier**{**enabled** | **disabled**}	Enables or disables IGMP snooping querier. By default, IGMP snooping querier is disabled for a multicast policy.
Step 4	UCS-A /org/mcast-policy* # **set querierip** *IGMP snooping querier IPv4 address*	Specifies the IPv4 address for the IGMP snooping querier.
Step 5	UCS-A /org/mcast-policy* # **set snooping**{**enabled** | **disabled**}	Enables or disables IGMP snooping. By default, IGMP snooping is enabled for a multicast policy.

	Command or Action	Purpose
Step 6	UCS-A /org/mcast-policy* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # scope mcast-policy policy1
UCS-A /org/mcast-policy* # set querier enabled
UCS-A /org/mcast-policy* # set querierip 1.2.3.4
UCS-A /org/mcast-policy* # set snooping enabled
UCS-A /org/mcast-policy* # commit-buffer
UCS-A /org/mcast-policy #
```

Assigning a VLAN Multicast Policy

You can set a multicast policy for a VLAN in the Ethernet uplink fabric mode. You cannot set a multicast policy for an isolated VLAN.

Before You Begin

Create a VLAN.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric**{**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric interconnect.
Step 3	UCS-A /eth-uplink/fabric # **scope vlan** *vlan-name*	Enters Ethernet uplink fabric VLAN mode.
Step 4	UCS-A /eth-uplink/fabric/vlan # **set mcastpolicy** *policy-name*	Assigns a multicast policy for the VLAN.
Step 5	UCS-A /eth-uplink/fabric/vlan # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets a named VLAN accessible to one fabric interconnect and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope vlan vlan1
UCS-A /eth-uplink/fabric/vlan # set mcastpolicy policy1
UCS-A /eth-uplink/fabric/vlan* # commit-buffer
UCS-A /eth-uplink/fabric/vlan #
```

Deleting a Multicast Policy

Note If you assigned a non-default (user-defined) multicast policy to a VLAN and then delete that multicast policy, the associated VLAN inherits the multicast policy settings from the default multicast policy until the deleted policy is re-created.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **delete mcast-policy** *policy-name*	Deletes a multicast policy with the specified policy name.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # delete mcast-policy policy1
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring LACP Policies

LACP Policy

Link Aggregation combines multiple network connections in parallel to increase throughput and to provide redundancy. Link aggregation control protocol (LACP) provides additional benefits for these link aggregation groups. Cisco UCS Manager enables you to configure LACP properties using LACP policy.

You can configure the following for a lacp policy:

- **Suspended-individual**: If you do not configure the ports on an upstream switch for lacp, the fabric interconnects treat all ports as uplink Ethernet ports to forward packets. You can place the lacp port in suspended state to avoid loops. When you set suspend-individual on a port-channel with LACP, if a port that is part of the port-channel does not receive PDUs from the peer port, it will go into suspended state.
- **Timer values**: You can configure rate-fast or rate-normal. In rate-fast configuration, the port is expected to receive 1 PDU every 1 second from the peer port. The time out for this is 3 seconds. In rate-normal configuration, the port is expected to receive 1 PDU every 30 seconds. The timeout for this is 90 seconds.

System creates a default LACP policy at system start up. You can modify this policy or create a new policy. You can also apply one LACP policy to multiple port-channels.

Creating a LACP Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters the root organization mode.
Step 2	UCS-A /org # **create lacppolicy***policy nam*.	Creates the specified lacp policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates the lacp policy and commits the transaction:

```
UCS-A# scope org
UCS-A /org # create lacppolicy lacp1
UCS-A /org* # commit-buffer
UCS-A /org #
```

Editing a LACP Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters the root organization mode.
Step 2	UCS-A /org # **scope lacppolicy** *policy-name* .	Enters the specified lacp policy.
Step 3	UCS-A /org/lacp policy/ policy-name # **set suspend-individual** *true* .	Sets suspend individual for the policy.
Step 4	UCS-A /org/lacp policy/ policy-name # **set lacp-rate** *fast* .	Sets LACP rate for the policy.
Step 5	UCS-A /org/lacp policy/ policy-name # **commit-buffer**	Commits the transaction to the system configuration.

The following example modifies the lacp policy and commits transaction:

```
UCS-A# scope org
UCS-A/org # scope lacppolicy policy-name
UCS-A /org/lacp policy policy-name# set suspend-individual true
UCS-A/prg/policy policy-name# set lacp-rate fast
UCS-A /org* # commit-buffer
UCS-A /org #
```

Assigning LACP Policy to Port-Channels

Default lacp policy is assigned to port channels by default. You can assign a different lacp policy to the port channel. If the assigned policy does not exist, system generates a fault. You can create the same policy to clear the fault.

Note You can assign lacp policy to port-channels, FCoE port-channels, and ethernet storage port-channels. This procedures describes assigning the lacp policy to port-channels.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric**	Enters the fabric mode.
Step 3	UCS-A /eth-uplink/fabric # **scope port-channel**	Enters the port-channel mode.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **set lacp-policy-name***policy-name*	Specifies the lacp policy for this port-channel.
Step 5	UCS-A /eth-uplink/ fabric/port-channel **commit-buffer**	Commits the transaction to the system.

The following example shows assigning a lacp policy to a port-channel:

```
UCS-A# scope eth-uplink
UCS-A UCS-A/eth-uplink # scope fabric
UCS-A UCS-A/eth-uplink/facric # scope port-channel
UCS-A UCS-A/eth-uplink/port-channel# set lacp-policy-name
UCS-A UCS-A/eth-uplink/port-channel# commit-buffer
```

Configuring UDLD Link Policies

Understanding UDLD

UniDirectional Link Detection (UDLD) is a Layer 2 protocol that enables devices connected through fiber-optic or twisted-pair Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists. All connected devices must support UDLD for the protocol to successfully identify and disable unidirectional links. When UDLD detects a unidirectional link, it marks the link as unidirectional. Unidirectional links can cause a variety of problems, including spanning-tree topology loops.

UDLD works with the Layer 1 mechanisms to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected interfaces. When you enable both autonegotiation and UDLD, the Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

A unidirectional link occurs whenever traffic sent by a local device is received by its neighbor but traffic from the neighbor is not received by the local device.

Modes of Operation

UDLD supports two modes of operation: normal (the default) and aggressive. In normal mode, UDLD can detect unidirectional links due to misconnected interfaces on fiber-optic connections. In aggressive mode, UDLD can also detect unidirectional links due to one-way traffic on fiber-optic and twisted-pair links and to misconnected interfaces on fiber-optic links.

In normal mode, UDLD detects a unidirectional link when fiber strands in a fiber-optic interface are misconnected and the Layer 1 mechanisms do not detect this misconnection. If the interfaces are connected correctly but the traffic is one way, UDLD does not detect the unidirectional link because the Layer 1 mechanism, which is supposed to detect this condition, does not do so. In case, the logical link is considered undetermined, and UDLD does not disable the interface. When UDLD is in normal mode, if one of the fiber strands in a pair is disconnected and autonegotiation is active, the link does not stay up because the Layer 1 mechanisms did not detect a physical problem with the link. In this case, UDLD does not take any action, and the logical link is considered undetermined.

UDLD aggressive mode is disabled by default. Configure UDLD aggressive mode only on point-to-point links between network devices that support UDLD aggressive mode. With UDLD aggressive mode enabled, when a port on a bidirectional link that has a UDLD neighbor relationship established stops receiving UDLD packets, UDLD tries to reestablish the connection with the neighbor and administratively shuts down the affected port. UDLD in aggressive mode can also detect a unidirectional link on a point-to-point link on which no failure between the two devices is allowed. It can also detect a unidirectional link when one of the following problems exists:

- On fiber-optic or twisted-pair links, one of the interfaces cannot send or receive traffic.
- On fiber-optic or twisted-pair links, one of the interfaces is down while the other is up.
- One of the fiber strands in the cable is disconnected.

Methods to Detect Unidirectional Links

UDLD operates by using two mechanisms:

- Neighbor database maintenance

  UDLD learns about other UDLD-capable neighbors by periodically sending a hello packet (also called an advertisement or probe) on every active interface to keep each device informed about its neighbors. When the switch receives a hello message, it caches the information until the age time (hold time or time-to-live) expires. If the switch receives a new hello message before an older cache entry ages, the switch replaces the older entry with the new one.

  UDLD clears all existing cache entries for the interfaces affected by the configuration change whenever an interface is disabled and UDLD is running, whenever UDLD is disabled on an interface, or whenever the switch is reset. UDLD sends at least one message to inform the neighbors to flush the part of their caches affected by the status change. The message is intended to keep the caches synchronized.

- Event-driven detection and echoing

  UDLD relies on echoing as its detection mechanism. Whenever a UDLD device learns about a new neighbor or receives a resynchronization request from an out-of-sync neighbor, it restarts the detection window on its side of the connection and sends echo messages in reply. Because this behavior is the same on all UDLD neighbors, the sender of the echoes expects to receive an echo in reply.

If the detection window ends and no valid reply message is received, the link might shut down, depending on the UDLD mode. When UDLD is in normal mode, the link might be considered undetermined and might not be shut down. When UDLD is in aggressive mode, the link is considered unidirectional, and the interface is shut down.

If UDLD in normal mode is in the advertisement or in the detection phase and all the neighbor cache entries are aged out, UDLD restarts the link-up sequence to resynchronize with any potentially out-of-sync neighbors.

If you enable aggressive mode when all the neighbors of a port have aged out either in the advertisement or in the detection phase, UDLD restarts the link-up sequence to resynchronize with any potentially out-of-sync neighbor. UDLD shuts down the port if, after the fast train of messages, the link state is still undetermined.

UDLD Configuration Guidelines

The following guidelines and recommendations apply when you configure UDLD:

- A UDLD-capable interface also cannot detect a unidirectional link if it is connected to a UDLD-incapable port of another switch.
- When configuring the mode (normal or aggressive), make sure that the same mode is configured on both sides of the link.
- UDLD should be enabled only on interfaces that are connected to UDLD capable devices. The following interface types are supported:
  - Ethernet uplink
  - FCoE uplink
  - Ethernet uplink port channel member
  - FCoE uplink port channel member

Configuring a Link Profile

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** /	Enters the root organization mode.
Step 2	UCS-A /org # **create eth-link-profile** *link-profile-name*	Creates a link profile with the specified name, and enters link profile mode.
Step 3	UCS-A /org/eth-link-profile # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /org/eth-link-profile # **exit**	Returns to the previous mode.
Step 5	UCS-A /org # **scope eth-link-profile** *link-profile-name*	Enters link profile mode for the specified link profile.

	Command or Action	Purpose
Step 6	UCS-A /org/eth-link-profile # **set udld-link-policy** *link-policy-name*	Assigns the specified UDLD link policy to the link profile.
Step 7	UCS-A /org/eth-link-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a link profile called LinkProfile1 and assign the default UDLD link policy.

```
UCS-A# scope org /
UCS-A /chassis/org # create eth-link-profile LinkProfile1
UCS-A /chassis/org/eth-link-profile* # commit-buffer
UCS-A /chassis/org/eth-link-profile # exit
UCS-A /chassis/org # scope eth-link-profile LinkProfile1
UCS-A /chassis/org/eth-link-profile # set udld-link-policy default
UCS-A /chassis/org/eth-link-profile* # commit-buffer
```

Configuring a UDLD Link Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** /	Enters the root organization mode.
Step 2	UCS-A /org # **create udld-link-policy** *link-policy-name*	Creates a UDLD link policy with the specified name, and enters UDLD link policy mode.
Step 3	UCS-A /org/udld-link-policy # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /org/udld-link-policy # **exit**	Returns to the previous mode.
Step 5	UCS-A /org # **scope udld-link-policy** *link-policy-name*	Enters UDLD link policy mode for the specified UDLD link policy.
Step 6	UCS-A /org/udld-link-policy # **set mode** {**aggressive** | **normal**}	Specifies the mode for the UDLD link policy.
Step 7	UCS-A /org/udld-link-policy # **set admin-state** {**disabled** | **enabled**}	Disables or enables UDLD on the interface.
Step 8	UCS-A /org/udld-link-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a link profile called UDLDPol1, sets the mode to aggressive, and enables UDLD on the interface.

```
UCS-A# scope org /
UCS-A /chassis/org # create udld-link-policy UDLDPol1
```

```
UCS-A /chassis/org/udld-link-policy* # commit-buffer
UCS-A /chassis/org/udld-link-policy # exit
UCS-A /chassis/org # scope udld-link-policy UDLDPol1
UCS-A /chassis/org/udld-link-policy # set mode aggressive
UCS-A /chassis/org/udld-link-policy* # set admin-state enabled
UCS-A /chassis/org/udld-link-policy* # commit-buffer
UCS-A /chassis/org/udld-link-policy #
```

Modifying the UDLD System Settings

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** /	Enters the root organization mode.
Step 2	UCS-A /org # **show udld-policy**	Displays the current UDLD system settings.
Step 3	UCS-A /org # **scope udld-policy default**	Enters UDLD policy mode for the global UDLD policy.
Step 4	UCS-A /org/udld-policy # **set message-interval** *seconds*	Specifies the time interval (in seconds) between UDLD probe messages on ports that are in advertisement mode. Enter an integer between 7 and 60. The default is 15 seconds.
Step 5	UCS-A /org/udld-policy # **set recovery-action** [**reset** | **none**]	Specifies the action to be taken on any ports that are disabled when UDLD aggressive mode is enabled. The default is none.
Step 6	UCS-A /org/udld-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to update the default UDLD system settings for a 30 second time interval.

```
UCS-A# scope org /
UCS-A /chassis/org # show udld-policy

UDLD system settings:
    Name       Message interval (sec) Recovery action
    ---------- ---------------------- ---------------
    default    15                     None

UCS-A /chassis/org # scope udld-policy default
UCS-A /chassis/org/udld-policy # set message-interval 30
UCS-A /chassis/org/udld-policy* # commit-buffer
UCS-A /chassis/org/udld-policy #
```

Assigning a Link Profile to a Port Channel Ethernet Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **scope port-channel** *port-chan-id*	Enters Ethernet uplink fabric port channel mode for the specified port channel.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **scope member-port** *slot-id port-id*	Enters Ethernet server fabric, fabric port channel mode for the specified member port.
Step 5	UCS-A /eth-uplink/fabric/port-channel/member-port # **set eth-link-profile** *link-profile-name*	Assigns the specified link profile.
Step 6	UCS-A /eth-uplink/fabric/port-channel/member-port # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to assign link profile LinkProfile1 to a port channel Ethernet interface:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope port-channel 88
UCS-A /eth-uplink/fabric/port-channel # scope member-port 1 31
UCS-A /eth-uplink/fabric/port-channel/member-port # set eth-link-profile LinkProfile1
UCS-A /eth-uplink/fabric/port-channel/member-port* # commit-buffer
UCS-A /eth-uplink/fabric/port-channel/member-port #
```

Assigning a Link Profile to a Port Channel FCoE Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope fcoe-port-channel** *port-chan-id*	Enters Fibre Channel uplink fabric port channel mode for the specified port channel.

	Command or Action	Purpose
Step 4	UCS-A /fc-uplink/fabric/fcoe-port-channel # **scope fcoe-member-port** *slot-id port-id*	Enters Fibre Channel server fabric, fabric port channel mode for the specified member port.
Step 5	UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # **set eth-link-profile** *link-profile-name*	Assigns the specified link profile.
Step 6	UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to assign link profile LinkProfile1 to a port channel FCoE interface:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope fcoe-port-channel 192
UCS-A /fc-uplink/fabric/fcoe-port-channel # scope fcoe-member-port 1 20
UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # set eth-link-profile LinkProfile1
UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port* # commit-buffer
UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port #
```

Assigning a Link Profile to an Uplink Ethernet Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **scope interface** *slot-num port num*	Enters the interface command mode for the specified uplink port.
Step 4	UCS-A /eth-uplink/fabric/interface # **set eth-link-profile** *link-profile-name*	Assigns the specified link profile.
Step 5	UCS-A /eth-uplink/fabric/interface # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to assign link profile LinkProfile1 to an uplink Ethernet interface:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope interface 2 2
UCS-A /eth-uplink/fabric/interface # set eth-link-profile LinkProfile1
UCS-A /eth-uplink/fabric/interface* # commit-buffer
UCS-A /eth-uplink/fabric/interface #
```

Assigning a Link Profile to an Uplink FCoE Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope fcoeinterface** *slot-num port num*	Enters the Fibre Channel interface command mode for the specified uplink port.
Step 4	UCS-A /fc-uplink/fabric/fcoeinterface # **set eth-link-profile** *link-profile-name*	Assigns the specified link profile.
Step 5	UCS-A /fc-uplink/fabric/fcoeinterface # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to assign link profile LinkProfile1 to an uplink FCoE interface:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope fcoeinterface 2 2
UCS-A /fc-uplink/fabric/fcoeinterface # set eth-link-profile LinkProfile1
UCS-A /fc-uplink/fabric/fcoeinterface* # commit-buffer
UCS-A /fc-uplink/fabric/fcoeinterface #
```

Configuring VMQ Connection Policies

VMQ Connection Policy

Cisco UCS Manager enables you to configure VMQ connection policy for a vNIC. VMQ provides improved network performance to the entire management operating system. Configuring a VMQ vNIC connection policy involves the following:

- Create a VMQ connection policy
- Create a static vNIC in a service profile
- Apply the VMQ connection policy to the vNIC

If you want to configure the VMQ vNIC on a service profile for a server, at least one adapter in the server must support VMQ. Make sure the servers have at least one the following adapters installed:

- UCS-VIC-M82-8P
- UCSB-MLOM-40G-01
- UCSC-PCIE-CSC-02

The following are the supported Operating Systems for VMQ:

- Windows 2012
- Windows 2012R2

You can apply only any one of the vNIC connection policies on a service profile at any one time. Make sure to select one of the three options such as Dynamic, usNIC or VMQ connection policy for the vNIC. When a VMQ vNIC is configured on service profile, make sure you have the following settings:

- Select SRIOV in the BIOS policy.
- Select Windows in the Adapter policy.

Creating a VMQ Connection Policy

Procedure

	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 # **create vmq-conn-policy** *policy-name*	Specifies the name for this VMQ connection policy.
Step 3	UCS-A /org/vmq-conn-policy* # **set queue-count***queue count*	Specifies the queue count for the VMQ connection policy.
Step 4	UCS-A /org/vmq-conn-policy* # **set interrupt-count***interrupt count*	Specifies the interrupt count for the VMQ connection policy.
Step 5	UCS-A /org/vmq-conn-policy* # **commit-buffer**	Commits the transaction to the system.

The following example creates a VMQ connection policy:

```
UCS-A# scope org
UCS-A /org # create vmq-conn-policy policy name
UCS-A /org/vmq-conn-policy* # set queue-count queue count (number)
UCS-A /org/vmq-conn-policy* # set interrupt-count queue count (number)
UCS-A /org/vmq-conn-policy* # commit-buffer
```

NetQueue

Information About NetQueue

NetQueue improves traffic performance by providing a network adapter with multiple receive queues. These queues allow the data interrupt processing that is associated with individual virtual machines to be grouped.

Note NetQueue is supported on servers running VMware ESXi operating systems.

Configuring NetQueue

Procedure

Step 1 Create a Virtual Machine Queue (VMQ) connection policy.

Step 2 Configure NetQueues in a service profile by selecting the VMQ connection policy.
Use the following when you are configuring NetQueue:

- The default ring size is rx512, tx256
- The interrupt count on each VNIC is VMQ count x 2 +2

  Note The number of interrupts depends on the number of NetQueues enabled.

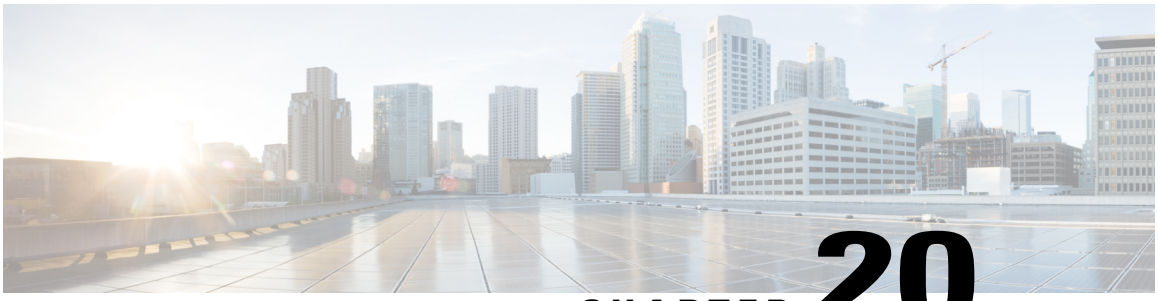
- The driver supports up to 16 NetQueues per port for standard frame configurations.

  Note VMware recommends that you use up to eight NetQueues per port for standard frame configurations.

- NetQueue should be enabled only on MSIX systems.
- You should disable NetQueue on 1 GB NICs.

Step 3 Enable the MSIX mode in the adapter policy for NetQueue.

Step 4 Associate the service profile with the server.



CHAPTER 20

Configuring Upstream Disjoint Layer-2 Networks

This chapter includes the following sections:

Upstream Disjoint Layer-2 Networks

Upstream disjoint layer-2 networks (disjoint L2 networks) are required if you have two or more Ethernet clouds that never connect, but must be accessed by servers or virtual machines located in the same Cisco UCS domain. For example, you could configure disjoint L2 networks if you require one of the following:

- Servers or virtual machines to access a public network and a backup network
- Servers or virtual machines for more than one customer are located in the same Cisco UCS domain, and that need to access the L2 networks for both customers in a multi-tenant system

Note By default, data traffic in Cisco UCS works on a principle of mutual inclusion. All traffic for all VLANs and upstream networks travels along all uplink ports and port channels. If you have upgraded from a release that does not support upstream disjoint layer-2 networks, you must assign the appropriate uplink interfaces to your VLANs, or traffic for those VLANs continues to flow along all uplink ports and port channels.

The configuration for disjoint L2 networks works on a principle of selective exclusion. Traffic for a VLAN that is designated as part of a disjoint network can only travel along an uplink Ethernet port or port channel that is specifically assigned to that VLAN, and is selectively excluded from all other uplink ports and port

channels. However, traffic for VLANs that are not specifically assigned to an uplink Ethernet port or port channel can still travel on all uplink ports or port channels, including those that carry traffic for the disjoint L2 networks.

In Cisco UCS, the VLAN represents the upstream disjoint L2 network. When you design your network topology for disjoint L2 networks, you must assign uplink interfaces to VLANs not the reverse.

For information about the maximum number of supported upstream disjoint L2 networks, see the appropriate *Cisco UCS Configuration Limits for Cisco UCS Manager Guide*.

Guidelines for Configuring Upstream Disjoint L2 Networks

When you plan your configuration for upstream disjoint L2 networks, consider the following:

Ethernet Switching Mode Must Be End-Host Mode

Cisco UCS only supports disjoint L2 networks when the Ethernet switching mode of the fabric interconnects is configured for end-host mode. You cannot connect to disjoint L2 networks if the Ethernet switching mode of the fabric interconnects is switch mode.

Symmetrical Configuration Is Recommended for High Availability

If a Cisco UCS domain is configured for high availability with two fabric interconnects, we recommend that both fabric interconnects are configured with the same set of VLANs.

VLAN Validity Criteria Are the Same for Uplink Ethernet Ports and Port Channels

The VLAN used for the disjoint L2 networks must be configured and assigned to an uplink Ethernet port or uplink Ethernet port channel. If the port or port channel does not include the VLAN, Cisco UCS Manager considers the VLAN invalid and does the following:

- Displays a configuration warning in the **Status Details** area for the server.
- Ignores the configuration for the port or port channel and drops all traffic for that VLAN.

Note The validity criteria are the same for uplink Ethernet ports and uplink Ethernet port channels. Cisco UCS Manager does not differentiate between the two.

Overlapping VLANs Are Not Supported

Cisco UCS does not support overlapping VLANs in disjoint L2 networks. You must ensure that each VLAN only connects to one upstream disjoint L2 domain.

Each vNIC Can Only Communicate with One Disjoint L2 Network

A vNIC can only communicate with one disjoint L2 network. If a server needs to communicate with multiple disjoint L2 networks, you must configure a vNIC for each of those networks.

To communicate with more than two disjoint L2 networks, a server must have a Cisco VIC adapter that supports more than two vNICs.

Appliance Port Must Be Configured with the Same VLAN as Uplink Ethernet Port or Port Channel

For an appliance port to communicate with a disjoint L2 network, you must ensure that at least one uplink Ethernet port or port channel is in the same network and is therefore assigned to the same VLANs that are used by the appliance port. If Cisco UCS Manager cannot identify an uplink Ethernet port or port channel that includes all VLANs that carry traffic for an appliance port, the appliance port experiences a pinning failure and goes down.

For example, a Cisco UCS domain includes a global VLAN named vlan500 with an ID of 500. vlan500 is created as a global VLAN on the uplink Ethernet port. However, Cisco UCS Manager does not propagate this VLAN to appliance ports. To configure an appliance port with vlan500, you must create another VLAN named vlan500 with an ID of 500 for the appliance port. You can create this duplicate VLAN in the **Appliances** node on the **LAN** tab of the Cisco UCS Manager GUI or the **eth-storage** scope in the Cisco UCS Manager CLI. If you are prompted to check for VLAN Overlap, accept the overlap and Cisco UCS Manager creates the duplicate VLAN for the appliance port.

Default VLAN 1 Cannot Be Configured Explicitly on an Uplink Ethernet Port or Port Channel

Cisco UCS Manager implicitly assigns default VLAN 1 to all uplink ports and port channels. Even if you do not configure any other VLANs, Cisco UCS uses default VLAN 1 to handle data traffic for all uplink ports and port channels.

Note After you configure VLANs in a Cisco UCS domain, default VLAN 1 remains implicitly on all uplink ports and port channels. You cannot explicitly assign default VLAN 1 to an uplink port or port channel, nor can you remove it from an uplink port or port channel.

If you attempt to assign default VLAN 1 to a specific port or port channel, Cisco UCS Manager raises an Update Failed fault.

Therefore, if you configure a Cisco UCS domain for disjoint L2 networks, do not configure any vNICs with default VLAN 1 unless you want all data traffic for that server to be carried on all uplink Ethernet ports and port channels and sent to all upstream networks.

VLANs for Both FIs Must be Concurrently Assigned

When you assign a port to a global VLAN, the VLAN is removed from all of the ports that are not explicitly assigned to the VLAN on both fabric interconnects. The ports on both FIs must be configured at the same time. If the ports are only configured on the first FI, traffic on the second FI will be disrupted.

Pinning Considerations for Upstream Disjoint L2 Networks

Communication with an upstream disjoint L2 network requires that you ensure that the pinning is properly configured. Whether you implement soft pinning or hard pinning, a VLAN membership mismatch causes traffic for one or more VLANs to be dropped.

Soft Pinning

Soft pinning is the default behavior in Cisco UCS. If you plan to implement soft pinning, you do not need to create LAN pin groups to specify a pin target for a vNIC. Instead, Cisco UCS Manager pins the vNIC to an uplink Ethernet port or port channel according to VLAN membership criteria.

With soft pinning, Cisco UCS Manager validates data traffic from a vNIC against the VLAN membership of all uplink Ethernet ports and port channels. If you have configured disjoint L2 networks, Cisco UCS Manager must be able to find an uplink Ethernet port or port channel that is assigned to all VLANS on the vNIC. If no uplink Ethernet port or port channel is configured with all VLANs on the vNIC, Cisco UCS Manager does the following:

- Brings the link down.
- Drops the traffic for all of the VLANs on the vNIC.
- Raises the following faults:
  - Link Down
  - VIF Down

Cisco UCS Manager does not raise a fault or warning about the VLAN configuration.

For example, a vNIC on a server is configured with VLANs 101, 102, and 103. Interface 1/3 is assigned only to VLAN 102. Interfaces 1/1 and 1/2 are not explicitly assigned to a VLAN, which makes them available for traffic on VLANs 101 and 103. As a result of this configuration, the Cisco UCS domain does not include a border port interface that can carry traffic for all three VLANS for which the vNIC is configured. As a result, Cisco UCS Manager brings down the vNIC, drops traffic for all three VLANs on the vNIC, and raises the Link Down and VIF Down faults.

Hard Pinning

Hard pinning occurs when you use LAN pin groups to specify the pinning target for the traffic intended for the disjoint L2 networks. In turn, the uplink Ethernet port or port channel that is the pinning target must be configured to communicate with the appropriate disjoint L2 network.

With hard pinning, Cisco UCS Manager validates data traffic from a vNIC against the VLAN membership of all uplink Ethernet ports and port channels, and validates the LAN pin group configuration to ensure it includes the VLAN and the uplink Ethernet port or port channel. If the validation fails at any point, Cisco UCS Manager does the following:

- Raises a Pinning VLAN Mismatch fault with a severity of Warning.
- Drops traffic for the VLAN.
- Does not bring the link down, so that traffic for other VLANs can continue to flow along it.

For example, if you want to configure hard pinning for an upstream disjoint L2 network that uses VLAN 177, do the following:

- Create a LAN pin group with the uplink Ethernet port or port channel that carries the traffic for the disjoint L2 network.
- Configure at least one vNIC in the service profile with VLAN 177 and the LAN pin group.
- Assign VLAN 177 to an uplink Ethernet port or port channel included in the LAN pin group

If the configuration fails at any of these three points, then Cisco UCS Manager warns for a VLAN mismatch for VLAN 177 and drops the traffic for that VLAN only.

Note If changes are made to soft pinning configurations resulting in vNIC VLANs not resolving with disjoint L2 uplink, a warning dialog box is displayed. The warning dialog box allows you to proceed with your configuration or cancel it. If you decide to proceed with the mis- configuration, you will experience a reduction is server traffic performance.

Configuring Cisco UCS for Upstream Disjoint L2 Networks

When you configure a Cisco UCS domain to connect with upstream disjoint L2 networks, you need to ensure that you complete all of the following steps.

Before You Begin

Before you begin this configuration, ensure that the ports on the fabric interconnects are properly cabled to support your disjoint L2 networks configuration.

Procedure

	Command or Action	Purpose
Step 1	Configure Ethernet switching mode for both fabric interconnects in Ethernet End-Host Mode.	The Ethernet switching mode must be in End-Host Mode for Cisco UCS to be able to communicate with upstream disjoint L2 networks. See Configuring Ethernet Switching Mode.
Step 2	Configure the ports and port channels that you require to carry traffic for the disjoint L2 networks.	See Configuring Ports and Port Channels, on page 61.
Step 3	Configure the LAN pin groups required to pin the traffic for the appropriate uplink Ethernet ports or port channels.	(Optional) See Configuring LAN Pin Groups, on page 259.
Step 4	Create one or more VLANs.	These can be named VLANs or private VLANs. For a cluster configuration, we recommend that you create the VLANs in Uplink Ethernet Mode and accessible to both fabric interconnects. See VLANs, on page 235.
Step 5	Assign the desired ports or port channels to the VLANs for the disjoint L2 networks.	When this step is completed, traffic for those VLANs can only be sent through the trunks for the assigned ports and/or port channels. Assigning Ports and Port Channels to VLANs, on page 326
Step 6	Ensure that the service profiles for all servers that need to communicate with the disjoint L2 networks include the correct LAN connectivity configuration to ensure the	You can complete this configuration through one or more vNIC templates or when you configure the networking options for the service profile. See Service Profiles, on page 593.

	Command or Action	Purpose
	vNICs send the traffic to the appropriate VLAN.	

Assigning Ports and Port Channels to VLANs

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope vlan** *vlan-name*	Enters Ethernet uplink VLAN mode for the specified VLAN.
Step 3	UCS-A /eth-uplink/vlan # **create member-port** *fabric-interconnect slot-id port-id*	Assigns the specified VLAN to the specified uplink Ethernet port.
Step 4	UCS-A /eth-uplink/vlan # **create member-port-channel** *fabric-interconnect member-port-chan-id*	Assigns the specified VLAN to the specified uplink Ethernet port channel.
Step 5	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration. After a port or port channel is assigned to one or more VLANs, it is removed from all other VLANs.

The following example assigns uplink Ethernet ports to a named VLAN called VLAN100 on fabric interconnect A and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope vlan VLAN100
UCS-A /eth-uplink/vlan # create member-port a 2
UCS-A /eth-uplink/vlan # create member-port a 4
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan #
```

Removing Ports and Port Channels from VLANs

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope vlan** *vlan-name*	Enters Ethernet uplink VLAN mode for the specified VLAN.
Step 3	UCS-A /eth-uplink/vlan # **delete member-port** *fabric-interconnect slot-id port-id*	Deletes the specified Uplink Ethernet member port assignment from the VLAN.
Step 4	UCS-A /eth-uplink/vlan # **delete member-port-channel** *fabric-interconnect member-port-chan-id*	Deletes the specified Uplink Ethernet port channel assignment from the VLAN.
Step 5	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration. **Important** If you remove all port or port channel interfaces from a VLAN, the VLAN returns to the default behavior and data traffic on that VLAN flows on all uplink ports and port channels. Based on the configuration in the Cisco UCS domain, this default behavior can cause Cisco UCS Manager to drop traffic for that VLAN. To avoid this occurrence, Cisco recommends that you assign at least one interface to the VLAN or delete the VLAN.

The following example deletes the association between uplink Ethernet port 2 on fabric interconnect A and the named VLAN called MyVLAN and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope vlan MyVLAN
UCS-A /eth-uplink/vlan # delete member-port a 2
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan #
```

Viewing Ports and Port Channels Assigned to VLANs

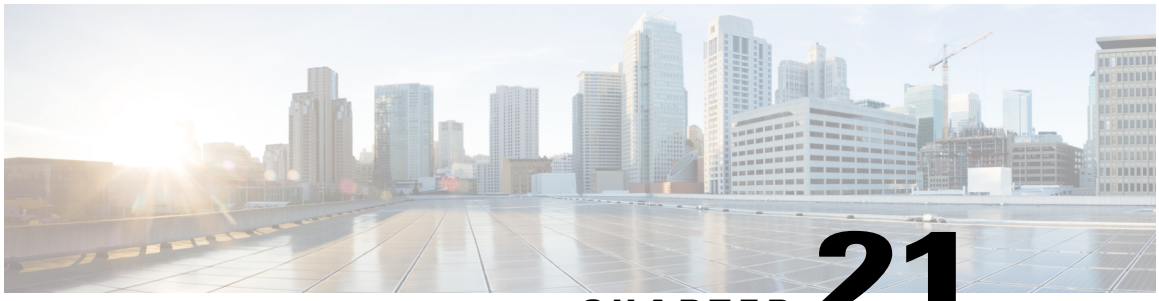
Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.

	Command or Action	Purpose
Step 2	UCS-A /eth-uplink # **scope vlan** *vlan-name*	Enters Ethernet uplink VLAN mode for the specified VLAN.
Step 3	UCS-A /eth-uplink/vlan # **show member-port** [**detail** | **expand**]	Shows member ports assigned to the specified VLAN.
Step 4	UCS-A /eth-uplink/vlan # **show member-port-channel** [**detail** | **expand**]	Shows member port channels assigned to the specified VLAN.
Step 5	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration.

The following example displays the full details for uplink Ethernet ports assigned to a named VLAN called MyVLAN:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope vlan MyVLAN
UCS-A /eth-uplink/vlan # show member-port detail
Member Port:
    Fabric ID: A
    Slot ID: 1
    Port ID: 2
    Mark Native Vlan: No
UCS-A /eth-uplink/vlan #
```



CHAPTER 21

Configuring Named VSANs

This chapter includes the following sections:

Named VSANs

A named VSAN creates a connection to a specific external SAN. The VSAN isolates traffic to that external SAN, including broadcast traffic. The traffic on one named VSAN knows that the traffic on another named VSAN exists, but cannot read or access that traffic.

Like a named VLAN, the name that you assign to a VSAN ID adds a layer of abstraction that allows you to globally update all servers associated with service profiles that use the named VSAN. You do not need to reconfigure the servers individually to maintain communication with the external SAN. You can create more than one named VSAN with the same VSAN ID.

Named VSANs in Cluster Configurations

In a cluster configuration, a named VSAN can be configured to be accessible only to the Fibre Channel uplink ports on one fabric interconnect or to the Fibre Channel uplink ports on both fabric interconnects.

Named VSANs and the FCoE VLAN ID

You must configure each named VSAN with an FCoE VLAN ID. This property determines which VLAN is used for transporting the VSAN and its Fibre Channel packets.

For FIP-capable, converged network adapters, such as the Cisco UCS CNA M72KR-Q and the Cisco UCS CNA M72KR-E, the named VSAN must be configured with a named VLAN that is not the native VLAN for the FCoE VLAN ID. This configuration ensures that FCoE traffic can pass through these adapters.

In the following sample configuration, a service profile with a vNIC and vHBA mapped to fabric A is associated with a server that has FIP capable, converged network adapters:

- The vNIC is configured to use VLAN 10.
- VLAN 10 is also designated as the native VLAN for the vNIC.
- The vHBA is configured to use VSAN 2.
- Therefore, VSAN 2 cannot be configured with VLAN 10 as the FCoE VLAN ID. VSAN 2 can be mapped to any other VLAN configured on fabric A.

Fibre Channel Uplink Trunking for Named VSANs

You can configure Fibre Channel uplink trunking for the named VSANs on each fabric interconnect. If you enable trunking on a fabric interconnect, all named VSANs in a Cisco UCS domain are allowed on all Fibre Channel uplink ports on that fabric interconnect.

Guidelines and Recommendations for VSANs

The following guidelines and recommendations apply to all named VSANs, including storage VSANs.

VSAN 4079 is a Reserved VSAN ID

Do not configure a VSAN as 4079. This VSAN is reserved and cannot be used in either FC switch mode or FC end-host mode.

If you create a named VSAN with ID 4079, Cisco UCS Manager marks that VSAN with an error and raises a fault.

Reserved VSAN Range for Named VSANs in FC Switch Mode

If you plan to use FC switch mode in a Cisco UCS domain, do not configure VSANs with an ID in the range from 3040 to 4078.

VSANs in that range are not operational if the fabric interconnects are configured to operate in FC switch mode. Cisco UCS Manager marks that VSAN with an error and raises a fault.

Reserved VSAN Range for Named VSANs in FC End-Host Mode

If you plan to use FC end-host mode in a Cisco UCS domain, do not configure VSANs with an ID in the range from 3840 to 4079.

VSANs in that range are not operational if the following conditions exist in a Cisco UCS domain:

- The fabric interconnects are configured to operate in FC end-host mode.
- The Cisco UCS domain is configured with Fibre Channel trunking or SAN port channels.

If these configurations exist, Cisco UCS Manager does the following:

1. Renders all VSANs with an ID in the range from 3840 to 4079 non-operational.
2. Raises a fault against the non-operational VSANs.
3. Transfers all non-operational VSANs to the default VSAN.
4. Transfers all vHBAs associated with the non-operational VSANs to the default VSAN.

If you disable Fibre Channel trunking and delete any existing SAN port channels, Cisco UCS Manager returns all VSANs in the range from 3840 to 4078 to an operational state and restores any associated vHBAs back to those VSANs.

Range Restrictions for Named VSAN IDs in FC Switch Mode

If you plan to use FC switch mode in a Cisco UCS domain, do not configure VSANs in the range from 3040 to 4078.

When a fabric interconnect operating in FC switch mode is connected to MDS as the upstream switch, VSANs configured in Cisco UCS Manager in the range from 3040 to 4078 and assigned as port VSANs cannot be created in MDS. This configuration results in a possible port VSAN mismatch.

Guidelines for FCoE VLAN IDs

Note FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

VLAN 4048 is user configurable. However, Cisco UCS Manager uses VLAN 4048 for the following default values. If you want to assign 4048 to a VLAN, you must reconfigure these values:

- After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.
- After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049.

Creating a Named VSAN Accessible to Both Fabric Interconnects (Fibre Channel Uplink Mode)

Note FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **create vsan** *vsan-name vsan-id fcoe-id*	Creates the specified named VSAN, specifies the VSAN name, VSAN ID and FCoE VLAN ID, and enters Fibre Channel uplink VSAN mode. • After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use. • After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049.
Step 3	UCS-A /fc-uplink/vsan # **set fc-zoning** {**disabled** | **enabled**}	Configures Fibre Channel zoning for the VSAN, as follows: • disabled—The upstream switch configures and controls the Fibre Channel zoning or Fibre Channel zoning is not implemented on this VSAN. • enabled—Cisco UCS Manager configures and controls Fibre Channel zoning.
Step 4	UCS-A /fc-uplink/vsan # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VSAN for both fabric interconnects, names the VSAN accounting, assigns the VSAN ID 2112, assigns the FCoE VLAN ID 4021, enables the VSAN for Cisco UCS Manager-based Fibre Channel zoning, and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink* # create vsan accounting 2112 4021
```

```
UCS-A /fc-uplink/vsan # set fc-zoning enabled
UCS-A /fc-uplink/vsan* # commit-buffer
UCS-A /fc-uplink/vsan #
```

Creating a Named VSAN Accessible to Both Fabric Interconnects (Fibre Channel Storage Mode)

Note FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-storage**	Enters Fibre Channel storage mode.
Step 2	UCS-A /fc-storage # **create vsan** *vsan-name vsan-id fcoe-id*	Creates the specified named VSAN, specifies the VSAN name, VSAN ID, and FCoE VLAN ID, and enters Fibre Channel storage VSAN mode. • After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use. • After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049.
Step 3	UCS-A /fc-storage/vsan # **create member-port** {**fc** | **fcoe**} {**a** | **b**} *slot-id port-id*	Creates a member port; specifies whether the port type, fabric, slot ID and port ID.
Step 4	UCS-A /fc-storage/vsan # **set fc-zoning** {**disabled** | **enabled**}	Configures Fibre Channel zoning for the VSAN, as follows: • disabled—The upstream switch configures and controls the Fibre Channel zoning or Fibre Channel zoning is not implemented on this VSAN. • enabled—Cisco UCS Manager configures and controls Fibre Channel zoning.
Step 5	UCS-A /fc-storage/vsan # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VSAN, names the VSAN finance, assigns the VSAN ID 3955, assigns the FCoE VLAN ID 4021, creates a member port and assigns it to member port A, slot 1 port 40, enables the VSAN for Cisco UCS Manager-based Fibre Channel zoning, and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage/ # create VSAN finance 3955 4021
UCS-A /fc-storage/vsan # create member-port fcoe a 1 40
UCS-A /fc-storage/vsan # set fc-zoning enabled
UCS-A /fc-storage/vsan/member-port* # commit-buffer
UCS-A /fc-storage/vsan/member-port #
```

Creating a Named VSAN Accessible to One Fabric Interconnect (Fibre Channel Uplink Mode)

Note FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A /fc-uplink/fabric # **create vsan** *vsan-name vsan-id fcoe-id*	Creates the specified named VSAN, specifies the VSAN name, VSAN ID, and FCoE VLAN ID, and enters Fibre Channel uplink VSAN mode. • After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use. • After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049.
Step 4	UCS-A /fc-uplink/vsan # **set fc-zoning** {**disabled** | **enabled**}	Configures Fibre Channel zoning for the VSAN, as follows: • disabled—The upstream switch configures and controls the Fibre Channel zoning or Fibre Channel zoning is not implemented on this VSAN.

	Command or Action	Purpose
		• enabled—Cisco UCS Manager configures and controls Fibre Channel zoning.
Step 5	UCS-A /fc-uplink/fabric/vsan # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VSAN for fabric interconnect A, names the VSAN finance, assigns the VSAN ID 3955, assigns the FCoE VLAN ID 2221, enables the VSAN for Cisco UCS Manager-based Fibre Channel zoning, and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create vsan finance 3955 2221
UCS-A /fc-uplink/vsan # set fc-zoning enabled
UCS-A /fc-uplink/fabric/vsan* # commit-buffer
UCS-A /fc-uplink/fabric/vsan #
```

Creating a Named VSAN Accessible to One Fabric Interconnect (Fibre Channel Storage Mode)

Note FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-storage**	Enters Fibre Channel storage mode.
Step 2	UCS-A /fc-storage # **scope fabric** {**a** | **b**}	Enters Fibre Channel storage mode for the specified fabric interconnect.
Step 3	UCS-A /fc-storage/fabric # **create vsan** *vsan-name vsan-id fcoe-id*	Creates the specified named VSAN, specifies the VSAN name, VSAN ID, and FCoE VLAN ID, and enters Fibre Channel storage VSAN mode. • After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.

	Command or Action	Purpose
		• After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049.
Step 4	UCS-A /fc-storage/fabric/vsan # **create member-port** {**fc** | **fcoe**} {**a** | **b**} *slot-id port-id*	Creates a member port on the specified VSAN.
Step 5	UCS-A /fc-storage/vsan # **set fc-zoning** {**disabled** | **enabled**}	Configures Fibre Channel zoning for the VSAN, as follows: • disabled—The upstream switch configures and controls the Fibre Channel zoning or Fibre Channel zoning is not implemented on this VSAN. • enabled—Cisco UCS Manager configures and controls Fibre Channel zoning.
Step 6	UCS-A /fc-storage/fabric/vsan # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a named VSAN on fabric A, names the VSAN finance, assigns the VSAN ID 3955, assigns the FCoE VLAN ID 2221, creates a member port and assigns the it to member port A, slot 1 port 40, and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage/ # scope fabric a
UCS-A /fc-storage/fabric # create VSAN finance 3955 2221
UCS-A /fc-storage/fabric/vsan # create member-port a 1 40
UCS-A /fc-storage/fabric/vsan # set fc-zoning enabled
UCS-A /fc-storage/fabric/vsan/member-port* # commit-buffer
UCS-A /fc-storage/fabric/vsan/member-port #
```

Deleting a Named VSAN

If Cisco UCS Manager includes a named VSAN with the same VSAN ID as the one you delete, the VSAN is not removed from the fabric interconnect configuration until all named VSANs with that ID are deleted.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **delete vsan** *vsan-name*	Deletes the specified named VSAN.
Step 3	UCS-A /fc-uplink # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete a named VSAN and commit the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # delete vsan finance
UCS-A /fc-uplink* # commit-buffer
UCS-A /fc-uplink #
```

Changing the VLAN ID for the FCoE Native VLAN for a Named VSAN

Note FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope vsan** *vsan-name*	Enters VSAN mode for the specified named VSAN.
Step 3	UCS-A /fc-uplink/vsan # **set fcoe-vlan** *fcoe-vlan-id*	Sets the unique identifier assigned to the VLAN used for Fibre Channel connections.
Step 4	UCS-A /fc-uplink/vsan # **commit-buffer**	Commits the transaction to the system configuration.

The following example changes the VLAN ID for the FCoE Native VLAN on a named VSAN called finance to 4000 and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope vsan finance
UCS-A /fc-uplink/vsan # set fcoe-vlan 4000
UCS-A /fc-uplink/vsan* # commit-buffer
UCS-A /fc-uplink/vsan #
```

Changing the VLAN ID for the FCoE Native VLAN for a Storage VSAN

Note FCoE VLANs in the SAN cloud and VLANs in the LAN cloud must have different IDs. Using the same ID for an FCoE VLAN in a VSAN and a VLAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that FCoE VLAN. Ethernet traffic is dropped on any VLAN with an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-storage**	Enters Fibre Channel storage mode.
Step 2	UCS-A /fc-storage # **set fcoe-storage-native-vlan** *fcoe-id*	Sets the unique identifier assigned to the VLAN used for Fibre Channel connections.
Step 3	UCS-A /fc-storage # **commit-buffer**	Commits the transaction to the system configuration.

The following example changes the VLAN ID for the FCoE Native VLAN on a storage VSAN called finance to 4000 and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage # set fcoe-storage-native-vlan 4000
UCS-A /fc-storage* # commit-buffer
UCS-A /fc-storage #
```

Enabling or Disabling Fibre Channel Uplink Trunking

Note If the fabric interconnects are configured for Fibre Channel end-host mode, enabling Fibre Channel uplink trunking renders all VSANs with an ID in the range from 3840 to 4079 non-operational.

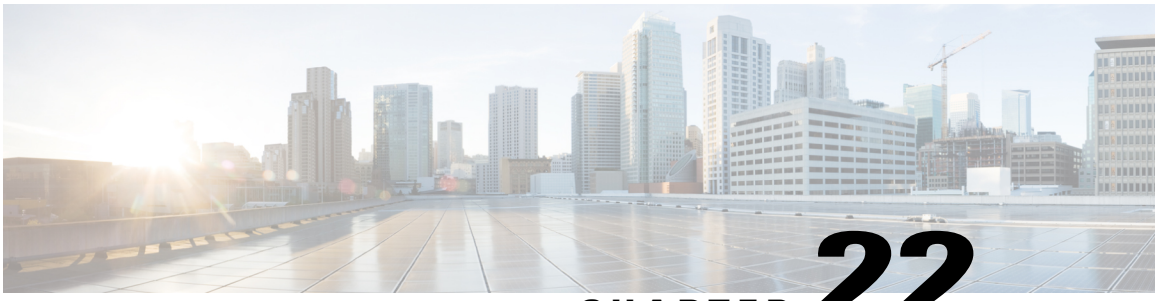
Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink mode for the specified fabric.

	Command or Action	Purpose
Step 3	UCS-A /fc-uplink/fabric # **set uplink-trunking** {**enabled** | **disabled** }	Enables or disables uplink trunking.
Step 4	UCS-A /fc-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

The following example enables Fibre Channel uplink trunking for fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # set uplink-trunking enabled
UCS-A /fc-uplink/fabric* # commit-buffer
UCS-A /fc-uplink/fabric #
```



CHAPTER 22

Configuring SAN Pin Groups

This chapter includes the following sections:

SAN Pin Groups

Cisco UCS uses SAN pin groups to pin Fibre Channel traffic from a vHBA on a server to an uplink Fibre Channel port on the fabric interconnect. You can use this pinning to manage the distribution of traffic from the servers.

Note In Fibre Channel switch mode, SAN pin groups are irrelevant. Any existing SAN pin groups will be ignored.

To configure pinning for a server, you must include the SAN pin group in a vHBA policy. The vHBA policy is then included in the service profile assigned to that server. All traffic from the vHBA will travel through the I/O module to the specified uplink Fibre Channel port.

You can assign the same pin group to multiple vHBA policies. As a result, you do not need to manually pin the traffic for each vHBA.

Important Changing the target interface for an existing SAN pin group disrupts traffic for all vHBAs which use that pin group. The fabric interconnect performs a log in and log out for the Fibre Channel protocols to re-pin the traffic.

Configuring a SAN Pin Group

In a system with two fabric interconnects, you can associate the pin group with only one fabric interconnect or with both fabric interconnects.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **create pin-group** *pin-group-name*	Creates a Fibre Channel (SAN) pin group with the specified name, and enters Fibre Channel uplink pin group mode.
Step 3	UCS-A /fc-uplink/pin-group # **set descr** *description*	(Optional) Provides a description for the pin group. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /fc-uplink/pin-group # **set target** {**a** | **b** | **dual**} **port** *slot-num* / *port-num*	(Optional) Sets the Fibre Channel pin target to the specified fabric and port.
Step 5	UCS-A /fc-uplink/pin-group # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a SAN pin group named fcpingroup12, provides a description for the pin group, sets the pin group target to slot 2, port 1, and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # create pin-group fcpingroup12
UCS-A /fc-uplink/pin-group* # set descr "This is my pin group #12"
UCS-A /fc-uplink/pin-group* # set target a port 2/1
UCS-A /fc-uplink/pin-group* # commit-buffer
UCS-A /fc-uplink/pin-group #
```

What to Do Next

Include the pin group in a vHBA template.

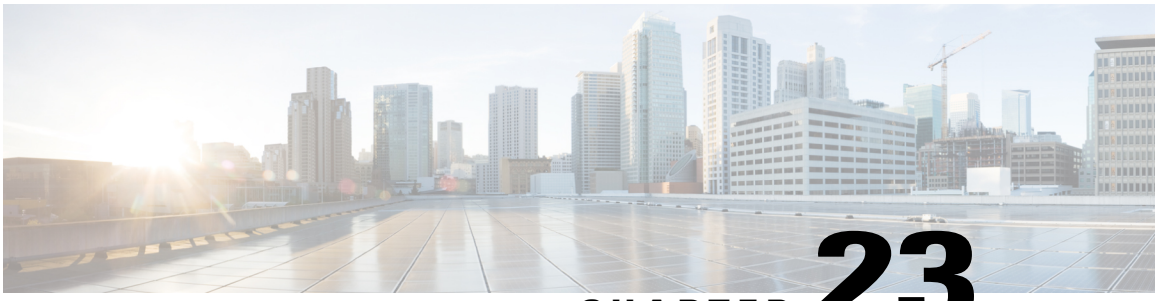
Configuring a FCoE Pin Group

You can create a FCoE pin group, and specify the FCoE uplink port as the pin group target.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters FC uplink mode.
Step 2	UCS-A /fc-uplink # **create pin-group fcoepingroup**	Creates a FCoE pin group with the specified name, and enters FCoE uplink pin group mode.
Step 3	UCS-A /fc-uplink/pin-group # **set target a fcoe-port 1/8**	Sets FCoE port 1/8 as the target port for this pin group.
Step 4	UCS-A /fc-uplink/pin-group # **commit-buffer**	Commits the transaction to the system configuration.

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # create pin-group fcoepingroup
UCS-A /fc-uplink/pin-group* #set target a fcoe-port 1/8
UCS-A /fc-uplink/pin-group* # commit-buffer
UCS-A /fc-uplink/pin-group #
```



CHAPTER 23

Configuring WWN Pools

This chapter includes the following sections:

WWN Pools

A World Wide Name (WWN) pool is a collection of WWNs for use by the Fibre Channel vHBAs in a Cisco UCS domain. You create separate pools for the following:

- WW node names assigned to the vHBA
- WW port names assigned to the vHBA
- Both WW node names and WW port names

Important A WWN pool can include only WWNNs or WWPNs in the ranges from 20:00:00:00:00:00:00:00 to 20:FF:FF:FF:FF:FF:FF:FF or from 50:00:00:00:00:00:00:00 to 5F:FF:FF:FF:FF:FF:FF:FF. All other WWN ranges are reserved. To ensure the uniqueness of the Cisco UCS WWNNs and WWPNs in the SAN fabric, Cisco recommends using the following WWN prefix for all blocks in a pool: 20:00:00:25:B5:XX:XX:XX

If you use WWN pools in service profiles, you do not have to manually configure the WWNs that will be used by the server associated with the service profile. In a system that implements multi-tenancy, you can use a WWN pool to control the WWNs used by each organization.

You assign WWNs to pools in blocks.

WWNN Pools

A WWNN pool is a WWN pool that contains only WW node names. If you include a pool of WWNNs in a service profile, the associated server is assigned a WWNN from that pool.

WWPN Pools

A WWPN pool is a WWN pool that contains only WW port names. If you include a pool of WWPNs in a service profile, the port on each vHBA of the associated server is assigned a WWPN from that pool.

WWxN Pools

A WWxN pool is a WWN pool that contains both WW node names and WW port names. You can specify how many ports per node are created with WWxN pools. The pool size must be a multiple of *ports-per-node* + 1. For example, if you specify 7 ports per node, the pool size must be a multiple of 8. If you specify 63 ports per node, the pool size must be a multiple of 64.

You can use a WWxN pool whenever you select a WWNN or WWPN pool. The WWxN pool must be created before it can be assigned.

- For WWNN pools, the WWxN pool is displayed as an option in the **WWNN Assignment** drop-down list.
- For WWPN pools, choose **Derived** in the **WWPN Assignment** drop-down list.

Creating a WWN Pool

Important A WWN pool can include only WWNNs or WWPNs in the ranges from 20:00:00:00:00:00:00:00 to 20:FF:FF:FF:FF:FF:FF:FF or from 50:00:00:00:00:00:00:00 to 5F:FF:FF:FF:FF:FF:FF:FF. All other WWN ranges are reserved. To ensure the uniqueness of the Cisco UCS WWNNs and WWPNs in the SAN fabric, Cisco recommends using the following WWN prefix for all blocks in a pool: 20:00:00:25:B5:XX:XX:XX

A WWNN pool with the last four digits ending in 00:01 causes the vHBA to not initialize, no output from the lunlist command, and displays the Waiting for Flogi error. This error occurs if the WWPN is in the same block as the WWNN ending in 00:01. To ensure that the WWNN and WWPN addresses do not overlap, we recommend using a unique WWN address.

Procedure

	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 wwn-pool** *wwn-pool-name* {**node-and-port-wwn-assignment** | **node-wwn-assignment** | **port-wwn-assignment**}	Creates a WWN pool with the specified name and purpose, and enters organization WWN pool mode. This can be one of the following: • **node-and-port-wwn-assignment—**Creates a WWxN pool that includes both world wide node names (WWNNs) and world wide port names (WWPNs). • **node-wwn-assignment—**Creates a WWNN pool that includes only WWNNs.

	Command or Action	Purpose
		• **port-wwn-assignment—**Creates a WWPN pool that includes only WWPNs. This name can be between 1 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	UCS-A /org/wwn-pool # **set descr** *description*	(Optional) Provides a description for the WWN pool. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/wwn-pool # **set assignmentorder** {**default** | **sequential**}	This can be one of the following: • **default—**Cisco UCS Manager selects a random identity from the pool. • **sequential—**Cisco UCS Manager selects the lowest available identity from the pool.
Step 5	UCS-A /org/wwn-pool # **set max-ports-per-node** {**15-ports-per-node** | **3-ports-per-node** | **31-ports-per-node** | **63-ports-per-node** | **7-ports-per-node**}	For WWxN pools, specify the maximum number of ports that can be assigned to each node name in this pool. The default value is **3-ports-per-node**. **Note** The pool size for WWxN pools must be a multiple of *ports-per-node* + 1. For example, if you specify **7-ports-per-node**, the pool size must be a multiple of 8. If you specify **63-ports-per-node**, the pool size must be a multiple of 64.
Step 6	UCS-A /org/wwn-pool # **create block** *first-wwn last-wwn*	Creates a block (range) of WWNs, and enters organization WWN pool block mode. You must specify the first and last WWN in the block using the form *nn*:*nn*:*nn*:*nn*:*nn*:*nn*:*nn*:*nn*, with the WWNs separated by a space. **Note** A WWN pool can contain more than one WWN block. To create multiple WWN blocks, you must enter multiple **create block** commands from organization WWN pool mode.
Step 7	UCS-A /org/wwn-pool/block # **exit**	Exits organization WWN pool block mode.
Step 8	UCS-A /org/wwn-pool # **create initiator** *wwn wwn*	Creates a single initiator for a WWNN or WWPN pool, and enters organization WWN pool initiator mode. You must specify the initiator using the form *nn*:*nn*:*nn*:*nn*:*nn*:*nn*:*nn*:*nn*.

	Command or Action	Purpose
		Note A WWNN or WWPN pool can contain more than one initiator. To create multiple initiators, you must enter multiple **create initiator** commands from organization WWN pool mode.
Step 9	UCS-A /org/wwn-pool/initiator # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a WWNN pool named sanpool, provide a description for the pool, specify a block of WWNs and an initiator to be used for the pool, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create wwn-pool sanpool node-wwn-assignment
UCS-A /org/wwn-pool* # set descr "This is my WWNN pool"
UCS-A /org/wwn-pool* # create block 20:00:00:25:B5:00:00:00 20:00:00:25:B5:00:00:01
UCS-A /org/wwn-pool/block* # exit
UCS-A /org/wwn-pool* # create initiator 23:00:00:05:AD:1E:02:00
UCS-A /org/wwn-pool/initiator* # commit-buffer
UCS-A /org/wwn-pool/initiator #
```

The following example shows how to create a WWxN pool named sanpool, provide a description for the pool, specify seven ports per node, specify a block of eight WWNs to be used for the pool, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create wwn-pool sanpool node-and-port-wwn-assignment
UCS-A /org/wwn-pool* # set descr "This is my WWxN pool"
UCS-A /org/wwn-pool* # set max-ports-per-node 7-ports-per-node
UCS-A /org/wwn-pool* # create block 20:00:00:25:B5:00:00:00 20:00:00:25:B5:00:00:08
UCS-A /org/wwn-pool/block* # commit-buffer
UCS-A /org/wwn-pool/block #
```

What to Do Next

- Include the WWPN pool in a vHBA template.
- Include the WWNN pool in a service profile and template.
- Include the WWxN pool in a service profile and template.

Deleting a WWN Pool

If you delete a pool, Cisco UCS Manager does not reallocate any addresses from that pool that were assigned to vNICs or vHBAs. All assigned addresses from a deleted pool remain with the vNIC or vHBA to which they are assigned until one of the following occurs:

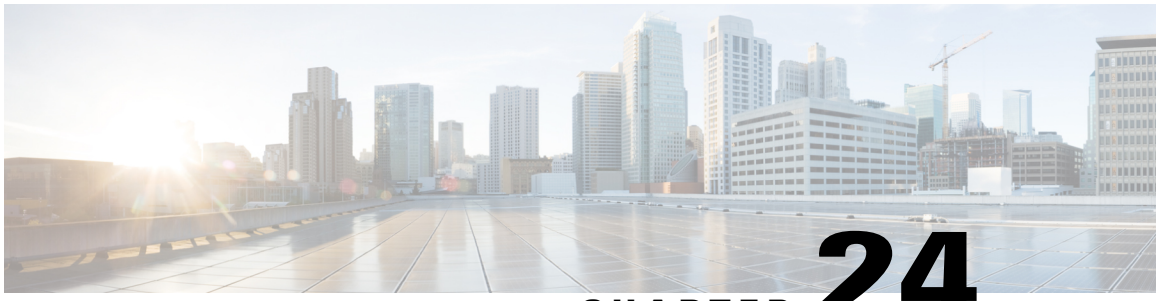
- The associated service profiles are deleted.
- The vNIC or vHBA to which the address is assigned is deleted.
- The vNIC or vHBA is assigned to a different pool.

Procedure

	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 wwn-pool** *pool-name*	Deletes the specified WWN pool.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the WWN pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete wwn-pool pool4
UCS-A /org* # commit-buffer
UCS-A /org #
```



CHAPTER 24

Configuring Storage-Related Policies

This chapter includes the following sections:

Configuring vHBA Templates

vHBA Template

This template is a policy that defines how a vHBA on a server connects to the SAN. It is also referred to as a vHBA SAN connectivity template.

You must include this policy in a service profile for it to take effect.

Configuring a vHBA Template

Procedure

	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 # **create vhba-templ** *vhba-templ-name* [**fabric** {**a** | **b**}] [**fc-if** *vsan-name*]	Creates a vHBA template and enters organization vHBA template mode.

	Command or Action	Purpose
Step 3	UCS-A /org/vhba-templ # **set descr** *description*	(Optional) Provides a description for the vHBA template.
Step 4	UCS-A /org/vhba-templ # **set fabric** {**a** | **b**}	(Optional) Specifies the fabric to use for the vHBA. If you did not specify the fabric when creating the vHBA template in Step 2, then you have the option to specify it with this command.
Step 5	UCS-A /org/vhba-templ # **set fc-if** *vsan-name*	(Optional) Specifies the Fibre Channel interface (named VSAN) to use for the vHBA template. If you did not specify the Fibre Channel interface when creating the vHBA template in Step 2, you have the option to specify it with this command.
Step 6	UCS-A /org/vhba-templ # **set max-field-size** *size-num*	Specifies the maximum size of the Fibre Channel frame payload (in bytes) that the vHBA supports.
Step 7	UCS-A /org/vhba-templ # **set pin-group** *group-name*	Specifies the pin group to use for the vHBA template.
Step 8	UCS-A /org/vhba-templ # **set qos-policy** *mac-pool-name*	Specifies the QoS policy to use for the vHBA template.
Step 9	UCS-A /org/vhba-templ # **set stats-policy** *policy-name*	Specifies the server and server component statistics threshold policy to use for the vHBA template.
Step 10	UCS-A /org/vhba-templ # **set type** {**initial-template** | **updating-template**}	Specifies the vHBA template update type. If you do not want vHBA instances created from this template to be automatically updated when the template is updated, use the **initial-template** keyword; otherwise, use the **updating-template** keyword to ensure that all vHBA instances are updated when the vHBA template is updated.
Step 11	UCS-A /org/vhba-templ # **set wwpn-pool** *pool-name*	Specifies the WWPN pool to use for the vHBA template.
Step 12	UCS-A /org/vhba-templ # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a vHBA template and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create vhba template VhbaTempFoo
UCS-A /org/vhba-templ* # set descr "This is a vHBA template example."
UCS-A /org/vhba-templ* # set fabric a
UCS-A /org/vhba-templ* # set fc-if accounting
UCS-A /org/vhba-templ* # set max-field-size 2112
UCS-A /org/vhba-templ* # set pin-group FcPinGroup12
UCS-A /org/vhba-templ* # set qos-policy policy34foo
UCS-A /org/vhba-templ* # set stats-policy ServStatsPolicy
UCS-A /org/vhba-templ* # set type updating-template
```

```
UCS-A /org/vhba-templ* # set wwpn-pool SanPool7
UCS-A /org/vhba-templ* # commit-buffer
UCS-A /org/vhba-templ #
```

Redundancy Template Pairs

Creating vNIC and vHBA template pairs enables you to group vNICs or vHBAs that belong to a specific server. For example, you can create a vNIC or a vHBA template and specify it as the Primary template, then create a different vNIC or vHBA template and specify it as the Secondary template. You can link the two templates to create a pair that share attributes that you define in the Primary template. The Secondary template inherits the attributes from the Primary template and any changes made to the Primary template are propagated to the Secondary template in the template pair. You can also modify any non-shared configurations on each individual template in the pair.

When creating the pair, you can assign one template, for example the Primary template to Fabric A and the other template, for example the Secondary template to Fabric B or vice versa. This feature eliminates the need to configure vNIC or vHBA pairs independently using one or more templates.

The number of vNIC and vHBA pairs that can be created using a template pair is only limited by the adapter's maximum capabilities.

Use the **Initial Template** type for one time provisioning.

Use the **Updating Template** type to have the Primary template drive the changes in the redundancy pair for shared configurations. See the shared configurations listed below.

Creating vHBA Template Pairs

Procedure

	Command or Action	Purpose
Step 1	UCS-A/ org # **create vhba-templ** *vhba-primary* .	Creates a Primary vHBA template.
Step 2	UCS-A/ # org vhba-templ **set type updating-template** .	Set the template type to updating, which drives the configurations in the Primary vNIC template for shared configurations to the peer vHBA template. See the shared configurations listed below.
Step 3	UCS-A/ # org vhba-templ [**set fabric** {**a** | **b**}] .	Specifies the fabric for the Primary vHBA template. If you specify Fabric A for the Primary vHBA template, the Secondary vHBA template must be Fabric B or vice versa.
Step 4	UCS-A/ # org vhba-templ **set redundancy-type** *primary* .	Sets the redundancy template type as the Primary template. See the **Redundancy Type** descriptions below.
Step 5	UCS-A/ # org vhba-templ **commit-buffer** .	Commits the transaction to the system configuration.
Step 6	UCS-A/ # org vhba-templ**create vhba-templ** *vhba-secondary* .	Creates a Secondary vHBA template.

	Command or Action	Purpose
Step 7	UCS-A/ # org vhba-templ **set redundancy-type** *secondary* .	Sets the secondary or peer redundancy type. Following is a list of the **Redundancy Types**. **Primary—**Creates configurations that can be shared with the Secondary vHBA template. Any shared changes on the Primary vHBA template are automatically synchronized to the Secondary vHBA template. **Secondary —** All shared configurations are inherited from the Primary template. **No Redundancy—** Legacy vHBA template behavior. Following is a list of shared configurations • **VSANS** • **Template Type** • **Maximum Data Field Size** • **QoS Policy** • **Stats Threshold Policy** Following is a list of non-shared configurations: • **Fabric ID** **Note** The Fabric ID must be mutually exclusive. If you assign the Primary template to Fabric A, then Fabric B is automatically assigned to the Secondary template as part of the synchronization from the Primary template. • **WWPN Pool** • **Description** • **Pin Group Policy**
Step 8	UCS-A/ # org vhba-templ **commit-buffer** .	Commits the transaction to the system configuration.
Step 9	UCS-A/ # org vhba-templ *vhba primary*.	Sets the Primary vHBA template as a redundancy pair template.
Step 10	UCS-A/ # org vhba-templ **scope vhba template** *vhba primary*.	Accesses the primary vhba template.
Step 11	UCS-A/ # org vhba-templ **set redundancy peer-template-name** *vhba-secondary*.	Sets the Secondary vHBA template as the peer to the Primary vHBA template.
Step 12	UCS-A/ # org vhba-templ **commit-buffer** .	Commits the transaction to the system configuration.

The following example configures a vHBA redundancy template pair and commits the transaction:

```
UCS-A /org* # create vhba-template vhba-primary
UCS-A /org/vhba-templ* # set type updating-template
UCS-A /org/vhba-templ* # set fabric a
UCS-A /org/vhba-templ* # set redundancy-type primary
UCS-A /org/vhba-templ* # commit-buffer
UCS-A /org/vhba-templ* # create vhba-template vhba-secondary
UCS-A /org/vhba-templ* # set redundancy-peer vhba-primary
UCS-A /org/vhba-templ* # commit-buffer
```

What to Do Next

After you create the vHBA redundancy template pair, you can use the redundancy template pair to create redundancy vHBA pairs for any service profile in the same organization or sub- organization.

Undo vHBA Template Pairs

You can undo the vHBA template pair by changing the Peer Redundancy Template so that there is no peer template for the Primary or the Secondary template. When you undo a vHBA template pair, the corresponding vHBA pairs also becomes undone.

Procedure

	Command or Action	Purpose
Step 1	UCS-A /org # **scope vhba-templ** *template1*.	Specifies the name of the vHBA template that you want to undo from the template pair.
Step 2	UCS-A /org/ vhba-templ # **set redundancy-type** *no redundancy*.	Removes the paring between the peer Primary or Secondary redundancy template used to perform the template pairing.
Step 3	UCS-A /org/vhba-templ* # **commit-buffer** .	Commits the transaction to the system configuration.

The following example shows how to undo a template pairing:

```
UCS-A /org # scope vhba-templ template1
UCS-A /org/vhba-templ # set redundancy-type no-redundancy
UCS-A /org/vhba-templ* # commit buffer
```

Deleting a vHBA Template

Procedure

	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 # **delete vhba-templ** *vhba-templ-name*	Deletes the specified vHBA template.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the vHBA template named VhbaTempFoo and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete vhba template VhbaTempFoo
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Fibre Channel Adapter Policies

Ethernet and Fibre Channel Adapter Policies

These policies govern the host-side behavior of the adapter, including how the adapter handles traffic. For example, you can use these policies to change default settings for the following:

- Queues
- Interrupt handling
- Performance enhancement
- RSS hash
- Failover in a cluster configuration with two fabric interconnects

Note For Fibre Channel adapter policies, the values displayed by Cisco UCS Manager may not match those displayed by applications such as QLogic SANsurfer. For example, the following values may result in an apparent mismatch between SANsurfer and Cisco UCS Manager:

- Max LUNs Per Target—SANsurfer has a maximum of 256 LUNs and does not display more than that number. Cisco UCS Manager supports a higher maximum number of LUNs.
- Link Down Timeout—In SANsurfer, you configure the timeout threshold for link down in seconds. In Cisco UCS Manager, you configure this value in milliseconds. Therefore, a value of 5500 ms in Cisco UCS Manager displays as 5s in SANsurfer.
- Max Data Field Size—SANsurfer has allowed values of 512, 1024, and 2048. Cisco UCS Manager allows you to set values of any size. Therefore, a value of 900 in Cisco UCS Manager displays as 512 in SANsurfer.
- LUN Queue Depth—The LUN queue depth setting is available for Windows system FC adapter policies. Queue depth is the number of commands that the HBA can send and receive in a single transmission per LUN. Windows Storport driver sets this to a default value of 20 for physical miniports and to 250 for virtual miniports. This setting adjusts the initial queue depth for all LUNs on the adapter. Valid range for this value is 1 to 254. The default LUN queue depth is 20. This feature only works with Cisco UCS Manager version 3.1(2) and higher.
- IO TimeOut Retry—When the target device is not responding to an IO request within the specified timeout, the FC adapter will abort the pending command then resend the same IO after the timer expires. The FC adapter valid range for this value is 1 to 59 seconds. The default IO retry timeout is 5 seconds. This feature only works with Cisco UCS Manager version 3.1(2) and higher.

Operating System Specific Adapter Policies

By default, Cisco UCS provides a set of Ethernet adapter policies and Fibre Channel adapter policies. These policies include the recommended settings for each supported server operating system. Operating systems are sensitive to the settings in these policies. Storage vendors typically require non-default adapter settings. You can find the details of these required settings on the support list provided by those vendors.

Important We recommend that you use the values in these policies for the applicable operating system. Do not modify any of the values in the default policies unless directed to do so by Cisco Technical Support.

However, if you are creating an Ethernet adapter policy for a Windows OS (instead of using the default Windows adapter policy), you must use the following formulas to calculate values that work with Windows:

Completion Queues = Transmit Queues + Receive Queues
Interrupt Count = (Completion Queues + 2) rounded up to nearest power of 2

For example, if Transmit Queues = 1 and Receive Queues = 8 then:

Completion Queues = 1 + 8 = 9
Interrupt Count = (9 + 2) rounded up to the nearest power of 2 = 16

Configuring a Fibre Channel Adapter Policy

Procedure

	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 # **create fc-policy** *policy-name*	Creates the specified Fibre Channel adapter policy and enters organization Fibre Channel policy mode.
Step 3	UCS-A /org/fc-policy # **set descr** *description*	(Optional) Provides a description for the policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/fc-policy # **set error-recovery** {**fcp-error-recovery** {**disabled** | **enabled**} | **link-down-timeout** *timeout-msec* | **port-down-io-retry-count** *retry-count* | **port-down-timeout** *timeout-msec*}	(Optional) Configures the Fibre Channel error recovery.
Step 5	UCS-A /org/fc-policy # **set interrupt mode** {**intx** | **msi** | **msi-x**}}	(Optional) Configures the driver interrupt mode.
Step 6	UCS-A /org/fc-policy # **set port** {**io-throttle-count** *throttle-count* | **max-luns** *max-num*}	(Optional) Configures the Fibre Channel port.
Step 7	UCS-A /org/fc-policy # **set port-f-logi** {**retries** *retry-count* | **timeout** *timeout-msec*}	(Optional) Configures the Fibre Channel port fabric login (FLOGI).
Step 8	UCS-A /org/fc-policy # **set port-p-logi** {**retries** *retry-count* | **timeout** *timeout-msec*}	(Optional) Configures the Fibre Channel port-to-port login (PLOGI).
Step 9	UCS-A /org/fc-policy # **set recv-queue** {**count** *count* | **ring-size** *size-num*}	(Optional) Configures the Fibre Channel receive queue.
Step 10	UCS-A /org/fc-policy # **set scsi-io** {**count** *count* | **ring-size** *size-num*}	(Optional) Configures the Fibre Channel SCSI I/O.

	Command or Action	Purpose
Step 11	UCS-A /org/fc-policy # **set trans-queue ring-size** *size-num*}	(Optional) Configures the Fibre Channel transmit queue.
Step 12	UCS-A /org/fc-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a Fibre Channel adapter policy and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create fc-policy FcPolicy42
UCS-A /org/fc-policy* # set descr "This is a Fibre Channel adapter policy example."
UCS-A /org/fc-policy* # set error-recovery error-detect-timeout 2500
UCS-A /org/fc-policy* # set port max-luns 4
UCS-A /org/fc-policy* # set port-f-logi retries 250
UCS-A /org/fc-policy* # set port-p-logi timeout 5000
UCS-A /org/fc-policy* # set recv-queue count 1
UCS-A /org/fc-policy* # set scsi-io ring-size 256
UCS-A /org/fc-policy* # set trans-queue ring-size 256
UCS-A /org/fc-policy* # commit-buffer
UCS-A /org/fc-policy #
```

Deleting a Fibre Channel Adapter Policy

Procedure

	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 # **delete fc-policy** *policy-name*	Deletes the specified Fibre Channel adapter policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the Fibre Channel adapter policy named FcPolicy42 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete fc-policy FcPolicy42
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring the Default vHBA Behavior Policy

Default vHBA Behavior Policy

Default vHBA behavior policy allow you to configure how vHBAs are created for a service profile. You can choose to create vHBAs manually, or you can allow them to be created automatically.

You can configure the default vHBA behavior policy to define how vHBAs are created. This can be one of the following:

- **None—**Cisco UCS Manager does not create default vHBAs for a service profile. All vHBAs must be explicitly created.
- **HW Inherit—**If a service profile requires vHBAs and none have been explicitly defined, Cisco UCS Manager creates the required vHBAs based on the adapter installed in the server associated with the service profile.

Note If you do not specify a default behavior policy for vHBAs, **none** is used by default.

Configuring a Default vHBA Behavior Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A/org # **scope vhba-beh-policy**	Enters default vHBA behavior policy mode.
Step 3	UCS-A/org/vhba-beh-policy # **set action** {**hw-inherit** [**template_name** *name*] | **none**}	Specifies the default vHBA behavior policy. This can be one of the following: • **hw-inherit—**If a service profile requires vHBAs and none have been explicitly defined, Cisco UCS Manager creates the required vHBAs based on the adapter installed in the server associated with the service profile. If you specify **hw-inherit**, you can also specify a vHBA template to create the vHBAs. • **none—**Cisco UCS Manager does not create default vHBAs for a service profile. All vHBAs must be explicitly created.
Step 4	UCS-A/org/vhba-beh-policy # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to set the default vHBA behavior policy to **hw-inherit**.

```
UCS-A # scope org /
UCS-A/org # scope vhba-beh-policy
UCS-A/org/vhba-beh-policy # set action hw-inherit
UCS-A/org/vhba-beh-policy* # commit-buffer
UCS-A/org/vhba-beh-policy #
```

Configuring SAN Connectivity Policies

About the LAN and SAN Connectivity Policies

Connectivity policies determine the connections and the network communication resources between the server and the LAN or SAN on the network. These policies use pools to assign MAC addresses, WWNs, and WWPNs to servers and to identify the vNICs and vHBAs that the servers use to communicate with the network.

Note We do not recommend that you use static IDs in connectivity policies, because these policies are included in service profiles and service profile templates and can be used to configure multiple servers.

Privileges Required for LAN and SAN Connectivity Policies

Connectivity policies enable users without network or storage privileges to create and modify service profiles and service profile templates with network and storage connections. However, users must have the appropriate network and storage privileges to create connectivity policies.

Privileges Required to Create Connectivity Policies

Connectivity policies require the same privileges as other network and storage configurations. For example, you must have at least one of the following privileges to create connectivity policies:

- admin—Can create LAN and SAN connectivity policies
- ls-server—Can create LAN and SAN connectivity policies
- ls-network—Can create LAN connectivity policies
- ls-storage—Can create SAN connectivity policies

Privileges Required to Add Connectivity Policies to Service Profiles

After the connectivity policies have been created, a user with ls-compute privileges can include them in a service profile or service profile template. However, a user with only ls-compute privileges cannot create connectivity policies.

Interactions between Service Profiles and Connectivity Policies

You can configure the LAN and SAN connectivity for a service profile through either of the following methods:

- LAN and SAN connectivity policies that are referenced in the service profile
- Local vNICs and vHBAs that are created in the service profile
- Local vNICs and a SAN connectivity policy
- Local vHBAs and a LAN connectivity policy

Cisco UCS maintains mutual exclusivity between connectivity policies and local vNIC and vHBA configuration in the service profile. You cannot have a combination of connectivity policies and locally created vNICs or vHBAs. When you include a LAN connectivity policy in a service profile, all existing vNIC configuration is erased, and when you include a SAN connectivity policy, all existing vHBA configuration in that service profile is erased.

Creating a SAN Connectivity Policy

Procedure

	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 san-connectivity-policy** *policy-name*	Creates the specified SAN connectivity policy, and enters organization network control policy mode. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	UCS-A /org/lan-connectivity-policy # **set descr** *policy-name*	(Optional) Adds a description to the policy. We recommend that you include information about where and how the policy should be used. Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote).
Step 4	UCS-A /org/service-profile # **set identity** {**dynamic-uuid** {*uuid* | **derived**} | **dynamic-wwnn** {*wwnn* | **derived**} | **uuid-pool** *pool-name* | **wwnn-pool** *pool-name*}	Specifies how the server acquires a UUID or WWNN. You can do one of the following: • Create a unique UUID in the form *nnnnnnnn-nnnn-nnnn-nnnnnnnnnnnn* • Derive the UUID from the one burned into the hardware at manufacture

	Command or Action	Purpose
		• Use a UUID pool • Create a unique WWNN in the form *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* • Derive the WWNN from one burned into the hardware at manufacture • Use a WWNN pool
Step 5	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy* # set descr "SAN connectivity policy"
UCS-A /org/san-connectivity-policy* # set identity wwnn-pool SanPool7
UCS-A /org/san-connectivity-policy* # commit-buffer
UCS-A /org/san-connectivity-policy #
```

What to Do Next

Add one or more vHBAs and/or initiator groups to this SAN connectivity policy.

Creating a vHBA for a SAN Connectivity Policy

If you are continuing from Creating a SAN Connectivity Policy, on page 362, begin this procedure at Step 3.

Procedure

	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 san-connectivity-policy** *policy-name*	Enters SAN connectivity policy mode for the specified SAN connectivity policy.
Step 3	UCS-A /org/san-connectivity-policy # **create vhba** *vhba-name* [**fabric** {**a** | **b**}] [**fc-if** *fc-if-name*]	Creates a vHBA for the specified SAN connectivity policy and enters vHBA mode. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.

	Command or Action	Purpose
Step 4	UCS-A /org/san-connectivity-policy/vhba # **set adapter-policy** *policy-name*	Specifies the adapter policy to use for the vHBA.
Step 5	UCS-A /org/san-connectivity-policy/vhba # **set identity** {**dynamic-wwpn** {*wwpn* | **derived**} | **wwpn-pool** *wwn-pool-name*}	Specifies the WWPN for the vHBA. You can set the storage identity using one of the following options: • Create a unique WWPN in the form *hh*:*hh*:*hh*:*hh*:*hh*:*hh*:*hh*:*hh*. You can specify a WWPN in the range from 20:00:00:00:00:00:00:00 to 20:FF:FF:FF:FF:FF:FF:FF or from 50:00:00:00:00:00:00:00 to 5F:FF:FF:FF:FF:FF:FF:FF. If you want the WWPN to be compatible with Cisco MDS Fibre Channel switches, use the WWPN template **20:00:00:25:B5:XX:XX:XX**. • Derive the WWPN from one burned into the hardware at manufacture. • Assign a WWPN from a WWN pool.
Step 6	UCS-A /org/san-connectivity-policy/vhba # **set max-field-size** *size-num*	Specifies the maximum size of the Fibre Channel frame payload (in bytes) that the vHBA supports. Enter an integer between 256 and 2112. The default is 2048.
Step 7	UCS-A /org/san-connectivity-policy/vhba # **set order** {*order-num* | **unspecified**}	Specifies the PCI scan order for the vHBA.
Step 8	UCS-A /org/san-connectivity-policy/vhba # **set pers-bind** {**disabled** | **enabled**}	Disables or enables persistent binding to Fibre Channel targets.
Step 9	UCS-A /org/san-connectivity-policy/vhba # **set pin-group** *group-name*	Specifies the SAN pin group to use for the vHBA.
Step 10	UCS-A /org/san-connectivity-policy/vhba # **set qos-policy** *policy-name*	Specifies the QoS policy to use for the vHBA.
Step 11	UCS-A /org/san-connectivity-policy/vhba # **set stats-policy** *policy-name*	Specifies the statistics threshold policy to use for the vHBA.

	Command or Action	Purpose
Step 12	UCS-A /org/san-connectivity-policy/vhba # **set template-name** *policy-name*	Specifies the vHBA template to use for the vHBA. If you choose to use a vHBA template for the vHBA, you must still complete all of the configuration not included in the vHBA template, including Steps 4, 7, and 8.
Step 13	UCS-A /org/san-connectivity-policy/vhba # **set vcon** {**1** | **2** | **3** | **4** | **any**}	Assigns the vHBA to one or all virtual network interface connections.
Step 14	UCS-A /org/san-connectivity-policy/vhba # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to configure a vHBA for a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy* # create vhba vhba3 fabric a
UCS-A /org/san-connectivity-policy/vhba* # set adapter-policy AdaptPol2
UCS-A /org/san-connectivity-policy/vhba* # set identity wwpn-pool SanPool7
UCS-A /org/san-connectivity-policy/vhba* # set max-field-size 2112
UCS-A /org/san-connectivity-policy/vhba* # set order 0
UCS-A /org/san-connectivity-policy/vhba* # set pers-bind enabled
UCS-A /org/san-connectivity-policy/vhba* # set pin-group FcPinGroup12
UCS-A /org/san-connectivity-policy/vhba* # set qos-policy QosPol5
UCS-A /org/san-connectivity-policy/vhba* # set stats-policy StatsPol2
UCS-A /org/san-connectivity-policy/vhba* # set template-name SanConnPol3
UCS-A /org/san-connectivity-policy/vhba* # set vcon any
UCS-A /org/san-connectivity-policy/vhba* # commit-buffer
UCS-A /org/san-connectivity-policy/vhba #
```

What to Do Next

If desired, add another vHBA or an initiator group to the SAN connectivity policy. If not, include the policy in a service profile or service profile template.

Deleting a vHBA from a SAN Connectivity Policy

Procedure

	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 san-connectivity-policy** *policy-name*	Enters SAN connectivity policy mode for the specified SAN connectivity policy.

	Command or Action	Purpose
Step 3	UCS-A /org/san-connectivity-policy # **delete vHBA** *vhba-name*	Deletes the specified vHBA from the SAN connectivity policy.
Step 4	UCS-A /org/san-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete a vHBA named vHBA3 from a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy # delete vHBA vHBA3
UCS-A /org/san-connectivity-policy* # commit-buffer
UCS-A /org/san-connectivity-policy #
```

Creating an Initiator Group for a SAN Connectivity Policy

If you are continuing from Creating a SAN Connectivity Policy, on page 362, begin this procedure at Step 3.

Procedure

	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 san-connectivity-policy** *policy-name*	Enters SAN connectivity policy mode for the specified SAN connectivity policy.
Step 3	UCS-A /org/san-connectivity-policy # **create initiator-group** *group-name* **fc**	Creates the specified initiator group for Fibre Channel zoning and enters initiator group mode. This name can be between 1 and 16 alphanumeric

	Command or Action	Purpose
		characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 4	UCS-A /org/san-connectivity-policy/initiator-group # **create initiator** *vhba-name*	Creates the specified vHBA initiator in the initiator group. If desired, repeat this step to add a second vHBA initiator to the group.
Step 5	UCS-A /org/san-connectivity-policy/initiator-group # **set storage-connection-policy** *policy-name*	Associates the specified storage connection policy with the SAN connectivity policy.

	Command or Action	Purpose
		Note This step assumes that you want to associate an existing storage connection policy to associate with the SAN connectivity policy. If you do, continue with Step 10. If you want to create a local storage definition for this policy instead, continue with Step 6.
Step 6	UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def # **create storage-target** *wwpn*	Creates a storage target endpoint with the specified WWPN, and enters storage target mode.
Step 7	UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target # **set target-path** {**a** | **b**}	Specifies which fabric interconnect is used for communications with the target endpoint.
Step 8	UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target # **set target-vsan** *vsan*	Specifies which VSAN is used for communications with the target endpoint.

	Command or Action	Purpose
Step 9	UCS-A /org/san-connectivity-policy/initiator-group # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to configure an initiator group named initGroupZone1 with two initiators for a a SAN connectivity policy named SanConnect242, configure a local storage connection policy definition named scPolicyZone1, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy # create initiator-group initGroupZone1 fc
UCS-A /org/san-connectivity-policy/initiator-group* # set zoning-type sist
UCS-A /org/san-connectivity-policy/initiator-group* # create initiator vhba1
UCS-A /org/san-connectivity-policy/initiator-group* # create initiator vhba2
UCS-A /org/san-connectivity-policy/initiator-group* # create storage-connection-def
scPolicyZone1
UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def* # create
storage-target
20:10:20:30:40:50:60:70
UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target*
# set
target-path a
UCS-A /org/san-connectivity-policy/initiator-group/storage-connection-def/storage-target*
# set
target-vsan default
UCS-A /org/san-connectivity-policy/initiator-group* # commit-buffer
UCS-A /org/san-connectivity-policy/initiator-group #
```

What to Do Next

If desired, add another initiator group or a vHBA to the SAN connectivity policy. If not, include the policy in a service profile or service profile template.

Deleting an Initiator Group from a SAN Connectivity Policy

Procedure

	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 san-connectivity-policy** *policy-name*	Enters SAN connectivity policy mode for the specified SAN connectivity policy.
Step 3	UCS-A /org/san-connectivity-policy # **delete initiator-group** *group-name*	Deletes the specified initiator group from the SAN connectivity policy.

	Command or Action	Purpose
Step 4	UCS-A /org/san-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an initiator group named initGroup3 from a SAN connectivity policy named SanConnect242 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope san-connectivity-policy SanConnect242
UCS-A /org/san-connectivity-policy # delete initiator-group initGroup3
UCS-A /org/san-connectivity-policy* # commit-buffer
UCS-A /org/san-connectivity-policy #
```

Deleting a SAN Connectivity Policy

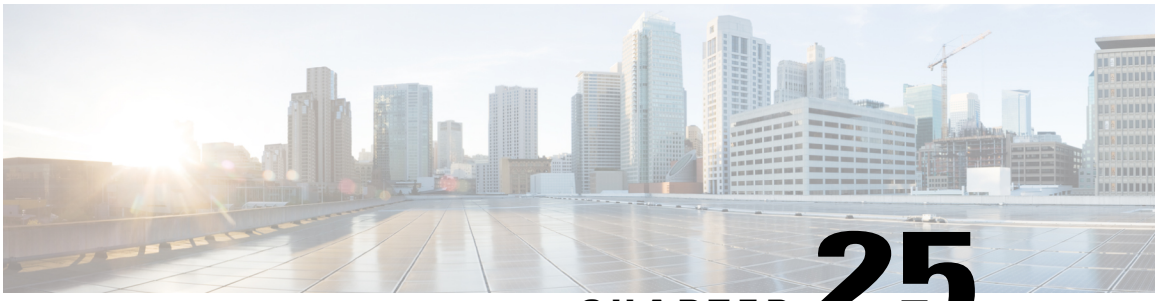
If you delete a SAN connectivity policy that is included in a service profile, it also deletes all vHBAs from that service profile and disrupts SAN data traffic for the server associated with the service profile.

Procedure

	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 san-connectivity-policy** *policy-name*	Deletes the specified SAN connectivity policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete a SAN connectivity policy named SanConnect52 from the root organization and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete san-connectivity-policy SanConnect52
UCS-A /org* # commit-buffer
UCS-A /org #
```



CHAPTER 25

Configuring Fibre Channel Zoning

This chapter includes the following sections:

Information About Fibre Channel Zoning

Fibre Channel zoning allows you to partition the Fibre Channel fabric into one or more zones. Each zone defines the set of Fibre Channel initiators and Fibre Channel targets that can communicate with each other in a VSAN. Zoning also enables you to set up access control between hosts and storage devices or user groups.

The access and data traffic control provided by zoning does the following:

- Enhances SAN network security
- Helps prevent data loss or corruption
- Reduces performance issues

Information About Zones

A zone consists of multiple zone members and has the following characteristics:

- Members in a zone can access each other; members in different zones cannot access each other.
- Zones can vary in size.
- Devices can belong to more than one zone.

- A physical fabric can have a maximum of 8,000 zones.

Information About Zone Sets

Each zone set consists of one or more zones. You can use zone sets to enforce access control within the Fibre Channel fabric. In addition, zone sets provide you with the following advantages:

- Only one zone set can be active at any time.
- All zones in a zone set can be activated or deactivated as a single entity across all switches in the fabric.
- A zone can be a member of more than one zone set.
- A switch in a zone can have a maximum of 500 zone sets.

Support for Fibre Channel Zoning in Cisco UCS Manager

Cisco UCS Manager supports switch-based Fibre Channel zoning and Cisco UCS Manager-based Fibre Channel zoning. You cannot configure a combination of zoning types in the same Cisco UCS domain. You can configure a Cisco UCS domain with one of the following types of zoning:

- Cisco UCS Manager-based Fibre Channel zoning—This configuration combines direct attach storage with local zoning. Fibre Channel or FCoE storage is directly connected to the fabric interconnects and zoning is performed in Cisco UCS Manager, using Cisco UCS local zoning. Any existing Fibre Channel or FCoE uplink connections need to be disabled. Cisco UCS does not currently support active Fibre Channel or FCoE uplink connections coexisting with the utilization of the UCS Local Zoning feature.
- Switch-based Fibre Channel zoning—This configuration combines direct attach storage with uplink zoning. The Fibre Channel or FCoE storage is directly connected to the fabric interconnects and zoning is performed externally to the Cisco UCS domain through an MDS or Nexus 5000 switch. This configuration does not support local zoning in the Cisco UCS domain.

Note Zoning is configured on a per-VSAN basis. You cannot enable zoning at the fabric level.

Cisco UCS Manager-Based Fibre Channel Zoning

With Cisco UCS Manager-based zoning, Cisco UCS Manager controls the Fibre Channel zoning configuration for the Cisco UCS domain, including creating and activating zones for all VSANs that you set up with this type of zoning. This type of zoning is also know as local zoning or direct attach storage with local zoning.

Note You cannot implement Cisco UCS Manager-based zoning if the VSAN is also configured to communicate with a VSAN on an upstream switch and includes Fibre Channel or FCoE uplink ports.

Supported Fibre Channel Zoning Modes

Cisco UCS Manager-based zoning supports the following types of zoning:

- Single initiator single target—Cisco UCS Manager automatically creates one zone for each vHBA and storage port pair. Each zone has two members. We recommend that you configure this type of zoning unless you expect the number of zones to exceed the maximum supported.
- Single initiator multiple targets—Cisco UCS Manager automatically creates one zone for each vHBA. We recommend that you configure this type of zoning if you expect the number of zones to reach or exceed the maximum supported.

vHBA Initiator Groups

vHBA initiator groups determine the Fibre Channel zoning configuration for all vHBAs in a service profile. Cisco UCS Manager does not include any default vHBA initiator groups. You must create vHBA initiator groups in any service profile that is to be assigned to servers included in a zone.

The configuration in a vHBA initiator group determines the following:

- The vHBAs included in the initiator group, which are sometimes referred to as vHBA initiators.
- A Fibre Channel storage connection policy, which includes the associated VSAN and the Fibre Channel target ports on the storage array.
- The type of Fibre Channel zoning to be configured for the vHBAs included in the group.

Fibre Channel Storage Connection Policy

The Fibre Channel storage connection policy contains a collection of target storage ports on storage arrays that you use to configure Cisco UCS Manager-based Fibre Channel zoning. You can create this policy under an organization or an initiator group.

The storage arrays in these zones must be directly connected to the fabric interconnects. The target storage ports on these arrays that you include in the Fibre Channel storage connection policy can be either Fibre Channel storage ports or FCoE storage ports. You use the WWN of a port to add it to the policy and to identify the port for the Fibre Channel zone.

Note Cisco UCS Manager does not create default Fibre Channel storage.

Fibre Channel Active Zone Set Configuration

In each VSAN that has been enabled for Fibre Channel zoning, Cisco UCS Manager automatically configures one zone set and multiple zones. The zone membership specifies the set of initiators and targets that are allowed to communicate with each other. Cisco UCS Manager automatically activates that zone set.

Cisco UCS Manager processes the user-configured vHBA initiator groups and their associated Fibre Channel storage connection policy to determine the desired connectivity between Fibre Channel initiators and targets. Cisco UCS Manager uses the following information to build pair-wise zone membership between initiators and targets:

- The port WWNs of the vHBA initiators derived from the vHBA initiator groups.
- The port WWNs of the storage array derived from the storage connection policy.

Switch-Based Fibre Channel Zoning

With switch-based zoning, a Cisco UCS domain inherits the zoning configuration from the upstream switch. You cannot configure or view information about your zoning configuration in Cisco UCS Manager. You have to disable zoning on a VSAN in Cisco UCS Manager to use switch-based zoning for that VSAN.

GuidelinesandrecommendationsforCiscoUCSManager-Based Fibre Channel Zoning

When you plan your configuration for Fibre Channel zoning, consider the following guidelines and recommendations:

Fibre Channel Switching Mode Must Be Switch Mode for Cisco UCS Manager Configurations

If you want Cisco UCS Manager to handle Fibre Channel zoning, the fabric interconnects must be in Fibre Channel Switch mode. You cannot configure Fibre Channel zoning in End-Host mode.

Symmetrical Configuration Is Recommended for High Availability

If a Cisco UCS domain is configured for high availability with two fabric interconnects, we recommend that both fabric interconnects are configured with the same set of VSANs.

Configuring Cisco UCS Manager Fibre Channel Zoning

Note This procedure provides a high level overview of the steps required to configure a Cisco UCS domain for Fibre Channel zoning that is controlled by Cisco UCS Manager. You must ensure that you complete all of the following steps.

Procedure

	Command or Action	Purpose
Step 1	If you have not already done so, disconnect the fabric interconnects in the Cisco UCS domain from any external Fibre Channel switches, such as an MDS.	

	Command or Action	Purpose
Step 2	If the Cisco UCS domain still includes zones that were managed by the external Fibre Channel switch, run the **clear-unmanaged-fc-zone-all** command on every affected VSAN to remove those zones.	This functionality is not currently available in the Cisco UCS Manager GUI. You must perform this step in the Cisco UCS Manager CLI.
Step 3	Configure the Fibre Channel switching mode for both fabric interconnects in Fibre Channel Switch mode.	You cannot configure Fibre Channel zoning in End-Host mode. See Configuring Fibre Channel Switching Mode, on page 59.
Step 4	Configure the Fibre Channel and FCoE storage ports that you require to carry traffic for the Fibre Channel zones.	See Configuring Ports and Port Channels, on page 61.
Step 5	Create one or more VSANs and enable Fibre Channel zoning on all VSANs that you require to carry traffic for the Fibre Channel zones.	For a cluster configuration, we recommend that you create the VSANs that you intend to include in a Fibre Channel zone in Fibre Channel storage mode and accessible to both fabric interconnects. See Configuring Named VSANs, on page 329.
Step 6	Create one or more Fibre Channel storage connection policies.	You can perform this step when you configure Fibre Channel zoning in the service profiles, if you prefer. See Creating a Fibre Channel Storage Connection Policy, on page 377.
Step 7	Configure zoning in service profiles or service profile templates for servers that need to communicate through Fibre Channel zones.	Complete the following steps to complete this configuration: • Enable zoning in the VSAN or VSANs assigned to the VHBAs. • Configure one or more vHBA initiator groups.

Removing Unmanaged Zones from a VSAN Accessible to Both Fabric Interconnects

After you disconnect the external Fibre Channel switch, the Fibre Channel zones that were managed by that switch might not been cleared from the Cisco UCS domain. This procedure removes those zones from each VSAN in the Cisco UCS domain so that you can configure Fibre Channel zoning in Cisco UCS.

Before You Begin

If you have not already done so, disconnect the fabric interconnects in the Cisco UCS domain from any external Fibre Channel switches, such as an MDS.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink mode for the specified fabric interconnect.
Step 3	UCS-A /fc-uplink/fabric # **scope vsan** *vsan-name*	Enters VSAN mode for the specified named VSAN.
Step 4	UCS-A /fc-uplink/fabric/vsan # **clear-unmanaged-fc-zones-all**	Clears all unmanaged Fibre Channel zones from the specified named VSAN. If desired, you can repeat Steps 2 through 4 to remove unmanaged zones from all VSANs that are accessible to the specified fabric interconnect before you commit the buffer.
Step 5	UCS-A /fc-uplink/fabric/vsan # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to remove unmanaged zones from a named VSAN accessible to fabric interconnect A and commit the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope vsan finance
UCS-A /fc-uplink/fabric/vsan # clear-unmanaged-fc-zones-all
UCS-A /fc-uplink/fabric/vsan* # commit-buffer
UCS-A /fc-uplink #
```

Removing Unmanaged Zones from a VSAN Accessible to One Fabric Interconnect

After you disconnect the external Fibre Channel switch, the Fibre Channel zones that were managed by that switch might not been cleared from the Cisco UCS domain. This procedure removes those zones from each VSAN in the Cisco UCS domain so that you can configure Fibre Channel zoning in Cisco UCS.

Before You Begin

If you have not already done so, disconnect the fabric interconnects in the Cisco UCS domain from any external Fibre Channel switches, such as an MDS.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope vsan** *vsan-name*	Enters VSAN mode for the specified named VSAN.
Step 3	UCS-A /fc-uplink/vsan # **clear-unmanaged-fc-zones-all**	Clears all unmanaged Fibre Channel zones from the specified named VSAN. If desired, you can repeat steps 2 and 3 to remove unmanaged zones from all VSANs that are accessible to both fabric interconnects before you commit the buffer.
Step 4	UCS-A /fc-uplink/vsan # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to remove unmanaged zones from a named VSAN and commit the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope vsan finance
UCS-A /fc-uplink/vsan # clear-unmanaged-fc-zones-all
UCS-A /fc-uplink/vsan* # commit-buffer
UCS-A /fc-uplink #
```

Configuring Fibre Channel Storage Connection Policies

Creating a Fibre Channel Storage Connection Policy

Procedure

	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-connection-policy** *policy-name*	Creates a storage connection policy with the specified policy name, and enters organization storage connection policy mode.
Step 3	UCS-A /org # **set zoning-type** {**none** | **simt** | **sist**}	• **None—**Cisco UCS Manager does not configure Fibre Channel zoning. • **Single Initiator Single Target—**Cisco UCS Manager automatically creates one zone for each vHBA and storage port pair. Each zone

	Command or Action	Purpose
		has two members. We recommend that you configure this type of zoning unless you expect the number of zones to exceed the maximum supported. • **Single Initiator Multiple Targets—**Cisco UCS Manager automatically creates one zone for each vHBA. We recommend that you configure this type of zoning if you expect the number of zones to reach or exceed the maximum supported.
Step 4	UCS-A /org/storage-connection-policy # **create storage-target** *wwpn*	Creates a storage target endpoint with the specified WWPN, and enters storage target mode.
Step 5	UCS-A /org/storage-connection-policy/storage-target # **set target-path** {**a** | **b**}	Specifies which fabric interconnect is used for communications with the target endpoint.
Step 6	UCS-A /org/storage-connection-policy/storage-target # **set target-vsan** *vsan*	Specifies which VSAN is used for communications with the target endpoint.
Step 7	UCS-A /org/storage-connection-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a Fibre Channel storage connection policy in the root organization named scPolicyZone1, using fabric interconnect A and the default VSAN, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create storage-connection-policy scPolicyZone1
UCS-A /org/storage-connection-policy* set zoning-type sist
UCS-A /org/storage-connection-policy* # create storage-target 20:10:20:30:40:50:60:70
UCS-A /org/storage-connection-policy/storage-target* # set target-path a
UCS-A /org/storage-connection-policy/storage-target* # set target-vsan default
UCS-A /org/storage-connection-policy* # commit-buffer
UCS-A /org/storage-connection-policy #
```

Deleting a Fibre Channel Storage Connection Policy

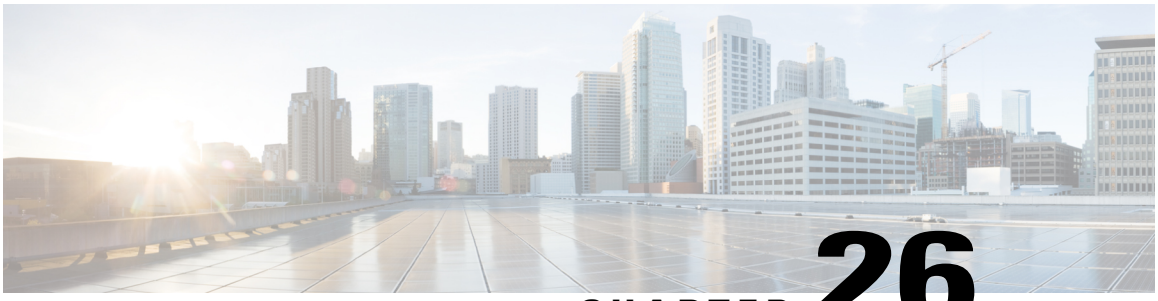
Procedure

	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-connection-policy** *policy-name*	Deletes the specified storage connection policy.

	Command or Action	Purpose
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the storage connection policy named scPolicyZone1 from the root organization and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete san-connectivity-policy scPolicyZone1
UCS-A /org* # commit-buffer
UCS-A /org #
```



CHAPTER 26

Configuring Server-Related Pools

This chapter includes the following sections:

Server Pool Configuration

Server Pools

A server pool contains a set of servers. These servers typically share the same characteristics. Those characteristics can be their location in the chassis, or an attribute such as server type, amount of memory, local storage, type of CPU, or local drive configuration. You can manually assign a server to a server pool, or use server pool policies and server pool policy qualifications to automate the assignment.

If your system implements multitenancy through organizations, you can designate one or more server pools to be used by a specific organization. For example, a pool that includes all servers with two CPUs could be assigned to the Marketing organization, while all servers with 64 GB memory could be assigned to the Finance organization.

A server pool can include servers from any chassis in the system. A given server can belong to multiple server pools.

Creating a Server Pool

Procedure

	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 # **create server-pool** *server-pool-name*	Creates a server pool with the specified name, and enters organization server pool mode.
Step 3	UCS-A /org/server-pool # **create server** *chassis-num*/*slot-num*	Creates a server for the server pool. **Note** A server pool can contain more than one server. To create multiple servers for the pool, you must enter multiple **create server** commands from organization server pool mode.
Step 4	UCS-A /org/server-pool # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a server pool named ServPool2, create two servers for the server pool, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create server-pool ServPool2
UCS-A /org/server-pool* # create server 1/1
UCS-A /org/server-pool* # create server 1/4
UCS-A /org/server-pool* # commit-buffer
UCS-A /org/server-pool #
```

Deleting a Server Pool

Procedure

	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 # **delete server-pool** *server-pool-name*	Deletes the specified server pool.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the server pool named ServPool2 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete server-pool ServPool2
UCS-A /org* # commit-buffer
UCS-A /org #
```

UUID Suffix Pool Configuration

UUID Suffix Pools

A UUID suffix pool is a collection of SMBIOS UUIDs that are available to be assigned to servers. The first number of digits that constitute the prefix of the UUID are fixed. The remaining digits, the UUID suffix, are variable. A UUID suffix pool ensures that these variable values are unique for each server associated with a service profile which uses that particular pool to avoid conflicts.

If you use UUID suffix pools in service profiles, you do not have to manually configure the UUID of the server associated with the service profile.

Creating a UUID Suffix Pool

Procedure

	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 uuid-suffix-pool** *pool-name*	Creates a UUID suffix pool with the specified pool name and enters organization UUID suffix pool mode. This name can be between 1 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	UCS-A /org/uuid-suffix-pool # **set descr** *description*	(Optional) Provides a description for the UUID suffix pool. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/uuid-suffix-pool # **set assignmentorder** {**default** | **sequential**}	This can be one of the following: • **default**—Cisco UCS Manager selects a random identity from the pool. • **sequential**—Cisco UCS Manager selects the lowest available identity from the pool.

	Command or Action	Purpose
Step 5	UCS-A /org/uuid-suffix-pool # **create block** *first-uuid last-uuid*	Creates a block (range) of UUID suffixes, and enters organization UUID suffix pool block mode. You must specify the first and last UUID suffixes in the block using the form *nnnn-nnnnnnnnnnnn*, with the UUID suffixes separated by a space. **Note** A UUID suffix pool can contain more than one UUID suffix block. To create multiple blocks, you must enter multiple **create block** commands from organization UUID suffix pool mode.
Step 6	UCS-A /org/uuid-suffix-pool/block # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a UUID suffix pool named pool4, provide a description for the pool, specify a block of UUID suffixes to be used for the pool, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create uuid-suffix-pool pool4
UCS-A /org/uuid-suffix-pool* # set descr "This is UUID suffix pool 4"
UCS-A /org/uuid-suffix-pool* # create block 1000-000000000001 1000-000000000010
UCS-A /org/uuid-suffix-pool/block* # commit-buffer
UCS-A /org/uuid-suffix-pool/block #
```

What to Do Next

Include the UUID suffix pool in a service profile and/or template.

Deleting a UUID Suffix Pool

If you delete a pool, Cisco UCS Manager does not reallocate any addresses from that pool that were assigned to vNICs or vHBAs. All assigned addresses from a deleted pool remain with the vNIC or vHBA to which they are assigned until one of the following occurs:

- The associated service profiles are deleted.
- The vNIC or vHBA to which the address is assigned is deleted.
- The vNIC or vHBA is assigned to a different pool.

Procedure

	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 uuid-suffix-pool** *pool-name*	Deletes the specified UUID suffix pool.

	Command or Action	Purpose
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the UUID suffix pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete uuid-suffix-pool pool4
UCS-A /org* # commit-buffer
UCS-A /org #
```

IP Pool Configuration

IP Pools

IP pools are collections of IP addresses that do not have a default purpose. You can create IPv4 or IPv6 address pools in Cisco UCS Manager to do the following:

-
- Replace the default management IP pool **ext-mgmt** for servers that have an associated service profile. Cisco UCS Manager reserves each block of IP addresses in the IP pool for external access that terminates in the Cisco Integrated Management Controller (CIMC) on a server. If there is no associated service profile, you must use the **ext-mgmt** IP pool for the CIMC to get an IP address.
- Replace the management inband or out-of-band IP addresses for the CIMC.

Note You cannot create iSCSI boot IPv6 pools in Cisco UCS Manager.

You can create IPv4 address pools in Cisco UCS Manager to do the following:

- Replace the default iSCSI boot IP pool **iscsi-initiator-pool**. Cisco UCS Manager reserves each block of IP addresses in the IP pool that you specify.
- Replace both the management IP address and iSCSI boot IP addresses.

Note The IP pool must not contain any IP addresses that were assigned as static IP addresses for a server or service profile.

Creating an Inband IP Pool

You can configure an inband IP pool with blocks of IPv4 and IPv6 addresses.

Procedure

	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 ip-pool** *pool-name*	Creates an IP pool with the specified name, and enters organization IP pool mode. This name can be between 1 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	UCS-A /org/ip-pool # **set descr** *description*	(Optional) Provides a description for the IP pool. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/ip-pool # **create block** *first-ip-addr last-ip-addr gateway-ip-addr subnet-mask*	Creates a block (range) of IP addresses, and enters organization IP pool block mode. You must specify the first and last IP addresses in the address range, the gateway IP address, and subnet mask.
Step 5	UCS-A /org/ip-pool/block # **commit-buffer**	Commits the transaction to the system configuration.
Step 6	UCS-A /org/ip-pool/block # **exit**	Exits organization IP pool block mode.
Step 7	UCS-A /org/ip-pool # **create ip6block** *first-ip6-addr last-ip6-addr gateway-ip6-addr prefix*	Creates a block of IPv6 addresses, and enters organization IPv6 pool block mode. You must specify the first and last IPv6 addresses in the address range, the gateway IPv6 address, and network prefix.
Step 8	UCS-A/org/ip-pool/ipv6-block # **commit-buffer**	Commits the transaction to the system configuration.

The example below creates an inband IP pool named inband-default, creates a block of IPv4 addresses, creates a block of IPv6 addresses, and commits the transaction:

```
UCS-A# scope org
UCS-A /org # create ip-pool inband_default
UCS-A /org/ip-pool* # create block 192.168.100.10 192.168.100.100 192.168.100.1 255.255.255.0
UCS-A /org/ip-pool/block* # commit-buffer
UCS-A /org/ip-pool/block # exit
UCS-A /org/ip-pool # create ipv6-block 2001:888::10 2001:888::100 2001:888::1 64
UCS-A /org/ip-pool/ipv6-block* # commit-buffer
UCS-A /org/ip-pool/ipv6-block #
```

What to Do Next

Include the IP pool in a service profile and template.

Adding Blocks to an IP Pool

Procedure

	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 ip-pool** *pool-name*	Enters organization IP pool mode for the specified pool.
Step 3	UCS-A /org/ip-pool # **create** *block first-ip-addr last-ip-addr gateway-ip-addr subnet-mask*	Creates a block (range) of IP addresses, and enters organization IP pool block mode. You must specify the first and last IP addresses in the address range, the gateway IP address, and subnet mask. **Note** An IP pool can contain more than one IP block. To create multiple blocks, enter multiple **create block** commands from organization IP pool mode.
Step 4	UCS-A /org/ip-pool/block # **commit-buffer**	Commits the transaction.
Step 5	UCS-A /org/ip-pool/block # **exit**	Exits IPv4 block configuration mode.
Step 6	UCS-A /org/ip-pool # **create ipv6-block** *first-ip6-addr last-ip6-addr gateway-ip6-addr prefix*	Creates a block (range) of IPv6 addresses, and enters organization IP pool IPv6 block mode. You must specify the first and last IPv6 addresses in the address range, the gateway IPv6 address, and network prefix. **Note** An IP pool can contain more than one IPv6 block. To create multiple IPv6 blocks, enter multiple **create ipv6-block** commands from organization IP pool mode.
Step 7	UCS-A /org/ip-pool/ ipv6-block # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to add blocks of IPv4 and IPv6 addresses to an IP pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope ip-pool pool4
UCS-A /org/ip-pool # create block 192.168.100.1 192.168.100.200 192.168.100.10 255.255.255.0
UCS-A /org/ip-pool/block* # commit-buffer
UCS-A /org/ip-pool/block #exit
UCS-A /org/ip-pool* # create ipv6-block 2001:888::10 2001:888::100 2001:888::1 64
UCS-A /org/ip-pool/ipv6-block* commit-buffer
```

Deleting a Block from an IP Pool

If you delete an address block from a pool, Cisco UCS Manager does not reallocate any addresses in that block that were assigned to vNICs or vHBAs. All assigned addresses from a deleted block remain with the vNIC or vHBA to which they are assigned until one of the following occurs:

- The associated service profiles are deleted.
- The vNIC or vHBA to which the address is assigned is deleted.
- The vNIC or vHBA is assigned to a different pool.

Note IPv6 address blocks are not applicable to vNICs or vHBAs.

Procedure

	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 ip-pool***pool-name*	Enters organization IP pool mode for the specified pool.
Step 3	UCS-A /org/ip-pool # **delete** {*ip-block*|*ipv6-block*} {*first-ip-addr*|*first-ip6-addr*} {*last-ip-addr*|*last-ip6-addr*}	Deletes the specified block (range) of IPv4 or IPv6 addresses.
Step 4	UCS-A /org/ip-pool # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to delete an IP address block from an IP pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope ip-pool pool4
UCS-A /org/ip-pool # delete block 192.168.100.1 192.168.100.200
UCS-A /org/ip-pool* # commit-buffer
UCS-A /org/ip-pool #
```

This example shows how to delete an IPv6 address block from an IP pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope ip-pool pool4
UCS-A /org/ip-pool # delete ipv6-block 2001::1 2001::10
UCS-A /org/ip-pool* # commit-buffer
UCS-A /org/ip-pool #
```

Deleting an IP Pool

If you delete a pool, Cisco UCS Manager does not reallocate any addresses from that pool that were assigned to vNICs or vHBAs. All assigned addresses from a deleted pool remain with the vNIC or vHBA to which they are assigned until one of the following occurs:

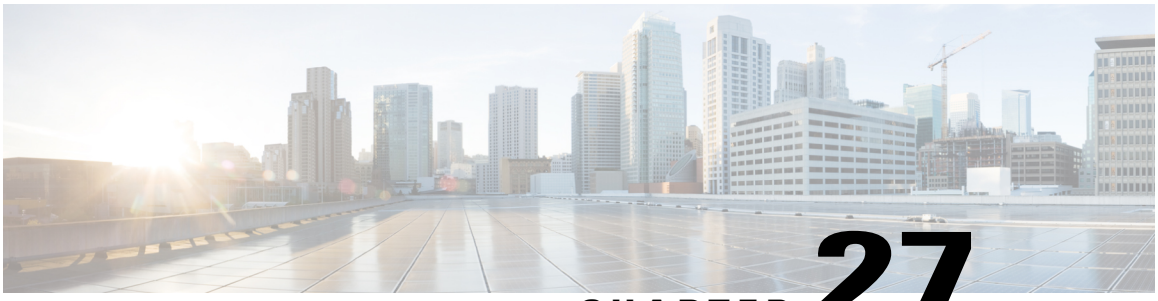
- The associated service profiles are deleted.
- The vNIC or vHBA to which the address is assigned is deleted.
- The vNIC or vHBA is assigned to a different pool.

Procedure

	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 ip-pool** *pool-name*	Deletes the specified IP pool.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the IP pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete ip-pool pool4
UCS-A /org* # commit-buffer
UCS-A /org #
```



CHAPTER 27

Setting the Management IP Address

This chapter includes the following sections:

Management IP Address

Each server in a Cisco UCS domain must have a one or more management IP addresses assigned to its Cisco Integrated Management Controller (CIMC) or to the service profile associated with the server. Cisco UCS Manager uses these IP addresses for external access that terminates in the CIMC. This external access can be through one of the following services:

- KVM console
- Serial over LAN
- An IPMI tool

The management IP addresses used to access the CIMC on a server can be out-of-band (OOB) addresses, through which traffic traverses the fabric interconnect via the management port, or inband addresses, through which traffic traverses the fabric interconnect via the fabric uplink port. Up to six IP addresses can be configured to access the CIMC on a server, two out-of-band (OOB) and four inband.

You can configure the following management IP addresses:

- A static OOB IPv4 address assigned directly to the server
- An OOB IPv4 address assigned to the server from a global ext-mgmt pool
- An inband IPv4 address derived from a service profile associated with the server
- An inband IPv4 address drawn from a management IP pool and assigned to a service profile or service profile template

- An static inband IPv6 address assigned directly to the server
- An inband IPv6 address derived from a service profile associated with the server

You can assign multiple management IP addresses to each CIMC on the server and to the service profile associated with the server. If you do so, you must use different IP addresses for each of them.

A management IP address that is assigned to a service profile moves with that service profile. If KVM or SoL sessions are active when you migrate the service profile to another server, Cisco UCS Manager terminates the sessions and does not restart them after the migration is completed. You configure the IP address when you create or modify a service profile.

Note You cannot assign a static IP address to a server or service profile if that IP address has already been assigned to a server or service profile in the Cisco UCS domain. If you attempt to do so, Cisco UCS Manager warns you that the IP address is already in use and rejects the configuration.

An ARP request will be sent to the gateway IP address every second from each server that is configured with an Inband IP address. This is to check if connectivity for the Inband traffic through the current Fabric Interconnect is up, and to initiate a failover to the other Fabric Interconnect if it is down. The path selected for Inband and the failover operations are completely independent of the server data traffic.

Configuring the Management IP Address on a Blade Server

Configuring a Blade Server to Use a Static IP Address

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-id* / *blade-id*	Enters chassis server mode for the specified server.
Step 2	UCS-A /chassis/server # **scope cimc**	Enters chassis server CIMC mode.
Step 3	UCS-A /chassis/server/cimc # **create ext-static-ip**	Creates a static management IP address for the specified server.
Step 4	UCS-A /chassis/server/cimc/ext-static-ip # **set addr** *ip-addr*	Specifies the static IPv4 address to be assigned to the server.
Step 5	UCS-A /chassis/server/cimc/ext-static-ip # **set default-gw** *ip-addr*	Specifies the default gateway that the IP address should use.
Step 6	UCS-A /chassis/server/cimc/ext-static-ip # **set subnet** *ip-addr*	Specifies the subnet mask for the IP address.
Step 7	UCS-A /chassis/server/cimc/ext-static-ip # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a static management IP address for chassis 1 server 1, sets the static IPv4 address, sets the default gateway, sets the subnet mask, and commits the transaction:

```
UCS-A# scope server 1/1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # create ext-static-ip
UCS-A /chassis/server/cimc/ext-static-ip* # set addr 192.168.10.10
UCS-A /chassis/server/cimc/ext-static-ip* # set default-gw 192.168.10.1
UCS-A /chassis/server/cimc/ext-static-ip* # set subnet 255.255.255.0
UCS-A /chassis/server/cimc/ext-static-ip* # commit-buffer
UCS-A /chassis/server/cimc/ext-static-ip #
```

Configuring a Blade Server to Use a Static IPv6 Address

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-id* / *blade-id*	Enters chassis server mode for the specified server.
Step 2	UCS-A /chassis/server # **scope cimc**	Enters chassis server CIMC mode.
Step 3	UCS-A /chassis/server/cimc # **create ext-static-ip6**	Creates a static management IPv6 address for the specified server.
Step 4	UCS-A /chassis/server/cimc/ext-static-ip6 # **set addr ipv6-addr**	Specifies the static IPv6 address to be assigned to the server.
Step 5	UCS-A /chassis/server/cimc/ext-static-ip6 # **set default-gw ip6-addr**	Specifies the default gateway that the IPv6 address should use.
Step 6	UCS-A /chassis/server/cimc/ext-static-ip6 # **set prefix ip6-addr**	Specifies the network prefix for an IPv6 address.
Step 7	UCS-A /chassis/server/cimc/ext-static-ip6 # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a static management IPv6 address for chassis 1 server 1, sets a static IPv6 address, sets the default gateway, sets the network prefix, and commits the transaction:

```
UCS-A# scope server 1/1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # create ext-static-ip6
UCS-A /chassis/server/cimc/ext-static-ip* # set addr 2001:888::10
UCS-A /chassis/server/cimc/ext-static-ip* # set default-gw 2001:888::100
UCS-A /chassis/server/cimc/ext-static-ip* # set prefix 64
UCS-A /chassis/server/cimc/ext-static-ip* # commit-buffer
UCS-A /chassis/server/cimc/ext-static-ip #
```

Configuring a Blade Server to Use the Management IP Pool

Deleting the static management IP address returns the specified server to the management IP pool.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-id* / *blade-id*	Enters chassis server mode for the specified server.
Step 2	UCS-A /chassis/server # **scope cimc**	Enters chassis server CIMC mode.
Step 3	UCS-A /chassis/server/cimc # **delete** {*ext-static-ip* | *ext-static-ip6*}	Deletes the external static IPv4 or IPv6 address and returns the blade server to the management IP pool.
Step 4	UCS-A /chassis/server/cimc/ # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the static management IP address for chassis 1 server 1 and commits the transaction:

```
UCS-A# scope server 1/1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # delete ext-static-ip
UCS-A /chassis/server/cimc* # commit-buffer
UCS-A /chassis/server/cimc/ #
```

The following example deletes the static management IPv6 address for chassis 1 server 1 and commits the transaction:

```
UCS-A# scope server 1/1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # delete ext-static-ip6
UCS-A /chassis/server/cimc* # commit-buffer
UCS-A /chassis/server/cimc/ #
```

Configuring the Management IP Address on a Rack Server

Configuring a Rack Server to Use a Static IP Address

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *blade-id*	Enters server mode for the specified server.
Step 2	UCS-A /server # **scope cimc**	Enters server CIMC mode.
Step 3	UCS-A /server/cimc # **create ext-static-ip**	Creates a static management IP address for the specified server.
Step 4	UCS-A /server/cimc/ext-static-ip # **set addr** *ip-addr*	Specifies the static IPv4 address to be assigned to the server.

	Command or Action	Purpose
Step 5	UCS-A /server/cimc/ext-static-ip # **set default-gw** *ip-addr*	Specifies the default gateway that the IP address should use.
Step 6	UCS-A /server/cimc/ext-static-ip # **set subnet** *ip-addr*	Specifies the subnet mask for the IP address.
Step 7	UCS-A /server/cimc/ext-static-ip # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a static management IP address for rack server 1, sets the static IPv4 address, sets the default gateway, sets the subnet mask, and commits the transaction:

```
UCS-A# scope server 1
UCS-A /server # scope cimc
UCS-A /server/cimc # create ext-static-ip
UCS-A /server/cimc/ext-static-ip* # set addr 192.168.10.10
UCS-A /server/cimc/ext-static-ip* # set default-gw 192.168.10.1
UCS-A /server/cimc/ext-static-ip* # set subnet 255.255.255.0
UCS-A /server/cimc/ext-static-ip* # commit-buffer
UCS-A /server/cimc/ext-static-ip #
```

Configuring a Rack Server to Use a Static IPv6 Address

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *blade-id*	Enters server mode for the specified server.
Step 2	UCS-A /server # **scope cimc**	Enters server CIMC mode.
Step 3	UCS-A /server/cimc # **create ext-static-ip6**	Creates a static management IPv6 address for the specified server.
Step 4	UCS-A /server/cimc/ext-static-ip6 # **set addr** *ip6-addr*	Specifies the static IPv6 address to be assigned to the server.
Step 5	UCS-A /server/cimc/ext-static-ip6 # **set default-gw** *ip6-addr*	Specifies the default gateway that the IP address should use.
Step 6	UCS-A /server/cimc/ext-static-ip6 # **set prefix** *ip6-addr*	Specifies the network prefix for the IPv6 address.
Step 7	UCS-A /server/cimc/ext-static-ip # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a static management IPv6 address for rack server 1, sets the static IPv4 address, sets the default gateway, sets the network prefix, and commits the transaction:

```
UCS-A# scope server 1
UCS-A /server # scope cimc
UCS-A /server/cimc # create ext-static-ip6
UCS-A /server/cimc/ext-static-ip6* # set addr 2001::8999
UCS-A /server/cimc/ext-static-ip6* # set default-gw 2001::1
UCS-A /server/cimc/ext-static-ip6* # set prefix 64
UCS-A /server/cimc/ext-static-ip6* # commit-buffer
UCS-A /server/cimc/ext-static-ip #
```

Configuring a Rack Server to Use the Management IP Pool

Deleting the static management IP address returns the specified server to the management IP pool.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *blade-id*	Enters server mode for the specified server.
Step 2	UCS-A /server # **scope cimc**	Enters server CIMC mode.
Step 3	UCS-A /server/cimc # **delete** {*ext-static-ip* | *ext-static-ip6*}	Deletes the external static IPv4 or IPv6 address and returns the rack server to the management IP pool.
Step 4	UCS-A /server/cimc/ # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the static management IP address for rack server 1 and commits the transaction:

```
UCS-A# scope server 1
UCS-A /server # scope cimc
UCS-A /server/cimc # delete ext-static-ip
UCS-A /server/cimc* # commit-buffer
UCS-A /server/cimc/ #
```

The following example deletes the static management IPv6 address for rack server 1 and commits the transaction:

```
UCS-A# scope server 1
UCS-A /server # scope cimc
UCS-A /server/cimc # delete ext-static-ip6
UCS-A /server/cimc* # commit-buffer
UCS-A /server/cimc/ #
```

Setting the Management IP Address on a Service Profile or Service Profile Template

Procedure

	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.
Step 3	UCS-A /org/service-profile # **set ext-mgmt-ip-state** {*none* | *ext-pooled-ip* | *ext-pooled-ip6*|*ext-static-ip*|*ext-static-ip6*}	Specifies how the management IPv4 or IPv6 address will be assigned to the service profile. You can set the management IP address policy using the following options: • None--The service profile is not assigned an IP address. • Pooled--The service profile is assigned an IP address from the management IPv4 or IPv6 pool. • Static--The service profile is assigned the configured static IPv4 or IPv6 address. **Note** Setting the ext-management-ip-state to static for a service profile template is not supported and will result in an error.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the management address policy for a service profile called accounting to static IPv4 and then commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # set ext-mgmt-ip-state ext-static-ip
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

What to Do Next

If you have set the management IP address to static, configure a server to use a static IP address.

Configuring the Management IP Pool

Management IP Pools

The default management IP pool, **IP Pool ext-mgmt** is a collection of external IPv4 and IPv6 addresses. Cisco UCS Manager reserves each block of IP addresses in the management IP pool for external access that terminates in the CIMC on a server.

By default, the **IP Pool ext-mgmt** is used to configure the CIMC outbound management IP address. You cannot change this IP pool if already a static IP address is assigned to the server from this pool. If you want to configure the outbound management IP address for CIMC from a static IP address, then you can delete the IP addresses from the default management IP pool.

You can configure separate out-of-band IPv4 address pools, and in-band IPv4 or IPv6 address pools. You can configure in-band pools that contain both IPv4 and IPv6 address blocks.

Tip To avoid assigning an IP pool that contains only IPv4 addresses as the in-band IPv6 policy, or assigning an IP pool that contains only IPv6 addresses as the in-band IPv4 policy to a server CIMC, it is suggested that you configure separate in-band address pools, each with only IPv4 or IPv6 addresses.

You can configure service profiles and service profile templates to use IP addresses from the management IP pools. You cannot configure servers to use the management IP pool.

All IP addresses in the management IP pool must be in the same IPv4 subnet, or have the same IPv6 network prefix as the IP address of the fabric interconnect.

Note The management IP pool must not contain any IP addresses that were assigned as static IP addresses for a server or service profile.

Configuring IP Address Blocks for the Management IP Pool

The management IP pool must not contain any IP addresses that were assigned as static IP addresses for a server or service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters root organization mode.
Step 2	UCS-A /org # **scope ip-pool ext-mgmt**	Enters organization IP pool mode. **Note** You cannot create (or delete) a management IP pool. You can only enter (scope to) the existing default pool.

	Command or Action	Purpose
Step 3	UCS-A /org/ip-pool # **set descr** *description*	(Optional) Provides a description for the management IP pool. This description applies to all address blocks in the management IP pool. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/ip-pool # **set assignmentorder** {**default** | **sequential**}	This can be one of the following: • **default—**Cisco UCS Manager selects a random identity from the pool. • **sequential—**Cisco UCS Manager selects the lowest available identity from the pool.
Step 5	UCS-A /org/ip-pool # **create block** *first-ip-addr last-ip-addr gateway-ip-addr subnet-mask*	Creates a block (range) of IP addresses, and enters organization IP pool block mode. You must specify the first and last IP addresses in the address range, the gateway IP address, and subnet mask. **Note** An IP pool can contain more than one IP block. To create multiple blocks, enter multiple **create block** commands from organization IP pool mode.
Step 6	UCS-A /org/ip-pool/block # **set primary-dns** *ip-addrress* |**secondary-dns** *ip-address*	Specifies the primary DNS and secondary DNS IP addresses.
Step 7	UCS-A /org/ip-pool/ ipv6-block # **commit-buffer**	Commits the transaction to the system configuration.
Step 8	UCS-A /org/ip-pool/block # **exit**	Exits IPv4 block configuration mode.
Step 9	UCS-A /org/ip-pool # **create ipv6-block** *first-ip6-addr last-ip6-addr gateway-ip6-addr prefix*	Creates a block (range) of IPv6 addresses, and enters organization IP pool IPv6 block mode. You must specify the first and last IPv6 addresses in the address range, the gateway IPv6 address, and network prefix. **Note** An IP pool can contain more than one IPv6 block. To create multiple IPv6 blocks, enter multiple **create ipv6-block** commands from organization IP pool mode.
Step 10	UCS-A /org/ip-pool/ipv6-block # **set primary-dns** *ip6-address* **secondary-dns** *ip6-address*	Specifies the primary DNS and secondary DNS IPv6 addresses.
Step 11	UCS-A /org/ip-pool/ipv6-block # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures an IPv4 address block for the management IP pool, specifies the primary and secondary IPv4 addresses, creates an IPv6 block, specifies the primary and secondary IPv6 addresses and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope ip-pool ext-mgmt-ip
UCS-A /org/ip-pool* # set descr "This is a management ip pool example."
UCS-A /org/ip-pool* # create block 192.168.100.1 192.168.100.200 192.168.100.10 255.255.255.0
UCS-A /org/ip-pool/block* # set primary-dns 192.168.100.1 secondary-dns 192.168.100.20
UCS-A /org/ip-pool/block* commit-buffer
UCS-A /org/ip-pool/block exit
UCS-A /org/ip-pool* # create ipv6-block 2001:888::10 2001:888::100 2001:888::1 64
UCS-A /org/ip-pool/ipv6- block* set primary-dns 2001:888::11 secondary-dns 2001:888::12
UCS-A /org/ip-pool/ipv6- block* commit-buffer
UCS-A /org/ip-pool/ipv6- block #UCS-A /org/ip-pool/block* # commit-buffer
UCS-A /org/ip-pool/block #
```

The following example configures an IPv6 address block for the management IP pool and commits the transaction:

```
UCS-A# scope org /
UCS-A /org #scope ip-pool ext-mgmt-ip
UCS-A /org/ip-pool* # set descr "This is a management IPv6 pool example."
UCS-A /org/ip-pool* # create ipv6-block 2001:888::10 2001:888::100 2001:888::1 64
UCS-A /org/ip-pool/ipv6-block* # commit-buffer
UCS-A /org/ip-pool/ipv6-block* #
```

What to Do Next

Configure one or more service profiles or service profile templates to obtain the CIMC IP address from the management IP pool.

Deleting an IP Address Block from the Management IP Pool

Procedure

	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 ip-pool ext-mgmt**	Enters the management IP pool.
Step 3	UCS-A /org/ip-pool # **delete** {*ip-block*|*ipv6-block*} {*first-ip-addr*|*first-ip6-addr*} {*last-ip-addr*| *last-ip6-addr*}	Deletes the specified block (range) of IPv4 or IPv6 addresses.
Step 4	UCS-A /org/ip-pool # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes an IP address block from the management IP pool and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope ip-pool ext-mgmt
```

```
UCS-A /org/ip-pool # delete block 192.168.100.1 192.168.100.200
UCS-A /org/ip-pool* # commit-buffer
UCS-A /org/ip-pool #
```

This example shows how to delete an IPv6 address block from the management IP pool and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope ip-pool pool4
UCS-A /org/ip-pool # delete ipv6-block 2001::1 2001::10
UCS-A /org/ip-pool* # commit-buffer
UCS-A /org/ip-pool #
```

CHAPTER 28

Configuring Server-Related Policies

This chapter includes the following sections:

Configuring BIOS Settings

Server BIOS Settings

Cisco UCS provides two methods for making global modifications to the BIOS settings on servers in an Cisco UCS domain. You can create one or more BIOS policies that include a specific grouping of BIOS settings that match the needs of a server or set of servers, or you can use the default BIOS settings for a specific server platform.

Both the BIOS policy and the default BIOS settings for a server platform enable you to fine tune the BIOS settings for a server managed by Cisco UCS Manager.

Depending upon the needs of the data center, you can configure BIOS policies for some service profiles and use the BIOS defaults in other service profiles in the same Cisco UCS domain, or you can use only one of them. You can also use Cisco UCS Manager to view the actual BIOS settings on a server and determine whether they are meeting current needs.

Note Cisco UCS Manager pushes BIOS configuration changes through a BIOS policy or default BIOS settings to the Cisco Integrated Management Controller (CIMC) buffer. These changes remain in the buffer and do not take effect until the server is rebooted.

We recommend that you verify the support for BIOS settings in the server that you want to configure. Some settings, such as Mirroring Mode for RAS Memory, are not supported by all Cisco UCS servers.

Main BIOS Settings

The following table lists the main server BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Reboot on BIOS Settings Change **set reboot-on-update**	When the server is rebooted after you change one or more BIOS settings. **yes—**If you enable this setting, the server is rebooted according to the maintenance policy in the server's service profile. For example, if the maintenance policy requires user acknowledgment, the server is not rebooted and the BIOS changes are not applied until a user acknowledges the pending activity. **no—**If you do not enable this setting, the BIOS changes are not applied until the next time the server is rebooted, whether as a result of another server configuration change or a manual reboot.
Quiet Boot **set quiet-boot-config quiet-boot**	What the BIOS displays during Power On Self-Test (POST). This can be one of the following: • **disabled—**The BIOS displays all messages and Option ROM information during boot. • **enabled—**The BIOS displays the logo screen, but does not display any messages or Option ROM information during boot. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Post Error Pause **set post-error-pause-config post-error-pause**	What happens when the server encounters a critical error during POST. This can be one of the following: • **disabled—**The BIOS continues to attempt to boot the server. • **enabled—**The BIOS pauses the attempt to boot the server and opens the Error Manager when a critical error occurs during POST. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Resume Ac On Power Loss **set resume-ac-on-power-loss-config resume-action**	How the server behaves when power is restored after an unexpected power loss. This can be one of the following: • **stay-off—**The server remains off until manually powered on. • **last-state—**The server is powered on and the system attempts to restore its last state. • **reset—**The server is powered on and automatically reset. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Front Panel Lockout **set front-panel-lockout-config front-panel-lockout**	Whether the power and reset buttons on the front panel are ignored by the server. This can be one of the following: • **disabled—**The power and reset buttons on the front panel are active and can be used to affect the server. • **enabled—**The power and reset buttons are locked out. The server can only be reset or powered on or off from the CIMC GUI. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Consistent Device Naming **set consistent-device-name-control cdn-name**	Consistent Device Naming allows Ethernet interfaces to be named in a consistent manner. This makes Ethernet interface names more uniform, easy to identify, and persistent when adapter or other configuration changes are made. Whether consistent device naming is enabled or not. This can be one of the following: • **disabled—**Consistent device naming is disabled for the BIOS policy. • **enabled—**Consistent device naming is enabled for the BIOS policy. This enables Ethernet interfaces to be named consistently. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Processor BIOS Settings

The following table lists the processor BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Turbo Boost **set intel-turbo-boost-config turbo-boost**	Whether the processor uses Intel Turbo Boost Technology, which allows the processor to automatically increase its frequency if it is running below power, temperature, or voltage specifications. This can be one of the following: • **disabled—**The processor does not increase its frequency automatically. • **enabled—**The processor uses Turbo Boost Technology if required. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Enhanced Intel Speedstep **set enhanced-intel-speedstep-config speed-step**	Whether the processor uses Enhanced Intel SpeedStep Technology, which allows the system to dynamically adjust processor voltage and core frequency. This technology can result in decreased average power consumption and decreased average heat production. This can be one of the following: • **disabled—**The processor never dynamically adjusts its voltage or frequency. • **enabled—**The processor utilizes Enhanced Intel SpeedStep Technology and enables all supported processor sleep states to further conserve power. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. We recommend that you contact your operating system vendor to make sure your operating system supports this feature.
Hyper Threading **set hyper-threading-config hyper-threading**	Whether the processor uses Intel Hyper-Threading Technology, which allows multithreaded software applications to execute threads in parallel within each processor. This can be one of the following: • **disabled—**The processor does not permit hyperthreading. • **enabled—**The processor allows for the parallel execution of multiple threads. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. We recommend that you contact your operating system vendor to make sure the operating system supports this feature.

Name	Description
Core Multi Processing **set core-multi-processing-config multi-processing**	Sets the state of logical processor cores per CPU in a package. If you disable this setting, Intel Hyper Threading technology is also disabled. This can be one of the following: • **all—**Enables multiprocessing on all logical processor cores. • **1** through ***n*—**Specifies the number of logical processor cores per CPU that can run on the server. To disable multiprocessing and have only one logical processor core per CPU running on the server, choose 1. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. We recommend that you contact your operating system vendor to make sure your operating system supports this feature.
Execute Disabled Bit **set execute-disable bit**	Classifies memory areas on the server to specify where the application code can execute. As a result of this classification, the processor disables code execution if a malicious worm attempts to insert code in the buffer. This setting helps to prevent damage, worm propagation, and certain classes of malicious buffer overflow attacks. This can be one of the following: • **disabled—**The processor does not classify memory areas. • **enabled—**The processor classifies memory areas. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. We recommend that you contact your operating system vendor to make sure your operating system supports this feature.
Virtualization Technology (VT) **set intel-vt-config vt**	Whether the processor uses Intel Virtualization Technology, which allows a platform to run multiple operating systems and applications in independent partitions. This can be one of the following: • **disabled—**The processor does not permit virtualization. • **enabled—**The processor allows multiple operating systems in independent partitions. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** If you change this option, you must power cycle the server before the setting takes effect.

Name	Description
Hardware Pre-fetcher **set processor-prefetch-config hardware-prefetch**	Whether the processor allows the Intel hardware prefetcher to fetch streams of data and instruction from memory into the unified second-level cache when necessary. This can be one of the following: • **disabled—**The hardware prefetcher is not used. • **enabled—**The processor uses the hardware prefetcher when cache issues are detected. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** **CPUPerformance** must be set to **Custom** in order to specify this value. For any value other than **Custom**, this option is overridden by the setting in the selected CPU performance profile.
Adjacent Cache Line Pre-fetcher **set processor-prefetch-config adjacent-cache-line-prefetch**	Whether the processor fetches cache lines in even/odd pairs instead of fetching just the required line. This can be one of the following: • **disabled—**The processor only fetches the required line. • **enabled—**The processor fetches both the required line and its paired line. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** **CPUPerformance** must be set to **Custom** in order to specify this value. For any value other than **Custom**, this option is overridden by the setting in the selected CPU performance profile.
DCU Streamer Pre-fetch **set processor-prefetch-config dcu-streamer-prefetch**	Whether the processor uses the DCU IP Prefetch mechanism to analyze historical cache access patterns and preload the most relevant lines in the L1 cache. This can be one of the following: • **disabled—**The processor does not try to anticipate cache read requirements and only fetches explicitly requested lines. • **enabled—**The DCU prefetcher analyzes the cache read pattern and prefetches the next line in the cache if it determines that it may be needed. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
DCU IP Pre-fetcher **set processor-prefetch-config dcu-ip-prefetch**	Whether the processor uses the DCU IP Prefetch mechanism to analyze historical cache access patterns and preload the most relevant lines in the L1 cache. This can be one of the following: • **disabled—**The processor does not preload any cache data. • **enabled—**The DCU IP prefetcher preloads the L1 cache with the data it determines to be the most relevant. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Direct Cache Access **set direct-cache-access-config access**	Allows processors to increase I/O performance by placing data from I/O devices directly into the processor cache. This setting helps to reduce cache misses. This can be one of the following: • **disabled—**Data from I/O devices is not placed directly into the processor cache. • **enabled—**Data from I/O devices is placed directly into the processor cache. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Processor C State **set processor-c-state-config c-state**	Whether the system can enter a power savings mode during idle periods. This can be one of the following: • **disabled—**The system remains in a high-performance state even when idle. • **enabled—**The system can reduce power to system components such as the DIMMs and CPUs. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. We recommend that you contact your operating system vendor to make sure your operating system supports this feature.

Name	Description
Processor C1E **set processor-c1e-config c1e**	Allows the processor to transition to its minimum frequency upon entering C1. This setting does not take effect until after you have rebooted the server. This can be one of the following: • **disabled—**The CPU continues to run at its maximum frequency in the C1 state. • **enabled—**The CPU transitions to its minimum frequency. This option saves the maximum amount of power in the C1 state. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Processor C3 Report **set processor-c3-report-config processor-c3-report**	Whether the processor sends the C3 report to the operating system. This can be one of the following: • **disabled—**The processor does not send the C3 report. • **acpi-c2—**The processor sends the C3 report using the advanced configuration and power interface (ACPI) C2 format. • **acpi-c3—**The processor sends the C3 report using the ACPI C3 format. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. On the Cisco UCS B440 Server, the BIOS Setup menu uses enabled and disabled for these options. If you specify acpi-c2 or acpi-c2, the server sets the BIOS value for that option to enabled.
Processor C6 Report **set processor-c6-report-config processor-c6-report**	Whether the processor sends the C6 report to the operating system. This can be one of the following: • **disabled—**The processor does not send the C6 report. • **enabled—**The processor sends the C6 report. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Processor C7 Report **set processor-c7-report-config processor-c7-report**	Whether the processor sends the C7 report to the operating system. This can be one of the following: • **disabled—**The processor does not send the C7 report. • **enabled—**The processor sends the C7 report. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Processor CMCI field	Enables CMCI generation.
CPU Performance **set cpu-performance-config cpu-performance**	Sets the CPU performance profile for the server. This can be one of the following: • **enterprise—**For M3 servers, all prefetchers and data reuse are enabled. For M1 and M2 servers, data reuse and the DCU IP prefetcher are enabled, and all other prefetchers are disabled. • **high-throughput—**Data reuse and the DCU IP prefetcher are enabled, and all other prefetchers are disabled. • **hpc—**All prefetchers are enabled and data reuse is disabled. This setting is also known as high-performance computing.
Max Variable MTRR Setting **set max-variable-mtrr-setting-config processor-mtrr**	Allows you to select the number of mean time to repair (MTRR) variables. This can be one of the following: • **auto-max—**BIOS uses the default value for the processor. • **8—**BIOS uses the number specified for the variable MTRR. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Local X2 APIC **set local-x2-apic-config localx2-apic**	Allows you to set the type of Application Policy Infrastructure Controller (APIC) architecture. This can be one of the following: • **xapic—**Uses the standard xAPIC architecture. • **x2apic—**Uses the enhanced x2APIC architecture to support 32 bit addressability of processors. • **auto—**Automatically uses the xAPIC architecture that is detected. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Power Technology **set processor-energy-config cpu-power-management**	Enables you to configure the CPU power management settings for the following options: • Enhanced Intel Speedstep Technology • Intel Turbo Boost Technology • Processor Power State C6 Power Technology can be one of the following: • **disabled—**The server does not perform any CPU power management and any settings for the BIOS parameters mentioned above are ignored. • **Energy_Efficient—**The server determines the best settings for the BIOS parameters mentioned above and ignores the individual settings for these parameters. • **performance—**The server automatically optimizes the performance for the BIOS parameters mentioned above. • **custom—**The server uses the individual settings for the BIOS parameters mentioned above. You must select this option if you want to change any of these BIOS parameters. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Energy Performance **set processor-energy-config energy-performance**	Allows you to determine whether system performance or energy efficiency is more important on this server. This can be one of the following: • **performance** • **balanced-performance** • **balanced-energy** • **energy-efficient** • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** **CPUPowerManagement** must be set to **Custom** or the server ignores the setting for this parameter.
Frequency Floor Override **set frequency-floor-override-config cpu-frequency**	Whether the CPU is allowed to drop below the maximum non-turbo frequency when idle. This can be one of the following: • **disabled—** The CPU can drop below the maximum non-turbo frequency when idle. This option decreases power consumption but may reduce system performance. • **enabled—** The CPU cannot drop below the maximum non-turbo frequency when idle. This option improves system performance but may increase power consumption. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
P-STATE Coordination **set p-state-coordination-config p-state**	Allows you to define how BIOS communicates the P-state support model to the operating system. There are 3 models as defined by the Advanced Configuration and Power Interface (ACPI) specification. • **hw-all—**The processor hardware is responsible for coordinating the P-state among logical processors with dependencies (all logical processors in a package). • **sw-all—**The OS Power Manager (OSPM) is responsible for coordinating the P-state among logical processors with dependencies (all logical processors in a physical package), and must initiate the transition on all of the logical processors. • **sw-all—**The OS Power Manager (OSPM) is responsible for coordinating the P-state among logical processors with dependencies (all logical processors in a package), and may initiate the transition on any of the logical processors in the domain. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** **CPUPowerManagement** must be set to **Custom** or the server ignores the setting for this parameter.
DRAM Clock Throttling **set dram-clock-throttling-config dram-clock-throttling**	Allows you to tune the system settings between the memory bandwidth and power consumption. This can be one of the following: • **balanced—** DRAM clock throttling is reduced, providing a balance between performance and power. • **performance—**DRAM clock throttling is disabled, providing increased memory bandwidth at the cost of additional power. • **Energy_Efficient—**DRAM clock throttling is increased to improve energy efficiency. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Channel Interleaving **set interleave-config channel-interleave**	Whether the CPU divides memory blocks and spreads contiguous portions of data across interleaved channels to enable simultaneous read operations. This can be one of the following: • **Auto—**The CPU determines what interleaving is done. • **1-way—**Some channel interleaving is used. • **2-way** • **3-way** • **4-way—**The maximum amount of channel interleaving is used. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Rank Interleaving **set interleave-config rank-interleave**	Whether the CPU interleaves physical ranks of memory so that one rank can be accessed while another is being refreshed. This can be one of the following: • **Auto—**The CPU determines what interleaving is done. • **1-way—**Some rank interleaving is used. • **2-way** • **4-way** • **8-way—**The maximum amount of rank interleaving is used. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Memory Interleaving **set interleave-config memory-interleave**	Whether the CPU interleaves the physical memory so that the memory can be accessed while another is being refreshed. This can be one of the following: • **Auto—**The CPU determines what interleaving is done. • **1-way—**Some memory interleaving is used. • **2-way** • **4-way** • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Demand Scrub **set scrub-policies-config demand-scrub**	Whether the system corrects single bit memory errors encountered when the CPU or I/O makes a demand read. This can be one of the following: • **disabled—** Single bit memory errors are not corrected. • **enabled—** Single bit memory errors are corrected in memory and the corrected data is set in response to the demand read. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Patrol Scrub **set scrub-policies-config patrol-scrub**	Whether the system actively searches for, and corrects, single bit memory errors even in unused portions of the memory on the server. This can be one of the following: • **disabled—**The system checks for memory ECC errors only when the CPU reads or writes a memory address. • **enabled—**The system periodically reads and writes memory searching for ECC errors. If any errors are found, the system attempts to fix them. This option may correct single bit errors before they become multi-bit errors, but it may adversely affect performance when the patrol scrub is running. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Altitude **set altitude altitude-config**	The approximate number of meters above sea level at which the physical server is installed. This can be one of the following: • **auto—**The CPU determines the physical elevation. • **300-m—**The server is approximately 300 meters above sea level. • **900-m—**The server is approximately 900 meters above sea level. • **1500-m—**The server is approximately 1500 meters above sea level. • **3000-m—**The server is approximately 3000 meters above sea level. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Package C State Limit **set package-c-state-limit-config package-c-state-limit**	The amount of power available to the server components when they are idle. This can be one of the following: • **auto** • **no-limit—**The server may enter any available C state. • **c0—**The server provides all server components with full power at all times. This option maintains the highest level of performance and requires the greatest amount of power. • **c1—**When the CPU is idle, the system slightly reduces the power consumption. This option requires less power than C0 and allows the server to return quickly to high performance mode. • **c3—**When the CPU is idle, the system reduces the power consumption further than with the C1 option. This requires less power than C1 or C0, but it takes the server slightly longer to return to high performance mode. • **c6—**When the CPU is idle, the system reduces the power consumption further than with the C3 option. This option saves more power than C0, C1, or C3, but there may be performance issues until the server returns to full power. • **c2—**When the CPU is idle, the system reduces the power consumption further than with the C1 option. This requires less power than C1 or C0, but it takes the server slightly longer to return to high performance mode. • **c7—**When the CPU is idle, the server makes a minimal amount of power available to the components. This option saves the maximum amount of power but it also requires the longest time for the server to return to high performance mode. • **c7s—**When the CPU is idle, the server makes a minimal amount of power available to the components. This option saves more power than C7, but it also requires the longest time for the server to return to high performance mode. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
CPU Hardware Power Management **set cpu-hardware-power-management-config cpu-hardware-power-management**	Enables processor Hardware Power Management (HWPM). This can be one of the following: • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. • **disabled—**HWPM is disabled. • **hwpm-native-mode—**HWPM native mode is enabled. • **hwpm-oob-mode—**HWPM Out-Of-Box mode is enabled.
Energy Performance Tuning	Determines if the BIOS or Operating System can turn on the energy performance bias tuning. The options are BIOS and OS.
Workload Configuration	This feature allows for workload optimization. The options are Balanced and I/O Sensitive. Cisco recommends using Balanced.

Intel Directed I/O BIOS Settings

The following table lists the Intel Directed I/O BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
VT for Directed IO **set intel-vt-directed-io-config vtd**	Whether the processor uses Intel Virtualization Technology for Directed I/O (VT-d). This can be one of the following: • **disabled—**The processor does not use virtualization technology. • **enabled—**The processor uses virtualization technology. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** This option must be enabled if you want to change any of the other Intel Directed I/O BIOS settings.

Name	Description
Interrupt Remap **set intel-vt-directed-io-config interrupt-remapping**	Whether the processor supports Intel VT-d Interrupt Remapping. This can be one of the following: • **disabled—**The processor does not support remapping. • **enabled—**The processor uses VT-d Interrupt Remapping as required. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Coherency Support **set intel-vt-directed-io-config coherency-support**	Whether the processor supports Intel VT-d Coherency. This can be one of the following: • **disabled—**The processor does not support coherency. • **enabled—**The processor uses VT-d Coherency as required. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
ATS Support **set intel-vt-directed-io-config ats-support**	Whether the processor supports Intel VT-d Address Translation Services (ATS). This can be one of the following: • **disabled—**The processor does not support ATS. • **enabled—**The processor uses VT-d ATS as required. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Pass Through DMA Support **set intel-vt-directed-io-config passthrough-dma**	Whether the processor supports Intel VT-d Pass-through DMA. This can be one of the following: • **disabled—**The processor does not support pass-through DMA. • **enabled—**The processor uses VT-d Pass-through DMA as required. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

RAS Memory BIOS Settings

The following table lists the RAS memory BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Memory RAS Config **set memory-ras-config ras-config**	How the memory reliability, availability, and serviceability (RAS) is configured for the server. This can be one of the following: • **maximum-performance—**System performance is optimized. • **mirroring—**System reliability is optimized by using half the system memory as backup. • **lockstep—**If the DIMM pairs in the server have an identical type, size, and organization and are populated across the SMI channels, you can enable lockstep mode to minimize memory access latency and provide better performance. Lockstep is enabled by default for B440 servers. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
NUMA **set numa-config numa-optimization**	Whether the BIOS supports NUMA. This can be one of the following: • **disabled—**The BIOS does not support NUMA. • **enabled—**The BIOS includes the ACPI tables that are required for NUMA-aware operating systems. If you enable this option, the system must disable Inter-Socket Memory interleaving on some platforms. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Mirroring Mode **set memory-mirroring-mode mirroring-mode**	Memory mirroring enhances system reliability by keeping two identical data images in memory. This option is only available if you choose the **mirroring** option for **Memory RAS Config**. It can be one of the following: • **inter-socket—**Memory is mirrored between two Integrated Memory Controllers (IMCs) across CPU sockets. • **intra-socket—**One IMC is mirrored with another IMC in the same socket. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Sparing Mode **set memory-sparing-mode sparing-mode**	Sparing optimizes reliability by holding memory in reserve so that it can be used in case other DIMMs fail. This option provides some memory redundancy, but does not provide as much redundancy as mirroring. The available sparing modes depend on the current memory population. This option is only available if you choose **sparing** option for **Memory RAS Config**. It can be one of the following: • **dimm-sparing—**One DIMM is held in reserve. If a DIMM fails, the contents of a failing DIMM are transferred to the spare DIMM. • **rank-sparing—**A spare rank of DIMMs is held in reserve. If a rank of DIMMs fails, the contents of the failing rank are transferred to the spare rank. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
LV DDR Mode **set lv-dimm-support-config lv-ddr-mode**	Whether the system prioritizes low voltage or high frequency memory operations. This can be one of the following: • **power-saving-mode—**The system prioritizes low voltage memory operations over high frequency memory operations. This mode may lower memory frequency in order to keep the voltage low. • **performance-mode—**The system prioritizes high frequency operations over low voltage operations. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
DRAM Refresh Rate **set dram-refresh-rate-config dram-refresh**	The refresh interval rate for internal memory. This can be one of the following: • **1x** • **2x** • **3x** • **4x** • **auto** • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
DDR3 Voltage Selection **set ddr3-voltage-config ddr3-voltage**	The voltage to be used by the dual-voltage RAM. This can be one of the following: • **ddr3-1500mv** • **ddr3-1350mv** • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Serial Port BIOS Settings

The following table lists the serial port BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Serial Port A **set serial-port-a-config serial-port-a**	Whether serial port A is enabled or disabled. This can be one of the following: • **disabled—**The serial port is disabled. • **enabled—**The serial port is enabled. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

USB BIOS Settings

The following table lists the USB BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Make Device Non Bootable **set usb-boot-config make-device-non-bootable**	Whether the server can boot from a USB device. This can be one of the following: • **disabled—**The server can boot from a USB device. • **enabled—**The server cannot boot from a USB device. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Legacy USB Support **set usb-boot-config legacy-support**	Whether the system supports legacy USB devices. This can be one of the following: • **disabled—**USB devices are only available to EFI applications. • **enabled—**Legacy USB support is always available. • **auto—**Disables legacy USB support if no USB devices are connected. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
USB System Idle Power Optimizing Setting **set usb-system-idle-power-optimizing-setting-config usb-idle-power-optimizing**	Whether the USB System Idle Power Optimizing setting is used to reduce USB EHCI idle power consumption. Depending upon the value you choose, this setting can have an impact on performance. This can be one of the following: • **high-performance—**The USB System Idle Power Optimizing setting is disabled, because optimal performance is preferred over power savings. Selecting this option can significantly improve performance. We recommend you select this option unless your site has server power restrictions. • **lower-idle-power—**The USB System Idle Power Optimizing setting is enabled, because power savings are preferred over optimal performance. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
USB Front Panel Access Lock **set usb-front-panel-access-lock-config usb-front-panel-lock**	USB front panel lock is configured to enable or disable the front panel access to USB ports. This can be one of the following: • **disabled** • **enabled** • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Port 60/64 Emulation **set usb-port-config usb-emulation**	Whether the system supports 60h/64h emulation for complete USB keyboard legacy support. This can be one of the following: • **disabled—**60h/64 emulation is not supported. • **enabled—**60h/64 emulation is supported. You should select this option if you are using a non-USB aware operating system on the server. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
USB Port:Front **set usb-port-config usb-front**	Whether the front panel USB devices are enabled or disabled. This can be one of the following: • **disabled—**Disables the front panel USB ports. Devices connected to these ports are not detected by the BIOS and operating system. • **enabled—**Enables the front panel USB ports. Devices connected to these ports are detected by the BIOS and operating system. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
USB Port:Internal **set usb-port-config usb-internal**	Whether the internal USB devices are enabled or disabled. This can be one of the following: • **disabled—**Disables the internal USB ports. Devices connected to these ports are not detected by the BIOS and operating system. • **enabled—**Enables the internal USB ports. Devices connected to these ports are detected by the BIOS and operating system. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
USB Port:KVM **set usb-port-config usb-kvm**	Whether the KVM ports are enabled or disabled. This can be one of the following: • **disabled—**Disables the KVM keyboard and/or mouse devices. Keyboard and/or mouse will not work in the KVM window. • **enabled—**Enables the KVM keyboard and/or mouse devices. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
USB Port:Rear **set usb-port-config usb-rear**	Whether the rear panel USB devices are enabled or disabled. This can be one of the following: • **disabled—**Disables the rear panel USB ports. Devices connected to these ports are not detected by the BIOS and operating system. • **enabled—**Enables the rear panel USB ports. Devices connected to these ports are detected by the BIOS and operating system. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
USB Port:SD Card **set usb-port-config usb-sdcard**	Whether the SD card drives are enabled or disabled. This can be one of the following: • **disabled—**Disables the SD card drives. The SD card drives are not detected by the BIOS and operating system. • **enabled—**Enables the SD card drives. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
USB Port:VMedia **set usb-port-config usb-vmedia**	Whether the virtual media devices are enabled or disabled. This can be one of the following: • **disabled—**Disables the vMedia devices. • **enabled—**Enables the vMedia devices. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
All USB Devices **set all-usb-devices-config all-usb**	Whether all physical and virtual USB devices are enabled or disabled. This can be one of the following: • **disabled—**All USB devices are disabled. • **enabled—**All USB devices are enabled. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
xHCI Mode Support **set usb-configuration-select-config xhci-enable-disable**	Whether xHCI mode support is enabled or disabled. This can be one of the following: • **disabled—**xHCI mode support is disabled. • **enabled—**xHCI mode support is enabled. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

PCI Configuration BIOS Settings

The following table lists the PCI configuration BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Max Memory Below 4G **set max-memory-below-4gb-config max-memory**	Whether the BIOS maximizes memory usage below 4GB for an operating system without PAE support, depending on the system configuration. This can be one of the following: • **disabled—**Does not maximize memory usage. Choose this option for all operating systems with PAE support. • **enabled—**Maximizes memory usage below 4GB for an operating system without PAE support. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Memory Mapped IO Above 4Gb Config **set memory-mapped-io-above-4gb-config memory-mapped-io**	Whether to enable or disable memory mapped I/O of 64-bit PCI devices to 4GB or greater address space. Legacy option ROMs are not able to access addresses above 4GB. PCI devices that are 64-bit compliant but use a legacy option ROM may not function correctly with this setting enabled. This can be one of the following: • **disabled—**Does not map I/O of 64-bit PCI devices to 4GB or greater address space. • **enabled—**Maps I/O of 64-bit PCI devices to 4GB or greater address space. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
VGA Priority **set vga-priority-config vga-priority**	Allows you to set the priority for VGA graphics devices if multiple VGA devices are found in the system. This can be one of the following: • **onboard—**Priority is given to the onboard VGA device. BIOS post screen and OS boot are driven through the onboard VGA port. • **offboard—**Priority is given to the PCIE Graphics adapter. BIOS post screen and OS boot are driven through the external graphics adapter port. • **onboard-vga-disabled—**Priority is given to the PCIE Graphics adapter, and the onboard VGA device is disabled. **Note** The vKVM does not function when the onboard VGA is disabled. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** Only onboard VGA devices are supported with Cisco UCS B-Series servers.
ASPM Support **set aspm-support-config aspm-support**	Allows you to set the level of ASPM (Active Power State Management) support in the BIOS. This can be one of the following: • **disabled—**ASPM support is disabled in the BIOS. • **auto—**The CPU determines the power state. • **forcel0—**Force all links to L0 standby (L0s) state. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

QPI BIOS Settings

The following table lists the QPI BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
QPI Link Frequency **set qpi-link-frequency-select-config qpi-link-freqency-mt-per-sec**	The Intel QuickPath Interconnect (QPI) link frequency, in megatransfers per second (MT/s). This can be one of the following: • **6400** • **7200** • **8000** • **9600** • **Auto—**The CPU determines the QPI link frequency. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
QPI Snoop Mode **set qpi-snoop-mode vpqpisnoopmode**	This can be one of the following: • **home-snoop—**The snoop is always spawned by the home agent (centralized ring stop) for the memory controller. This mode has a higher local latency than early snoop, but it provides extra resources for a larger number of outstanding transactions. • **cluster-on-die—**This mode is available only for processors that have 10 or more cores. It is the best mode for highly NUMA optimized workloads. • **home-directory-snoop-with-osb** • **early-snoop—**The distributed cache ring stops can send a snoop probe or a request to another caching agent directly. This mode has lower latency and it is best for workloads that have shared data sets across threads and can benefit from a cache-to-cache transfer, or for workloads that are not NUMA optimized. • **auto —**The CPU determines the QPI Snoop mode. • **platform-default —**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

LOM and PCIe Slots BIOS Settings

The following table lists the USB BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
PCIe Slot:SAS OptionROM **set slot-option-rom-enable-config pcie-sas**	Whether Option ROM is available on the SAS port. This can be one of the following: • **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
PCIe Slot:*n* Link Speed **set slot-link-speed-config pcie-slot*n*-link-speed**	This option allows you to restrict the maximum speed of an adapter card installed in PCIe slot ***n***. This can be one of the following: • **gen1—**2.5GT/s (gigatransfers per second) is the maximum speed allowed. • **gen2—**5GT/s is the maximum speed allowed. • **gen3—**8GT/s is the maximum speed allowed. • **auto—**The maximum speed is set automatically. • **disabled—**The maximum speed is not restricted. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
PCIe Slot:*n* OptionROM **set slot-option-rom-enable-config slot*n*-option-rom-enable**	Whether Option ROM is available on the port. This can be one of the following: • **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
PCIe Slot:HBA OptionROM **set slot-option-rom-enable-config pcie-hba**	Whether Option ROM is available on the HBA port. This can be one of the following: • **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
PCIe Slot:MLOM OptionROM **set slot-option-rom-enable-config pcie-mlom**	Whether Option ROM is available on the MLOM port. This can be one of the following: • **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
PCIe Slot:N1 OptionROM **set slot-option-rom-enable-config pcie-n1**	Whether Option ROM is available on the port. This can be one of the following: • **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
PCIe Slot:N2 OptionROM **set slot-option-rom-enable-config pcie-n2**	Whether Option ROM is available on the port. This can be one of the following: • **Disabled—**The expansion slot is not available. • **Enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
PCIe 10G LOM 2 Link **set lom-ports-config pcie-lom2-link**	Whether Option ROM is available on the 10G LOM port. This can be one of the following: • **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
PCI ROM CLP **set pci-rom-clp-support pci-rom-clp-config**	• **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
SIOC1 Option ROM **set sioc1-optionrom-config sioc1-optionrom**	• **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
SIOC2 Option ROM **set sioc2-optionrom-config sioc2-optionrom**	• **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
SB MEZZ1 Option ROM **set sbmezz1-optionrom-config sbmezz1-optionrom**	• **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
IOE Slot1 Option ROM **set ioeslot1-optionrom-config ioeslot1-optionrom**	• **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
IOE MEZZ 1 Option ROM **set ioemezz1-optionrom-config ioemezz1-optionrom**	• **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
IOE Slot2 Option ROM **set ioeslot2-optionrom-config ioeslot2-optionrom**	• **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
IO ENVME1 Option ROM **set ioenvme1-optionrom-config ioenvme1-optionrom**	• **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
IO ENVME2 Option ROM **set ioenvme2-optionrom-config ioenvme2-optionrom**	• **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
SBNVME1 Option ROM **set sbnvme1-optionrom-config sbnvme1-optionrom**	• **disabled—**The expansion slot is not available. • **enabled—**The expansion slot is available. • **uefi-only —**The expansion slot is available for UEFI only. • **legacy-only—**The expansion slot is available for legacy only. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Graphics Configuration BIOS Settings

The following tables list the graphics configuration BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Integrated Graphics **set integrated-graphics-config integrated-graphics**	Enables integrated graphics. This can be one of the following: • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. • **enabled—**Integrated graphic is enabled. • **disabled—**Integrated graphics is disabled.
Aperture Size **set integrated-graphics-aperture-config integrated-graphics-aperture**	Allows you to set the size of mapped memory for the integrated graphics controller. This can be one of the following: • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. • **128mb** • **256mb** • **512mb** • **1024mb** • **2048mb** • **4096mb**

Name	Description
Onboard Graphics **set onboard-graphics-config onboard-graphics**	Enables onboard graphics (KVM). This can be one of the following: • **platform-default**—The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. • **enabled**—Onboard graphics is enabled. • **disabled**—Onboard graphics is disabled.

Boot Options BIOS Settings

The following table lists the boot options BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

Name	Description
Boot Option Retry **set boot-option-retry-config retry**	Whether the BIOS retries NON-EFI based boot options without waiting for user input. This can be one of the following: • **disabled**—Waits for user input before retrying NON-EFI based boot options. • **enabled**—Continually retries NON-EFI based boot options without waiting for user input. • **platform-default**—The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Intel Entry SAS RAID **set intel-entry-sas-raid-config sas-raid**	Whether the Intel SAS Entry RAID Module is enabled. This can be one of the following: • **disabled**—The Intel SAS Entry RAID Module is disabled. • **enabled**—The Intel SAS Entry RAID Module is enabled. • **platform-default**—The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Intel Entry SAS RAID Module **set intel-entry-sas-raid-config sas-raid-module**	How the Intel SAS Entry RAID Module is configured. This can be one of the following: • **it-ir-raid—**Configures the RAID module to use Intel IT/IR RAID. • **intel-esrtii—**Configures the RAID module to use Intel Embedded Server RAID Technology II. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Onboard SCU Storage Support **set onboard-sas-storage-config onboard-sas-ctrl**	Whether the onboard software RAID controller is available to the server. This can be one of the following: • **disabled—**The software RAID controller is not available. • **enabled—**The software RAID controller is available. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Note BIOS parameter virtualization capability in Cisco UCS Manager maps a unified set of BIOS settings in a service profile to the actual BIOS supporting parameters. However, not all BIOS setting items are applicable to every server model/platform. When you create a custom BIOS policy and have the **Boot Option Retry** selected, and when there is no bootable option available, the reboot fails on the Cisco UCS B420 M3 or Cisco UCS B420 M4 servers and Cisco UCS Manager displays this message : *Reboot and Select proper Boot device or Insert Boot Media in selected Boot device and press a key*. You must manually set a boot option after the boot path is corrected, in order to enable the servers to reboot after a power outage. For more information about BIOS default server policies and the BIOS options and their default settings, see BIOS Policy, on page 444 and Server BIOS Settings, on page 403.

Server Management BIOS Settings

The following tables list the server management BIOS settings that you can configure through a BIOS policy or the default BIOS settings:

General Settings

Name	Description
Assert Nmi on Serr **set assert-nmi-on-serr-config assertion**	Whether the BIOS generates a non-maskable interrupt (NMI) and logs an error when a system error (SERR) occurs. This can be one of the following: • **disabled—**The BIOS does not generate an NMI or log an error when a SERR occurs. • **enabled—**The BIOS generates an NMI and logs an error when a SERR occurs. You must enable this setting if you want to enable **Assert Nmi on Perr**. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Assert Nmi on Perr **set assert-nmi-on-perr-config assertion**	Whether the BIOS generates a non-maskable interrupt (NMI) and logs an error when a processor bus parity error (PERR) occurs. This can be one of the following: • **disabled—**The BIOS does not generate an NMI or log an error when a PERR occurs. • **enabled—**The BIOS generates an NMI and logs an error when a PERR occurs. You must enable **Assert Nmi on Serr** to use this setting. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
OS Boot Watchdog Timer **set os-boot-watchdog-timer-config os-boot-watchdog-timer**	Whether the BIOS programs the watchdog timer with a predefined timeout value. If the operating system does not complete booting before the timer expires, the CIMC resets the system and an error is logged. This can be one of the following: • **disabled—**The watchdog timer is not used to track how long the server takes to boot. • **enabled—**The watchdog timer tracks how long the server takes to boot. If the server does not boot within the predefined length of time, the CIMC resets the system and logs an error. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. This feature requires either operating system support or Intel Management software.

Name	Description
OS Boot Watchdog Timer Timeout Policy **set os-boot-watchdog-timer-policy-config os-boot-watchdog-timer-policy**	What action the system takes if the watchdog timer expires. This can be one of the following: • **power-off—**The server is powered off if the watchdog timer expires during OS boot. • **reset—**The server is reset if the watchdog timer expires during OS boot. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. This option is only available if you enable the OS Boot Watchdog Timer.
OS Boot Watchdog Timer Timeout **set os-boot-watchdog-timer-timeout-config os-boot-watchdog-timer-timeout**	What timeout value the BIOS uses to configure the watchdog timer. This can be one of the following: • **5-minutes—**The watchdog timer expires 5 minutes after the OS begins to boot. • **10-minutes—**The watchdog timer expires 10 minutes after the OS begins to boot. • **15-minutes—**The watchdog timer expires 15 minutes after the OS begins to boot. • **20-minutes—**The watchdog timer expires 20 minutes after the OS begins to boot. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. This option is only available if you enable the OS Boot Watchdog Timer.
FRB-2 Timer **set frb-2-timer-config frb-2-timer**	Whether the FRB-2 timer is used to recover the system if it hangs during POST. This can be one of the following: • **Disabled—**The FRB-2 timer is not used. • **Enabled—**The FRB-2 timer is started during POST and used to recover the system if necessary. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Console Redirection Settings

Name	Description
Console Redirection **set console-redir-config console-redir**	Allows a serial port to be used for console redirection during POST and BIOS booting. After the BIOS has booted and the operating system is responsible for the server, console redirection is irrelevant and has no effect. This can be one of the following: • **disabled—**No console redirection occurs during POST. • **serial-port-a—**Enables serial port A for console redirection during POST. This option is valid for blade servers and rack-mount servers. • **serial-port-b—**Enables serial port B for console redirection and allows it to perform server management tasks. This option is only valid for rack-mount servers. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** If you enable this option, you also disable the display of the Quiet Boot logo screen during POST.
Flow Control **set console-redir-config flow-control**	Whether a handshake protocol is used for flow control. Request to Send / Clear to Send (RTS/CTS) helps to reduce frame collisions that can be introduced by a hidden terminal problem. This can be one of the following: • **none—**No flow control is used. • **rts-cts—**RTS/CTS is used for flow control. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** This setting must match the setting on the remote terminal application.

Name	Description
BAUD Rate **set console-redir-config baud-rate**	What BAUD rate is used for the serial port transmission speed. If you disable Console Redirection, this option is not available. This can be one of the following: • **9600—**A 9600 BAUD rate is used. • **19200—**A 19200 BAUD rate is used. • **38400—**A 38400 BAUD rate is used. • **57600—**A 57600 BAUD rate is used. • **115200—**A 115200 BAUD rate is used. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** This setting must match the setting on the remote terminal application.
Terminal Type **set console-redir-config terminal-type**	What type of character formatting is used for console redirection. This can be one of the following: • **pc-ansi—**The PC-ANSI terminal font is used. • **vt100—**A supported vt100 video terminal and its character set are used. • **vt100-plus—**A supported vt100-plus video terminal and its character set are used. • **vt-utf8—**A video terminal with the UTF-8 character set is used. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor. **Note** This setting must match the setting on the remote terminal application.
Legacy OS Redirect **set console-redir-config legacy-os-redir**	Whether redirection from a legacy operating system, such as DOS, is enabled on the serial port. This can be one of the following: • **disabled—**The serial port enabled for console redirection is hidden from the legacy operating system. • **enabled—** The serial port enabled for console redirection is visible to the legacy operating system. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.

Name	Description
Putty KeyPad **set console-redir-config putty-function-keypad**	Allows you to change the action of the PuTTY function keys and the top row of the numeric keypad. This can be one of the following: • **vt100—**The function keys generate ESC OP through ESC O[. • **linux—**Mimics the Linux virtual console. Function keys F6 to F12 behave like the default mode, but F1 to F5 generate ESC [[A through ESC [[E. • **xtermr6—**Function keys F5 to F12 behave like the default mode. Function keys F1 to F4 generate **ESC OP** through ESC OS, which are the sequences produced by the top row of the keypad on Digital terminals. • **sco—**The function keys F1 to F12 generate ESC [M through ESC [X. The function and shift keys generate ESC [Y through ESC [j. The control and function keys generate ESC [k through ESC [v. The shift, control and function keys generate ESC [w through ESC [{. • **escn—**The default mode. The function keys match the general behavior of Digital terminals. The function keys generate sequences such as ESC [11~ and ESC [12~. • **vt400—**The function keys behave like the default mode. The top row of the numeric keypad generates ESC OP through ESC OS. • **platform-default—**The BIOS uses the value for this attribute contained in the BIOS defaults for the server type and vendor.
Out of Band Management	Used for Windows Special Administration Control (SAC).
Redirection After BIOS POST	

BIOS Policy

The BIOS policy is a policy that automates the configuration of BIOS settings for a server or group of servers. You can create global BIOS policies available to all servers in the root organization, or you can create BIOS policies in sub-organizations that are only available to that hierarchy.

To use a BIOS policy, do the following:

1. Create the BIOS policy in Cisco UCS Manager.
2. Assign the BIOS policy to one or more service profiles.
3. Associate the service profile with a server.

During service profile association, Cisco UCS Manager modifies the BIOS settings on the server to match the configuration in the BIOS policy. If you do not create and assign a BIOS policy to a service profile, the server uses the default BIOS settings for that server platform.

Default BIOS Settings

Cisco UCS Manager includes a set of default BIOS settings for each type of server supported by Cisco UCS. The default BIOS settings are available only in the root organization and are global. Only one set of default BIOS settings can exist for each server platform supported by Cisco UCS. You can modify the default BIOS settings, but you cannot create an additional set of default BIOS settings.

Each set of default BIOS settings are designed for a particular type of supported server and are applied to all servers of that specific type which do not have a BIOS policy included in their service profiles.

Unless a Cisco UCS implementation has specific needs that are not met by the server-specific settings, we recommend that you use the default BIOS settings that are designed for each type of server in the Cisco UCS domain.

Cisco UCS Manager applies these server platform-specific BIOS settings as follows:

- The service profile associated with a server does not include a BIOS policy.
- The BIOS policy is configured with the platform-default option for a specific setting.

You can modify the default BIOS settings provided by Cisco UCS Manager. However, any changes to the default BIOS settings apply to all servers of that particular type or platform. If you want to modify the BIOS settings for only certain servers, we recommend that you use a BIOS policy.

Creating a BIOS Policy

Note Cisco UCS Manager pushes BIOS configuration changes through a BIOS policy or default BIOS settings to the Cisco Integrated Management Controller (CIMC) buffer. These changes remain in the buffer and do not take effect until the server is rebooted.

We recommend that you verify the support for BIOS settings in the server that you want to configure. Some settings, such as Mirroring Mode for RAS Memory, are not supported by all Cisco UCS servers.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters org mode for the specified organization. To enter the default org mode, type / 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	Configure the BIOS settings.	For the CLI commands, descriptions and information about the options for each BIOS setting, see the following topics: • **Main** page: Main BIOS Settings, on page 404

	Command or Action	Purpose
		• **Processor** page: Processor BIOS Settings, on page 406 • **Intel Directed IO** page: Intel Directed I/O BIOS Settings, on page 419 • **RAS Memory** page: RAS Memory BIOS Settings, on page 421 • **Serial Port** page: Serial Port BIOS Settings, on page 423 • **USB** page: USB BIOS Settings, on page 424 • **PCI Configuration** page: PCI Configuration BIOS Settings, on page 427 • **Boot Options** page: Boot Options BIOS Settings, on page 438 • **Server Management** page: Server Management BIOS Settings, on page 439
Step 4	UCS-A /org/bios-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a BIOS policy under the root organization and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create bios-policy biosPolicy3
UCS-A /org/bios-policy* # set numa-config numa-optimization enabled
UCS-A /org/bios-policy* # commit-buffer
UCS-A /org/bios-policy #
```

Modifying BIOS Defaults

We recommend that you verify the support for BIOS settings in the server that you want to configure. Some settings, such as Mirroring Mode for RAS Memory, are not supported by all Cisco UCS servers.

Unless a Cisco UCS implementation has specific needs that are not met by the server-specific settings, we recommend that you use the default BIOS settings that are designed for each type of server in the Cisco UCS domain.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope server-defaults**	Enters server defaults mode.
Step 3	UCS-A /system/server-defaults # **show platform**	(Optional) Displays platform descriptions for all servers.

	Command or Action	Purpose
Step 4	UCS-A /system/server-defaults # **scope platform** *platform-description*	Enters server defaults mode for the server specified. For the *platform-description* argument, enter the server description displayed by the **show platform** command using the following format: "*vendor*" *model revision*. **Tip** You must enter the vendor exactly as shown in the **show platform** command, including all punctuation marks.
Step 5	UCS-A /system/server-defaults/platform # **scope bios-settings**	Enters server defaults BIOS settings mode for the server.
Step 6	Reconfigure the BIOS settings.	For the CLI commands, descriptions and information about the options for each BIOS setting, see the following topics: • **Main** page: Main BIOS Settings, on page 404 • **Processor** page: Processor BIOS Settings, on page 406 • **Intel Directed IO** page: Intel Directed I/O BIOS Settings, on page 419 • **RAS Memory** page: RAS Memory BIOS Settings, on page 421 • **Serial Port** page: Serial Port BIOS Settings, on page 423 • **USB** page: USB BIOS Settings, on page 424 • **PCI Configuration** page: PCI Configuration BIOS Settings, on page 427 • **Boot Options** page: Boot Options BIOS Settings, on page 438 • **Server Management** page: Server Management BIOS Settings, on page 439
Step 7	UCS-A /system/server-defaults/platform/bios-settings # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to change the NUMA default BIOS setting for a platform and commit the transaction:

```
UCS-A# scope system
UCS-A /system # scope server-defaults
UCS-A /system/server-defaults # show platform

Platform:
    Product Name Vendor     Model      Revision
```

```
    ------------ ---------- ---------- --------
    Cisco B200-M1
                Cisco Systems, Inc.
                           N20-B6620-1
                                      0

UCS-A /system/server-defaults # scope platform "Cisco Systems, Inc." N20-B6620-1 0
UCS-A /system/server-defaults/platform # scope bios-settings
UCS-A /system/server-defaults/platform/bios-settings # set numa-config numa-optimization
disabled
UCS-A /system/server-defaults/platform/bios-settings* # commit-buffer
UCS-A /system/server-defaults/platform/bios-settings #
```

Viewing the Actual BIOS Settings for a Server

Follow this procedure to see the actual BIOS settings on a server.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-id* / *server-id*	Enters chassis server mode for the specified server.
Step 2	UCS-A /chassis/server # **scope bios**	Enters BIOS mode for the specified server.
Step 3	UCS-A /chassis/server/bios # **scope bios-settings**	Enters BIOS settings mode for the specified server.
Step 4	UCS-A /chassis/server/bios/bios-settings # **show** *setting*	Displays the BIOS setting. Enter **show ?** to display a list of allowed values for *setting* .

The following example displays a BIOS setting for blade 3 in chassis 1:

```
UCS-A# scope server 1/3
UCS-A /chassis/server # scope bios
UCS-A /chassis/server/bios # scope  bios-settings
UCS-A /chassis/server/bios/bios-settings # show intel-vt-config

Intel Vt Config:
    Vt
    --
    Enabled

UCS-A /chassis/server/bios/bios-settings #
```

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. Cisco UCS M4 blade and rack-mount servers include support for TPM. TPM is enabled by default on these servers.

Important

- If you upgrade Cisco UCS Manager to Release 2.2(4), TPM is enabled.
- When TPM is enabled and you downgrade Cisco UCS Manager from Release 2.2(4), TPM is disabled.

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. Cisco UCS M4 blade and rack-mount servers 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, on page 449
- Enabling or Disabling TXT, on page 450
- Clearing TPM for a Blade Server, on page 717 or Clearing TPM for a Rack-Mount Server, on page 731

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

Enabling or Disabling TPM

Procedure

	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 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-support** {**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.
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 / slot-id*	Associates the service profile with a single server.

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-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 1/2
```

Enabling or Disabling TXT

Procedure

	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 / slot-id*	Associates the service profile with a single server.

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 1/2
```

Consistent Device Naming

When there is no mechanism for the Operating System to label Ethernet interfaces in a consistent manner, it becomes difficult to manage network connections with server configuration changes. Consistent Device Naming (CDN), introduced in Cisco UCS Manager Release 2.2(4), allows Ethernet interfaces to be named in a consistent manner. This makes Ethernet interface names more persistent when adapter or other configuration changes are made.

To configure CDN for a vNIC, do the following:

- Enable consistent device naming in the BIOS policy.
- Associate the BIOS policy with a service profile.
- Configure consistent naming for a vNIC.

Guidelines and Limitations for Consistent Device Naming

- CDN is supported only on Windows 2012 R2. It is not supported on any other Operating System.
- Consistent device naming (CDN) is supported on all M3 and higher blade and rack-mount servers.
- BIOS and adapter firmware must be part of the Release 2.2(4) bundle to support CDN.
- In Cisco UCS Manager Release 2.2(4), CDN is supported only on the following adapters:
  - Cisco UCS VIC 1225 (UCSC-PCIE-CSC-02)
  - Cisco UCS MLOM 1227 (UCSC-MLOM-CSC-02)
  - Cisco UCS VIC 1225T (UCSC-PCIE-C10T-02)
  - Cisco UCS MLOM 1227T (UCSC-MLOM-C10T-02)
  - Cisco UCS VIC 1240 (UCSB-MLOM-40G-01)
  - Cisco UCS VIC 1280 (UCS-VIC-M82-8P)
  - Cisco UCS VIC 1340 (UCSB-MLOM-40G-03)
  - Cisco UCS VIC 1380 (UCSB-VIC-M83-8P)
- CDN is not supported for vNIC template and dynamic vNIC.
- Multiple vNICs within the same service profile cannot have the same CDN name.
- When a CDN name is not specified for a vNIC, the vNIC name is used as the CDN name.
- The CDN name that you configure for a vNIC appears as **Admin CDN Name**. The CDN name that is finally applied to the vNIC appears as **Oper CDN Name**. For example, if the **Admin CDN Name** for a vNIC called "vnic0" is cdn0, then the **Oper CDN Name** for this vNIC will be cdn0, but if the **Admin CDN Name** for the same vNIC is not specified, the **Oper CDN Name** will be vnic0.
- In Cisco UCS Manager Release 2.2(4), downgrade of Cisco UCS Manager is prevented if CDN is enabled in a BIOS policy that is assigned to an associated server.
- In Cisco UCS Manager Release 2.2(4), downgrade of the BIOS firmware is prevented if a CDN-enabled BIOS policy is assigned to a server.
- In Cisco UCS Manager Release 2.2(4), downgrade of the adapter firmware is prevented if a CDN-enabled BIOS policy is assigned to a server.
- When the applied BIOS policy is changed from CDN-disabled to CDN-enabled or from CDN-enabled to CDN-disabled, the host reboots with a warning, irrespective of whether reboot on BIOS update is enabled or not.
- It is recommended that you enable CDN in the BIOS policy and add CDN names to the vNICS before the Windows Operating System is installed.
- If the Windows Operating System is already installed on the server and CDN is then enabled in the BIOS policy, do the following:
  1. Uninstall the network drivers.
  2. Scan the system for hidden devices and uninstall them.
  3. Rescan the system for new hardware and install the network drivers again.

If this is not done, the vNICs will not come up with the configured CDN names.

- When the applied BIOS policy is changed from CDN-disabled to CDN-enabled or from CDN-enabled to CDN-disabled on a service profile, do the following:

  1. Uninstall the network drivers.
  2. Scan the system for hidden devices and delete them.
  3. Rescan the system for new hardware and install the network drivers again.

  Note When the BIOS policy is changed from CDN-enabled to CDN-disabled, ensure that the CDN names are removed from all the vNICs on the system.

- If any change is made to the vNICs, the BDF of all the devices on the system also changes. Following are some of the scenarios that trigger a change in the BDF of all the vNICs present on the system:
  - When a vNIC is added or deleted
  - When a vNIC is moved from one adapter on the system to another adapter on the system

  When these changes are made to the system, do the following:

  1. Uninstall the network driver from all the present network interfaces.
  2. Scan the system for hidden devices and uninstall them.
  3. Rescan the system for new hardware and install the network driver on the network controllers again.

  If the hidden devices are not deleted, the CDN names of the network adapters will not appear as configured on Cisco UCS Manager.

CDN with a Mixed Set of Adapters

When a CDN name is configured for a vNIC in a system with a mixed set of CDN-supported adapters and CDN-unsupported adapters, then system placement may not place CDN-configured vNICs on adapters that support CDN.

If CDN is enabled in the BIOS policy, and system placement places a CDN-configured vNIC (Admin CDN configured) on an adapter that does not support CDN, an info fault will be raised, but the configuration issue for the service profile will be ignored.

If CDN is enabled in the BIOS policy, and system placement places a vNIC (Admin CDN not configured) on an adapter that does not support CDN, an info fault will be raised, but the configuration issue for the service profile will be ignored. The **Oper CDN Name** in this case will be empty and will not be derived from the vNIC name.

If you want to deploy the CDN name as the host network interface name for a server, you must manually place a vNIC on a supported adapter.

Enabling Consistent Device Naming in a BIOS Policy

Procedure

	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 consistent-device-name-control cdn-name** {**enabled** | **disabled** | **platform-default**}	Specifies whether consistent device naming (CDN) is **enabled** or **disabled**.
Step 4	UCS-A /org/bios-policy* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to enable CDN in a BIOS policy:

```
UCS-A # scope org
UCS-A /org # create bios-policy cdn-bios-policy
UCS-A /org/bios-policy* # set consistent-device-name-control cdn-name enabled
UCS-A /org/bios-policy* # commit-buffer
```

Associating a BIOS Policy with a Service Profile

Procedure

	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** *sp-name*}	Enters service profile configuration mode for the specified service profile.
Step 3	UCS-A /org/service-profile # **set bios-policy** *policy-name*	Associates the specified BIOS policy with the service profile.
Step 4	UCS-A /org/service-profile* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to associate a CDN-enabled BIOS policy with a service profile:

```
UCS-A # scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile # set bios-policy cdn-bios-policy
UCS-A /org/service-profile* # commit-buffer
```

Configuring Consistent Device Naming for a vNIC

Procedure

	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** *sp-name*	Enters service profile configuration mode for the specified service profile.
Step 3	UCS-A /org/service-profile # **scope vnic** *vnic-name*	Enters vNIC configuration mode for the specified vNIC.
Step 4	UCS-A /org/service-profile/vnic # **set cdn-name** *cdn-name*	Specifies the CDN name for the vNIC.
Step 5	UCS-A /org/service-profile/vnic* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to configure CDN for a vNIC:

```
UCS-A # scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile # scope vnic vn1
UCS-A /org/service-profile/vnic # set cdn-name eth0
UCS-A /org/service-profile/vnic* # commit-buffer
```

Displaying the CDN Name of a vNIC

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *server-num*	Enters server mode for the specified server.
Step 2	UCS-A /server # **scope adapter** *adapter-id*	Enters adapter mode for the specified adapter.

	Command or Action	Purpose
Step 3	UCS-A /server/adapter # **show host-eth-if** [**detail**] [**expand**]	Displays the details of the host Ethernet interface for the specified adapter.

The following example shows how to display the CDN name of a vNIC:

```
UCS-A # scope server 3
UCS-A /server # scope adapter 1
UCS-A /server/adapter # show host-eth-if detail expand

Eth Interface:
    ID: 1
    Dynamic MAC Address: 00:25:B5:00:00:99
    Burned-In MAC Address: 00:00:00:00:00:00
    Model: UCSC-PCIE-CSC-02
    Name: vnic1
    Cdn Name: cdn0
    Admin State: Enabled
    Operability: Operable
    Order: 1
```

Displaying the Status of a vNIC

Procedure

	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** *sp-name*	Enters service profile configuration mode for the specified service profile.
Step 3	UCS-A /org/service-profile # **show vnic** [**detail**] [**expand**]	Displays the details of the vNIC in the specified service profile.

This example shows how to display the status of a vNIC.

Note The CDN name that you configured for the vNIC appears as the **Admin CDN Name**. The CDN name that is finally applied to the BIOS policy appears as the **Oper CDN Name**.

```
UCS-A# scope org
UCS-A /org # scope service-profile sp1
UCS-A /org/service-profile # show vnic detail expand

vNIC:
    Name: vnic1
    Fabric ID: B
```

```
Dynamic MAC Addr: 00:25:B5:17:47:01
Desired Order: Unspecified
Actual Order: 1
Desired VCon Placement: 2
Actual VCon Placement: 2
Desired Host Port: ANY
Actual Host Port: NONE
Equipment: sys/chassis-2/blade-5/adaptor-3/host-eth-2
Host Interface Ethernet MTU: 1500
Ethernet Interface Admin CDN Name:cdn0
Ethernet Interface Oper CDN Name:cdn0
Template Name:
```

CIMC Security Policies

Cisco UCS Manager provides the following policies to increase security:

- KVM Management Policy
- IPMI Access Profile

IPMI Access Profile

This policy allows you to determine whether IPMI commands can be sent directly to the server, using the IP address. For example, you can send commands to retrieve sensor data from the CIMC. This policy defines the IPMI access, including a username and password that can be authenticated locally on the server, and whether the access is read-only or read-write.

You can also restrict remote connectivity by disabling or enabling IPMI over LAN in the IPMI access profile. IPMI over LAN is disabled by default on all unassociated servers, and on all servers without an IPMI access policy. When an IPMI access policy is created, the IPMI over LAN is set to enabled by default. If you do not change the value to disabled, IPMI over LAN will be enabled on all associated servers.

You must include this policy in a service profile and that service profile must be associated with a server for it to take effect.

Configuring an IPMI Access Profile

Before You Begin

Obtain the following:

- Username with appropriate permissions that can be authenticated by the operating system of the server
- Password for the username
- Permissions associated with the username

Procedure

	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 # **create ipmi-access-profile** *profile-name*	Creates the specified IPMI access profile and enters organization IPMI access profile mode.
Step 3	UCS-A /org/ipmi-access-profile # **set ipmi-over-lan** {**disable** | **enable**}	Determines whether remote connectivity can be established. **Note** IPMI over LAN is disabled by default on all unassociated servers, and on all servers without an IPMI access policy. When an IPMI access policy is created, the IPMI over LAN is set to enabled by default. If you do not change the value to disabled, IPMI over LAN will be enabled on all associated servers.
Step 4	UCS-A /org/ipmi-access-profile # **create ipmi-user** *ipmi-user-name*	Creates the specified endpoint user and enters organization IPMI access profile endpoint user mode. **Note** More than one endpoint user can be created within an IPMI access profile, with each endpoint user having its own password and privileges.
Step 5	UCS-A /org/ipmi-access-profile/ipmi-user # **set password**	Sets the password for the endpoint user. After entering the **set password** command, you are prompted to enter and confirm the password. For security purposes, the password that you type does not appear in the CLI.
Step 6	UCS-A /org/ipmi-access-profile/ipmi-user # **set privilege** {**admin** | **readonly**}	Specifies whether the endpoint user has administrative or read-only privileges.
Step 7	UCS-A /org/ipmi-access-profile/ipmi-user # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates an IPMI access profile named ReadOnly, creates an endpoint user named bob, sets the password and the privileges for bob, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create ipmi-access-profile ReadOnly
UCS-A /org/ipmi-access-profile* # create ipmi-user bob
UCS-A /org/ipmi-access-profile/ipmi-user* # set password
Enter a password:
Confirm the password:
UCS-A /org/ipmi-access-profile/ipmi-user* # set privilege readonly
UCS-A /org/ipmi-access-profile/ipmi-user* # commit-buffer
UCS-A /org/ipmi-access-profile/ipmi-user #
```

What to Do Next

Include the IPMI profile in a service profile and/or template.

Deleting an IPMI Access Profile

Procedure

	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 # **delete ipmi-access-profile** *profile-name*	Deletes the specified IPMI access profile.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the IPMI access profile named ReadOnly and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete ipmi-access-profile ReadOnly
UCS-A /org* # commit-buffer
UCS-A /org #
```

Adding an Endpoint User to an IPMI Access Profile

Procedure

	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 ipmi-access-profile** *profile-name*	Enters organization IPMI access profile mode for the specified IPMI access profile.
Step 3	UCS-A /org/ipmi-access-profile # **create ipmi-user** *ipmi-user-name*	Creates the specified endpoint user and enters organization IPMI access profile endpoint user mode. **Note** More than one endpoint user can be created within an IPMI access profile, with each endpoint user having its own password and privileges.
Step 4	UCS-A /org/ipmi-access-profile/ipmi-user # **set password**	Sets the password for the endpoint user. After entering the **set password** command, you are prompted to enter and confirm the password. For security

	Command or Action	Purpose
		purposes, the password that you type does not appear in the CLI.
Step 5	UCS-A /org/ipmi-access-profile/ipmi-user # **set privilege** {**admin** | **readonly**}	Specifies whether the endpoint user has administrative or read-only privileges.
Step 6	UCS-A /org/ipmi-access-profile/ipmi-user # **commit-buffer**	Commits the transaction to the system configuration.

The following example adds an endpoint user named alice to the IPMI access profile named ReadOnly and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope ipmi-access-profile ReadOnly
UCS-A /org/ipmi-access-profile* # create ipmi-user alice
UCS-A /org/ipmi-access-profile/ipmi-user* # set password
Enter a password:
Confirm the password:
UCS-A /org/ipmi-access-profile/ipmi-user* # set privilege readonly
UCS-A /org/ipmi-access-profile/ipmi-user* # commit-buffer
UCS-A /org/ipmi-access-profile/ipmi-user #
```

Deleting an Endpoint User from an IPMI Access Profile

Procedure

	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 ipmi-access-profile** *profile-name*	Enters organization IPMI access profile mode for the specified IPMI access profile.
Step 3	UCS-A /org/ipmi-access-profile # **delete ipmi-user** *epuser-name*	Deletes the specified endpoint user from the IPMI access profile.
Step 4	UCS-A /org/ipmi-access-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the endpoint user named alice from the IPMI access profile named ReadOnly and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope ipmi-access-profile ReadOnly
UCS-A /org/ipmi-access-profile # delete ipmi-user alice
UCS-A /org/ipmi-access-profile* # commit-buffer
UCS-A /org/ipmi-access-profile #
```

KVM Management Policy

The KVM Management policy allows you to determine whether vMedia encryption is enabled when you access a server via KVM.

You must include this policy in a service profile and that service profile must be associated with a server for it to take effect.

Note After a KVM vMedia session is mapped, if you change the KVM management policy, it will result in a loss of the vMedia session. You must re-map the KVM vMedia session again.

Configuring a KVM Management Policy

Procedure

	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 # **create kvm-mgmt-policy** *policy-name*	Creates the specified KVM management policy and enters organization KVM management policy mode.
Step 3	UCS-A /org/kvm-mgmt-policy # **set descr** *description*	(Optional) Provides a description for the policy.
Step 4	UCS-A /org/kvm-mgmt-policy # **set vmedia-encryption** {**disable** | **enable**}	Specifies vMedia encryption is enabled or disabled.
Step 5	UCS-A /org/ipmi-access-profile/ipmi-user # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a KVM management policy named KVM_Policy1, enable vMedia encryption, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create kvm-mgmt-policy KVM_Policy1
UCS-A /org/kvm-mgmt-policy* # set vmedia-encryption enable
UCS-A /org/kvm-mgmt-policy* # commit-buffer
UCS-A /org/kvm-mgmt-policy #
```

Configuring Local Disk Configuration Policies

Local Disk Configuration Policy

This policy configures any optional SAS local drives that have been installed on a server through the onboard RAID controller of the local drive. This policy enables you to set a local disk mode for all servers that are associated with a service profile that includes the local disk configuration policy.

The local disk modes include the following:

- **No Local Storage—**For a diskless server or a SAN only configuration. If you select this option, you cannot associate any service profile which uses this policy with a server that has a local disk.
- **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.
- **Any Configuration—**For a server configuration that carries forward the local disk configuration without any changes.
- **No RAID—**For a server configuration that removes the RAID and leaves the disk MBR and payload unaltered.

  If you choose **No RAID** and you apply this policy to a server that already has an operating system with RAID storage configured, the system does not remove the disk contents. Therefore, there may be no visible differences on the server after you apply the **No RAID** mode. This can lead to a mismatch between the RAID configuration in the policy and the actual disk configuration shown in the **Inventory** > **Storage** tab for the server.

  To make sure that any previous RAID configuration information is removed from a disk, apply a scrub policy that removes all disk information after you apply the **No RAID** configuration mode.
- **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 6 Striped Dual Parity—**Data is striped across all disks in the array and two parity disks are used to provide protection against the failure of up to two physical disks. In each row of data blocks, two sets of parity data are stored.
- **RAID 10 Mirrored and Striped—**RAID 10 uses mirrored pairs of disks to provide complete data redundancy and high throughput rates.
- **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.

You must include this policy in a service profile and that service profile must be associated with a server for the policy to take effect.

Note For a Cisco UCS C-Series server integrated with Cisco UCS Manager, with an embedded on-board RAID controller, the local disk mode should always be **Any Configuration**, and the RAID must be configured directly on the controller.

Guidelines for all Local Disk Configuration Policies

Before you create a local disk configuration policy, consider the following guidelines:

No Mixed HDDs and SSDs

Do not include HDDs and SSDs in a single server or RAID configuration.

Do Not Assign a Service Profile with the Default Local Disk Configuration Policy from a B200 M1 or M2 to a B200 M3

Due to the differences in the RAID/JBOD support provided by the storage controllers of B200 M1 and M2 servers and those of the B200 M3 server, you cannot assign or re-assign a service profile that includes the default local disk configuration policy from a B200M1 or M2 server to a B200 M3 server. The default local disk configuration policy includes those with Any Configuration or JBOD configuration.

JBOD Mode Support

The B200 M3 server supports JBOD mode for local disks.

Note Only B200 M1, B200 M2, B200 M3, B250 M1, B250 M2 and B22 M3 blade servers support the JBOD mode for local disks.

Guidelines for Local Disk Configuration Policies Configured for RAID

Configure RAID Settings in Local Disk Configuration Policy for Servers with MegaRAID Storage Controllers

If a blade server or integrated rack-mount server has a MegaRAID controller, you must configure RAID settings for the drives in the Local Disk Configuration policy included in the service profile for that server. You can do this either by configuring the local disk configuration policy in the service profile using one of the defined RAID modes for that server, or you can use the **Any Configuration** mode with the LSI Utilities toolset to create the RAID volumes.

If you do not configure your RAID LUNs before installing the OS, disk discovery failures might occur during the installation and you might see error messages such as “No Device Found.”

Server May Not Boot After RAID1 Cluster Migration if Any Configuration Mode Specified in Service Profile

After RAID1 clusters are migrated, you need to associate a service profile with the server. If the local disk configuration policy in the service profile is configured with **Any Configuration** mode rather than **RAID1**, the RAID LUN remains in "inactive" state during and after association. As a result, the server cannot boot.

To avoid this issue, ensure that the service profile you associate with the server contains the identical local disk configuration policy as the original service profile before the migration and does not include the **Any Configuration** mode.

Do Not Use JBOD Mode on Servers with MegaRAID Storage Controllers

Do not configure or use JBOD mode or JBOD operations on any blade server or integrated rack-mount server with a MegaRAID storage controllers. JBOD mode and operations are not intended for nor are they fully functional on these servers.

Maximum of One RAID Volume and One RAID Controller in Integrated Rack-Mount Servers

A rack-mount server that has been integrated with Cisco UCS Manager can have a maximum of one RAID volume irrespective of how many hard drives are present on the server.

All the local hard drives in an integrated rack-mount server must be connected to only one RAID Controller. Integration with Cisco UCS Manager does not support the connection of local hard drives to multiple RAID Controllers in a single rack-mount server. We therefore recommend that you request a single RAID Controller configuration when you order rack-mount servers to be integrated with Cisco UCS Manager.

In addition, do not use third party tools to create multiple RAID LUNs on rack-mount servers. Cisco UCS Manager does not support that configuration.

Maximum of One RAID Volume and One RAID Controller in Blade Servers

A blade server can have a maximum of one RAID volume irrespective of how many drives are present in the server. All the local hard drives must be connected to only one RAID controller. For example, a B200 M3 server has an LSI controller and an Intel Patsburg controller, but only the LSI controller can be used as a RAID controller.

In addition, do not use third party tools to create multiple RAID LUNs on blade servers. Cisco UCS Manager does not support that configuration.

Number of Disks Selected in Mirrored RAID Should Not Exceed Two

If the number of disks selected in the Mirrored RAID exceed two, RAID 1 is created as a RAID 10 LUN. This issue can occur with the Cisco UCS B440 M1 and B440 M2 servers.

License Required for Certain RAID Configuration Options on Some Servers

Some Cisco UCS servers require a license for certain RAID configuration options. When Cisco UCS Manager associates a service profile containing this local disk policy with a server, Cisco UCS Manager verifies that the selected RAID option is properly licensed. If there are issues, Cisco UCS Manager displays a configuration error during the service profile association.

For RAID license information for a specific Cisco UCS server, see the *Hardware Installation Guide* for that server.

B420 M3 Server Does Not Support All Configuration Modes

The B420 M3 server does not support the following configuration modes in a local disk configuration policy:

- No RAID
- RAID 6 Striped Dual Parity

In addition, the B420 M3 does not support JBOD modes or operations.

Single-Disk RAID 0 Configurations Not Supported on Some Blade Servers

A single-disk RAID 0 configuration is not supported in the following blade servers:

- Cisco UCS B200 M1
- Cisco UCS B200 M2
- Cisco UCS B250 M1
- Cisco UCS B250 M2

Creating a Local Disk Configuration Policy

Procedure

	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 # **create local-disk-config-policy** *policy-name*	Creates a local disk configuration policy and enters local disk configuration policy mode.
Step 3	UCS-A /org/local-disk-config-policy # **set descr** *description*	(Optional) Provides a description for the local disk configuration policy.
Step 4	UCS-A /org/local-disk-config-policy # **set mode** {**any-configuration** | **no-local-storage** | **no-raid** | **raid-0-striped** | **raid-1-mirrored** | **raid-5-striped-parity** | **raid-6-striped-dual-parity** | **raid-10-mirrored-and-striped**}	Specifies the mode for the local disk configuration policy.
Step 5	UCS-A /org/local-disk-config-policy # **set protect** {**yes** | **no**}	Specifies whether the server retains the configuration in the local disk configuration policy even if the server is disassociated from the service profile. **Caution** Protect Configuration becomes non-functional if one or more disks in the server are defective or faulty. When a service profile is disassociated from a server and a new service profile associated, the setting for the Protect Configuration property in the new service profile takes precedence and overwrites the setting in the previous service profile. With this option enabled, the data on the disk is protected even after the server is decommissioned and then recommissioned. Hence, reassociation of the server with a service profile fails.

	Command or Action	Purpose
		Note If you disassociate the server from a service profile with this option enabled and then associate it with a new service profile that includes a local disk configuration policy with different properties, the server returns a configuration mismatch error and the association fails.
Step 6	UCS-A /org/local-disk-config-policy # **set flexflash-state** {**enable** | **disable**}	Specifies whether FlexFlash SD card support is enabled.
Step 7	UCS-A /org/local-disk-config-policy # **set flexflash-raid-reporting-state** {**enable** | **disable**}	Specifies whether FlexFlash RAID reporting support is enabled. **Note** If only one SD card is installed, the FlexFlash inventory displays the RAID State as Disabled and the RAID Health as NA.
Step 8	UCS-A /org/local-disk-config-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a local disk configuration policy and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create local-disk-config-policy DiskPolicy7
UCS-A /org/local-disk-config-policy* # set mode raid-1-mirrored
UCS-A /org/local-disk-config-policy* # set protect yes
UCS-A /org/local-disk-config-policy* # commit-buffer
UCS-A /org/local-disk-config-policy #
```

Viewing a Local Disk Configuration Policy

Procedure

	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 # **show local-disk-config-policy** *policy-name*	Displays the local disk policy. If you have not configured a local disk policy, the local disk configuration (created by the create local-disk-config command) displays. Displays the local disk definition (set by the create local-disk-config command). If the serial over LAN definition is not set, and if a policy is set (using the set local-disk-config-policy command), then the policy will be displayed.

The following example shows how to display local disk policy information for a local disk configuration policy called DiskPolicy7:

```
UCS-A# scope org /
UCS-A /org # show local-disk-config-policy DiskPolicy7

Local Disk Config Policy:
Name: DiskPolicy7
Mode: Raid 1 Mirrored
Description:
Protect Configuration: Yes
```

Deleting a Local Disk Configuration Policy

Procedure

	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 # **delete local-disk-config-policy** *policy-name*	Deletes the specified local disk configuration policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the local disk configuration policy named DiskPolicy7 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete local-disk-config-policy DiskPolicy7
UCS-A /org* # commit-buffer
UCS-A /org #
```

FlexFlash Support

Overview

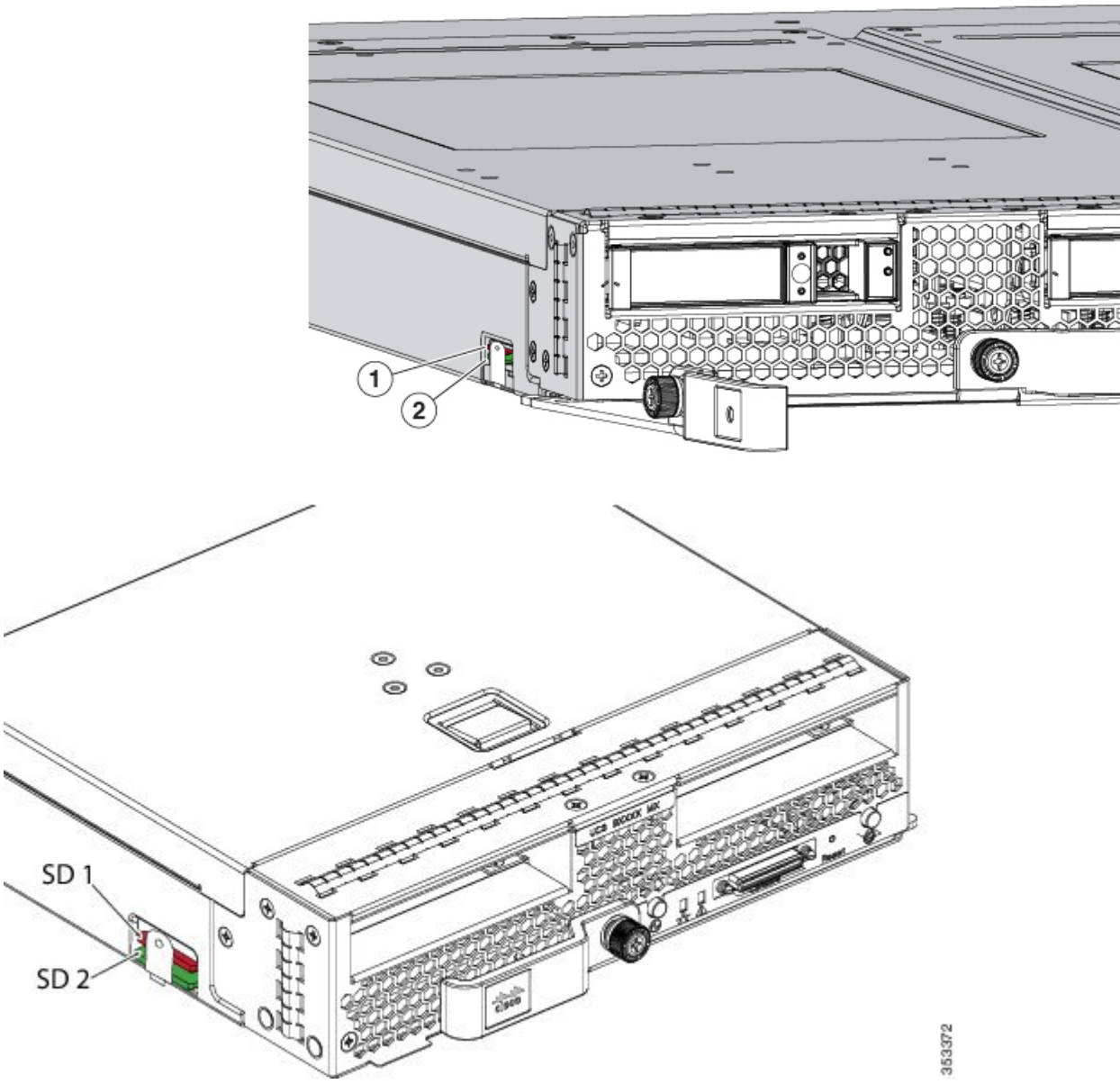
Cisco UCS B-Series, C-Series M3 and higher, and S-Series M4 servers support internal Secure Digital (SD) memory cards. The SD cards are hosted by the Cisco Flexible Flash storage controller, a PCI-based controller which has two slots for SD cards. The cards contain a single partition called HV. When FlexFlash is enabled, Cisco UCS Manager displays the HV partition as a USB drive to both the BIOS and the host operating system.

You can populate one or both the SD card slots that are provided. If two SD cards are populated, you can use them in a mirrored mode.

Note Do not mix different capacity cards in the same server.

The SD cards can be used to store operating system boot images or other information. The following figure illustrates the SD card slots.

Figure 2: SD Card Slots



FlexFlash is disabled by default. You can enable FlexFlash in a local disk policy used in a service profile. When FlexFlash is enabled in a local disk policy, and the server is capable of supporting SD cards, the FlexFlash controller is enabled during service profile association. If a server is not capable of supporting SD cards or has an older CIMC version, a config failure message is displayed.

If you disable FlexFlash in a supported server, the Hypervisor or HV partition is immediately disconnected from the host. The FlexFlash controller will also be disabled as part of a related service profile disassociation.

The FlexFlash controller supports RAID-1 for dual SD cards. The FlexFlash scrub policy erases the HV partition in both cards, and brings the cards to a healthy RAID state.

You can configure new SD cards in a RAID pair and format them using one of the following methods:

- Format the SD cards.
- For an associated server, create a FlexFlash scrub policy and disassociate the service profile from the server. For an unassociated server, create a FlexFlash scrub policy and reacknowledge the server after modifying the default scrub policy.

  The *Scrub Policy Settings* section in the *Cisco UCS Manager Server Management Guide* provides more details about the usage of the scrub policy.

Note Disable the scrub policy as soon as the pairing is complete.

To boot from the HV partition, the SD card must be present in the boot policy used in the service profile.

FlexFlash Firmware Management

The FlexFlash controller firmware is bundled as part of the CIMC image. When you upgrade the CIMC, if a newer firmware version is available for the FlexFlash controller, the controller can no longer be managed, and the FlexFlash inventory displays the **Controller State** as **Waiting For User Action** and the **Controller Health** as **Old Firmware Running**. To upgrade the FlexFlash controller firmware, you need to perform a board controller update. For more information, see the appropriate *Cisco UCS B-Series Firmware Management Guide*, available at the following URL: http://www.cisco.com/en/US/products/ps10281/products_installation_and_configuration_guides_list.html.

Limitations for the Cisco Flexible Flash Storage Controller:

- The Cisco Flexible Flash storage controller only supports 16 GB, 32 GB, and 64 GB SD cards.

  Note 16 GB and 32 GB cards are supported only on the B200-M3 blade servers, and the 64 GB SD cards are supported only on the B200-M4 blade servers.

- We do not recommend using an SD card from a rack server in a blade server, or using an SD card from a blade server in a rack server. Switching SD cards between server types might result in data loss from the SD card.
- Some Cisco UCS C-Series rack-mount servers have SD cards with four partitions: HV, HUU, SCU, and Drivers. Only the HV partition is visible in Cisco UCS Manager. You can migrate a four-partition SD card to a single HV partition card with a FlexFlash scrub policy.
- The FlexFlash controller does not support RAID-1 sync (mirror rebuild). If the SD cards are in a degraded RAID state, or if any metadata errors are reported by the controller, you must run the FlexFlash scrub policy to pair the cards for RAID. For more information about the FlexFlash scrub policy, see Server-Related Policies. The following conditions might result in degraded RAID or metadata errors:
  - Inserting a new or used SD card in one slot, when the server already has an SD card populated in the second slot.
  - Inserting two SD cards from different servers.

- The server firmware version must be at 2.2(1a) or higher.

FlexFlash FX3S Support

Beginning with Release 2.2(3), Cisco UCS Manager allows additional FlexFlash support with the FX3S controller. The FX3S controller is present on the following servers:

- Cisco UCS B200 M4 blade server
- Cisco UCS C220 M4 rack server
- Cisco UCS C240 M4 rack server

FlexFlash operations with the FX3S control are similar to those with the Cisco Flexible Flash storage controller. FlexFlash is disabled by default, and is enabled using a local disk policy. You can also reset the controller, format the SD cards, and enable automatic synchronization of your paired SD cards.

The SD cards for the FX3S controller contain a single partition called Hypervisor.

Limitations for the Cisco FX3S Controller:

- The FX3S controller supports only 32 GB and 64 GB SD cards. 16 GB cards are not supported.
- We do not recommend using an SD card from a rack server in a blade server, or using an SD card from a blade server in a rack server. Switching SD cards between server types might result in data loss from the SD card.
- The server firmware version must be at 2.2(3a) or higher.

Enabling or Disabling FlexFlash SD Card Support

Procedure

	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 local-disk-config-policy** *policy-name*	Enters the specified local disk configuration policy mode.
Step 3	UCS-A /org/local-disk-config-policy # **set flexflash-state** {**enable** | **disable**}	Specifies whether FlexFlash SD card support is enabled.
Step 4	UCS-A /org/local-disk-config-policy # **set flexflash-raid-reporting-state** {**enable** | **disable**}	Specifies whether FlexFlash RAID reporting support is enabled. **Note** If only one SD card is installed, the FlexFlash inventory displays the RAID State as Disabled and the RAID Health as NA.

	Command or Action	Purpose
Step 5	UCS-A /org/local-disk-config-policy # **commit-buffer**	Commits the transaction to the system.

The following example shows how to enable FlexFlash SD card support and FlexFlash RAID reporting state on the local disk config policy default, and commits the transaction to the system:

```
UCS-A# scope org/
UCS-A /org # scope local-disk-config-policy default
UCS-A /org/local-disk-config-policy #set flexflash-state enable
UCS-A /org/local-disk-config-policy# #set flexflash-raid-reporting-state enable
UCS-A /org/local-disk-config-policy* # commit-buffer
UCS-A /org/local-disk-config-policy #
```

Enabling Auto-Sync

Procedure

	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 server** *server-num*	Enters server chassis mode.
Step 3	UCS-A /chassis/server # **scope flexflash-controller** *controller-id*	Enters flexflash controller server chassis mode.
Step 4	UCS-A /chassis/server/flexflash-controller # **pair** *primary_slot_number*	Resyncs the SD cards if they are out of sync, using the card in the selected slot number as the primary. This can be one of the following: • **1—**The SD card in slot 1 will be used as the primary. • **2—**The SD card in slot 2 will be used as the primary.
Step 5	UCS-A /chassis/server/flexflash-controller # **commit-buffer**	Commits the transaction to the system configuration.

The following example resyncs the SD cards using the SD card in slot 2 as the primary:

```
UCS-A# scope chassis 1
UCS-A /chassis # scope server 1
UCS-A /chassis/server # scope flexflash-controller 1
UCS-A /chassis/server/flexflash-controller # pair 2
UCS-A /chassis/server/flexflash-controller* # commit-buffer
UCS-A /chassis/server/flexflash-controller #
```

Formatting the FlexFlash Cards

Procedure

	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 server** *server-num*	Enters server chassis mode.
Step 3	UCS-A /chassis/server # **scope flexflash-controller** *controller-id*	Enters flexflash controller server chassis mode.
Step 4	UCS-A /chassis/server/flexflash-controller # **format**	Formats the SD cards.
Step 5	UCS-A /chassis/server/flexflash-controller # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to format the FlexFlash controller:

```
UCS-A# scope chassis 1
UCS-A /chassis # scope server 1
UCS-A /chassis/server # scope flexflash-controller 1
UCS-A /chassis/server/flexflash-controller # format
Warning: When commited, UCSM will format the SD Cards.
This will completely erase the data on the SD Cards!!

UCS-A /chassis/server/flexflash-controller* # commit-buffer
UCS-A /chassis/server/flexflash-controller #
```

Resetting the FlexFlash Controller

Procedure

	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 server** *server-num*	Enters server chassis mode.
Step 3	UCS-A /chassis/server # **scope flexflash-controller** *controller-id*	Enters flexflash controller server chassis mode.
Step 4	UCS-A /chassis/server/flexflash-controller # **reset**	Resets the specified FlexFlash controller.
Step 5	UCS-A /chassis/server/flexflash-controller # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to reset the FlexFlash controller:

```
UCS-A# scope chassis 1
UCS-A /chassis # scope server 1
UCS-A /chassis/server # scope flexflash-controller 1
UCS-A /chassis/server/flexflash-controller # reset
Warning: When commited, UCSM will reset the FlexFlash Controller.
This will cause the host OS to lose connectivity to the SD Cards.

UCS-A /chassis/server/flexflash-controller* # commit-buffer
UCS-A /chassis/server/flexflash-controller #
```

Viewing the FlexFlash Controller Status

Procedure

	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 server** *server-num*	Enters server chassis mode.
Step 3	UCS-A /chassis/server # **scope flexflash-controller** *controller-id*	Enters flexflash controller server chassis mode.
Step 4	UCS-A /chassis/server/flexflash-controller # **show detail expand**	Displays the detailed FlexFlash controller properties.

The following example shows the status of the FlexFlash controller and SD cards:

```
UCS-A# scope chassis 1
UCS-A /chassis # scope server 1
UCS-A /chassis/server # scope flexflash-controller 1
UCS-A /chassis/server/flexflash-controller # show detail expand

FlexFlash Controller:
    ID: 1
    Type: SD
    FlexFlash Type: FX3S
    Vendor: Cypress
    Model: FX3S
    Serial: NA
    Firmware Version: 1.3.2 build 158
    Controller State: Connected Partition Over USB To Host
    Controller Health: Old Firmware Running
    RAID State: Enabled Paired
    RAID Health: OK
    Physical Drive Count: 2
    Virtual Drive Count: 1
    RAID Sync Support: Supported
    Operability: Operable
    Oper Qualifier Reason:
    Presence: Equipped
    Current Task:

    FlexFlash Card:
        Controller Index: 1
        Slot Number: 1
        Vendor: SE32G
        Model: SE32G
        HW Rev: 8.0
```

```
            Serial: 0xa2140794
            Manufacturer ID: 3
            OEM ID: SD
            Manufacturer Date: 2/14
            Size (MB): 30436
            Block Size: 512
            Card Type: FX3S configured
            Write Enabled: Not Write Protected
            Card Health: OK
            Card Mode: Secondary Active
            Operation State: Raid Partition
            Card State: Active
            Write IO Error Count: 0
            Read IO Error Count: 0
            Operability: Operable
            Oper Qualifier Reason:
            Presence: Equipped

            FlexFlash Card Drive:
                Name: Hypervisor
                Size (MB): 30432
                Removable: Yes
                Operability: Operable
                Operation State: Raid Partition

            Controller Index: 1
            Slot Number: 2
            Vendor: SE32G
            Model: SE32G
            HW Rev: 8.0
            Serial: 0xa2140742
            Manufacturer ID: 3
            OEM ID: SD
            Manufacturer Date: 2/14
            Size (MB): 30436
            Block Size: 512
            Card Type: FX3S configured
            Write Enabled: Not Write Protected
            Card Health: OK
            Card Mode: Primary
            Operation State: Raid Partition
            Card State: Active
            Write IO Error Count: 0
            Read IO Error Count: 0
            Operability: Operable
            Oper Qualifier Reason:
            Presence: Equipped

            FlexFlash Card Drive:
                Name: Hypervisor
                Size (MB): 30432
                Removable: Yes
                Operability: Operable
                Operation State: Raid Partition

        Local Disk Config Definition:
            Mode: Any Configuration
            Description:
            Protect Configuration: Yes

    UCS-A /chassis/server/flexflash-controller #
```

Configuring Scrub Policies

Scrub Policy Settings

This policy determines what happens to local data and to the BIOS settings on a server during the discovery process, when the server is re-acknowledged, or when the server is disassociated from a service profile.

Note Local disk scrub policies only apply to hard drives that are managed by Cisco UCS Manager and do not apply to other devices such as USB drives.

Depending upon how you configure a scrub policy, the following can occur at those times:

Disk scrub

One of the following occurs to the data on any local drives on disassociation:

- If enabled, destroys all data on any local drives.
- If disabled, preserves all data on any local drives, including local storage configuration.

BIOS Settings Scrub

One of the following occurs to the BIOS settings when a service profile containing the scrub policy is disassociated from a server:

- If enabled, erases all BIOS settings for the server and resets them to the BIOS defaults for that server type and vendor.
- If disabled, preserves the existing BIOS settings on the server.

FlexFlash Scrub

FlexFlash Scrub enables you to pair new or degraded SD cards, resolve FlexFlash metadata configuration failures, and migrate older SD cards with 4 partitions to single partition SD cards. One of the following occurs to the SD card when a service profile containing the scrub policy is disassociated from a server, or when the server is reacknowledged:

- If enabled, the HV partition on the SD card is formatted using the PNUOS formatting utility. If two SD cards are present, the cards are RAID-1 paired, and the HV partitions in both cards are marked as valid. The card in slot 1 is marked as primary, and the card in slot 2 is marked as secondary.
- If disabled, preserves the existing SD card settings.

Note

- Because the FlexFlash scrub erases the HV partition on the SD cards, we recommend that you take a full backup of the SD card(s) using your preferred host operating system utilities before performing the FlexFlash Scrub.
- To resolve metadata config failures in a service profile, you need to disable FlexFlash in the local disk config policy before you run the FlexFlash scrub, then enable FlexFlash after the server is reacknowledged.
- Disable the scrub policy as soon as the pairing is complete or the metadata failures are resolved.
- FlexFlash scrub is not supported for Cisco UCS S3260 Storage Server.

Creating a Scrub Policy

Procedure

	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 # **create scrub-policy** *policy-name*	Creates a scrub policy with the specified policy name, and enters organization scrub policy mode.
Step 3	UCS-A /org/scrub-policy # **set descr** *description*	(Optional) Provides a description for the scrub policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/scrub-policy # **set disk-scrub** {**no** | **yes**}	Disables or enables disk scrubbing on servers using this scrub policy as follows: • If enabled, destroys all data on any local drives. • If disabled, preserves all data on any local drives, including local storage configuration.
Step 5	UCS-A /org/scrub-policy # **set bios-settings-scrub** {**no** | **yes**}	Disables or enables BIOS settings scrubbing on servers using this scrub policy as follows: • If enabled, erases all BIOS settings for the server and resets them to the BIOS defaults for that server type and vendor. • If disabled, preserves the existing BIOS settings on the server.

	Command or Action	Purpose
Step 6	UCS-A /org/scrub-policy # **set flexflash-scrub** {**no** | **yes**}	Disables or enables flexflash scrubbing on servers using this scrub policy as follows: • If enabled, the HV partition on the SD card is formatted using the PNUOS formatting utility. If two SD cards are present, the cards are RAID-1 paired, and the HV partitions in both cards are marked as valid. The card in slot 1 is marked as primary, and the card in slot 2 is marked as secondary. • If disabled, preserves the existing SD card settings.
Step 7	UCS-A /org/scrub-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a scrub policy named ScrubPolicy2, enables disk scrubbing on servers using the scrub policy, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create scrub-policy ScrubPolicy2
UCS-A /org/scrub-policy* # set descr "Scrub disk but not BIOS."
UCS-A /org/scrub-policy* # set disk-scrub yes
UCS-A /org/scrub-policy* # set bios-settings-scrub no
UCS-A /org/scrub-policy* # set flexflash-scrub no
UCS-A /org/scrub-policy* # commit-buffer
UCS-A /org/scrub-policy #
```

Deleting a Scrub Policy

Procedure

	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 # **delete scrub-policy** *policy-name*	Deletes the specified scrub policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the scrub policy named ScrubPolicy2 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete scrub-policy ScrubPolicy2
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring DIMM Error Management

DIMM Correctable Error Handling

In Cisco UCS Manager, when a DIMM encounters a significant correctable error in a given predefined window, it is stated as degraded and considered as a non-functional device.

The DIMM correctable error handling feature enables you to reset all the correctable and uncorrectable memory errors on all the DIMMs in a server. When you reset the error configuration, the error count of a given DIMM is cleared, the status changes to operable, and it resets the sensor state of the given DIMM.

Resetting Memory Errors

Use this procedure to reset all correctable and uncorrectable memory errors encountered by Cisco UCS Manager and the baseboard management controller (BMC).

Procedure

	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 server** *server-num*	Enters server mode for the specified server.
Step 3	UCS-A/chassis/server # **reset-all-memory-errors**	Resets the correctable and uncorrectable errors on all the DIMMs in a server.
Step 4	UCS-A /chassis/server* # **commit-buffer**	Commits any pending transactions.

This example shows how to reset the memory errors for the selected memory unit(s):

```
UCS-A# scope chassis 1
UCS-A/chassis # scope server 1
UCS-A/chassis/server # reset-all-memory-errors
UCS-A/chassis/server* # commit-buffer
UCS-A/chassis/server #
```

DIMM Blacklisting

In Cisco UCS Manager, the state of the Dual In-line Memory Module (DIMM) is based on SEL event records. When the BIOS encounters a noncorrectable memory error during memory test execution, the DIMM is marked as faulty. A faulty DIMM is a considered a nonfunctional device.

If you enable DIMM blacklisting, Cisco UCS Manager monitors the memory test execution messages and blacklists any DIMMs that encounter memory errors in the DIMM SPD data. To allow the host to map out any DIMMs that encounter uncorrectable ECC errors.

Enabling DIMM Blacklisting

The memory policy is a global policy that you can apply to existing servers on a Cisco UCS domain and also to the servers that are added after you set the memory policy.

Note

- This feature is supported both on the Cisco UCS B-Series blade servers and UCS C-Series rack servers.

  Note Cisco UCS C-Series 420 M3 rack server do not support this feature.

- This global policy cannot be added to a service profile.

Before You Begin

- For Cisco B-Series blade server, the server firmware must be at Release 2.2(1) or a later release.
- For Cisco C-Series rack server, the server firmware must be at Release 2.2(3).
- You must be logged in with one of the following privileges:
  - Admin
  - Server policy
  - Server profile server policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** /	Enters root organization mode.
Step 2	UCS-A /org # **scope memory-config-policy default**	Enters memory policy mode for the global memory policy.
Step 3	UCS-A /org/memory-config-policy # **set blacklisting enabled**	Enables DIMM blacklisting for the domain level policy and these changes applies to all the servers on that particular domain. **Note** If the Cisco IMC of a server does not support DIMM blacklisting, an information level fault is generated.
Step 4	UCS-A /org/memory-config-policy* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to enable DIMM blacklisting:

```
UCS-A# scope org /
UCS-A /chassis/org # scope memory-config-policy default
UCS-A /chassis/org/memory-config-policy # set blacklisting enabled
```

```
UCS-A /chassis/org/memory-config-policy* # commit-buffer
UCS-A /chassis/org/memory-config-policy #
UCS-A /chassis/org/memory-config-policy # show detail

Memory Config Policy:
    Blacklisting: enabled
```

Configuring Serial over LAN Policies

Serial over LAN Policy Overview

This policy sets the configuration for the serial over LAN connection for all servers associated with service profiles that use the policy. By default, the serial over LAN connection is disabled.

If you implement a serial over LAN policy, we recommend that you also create an IPMI profile.

You must include this policy in a service profile and that service profile must be associated with a server for it to take effect.

Configuring a Serial over LAN Policy

Procedure

	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 # **create sol-policy** *policy-name*	Creates a serial over LAN policy and enters organization serial over LAN policy mode.
Step 3	UCS-A /org/sol-policy # **set descr** *description*	(Optional) Provides a description for the policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/sol-policy # **set speed** {**115200** | **19200** | **38400** | **57600** | **9600**}	Specifies the serial baud rate.
Step 5	UCS-A /org/sol-policy # {**disable** | **enable**}	Disables or enables the serial over LAN policy. By default, the serial over LAN policy is disabled; you must enable it before it can be applied.
Step 6	UCS-A /org/sol-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a serial over LAN policy named Sol9600, provides a description for the policy, sets the speed to 9,600 baud, enables the policy, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create sol-policy Sol9600
UCS-A /org/sol-policy* # set descr "Sets serial over LAN policy to 9600 baud."
UCS-A /org/sol-policy* # set speed 9600
UCS-A /org/sol-policy* # enable
UCS-A /org/sol-policy* # commit-buffer
UCS-A /org/sol-policy #
```

Viewing a Serial over LAN Policy

Procedure

	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 # **show sol-policy** *policy-name*	Displays the serial over LAN definition (set by the create sol-config command). If the serial over LAN definition is not set, and if a policy is set (using the set sol-policy command), then the policy will be displayed.

The following example shows how to display serial over LAN information for a serial over LAN policy called Sol115200:

```
UCS-A# scope org /
UCS-A /org # show sol-policy Sol115200

SOL Policy:
    Name: sol115200
    SOL State: Enable
    Speed: 115200
    Description:
    Policy Owner: Local
```

Deleting a Serial over LAN Policy

Procedure

	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 # **delete sol-policy** *policy-name*	Deletes the specified serial over LAN policy.

	Command or Action	Purpose
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the serial over LAN policy named Sol9600 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # delete sol-policy Sol9600
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Server Autoconfiguration Policies

Server Autoconfiguration Policy Overview

Cisco UCS Manager uses this policy to determine how to configure a new server. If you create a server autoconfiguration policy, the following occurs when a new server starts:

1. The qualification in the server autoconfiguration policy is executed against the server.
2. If the server meets the required qualifications, the server is associated with a service profile created from the service profile template configured in the server autoconfiguration policy. The name of that service profile is based on the name given to the server by Cisco UCS Manager.
3. The service profile is assigned to the organization configured in the server autoconfiguration policy.

Configuring a Server Autoconfiguration Policy

Procedure

	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 # **create server-autoconfig-policy** *policy-name*	Creates a server autoconfiguration policy with the specified policy name, and enters organization server autoconfiguration policy mode.
Step 3	UCS-A /org/server-autoconfig-policy # **set descr** *description*	(Optional) Provides a description for the policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.

	Command or Action	Purpose
Step 4	UCS-A /org/server-autoconfig-policy # **set destination org** *org-name*	(Optional) Specifies the organization for which the server is to be used.
Step 5	UCS-A /org/server-autoconfig-policy # **set qualifier** *server-qual-name*	(Optional) Specifies server pool policy qualification to use for qualifying the server.
Step 6	UCS-A /org/server-autoconfig-policy # **set template** *profile-name*	(Optional) Specifies a service profile template to use for creating a service profile instance for the server.
Step 7	UCS-A /org/server-autoconfig-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a server autoconfiguration policy named AutoConfigFinance, provides a description for the policy, specifies finance as the destination organization, ServPoolQual22 as the server pool policy qualification, and ServTemp2 as the service profile template, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create server-autoconfig-policy AutoConfigFinance
UCS-A /org/server-autoconfig-policy* # set descr "Server Autoconfiguration Policy for
Finance"
UCS-A /org/server-autoconfig-policy* # set destination org finance
UCS-A /org/server-autoconfig-policy* # set qualifier ServPoolQual22
UCS-A /org/server-autoconfig-policy* # set template ServTemp2
UCS-A /org/server-autoconfig-policy* # commit-buffer
UCS-A /org/server-autoconfig-policy #
```

Deleting a Server Autoconfiguration Policy

Procedure

	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 # **delete server-autoconfig-policy** *policy-name*	Deletes the specified server autoconfiguration policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the server autoconfiguration policy named AutoConfigFinance and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # delete server-autoconfig-policy AutoConfigFinance
```

```
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Server Discovery Policies

Server Discovery Policy Overview

The server discovery policy determines how the UCS Manager reacts when you add a new UCS Blade Server and UCS Mini. If you create a server discovery policy, you can control whether the system conducts a deep discovery when a server is added to a chassis, or whether a user must first acknowledge the new server. By default, the system conducts a full discovery.

If you create a server discovery policy, the following occurs when a new server starts:

1. The server discovery policy qualification is executed against the server.
2. If the server meets the required qualifications, Cisco UCS Manager applies the following to the server:
   - Depending on the option that you select for the action, UCS Manager discovers the new server immediately, or waits for a user acknowledgment of the new server
   - Applies the scrub policy to the server

If automatic deep discovery is triggered by any hardware insertion, removal, or replacement, the following occurs:

1. The server is moved to a “pending activities” list.
2. A critical hardware mismatch fault is raised on the server, indicating that UCSM has detected a hardware mismatch.
3. User must explicitly acknowledge the server to trigger the deep discovery.

Important 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
```

If this error occurs, do the following:

1. Remove the 4K drive.
2. Reacknowledge the server.

Reacknowledging the server causes the server to reboot and results in loss of service.

Configuring a Server Discovery Policy

Before You Begin

If you plan to associate this policy with a server pool, create server pool policy qualifications.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode. **Note** Chassis discovery policies can only be accessed from the root organization.
Step 2	UCS-A /org # **create server-disc-policy** *policy-name*	Creates a server discovery policy with the specified policy name, and enters org server discovery policy mode.
Step 3	UCS-A /org/server-disc-policy # **set action** {**diag** | **immediate** | **user-acknowledged**}	Specifies when the system will attempt to discover new servers.
Step 4	UCS-A /org/chassis-disc-policy # **set descr** *description*	(Optional) Provides a description for the server discovery policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 5	UCS-A /org/server-disc-policy # **set qualifier** *qualifier*	(Optional) Uses the specified server pool policy qualifications to associates this policy with a server pool.
Step 6	UCS-A /org/server-disc-policy # **set scrub-policy**	Specifies the scrub policy to be used by this policy. The scrub policy defines whether the disk drive on a server should be scrubbed clean upon discovery.
Step 7	UCS-A /org/server-disc-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a server discovery policy named ServDiscPolExample, sets it to immediately discover new servers, provides a description for the policy, specifies the server pool policy qualifications and scrub policy, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create server-disc-policy ServDiscPolExample
UCS-A /org/server-disc-policy* # set action immediate
UCS-A /org/server-disc-policy* # set descr "This is an example server discovery policy."
UCS-A /org/server-disc-policy* # set qualifier ExampleQual
UCS-A /org/server-disc-policy* # set scrub-policy NoScrub
UCS-A /org/server-disc-policy # commit-buffer
```

What to Do Next

Include the server discovery policy in a service profile and/or template.

Deleting a Server Discovery Policy

Procedure

	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 # **Delete server-disc-policy** *policy-name*	Deletes the specified server discovery policy.
Step 3	UCS-A /org/server-disc-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the server discovery policy named ServDiscPolExample and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete server-disc-policy ServDiscPolExample
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Server Inheritance Policies

Server Inheritance Policy Overview

This policy is invoked during the server discovery process to create a service profile for the server. All service profiles created from this policy use the values burned into the blade at manufacture. The policy performs the following:

- Analyzes the inventory of the server
- If configured, assigns the server to the selected organization
- Creates a service profile for the server with the identity burned into the server at manufacture

You cannot migrate a service profile created with this policy to another server.

Configuring a Server Inheritance Policy

A blade server or rack-mount server with a VIC adapter, such as the Cisco UCS M81KR Virtual Interface Card, does not have server identity values burned into the server hardware at manufacture. As a result, the

identity of the adapter must be derived from default pools. If the default pools do not include sufficient entries for one to be assigned to the server, service profile association fails with a configuration error.

Procedure

	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 # **create server-inherit-policy** *policy-name*	Creates a server inheritance policy with the specified policy name, and enters organization server inheritance policy mode.
Step 3	UCS-A /org/server-inherit-policy # **set descr** *description*	(Optional) Provides a description for the policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/server-inherit-policy # **set destination org** *org-name*	(Optional) Specifies the organization for which the server is to be used.
Step 5	UCS-A /org/server-inherit-policy # **set qualifier** *server-qual-name*	(Optional) Specifies server pool policy qualification to use for qualifying the server.
Step 6	UCS-A /org/server-inherit-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a server inheritance policy named InheritEngineering, provides a description for the policy, specifies engineering as the destination organization and ServPoolQual22 as the server pool policy qualification, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create server-inherit-policy InheritEngineering
UCS-A /org/server-inherit-policy* # set descr "Server Inheritance Policy for Engineering"
UCS-A /org/server-inherit-policy* # set destination org engineering
UCS-A /org/server-inherit-policy* # set qualifier ServPoolQual22
UCS-A /org/server-inherit-policy* # commit-buffer
UCS-A /org/server-inherit-policy #
```

Deleting a Server Inheritance Policy

Procedure

	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 # **delete server-inherit-policy** *policy-name*	Deletes the specified server inheritance policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the server inheritance policy named InheritEngineering and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # delete server-inherit-policy InheritEngineering
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Server Pool Policies

Server Pool Policy Overview

This policy is invoked during the server discovery process. It determines what happens if server pool policy qualifications match a server to the target pool specified in the policy.

If a server qualifies for more than one pool and those pools have server pool policies, the server is added to all those pools.

Configuring a Server Pool Policy

Procedure

	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 # **create pooling-policy** *policy-name*	Creates a server pool policy with the specified name, and enters organization pooling policy mode.

	Command or Action	Purpose
Step 3	UCS-A /org/pooling-policy # **set descr** *description*	(Optional) Provides a description for the server pool policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/pooling-policy # **set pool** *pool-distinguished-name*	Specifies the server pool to use with the server pool policy. You must specify the full distinguished name for the pool.
Step 5	UCS-A /org/pooling-policy # **set qualifier** *qualifier-name*	Specifies the server pool qualifier to use with the server pool policy.
Step 6	UCS-A /org/pooling-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a server pool policy named ServerPoolPolicy4 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create pooling-policy ServerPoolPolicy4
UCS-A /org/pooling-policy* # set pool org-root/compute-pool-pool3
UCS-A /org/pooling-policy* # set qualifier ServPoolQual8
UCS-A /org/pooling-policy* # commit-buffer
UCS-A /org/pooling-policy #
```

Deleting a Server Pool Policy

Procedure

	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 # **delete pooling-policy** *policy-name*	Deletes the specified server pool policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the server pool policy named ServerPoolPolicy4 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete pooling-policy ServerPoolPolicy4
UCS-A /org/pooling-policy* # commit-buffer
UCS-A /org/pooling-policy #
```

Configuring Server Pool Policy Qualifications

Server Pool Policy Qualification Overview

This policy qualifies servers based on the inventory of a server conducted during the discovery process. The qualifications are individual rules that you configure in the policy to determine whether a server meets the selection criteria. For example, you can create a rule that specifies the minimum memory capacity for servers in a data center pool.

Qualifications are used in other policies to place servers, not just by the server pool policies. For example, if a server meets the criteria in a qualification policy, it can be added to one or more server pools or have a service profile automatically associated with it.

You can use the server pool policy qualifications to qualify servers according to the following criteria:

- Adapter type
- Chassis location
- Memory type and configuration
- Power group
- CPU cores, type, and configuration
- Storage configuration and capacity
- Server model

Depending upon the implementation, you might need to configure several policies with server pool policy qualifications including the following:

- Autoconfiguration policy
- Chassis discovery policy
- Server discovery policy
- Server inheritance policy
- Server pool policy

Creating a Server Pool Policy Qualification

Procedure

	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* .

	Command or Action	Purpose
Step 2	UCS-A /org # **create server-qual** *server-qual-name*	Creates a server pool qualification with the specified name, and enters organization server qualification mode.
Step 3	UCS-A /org/server-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a server pool qualification named ServPoolQual22 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create server-qual ServPoolQual22
UCS-A /org/server-qual* # commit-buffer
UCS-A /org/server-qual #
```

What to Do Next

Configure one or more of the following server component qualifications:

- Adapter qualification
- Chassis qualification
- Memory qualification
- Power group qualification
- Processor qualification
- Storage qualification

Deleting a Server Pool Policy Qualification

Procedure

	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 # **delete server-qual** *server-qual-name*	Deletes the specified server pool qualification.
Step 3	UCS-A /org/server-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the server pool qualification named ServPoolQual22 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # delete server-qual ServPoolQual22
```

```
UCS-A /org* # commit-buffer
UCS-A /org #
```

Creating an Adapter Qualification

Before You Begin

Create a server pool policy qualification.

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **create adapter**	Creates an adapter qualification and enters organization server qualification adapter mode.
Step 4	UCS-A /org/server-qual/adapter # **create cap-qual** *adapter-type*	Creates an adapter capacity qualification for the specified adapter type and enters organization server qualification adapter capacity qualification mode. The *adapter-type* argument can be any of the following values: • **fcoe —**Fibre Channel over Ethernet • **non-virtualized-eth-if —**Non-virtualized Ethernet interface • **non-virtualized-fc-if —**Non-virtualized Fibre Channel interface • **path-encap-consolidated —**Path encapsulation consolidated • **path-encap-virtual —**Path encapsulation virtual • **protected-eth-if —**Protected Ethernet interface • **protected-fc-if —**Protected Fibre Channel interface • **protected-fcoe —**Protected Fibre Channel over Ethernet • **virtualized-eth-if —**Virtualized Ethernet interface • **virtualized-fc-if —**Virtualized Fibre Channel interface • **virtualized-scsi-if —**Virtualized SCSI interface
Step 5	UCS-A /org/server-qual/adapter/cap-qual # **set maximum** {*max-cap* | **unspecified**}	Specifies the maximum capacity for the selected adapter type.

	Command or Action	Purpose
Step 6	UCS-A /org/server-qual/adapter/cap-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates and configures an adapter qualification for a non-virtualized Ethernet interface and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # create adapter
UCS-A /org/server-qual/adapter* # create cap-qual non-virtualized-eth-if
UCS-A /org/server-qual/adapter/cap-qual* # set maximum 2500000000
UCS-A /org/server-qual/adapter/cap-qual* # commit-buffer
UCS-A /org/server-qual/adapter/cap-qual #
```

Deleting an Adapter Qualification

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **delete adapter**	Deletes the adapter qualification from the server pool policy qualification.
Step 4	UCS-A /org/server-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the adapter qualification from the server pool policy qualification named ServPoolQual22 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # delete adapter
UCS-A /org/server-qual* # commit-buffer
UCS-A /org/server-qual #
```

Configuring a Chassis Qualification

Before You Begin

Create a server pool policy qualification.

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **create chassis** *min-chassis-num max-chassis-num*	Creates a chassis qualification for the specified chassis range and enters organization server qualification chassis mode.
Step 4	UCS-A /org/server-qual/chassis # **create slot** *min-slot-num max-slot-num*	Creates a chassis slot qualification for the specified slot range and enters organization server qualification chassis slot mode.
Step 5	UCS-A /org/server-qual/chassis/slot # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a chassis qualification for slots 1 to 4 on chassis 1 and 2 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope server-qual ServPoolQual22
UCS-A /org/server-qual* # create chassis 1 2
UCS-A /org/server-qual/chassis* # create slot 1 4
UCS-A /org/server-qual/chassis/slot* # commit-buffer
UCS-A /org/server-qual/chassis/slot #
```

Deleting a Chassis Qualification

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **delete chassis** *min-chassis-num max-chassis-num*	Deletes the chassis qualification for the specified chassis range.
Step 4	UCS-A /org/server-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the chassis qualification for chassis 1 and 2 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # delete chassis 1 2
UCS-A /org/server-qual* # commit-buffer
UCS-A /org/server-qual #
```

Creating a CPU Qualification

Before You Begin

Create a server pool policy qualification.

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **create cpu**	Creates a CPU qualification and enters organization server qualification processor mode.
Step 4	UCS-A /org/server-qual/cpu # **set arch** {**any** | **dual-core-opteron** | **intel-p4-c** | **opteron** | **pentium-4** | **turion-64** | **xeon** | **xeon-mp**}	Specifies the processor architecture type.
Step 5	UCS-A /org/server-qual/cpu # **set maxcores** {*max-core-num* | **unspecified**}	Specifies the maximum number of processor cores.
Step 6	UCS-A /org/server-qual/cpu # **set mincores** {*min-core-num* | **unspecified**}	Specifies the minimum number of processor cores.
Step 7	UCS-A /org/server-qual/cpu # **set maxprocs** {*max-proc-num* | **unspecified**}	Specifies the maximum number of processors.
Step 8	UCS-A /org/server-qual/cpu # **set minprocs** {*min-proc-num* | **unspecified**}	Specifies the minimum number of processors.
Step 9	UCS-A /org/server-qual/cpu # **set maxthreads** {*max-thread-num* | **unspecified**}	Specifies the maximum number of threads.
Step 10	UCS-A /org/server-qual/cpu # **set minthreads** {*min-thread-num* | **unspecified**}	Specifies the minimum number of threads.
Step 11	UCS-A /org/server-qual/cpu # **set stepping** {*step-num* | **unspecified**}	Specifies the processor stepping number.

	Command or Action	Purpose
Step 12	UCS-A /org/server-qual/cpu # **set model-regex** *regex*	Specifies a regular expression that the processor name must match.
Step 13	UCS-A /org/server-qual/cpu # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates and configures a CPU qualification and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # create processor
UCS-A /org/server-qual/cpu* # set arch xeon
UCS-A /org/server-qual/cpu* # set maxcores 8
UCS-A /org/server-qual/cpu* # set mincores 4
UCS-A /org/server-qual/cpu* # set maxprocs 2
UCS-A /org/server-qual/cpu* # set minprocs 1
UCS-A /org/server-qual/cpu* # set maxthreads 16
UCS-A /org/server-qual/cpu* # set minthreads 8
UCS-A /org/server-qual/cpu* # set stepping 5
UCS-A /org/server-qual/cpu* # commit-buffer
UCS-A /org/server-qual/cpu #
```

Deleting a CPU Qualification

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **delete cpu**	Deletes the processor qualification.
Step 4	UCS-A /org/server-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the processor qualification and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # delete cpu
UCS-A /org/server-qual* # commit-buffer
UCS-A /org/server-qual #
```

Creating a Power Group Qualification

Before You Begin

Create a server pool policy qualification.

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **create power-group** *power-group-name*	Creates a power group qualification for the specified power group name.
Step 4	UCS-A /org/server-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a power group qualification for a power group called powergroup1 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # create power-group powergroup1
UCS-A /org/server-qual* # commit-buffer
UCS-A /org/server-qual #
```

Deleting a Power Group Qualification

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **delete power-group** *power-group-name*	Deletes the specified power group qualification.
Step 4	UCS-A /org/server-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a power group qualification for a power group called powergroup1 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # delete power-group powergroup1
UCS-A /org/server-qual* # commit-buffer
UCS-A /org/server-qual #
```

Creating a Memory Qualification

Before You Begin

Create a server pool policy qualification.

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **create memory**	Creates a memory qualification and enters organization server qualification memory mode.
Step 4	UCS-A /org/server-qual/memory # **set clock** {*clock-num* | **unspec**}	Specifies the memory clock speed.
Step 5	UCS-A /org/server-qual/memory # **set maxcap** {*max-cap-num* | **unspec**}	Specifies the maximum capacity of the memory array.
Step 6	UCS-A /org/server-qual/memory # **set mincap** {*min-cap-num* | **unspec**}	Specifies the minimum capacity of the memory array.
Step 7	UCS-A /org/server-qual/memory # **set speed** {*speed-num* | **unspec**}	Specifies the memory data rate.
Step 8	UCS-A /org/server-qual/memory # **set units** {*unit-num* | **unspec**}	Specifies the number of memory units (DRAM chips mounted to the memory board).
Step 9	UCS-A /org/server-qual/memory # **set width** {*width-num* | **unspec**}	Specifies the bit width of the data bus.
Step 10	UCS-A /org/server-qual/memory # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates and configures a memory qualification and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # create memory
UCS-A /org/server-qual/memory* # set clock 1067
UCS-A /org/server-qual/memory* # set maxcap 4096
UCS-A /org/server-qual/memory* # set mincap 2048
UCS-A /org/server-qual/memory* # set speed unspec
UCS-A /org/server-qual/memory* # set units 16
UCS-A /org/server-qual/memory* # set width 64
UCS-A /org/server-qual/memory* # commit-buffer
UCS-A /org/server-qual/memory #
```

Deleting a Memory Qualification

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **delete memory**	Deletes the memory qualification.
Step 4	UCS-A /org/server-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the memory qualification and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # delete memory
UCS-A /org/server-qual* # commit-buffer
UCS-A /org/server-qual #
```

Creating a Physical Qualification

Before You Begin

Create a server pool policy qualification.

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **create physical-qual**	Creates a physical qualification and enters organization server qualification physical mode.
Step 4	UCS-A /org/server-qual/physical-qual # **set model-regex** *regex*	Specifies a regular expression that the model name must match.
Step 5	UCS-A /org/server-qual/physical-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates and configures a physical qualification and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # create physical-qual
UCS-A /org/server-qual/physical-qual* # set model-regex
UCS-A /org/server-qual/physical-qual* # commit-buffer
UCS-A /org/server-qual/physical-qual #
```

Deleting a Physical Qualification

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **delete physical-qual**	Deletes the physical qualification.
Step 4	UCS-A /org/server-qual # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a physical qualification and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
```

```
UCS-A /org/server-qual # delete physical-qual
UCS-A /org/server-qual* # commit-buffer
UCS-A /org/server-qual #
```

Creating a Storage Qualification

Before You Begin

Create a server pool policy qualification.

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **create storage**	Creates a storage qualification and enters organization server qualification storage mode.
Step 4	UCS-A /org/server-qual/storage # **set blocksize** {*block-size-num* | **unknown**}	Specifies the storage block size.
Step 5	UCS-A /org/server-qual/storage # **set diskless** {**no** | **unspecified** | **yes** }	Specifies whether the available storage must be diskless.
Step 6	UCS-A /org/server-qual/storage # **set disktype** {**hdd** | **ssd** | **unspecified**}	Specifies the type of disk that can be used. The options are: • **Unspecified—**Either disk type is acceptable. • **HDD—**The disk must be HDD. • **SSD—**The disk must be SSD (SATA or SAS).
Step 7	UCS-A /org/server-qual/storage # **set flexflash-num-cards** {*ff_card-num* | **unknown**}	Specifies the number of FlexFlash cards.
Step 8	UCS-A /org/server-qual/storage # **set maxcap** {*max-cap-num* | **unknown**}	Specifies the maximum capacity of the storage array.
Step 9	UCS-A /org/server-qual/storage # **set mincap** {*min-cap-num* | **unknown**}	Specifies the minimum capacity of the storage array.
Step 10	UCS-A /org/server-qual/storage # **set numberofblocks** {*block-num* | **unknown**}	Specifies the number of blocks.

	Command or Action	Purpose
Step 11	UCS-A /org/server-qual/storage # **set perdiskcap** {*disk-cap-num* | **unknown**}	Specifies the per-disk capacity.
Step 12	UCS-A /org/server-qual/storage # **set units** {*unit-num* | **unspecified**}	Specifies the number of storage units.
Step 13	UCS-A /org/server-qual/storage # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create and configure a storage qualification and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # create storage
UCS-A /org/server-qual/storage* # set blocksize 512
UCS-A /org/server-qual/storage* # set disktype hdd
UCS-A /org/server-qual/storage* # set maxcap 420000
UCS-A /org/server-qual/storage* # set mincap 140000
UCS-A /org/server-qual/storage* # set numberofblocks 287277984
UCS-A /org/server-qual/storage* # set perdiskcap 140000
UCS-A /org/server-qual/storage* # set units 1
UCS-A /org/server-qual/storage* # set flexflash-num-cards 2
UCS-A /org/server-qual/storage* # commit-buffer
UCS-A /org/server-qual/storage #
```

Deleting a Storage Qualification

Procedure

	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 server-qual** *server-qual-name*	Enters organization server qualification mode for the specified server pool policy qualification.
Step 3	UCS-A /org/server-qual # **delete storage**	Deletes the storage qualification.
Step 4	UCS-A /org/server-qual/ # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the storage qualification and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope server-qual ServPoolQual22
UCS-A /org/server-qual # delete storage
UCS-A /org/server-qual* # commit-buffer
UCS-A /org/server-qual #
```

Configuring vNIC/vHBA Placement Policies

vNIC/vHBA Placement Policies

vNIC/vHBA placement policies are used to determine the following:

- How the virtual network interface connections (vCons) are mapped to the physical adapters on a server.
- What types of vNICs or vHBAs can be assigned to each vCon.

Each vNIC/vHBA placement policy contains four vCons that are virtual representations of the physical adapters. When a vNIC/vHBA placement policy is assigned to a service profile, and the service profile is associated with a server, the vCons in the vNIC/vHBA placement policy are assigned to the physical adapters and the vNICs and vHBAs are assigned to those vCons.

For blade or rack servers that contain one adapter, Cisco UCS assigns all vCons to that adapter. For servers that contain four adapters, Cisco UCS assigns vCon1 to Adapter1, vCon2 to Adapter2, vCon3 to Adapter3, and vCon4 to Adapter4.

For blade or rack servers that contain two or three adapters, Cisco UCS assigns the vCons based on the type of server and the selected virtual slot mapping scheme, which can be **Round Robin** or **Linear Ordered**. For details about the available mapping schemes, see vCon to Adapter Placement, on page 504.

After Cisco UCS assigns the vCons, it assigns the vNICs and vHBAs based on the **Selection Preference** for each vCon. This can be one of the following:

Note You can specify the PCI order for the vHBA; however, the desired order works within a class of devices, such as vNICs or vHBAs and not across them. Within an adapter, vNICs are always placed ahead of the vHBAs.

- **all—**All configured vNICs and vHBAs can be assigned to the vCon, whether they are explicitly assigned to it, unassigned, or dynamic. This is the default.
- **assigned-only—**vNICs and vHBAs must be explicitly assigned to the vCon. You can assign them explicitly through the service profile or the properties of the vNIC or vHBA.
- **exclude-dynamic—**Dynamic vNICs and vHBAs cannot be assigned to the vCon. The vCon can be used for all static vNICs and vHBAs, whether they are unassigned or explicitly assigned to it.
- **exclude-unassigned—**Unassigned vNICs and vHBAs cannot be assigned to the vCon. The vCon can be used for dynamic vNICs and vHBAs and for static vNICs and vHBAs that are explicitly assigned to it.
- **exclude-usnic—**Cisco usNICs cannot be assigned to the vCon. The vCon can be used for all other configured vNICs and vHBAs, whether they are explicitly assigned to it, unassigned, or dynamic.

Note An SRIOV usNIC that is explicitly assigned to a vCon set to **exclude-usnic** will remain assigned to that vCon.

If you do not include a vNIC/vHBA placement policy in the service profile, Cisco UCS Manager defaults to the **Round Robin** vCon mapping scheme and the **All** vNIC/vHBA selection preference, distributing the vNICs and vHBAs between the adapters based on the capabilities and relative capacities of each adapter.

vCon to Adapter Placement

Cisco UCS maps every vCon in a service profile to a physical adapter on the server. How that mapping occurs and how the vCons are assigned to a specific adapter in a server depends on the following:

- The type of server. N20-B6620-2 and N20-B6625-2 blade servers with two adapter cards use a different mapping scheme than other supported rack or blade servers.
- The number of adapters in the server.
- The setting of the virtual slot mapping scheme in the vNIC/vHBA placement policy, if applicable.

You must consider this placement when you configure the vNIC/vHBA selection preference to assign vNICs and vHBAs to vCons.

Note vCon to adapter placement is not dependent upon the PCIE slot number of the adapter. The adapter numbers used for the purpose of vCon placement are not the PCIE slot numbers of the adapters, but the ID assigned to them during server discovery.

vCon to Adapter Placement for N20-B6620-2 and N20-B6625-2 Blade Servers

In N20-B6620-2 and N20-B6625-2 blade servers, the two adapters are numbered left to right while vCons are numbered right to left. If one of these blade servers has a single adapter, Cisco UCS assigns all vCons to that adapter. If the server has two adapters, the vCon assignment depends upon the virtual slot mapping scheme:

- **round-robin—**Cisco UCS assigns vCon2 and vCon4 to Adapter1 and vCon1 and vCon3 to Adapter2. This is the default.
- **linear-ordered—**Cisco UCS assigns vCon3 and vCon4 to Adapter1 and vCon1 and vCon2 to Adapter2.

vCon to Adapter Placement for All Other Supported Servers

For all other servers supported by Cisco UCS in addition to the N20-B6620-2 and N20-B6625-2 blade servers, the vCon assignment depends on the number of adapters in the server and the virtual slot mapping scheme.

For blade or rack servers that contain one adapter, Cisco UCS assigns all vCons to that adapter. For servers that contain four adapters, Cisco UCS assigns vCon1 to Adapter1, vCon2 to Adapter2, vCon3 to Adapter3, and vCon4 to Adapter4.

For blade or rack servers that contain two or three adapters, Cisco UCS assigns the vCons based on the selected virtual slot mapping scheme: Round Robin or Linear Ordered.

Table 10: vCon to Adapter Placement Using the Round - Robin Mapping Scheme

Number of Adapters	vCon1 Assignment	vCon2 Assignment	vCon3 Assignment	vCon4 Assignment
1	Adapter1	Adapter1	Adapter1	Adapter1
2	Adapter1	Adapter2	Adapter1	Adapter2
3	Adapter1	Adapter2	Adapter3	Adapter2
4	Adapter1	Adapter2	Adapter3	Adapter4

Round Robin is the default mapping scheme.

Table 11: vCon to Adapter Placement Using the Linear Ordered Mapping Scheme

Number of Adapters	vCon1 Assignment	vCon2 Assignment	vCon3 Assignment	vCon4 Assignment
1	Adapter1	Adapter1	Adapter1	Adapter1
2	Adapter1	Adapter1	Adapter2	Adapter2
3	Adapter1	Adapter2	Adapter3	Adapter3
4	Adapter1	Adapter2	Adapter3	Adapter4

Note If you are using a vCon policy with two adapters in the Cisco UCS B440 M2 Blade Server, be aware of the following mapping.

- vCon 2 to adapter 1 maps first
- vCon 1 to adapter 2 maps second ZXA Q

vNIC/vHBA to vCon Assignment

Cisco UCS Manager provides two options for assigning vNICs and vHBAs to vCons through the vNIC/vHBA placement policy: explicit assignment and implicit assignment.

Explicit Assignment of vNICs and vHBAs

With explicit assignment, you specify the vCon and, therefore, the adapter to which a vNIC or vHBA is assigned. Use this assignment option when you need to determine how the vNICs and vHBAs are distributed between the adapters on a server.

To configure a vCon and the associated vNICs and vHBAs for explicit assignment, do the following:

- Set the vCon configuration to any of the available options. You can configure the vCons through a vNIC/vHBA placement policy or in the service profile associated with the server. If a vCon is configured for **All**, you can still explicitly assign a vNIC or vHBA to that vCon.
- Assign the vNICs and vHBAs to a vCon. You can make this assignment through the virtual host interface placement properties of the vNIC or vHBA or in the service profile associated with the server.

If you attempt to assign a vNIC or vHBA to a vCon that is not configured for that type of vNIC or vHBA, Cisco UCS Manager displays a message advising you of the configuration error.

During service profile association, Cisco UCS Manager validates the configured placement of the vNICs and vHBAs against the number and capabilities of the physical adapters in the server before assigning the vNICs and vHBAs according to the configuration in the policy. Load distribution is based upon the explicit assignments to the vCons and adapters configured in this policy.

If the adapters do not support the assignment of one or more vNICs or vHBAs, Cisco UCS Manager raises a fault against the service profile.

Note You can specify the PCI order for the vHBA; however, the desired order works within a class of devices, such as vNICs or vHBAs and not across them. Within an adapter, vNICs are always placed ahead of the vHBAs.

Implicit Assignment of vNICs and vHBAs

With implicit assignment, Cisco UCS Manager determines the vCon and, therefore, the adapter to which a vNIC or vHBA is assigned according to the capability of the adapters and their relative capacity. Use this assignment option if the adapter to which a vNIC or vHBA is assigned is not important to your system configuration.

To configure a vCon for implicit assignment, do the following:

- Set the vCon configuration to **All**, **Exclude Dynamic**, or **Exclude Unassigned**. You can configure the vCons through a vNIC/vHBA placement policy or in the service profile associated with the server.
- Do not set the vCon configuration to **Assigned Only**. Implicit assignment cannot be performed with this setting.
- Do not assign any vNICs or vHBAs to a vCon.

During service profile association, Cisco UCS Manager verifies the number and capabilities of the physical adapters in the server and assigns the vNICs and vHBAs accordingly. Load distribution is based upon the capabilities of the adapters, and placement of the vNICs and vHBAs is performed according to the actual order determined by the system. For example, if one adapter can accommodate more vNICs than another, that adapter is assigned more vNICs.

If the adapters cannot support the number of vNICs and vHBAs configured for that server, Cisco UCS Manager raises a fault against the service profile.

Implicit Assignment of vNICs in a Dual Adapter Environment

When you use implicit vNIC assignment for a dual slot server with an adapter card in each slot, Cisco UCS Manager typically assigns the vNICs/vHBAs as follows:

- If the server has the same adapter in both slots, Cisco UCS Manager assigns half the vNICs and half the vHBAs to each adapter.
- If the server has one non-VIC adapter and one VIC adapter, Cisco UCS Manager assigns two vNICs and two vHBAs to the non-VIC adapter and the remaining vNICs and vHBAs to the VIC adapter.
- If the server has two different VIC adapters, Cisco UCS Manager assigns the vNICs and vHBAs proportionally, based on the relative capabilities of the two adapters.

The following examples show how Cisco UCS Manager would typically assign the vNICs and vHBAs with different combinations of supported adapter cards:

- If you want to configure four vNICs and the server contains two Cisco UCS M51KR-B Broadcom BCM57711 adapters (with two vNICs each), Cisco UCS Manager assigns two vNICs to each adapter.
- If you want to configure 50 vNICs and the server contains a Cisco UCS CNA M72KR-E adapter (2 vNICs) and a Cisco UCS M81KR Virtual Interface Card adapter (128 vNICs), Cisco UCS Manager assigns two vNICs to the Cisco UCS CNA M72KR-E adapter and 48 vNICs to the Cisco UCS M81KR Virtual Interface Card adapter.
- If you want to configure 150 vNICs and the server contains a Cisco UCS M81KR Virtual Interface Card adapter (128 vNICs) and a Cisco UCS VIC-1240 Virtual Interface Card adapter (256 vNICs), Cisco UCS Manager assigns 50 vNICs to the Cisco UCS M81KR Virtual Interface Card adapter and 100 vNICs to the Cisco UCS VIC-1240 Virtual Interface Card adapter.

Note Exceptions to this implicit assignment occur if you configure the vNICs for fabric failover and if you configure dynamic vNICs for the server.

For a configuration that includes vNIC fabric failover where one adapter does not support vNIC failover, Cisco UCS Manager implicitly assigns all vNICs that have fabric failover enabled to the adapter that supports them. If the configuration includes only vNICs that are configured for fabric failover, no vNICs are implicitly assigned to the adapter that does not support them. If some vNICs are configured for fabric failover and some are not, Cisco UCS Manager assigns all failover vNICs to the adapter that supports them and a minimum of one nonfailover vNIC to the adapter that does not support them, according to the ratio above.

For a configuration that includes dynamic vNICs, the same implicit assignment would occur. Cisco UCS Manager assigns all dynamic vNICs to the adapter that supports them. However, with a combination of dynamic vNICs and static vNICs, at least one static vNIC is assigned to the adapter that does not support dynamic vNICs.

Configuring a vNIC/vHBA Placement Policy

Procedure

	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*.

	Command or Action	Purpose
Step 2	UCS-A /org # **create vcon-policy** *policy-name*	Creates the specified vNIC/vHBA placement profile and enters organization vcon policy mode.
Step 3	UCS-A /org/vcon-policy # **set descr** *description*	(Optional) Provides a description for the vNIC/vHBA Placement Profile. Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote). **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/vcon-policy # **set mapping-scheme** {**round-robin** | **linear-ordered**}	(Optional) For blade or rack servers that contain one adapter, Cisco UCS assigns all vCons to that adapter. For servers that contain four adapters, Cisco UCS assigns vCon1 to Adapter1, vCon2 to Adapter2, vCon3 to Adapter3, and vCon4 to Adapter4. For blade or rack servers that contain two or three adapters, Cisco UCS assigns the vCons based on the selected virtual slot mapping scheme. This can be one of the following: • **round-robin—** In a server with two adapter cards, Cisco UCS assigns vCon1 and vCon3 to Adapter1, then assigns vCon2 and vCon4 to Adapter2. In a server with three adapter cards, Cisco UCS assigns vCon1 to Adapter1, vCon2 and vCon4 to Adapter2, and vCon3 to Adapter3. This is the default scheme. • **linear-ordered—** In a server with two adapter cards, Cisco UCS assigns vCon1 and vCon2 to Adapter1, then assigns vCon3 and vCon4 to Adapter2. In a server with three adapter cards, Cisco UCS assigns vCon1 to Adapter1 and vCon2 to Adapter2, then assigns vCon3 and vCon4 to Adapter3. In N20-B6620-2 and N20-B6625-2 blade servers, the two adapters are numbered left to right while vCons are numbered right to left. If one of these blade servers has a single adapter, Cisco UCS assigns all vCons to that adapter. If the server has two adapters, the vCon assignment depends upon the virtual slot mapping scheme: • **round-robin—**Cisco UCS assigns vCon2 and vCon4 to Adapter1 and vCon1 and vCon3 to Adapter2. This is the default. • **linear-ordered—**Cisco UCS assigns vCon3 and vCon4 to Adapter1 and vCon1 and vCon2 to Adapter2.

	Command or Action	Purpose
Step 5	UCS-A /org/vcon-policy # **set vcon** {**1** | **2** | **3** | **4**} **selection** {**all** | **assigned-only** | **exclude-dynamic** | **exclude-unassigned**}	Specifies the selection preference for the specified vCon. The options are: • **all—**All configured vNICs and vHBAs can be assigned to the vCon, whether they are explicitly assigned to it, unassigned, or dynamic. This is the default. • **assigned-only—**vNICs and vHBAs must be explicitly assigned to the vCon. You can assign them explicitly through the service profile or the properties of the vNIC or vHBA. • **exclude-dynamic—**Dynamic vNICs and vHBAs cannot be assigned to the vCon. The vCon can be used for all static vNICs and vHBAs, whether they are unassigned or explicitly assigned to it. • **exclude-unassigned—**Unassigned vNICs and vHBAs cannot be assigned to the vCon. The vCon can be used for dynamic vNICs and vHBAs and for static vNICs and vHBAs that are explicitly assigned to it. • **exclude-usnic—**Cisco usNICs cannot be assigned to the vCon. The vCon can be used for all other configured vNICs and vHBAs, whether they are explicitly assigned to it, unassigned, or dynamic. **Note** An SRIOV usNIC that is explicitly assigned to a vCon set to **exclude-usnic** will remain assigned to that vCon.
Step 6	UCS-A /org/vcon-policy # **commit-buffer**	Commits the transaction.

The following example creates a vNIC/vHBA placement policy named Adapter1All, sets the vCon mapping scheme to Linear Ordered, specifies that only assigned vNICs and vHBAs can be placed on adapter 1, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create vcon-policy Adapter1
UCS-A /org/vcon-policy* # set descr "This profile places all vNICs and vHBAs on adapter 1."
UCS-A /org/vcon-policy* # set mapping-scheme linear-ordered
UCS-A /org/vcon-policy* # set vcon 1 selection assigned-only
UCS-A /org/vcon-policy* # commit-buffer
UCS-A /org/vcon-policy* #
UCS-A /org #
```

Deleting a vNIC/vHBA Placement Policy

Procedure

	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 # **delete vcon-policy** *policy-name*	Deletes the specified vNIC/vHBA placement profile.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction.

The following example deletes the vNIC/vHBA placement profile named Adapter1All and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete vcon-policy Adapter1All
UCS-A /org* # commit-buffer
UCS-A /org #
```

Explicitly Assigning a vNIC to a vCon

Before You Begin

Configure the vCons through a vNIC/vHBA placement policy or in the service profile with one of the following values:

- **Assigned Only**
- **Exclude Dynamic**
- **Exclude Unassigned**

If a vCon is configured for **All**, you can explicitly assign a vNIC or vHBA to that vCon. However, there is less control with this configuration.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the organization which contains the service profile whose vNICs you want to explicitly assign to a vCon. To enter the root organization mode, type / as the *org-name* .

	Command or Action	Purpose
Step 2	UCS-A /org # **scope service-profile** *profile-name*	Enters organization service profile mode for the specified service.
Step 3	UCS-A /org/service-profile # **scope vnic** *vnic-name*	Enters organization service profile mode for the specified vnic.
Step 4	UCS-A /org/service-profile/vnic # **set vcon** {**1** | **2** | **3** | **4** | **any**}	Sets the virtual network interface connection (vCon) placement for the specified vNIC. Entering a value of any allows Cisco UCS Manager to determine the vCon to which the vNIC is assigned.
Step 5	UCS-A /org/service-profile/vnic # **set order** {*order-num* | **unspecified**}	Specifies the desired PCI order for the vNIC. Valid values include 0-128 and unspecified.
Step 6	UCS-A /org/service-profile/vnic # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the vCon placement for a vNIC called vnic3 to 2, sets the desired order to 10, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope vnic vnic3
UCS-A /org/service-profile/vnic # set vcon 2
UCS-A /org/service-profile/vnic* # set order 10
UCS-A /org/service-profile/vnic* # commit-buffer
UCS-A /org/service-profile/vnic #
```

Explicitly Assigning a vHBA to a vCon

Before You Begin

Configure the vCons through a vNIC/vHBA placement policy or in the service profile with one of the following values:

- **Assigned Only**
- **Exclude Dynamic**
- **Exclude Unassigned**

If a vCon is configured for **All**, you can explicitly assign a vNIC or vHBA to that vCon. However, there is less control with this configuration.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the organization which contains the service profile whose vHBAs you want to

	Command or Action	Purpose
		explicitly assign to a vCon. 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.
Step 3	UCS-A /org/service-profile # **scope vhba** *vhba-name*	Enters organization service profile mode for the specified vHBA.
Step 4	UCS-A /org/service-profile/vhba # **set vcon** {**1** | **2** | **3** | **4** | **any**}	Sets the virtual network interface connection (vCon) placement for the specified vHBA. Entering a value of any allows Cisco UCS Manager to determine the vCon to which the vHBA is assigned.
Step 5	UCS-A /org/service-profile/vhba # **set order** {*order-num* | **unspecified**}	Specifies the desired PCI order for the vHBA. Valid desired order number values include 0-128 and unspecified.
Step 6	UCS-A /org/service-profile/vhba # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the vCon placement for a vHBA called vhba3 to 2, sets the desired order to 10, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope vhba vhba3
UCS-A /org/service-profile/vhba # set vcon 2
UCS-A /org/service-profile/vhba* # set order 10
UCS-A /org/service-profile/vhba* # commit-buffer
UCS-A /org/service-profile/vhba #
```

Placing Static vNICs Before Dynamic vNICs

For optimal performance, static vNICs and vHBAs should be placed before dynamic vNICs on the PCIe bus. Static vNICs refer to both static vNICs and vHBAs. Cisco UCS Manager Release 2.1 provides the following functionality regarding the order of static and dynamic vNICs:

- After upgrading to Cisco UCS Manager Release 2.1, if no change is made to existing service profiles (profiles that are defined in releases prior to Cisco UCS Manager Release 2.1), the vNIC order does not change.
- After an upgrade to Cisco UCS Manager Release 2.1, any vNIC-related change would reorder the vNIC map. As a result, all dynamic vNICs would be placed after the static vNICs.
- For newly created service profiles in Cisco UCS Manager Release 2.1, static vNICs are always ordered before dynamic vNICs.
- The above behavior is independent of the sequence of creating or deleting static or dynamic vNICs.

- For SRIOV-enabled service profiles, UCSM places the vNIC Physical Function(PF) before the corresponding Virtual Functions (VFs). This scheme guarantees that the VFs are placed close to the parent PF vNIC on the PCIe bus and BDFs are in successive incremental order for the VFs.

Example

Beginning Device Order in Cisco UCS Manager Release 2.0:

```
dyn-vNIC-1 1
dyn-vNIC-2 2
```

New Device Order in Cisco UCS Manager Release 2.0 (Add 2 static vNICs):

```
dyn-vNIC-1 1
dyn-vNIC-2 2
eth-vNIC-1 3
eth-vNIC-2 4
```

After upgrading to Cisco UCS Manager Release 2.1, (Before any vNIC-related change is made to the service profile.)

```
dyn-vNIC-1 1
dyn-vNIC-2 2
eth-vNIC-1 3
eth-vNIC-2 4
```

New Device Order in Cisco UCS Manager Release 2.1 (Add 2 dynamic vNICs by changing the policy count from 2 to 4.)

```
dyn-vNIC-1 3
dyn-vNIC-2 4
eth-vNIC-1 1
eth-vNIC-2 2
dyn-vNIC-3 5
dyn-vNIC-4 6
```

Dynamic vNICs as Multifunction PCIe Devices

Cisco UCS Manager Version 2.1 provisions static vNICs as 0-function devices (new BUS for every static vNIC). Multifunction dynamic vNICs are placed from the new Bus-slot after the last static vNIC/vHBA.

Note Cisco UCS Manager Version 2.1 supports the new StaticZero mode.

Table 12: Version Compatibility

Cisco UCS Manager		
Version 1.4 **Scheme: ZeroFunction**	**Version 2.0** **Scheme: ZeroFunction / MultiFunction**	**Version 2.1** **Scheme: ZeroFunction / MultiFunction / StaticZero**
Static and Dynamic vNICs are all on Bus [0-57], Function [0] < ZeroFunction Mode >	Static vNICs and Dynamic vNICs are on Bus [0-57], Function [0-7]. Bus 0, Function 0 Bus 0, Function 7 Bus 1, Function 0 < MultiFunction Mode >	Static vNICs or PFs will be on Bus [0-57], Function [0]. SRIOV: Corresponding VFs will be on the same Bus and Functions [1-255] No-SRIOV: Dynamic vNICs are on Bus [0-57], Function [0-7] < StaticZero Mode >

Cisco UCS Manager		
Version 1.4 **Scheme: ZeroFunction**	**Version 2.0** **Scheme: ZeroFunction / MultiFunction**	**Version 2.1** **Scheme: ZeroFunction / MultiFunction / StaticZero**
	Upgrade from Balboa will not renumber BDFs (remain in ZeroFunction mode) until Bus <= 57. Once devices exceed 58, switch to MultiFunction mode.	Upgrade from Balboa will not renumber BDFs (remain in ZeroFunction mode) until Bus <=57. Once devices exceed 58 or Platform specific maximum PCIe Bus number or change to SRIOV configuration, switch to StaticZero mode.
		Upgrade from Cisco UCS Manager Version 2.0 will not renumber BDFs (remain in ZeroFunction / MultiFunction mode). Once devices exceed 58 or Platfor specific maximum PCIe Bus number OR Change to SRIOV configuration, switch to StaticZero mode.

vNIC/vHBA Host Port Placement

After a vNIC/vHBA is assigned to a vCON, it can be placed on one of the host ports of specific adapters. You can either explicitly specify the host port for placement, or allow Cisco UCS Manager to automatically assign vNICs/vHBAs to host ports.

Note You can perform vNIC/vHBA host port placement on servers that support Cisco UCS VIC 1340 and VIC 1380 adapters.

The host port placement of the vNIC/vHBA determines the order of the vNIC/vHBA on the adapter. The vNICs/vHBAs placed on the first host port will be enumerated first, followed by the vNICs/vHBAs on the second host port.

Configuring Host Port Placement

You can configure host port placement for vNICs on servers that support Cisco UCS VIC 1340 and VIC 1380 adapters.

Procedure

	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** *profile-name*	Enters service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # **scope vnic** *vnic-name*	Enters organization service profile mode for the specified vNIC.
Step 4	UCS-A /org/service-profile/vnic # **set host-port** {1 | 2 | any}	Sets the host port for the specified vNIC. Entering a value of **any** allows Cisco UCS Manager to determine the host port to which the vNIC is assigned. If you set the host port for a vNIC on an adapter that does not support host port placement, the **Actual Host Port** parameter displays **None**.
Step 5	UCS-A /org/service-profile/vnic* # **commit-buffer**	Commits the transaction to the system configuration.
Step 6	UCS-A /org/service-profile/vnic # **show detail**	Displays details about the specified vNIC.

The following example places a vNIC called vnic3 to host port 2, commits the transaction, and displays the host port information:

```
UCS-A# scope org
UCS-A /org # scope service-profile SP-2
UCS-A /org/service-profile # scope vnic vnic3
UCS-A /org/service-profile/vnic # set host-port 2
UCS-A /org/service-profile/vnic* # commit-buffer
UCS-A /org/service-profile/vnic # show detail
vNIC:
    Name: vnic3
    Fabric ID: A
    Dynamic MAC Addr: 00:25:B5:13:13:11
    Desired Order: 2
    Actual Order: 3
    Desired VCon Placement: 1
    Actual VCon Placement: 1
    Desired Host Port: 2
    Actual Host Port: 2
...

UCS-A /org/service-profile/vnic #
```

CIMC Mounted vMedia

Using Scriptable vMedia

Cisco UCS Manager allows provisioning of vMedia devices iso images for remote UCS servers. Using Scriptable vMedia, you can programmatically mount an IMG or an ISO image on a remote server. CIMC mounted vMedia provide communications between other mounted media inside your datacenter with no additional requirements media connection. Scriptable vMedia allows you to control virtual media devices without using a browser to manually map each UCS server individually.

Scriptable vMedia supports multiple share types including NFS, CIFS, HTTP, and HTTPS shares. **Scriptable vMedia** is enabled through BIOS configuration and configured through a Web GUI and CLI interface.

Cisco UCS Manager Scriptable vMedia supports the following functionality:

- Booting from a specific vMedia device
- Copying files from a mounted share to a local disk
- Installation and updating OS drivers

Note Cisco UCS Manager support for Scriptable vMedia is applicable for CIMC mapped devices only. Existing KVM based vMedia devices are not supported.

vMedia mount fails when the following conditions are met:

1. The remote vMedia image filename in the vMedia policy is set to **Service-Profile-Name**.
2. The service profile is renamed.

This is because the change in the name of the service profile does not change the remote vMedia image filename in the vMedia policy. The image filename still points to the older image on the remote device, which cannot be found.

Note Cisco UCS B200M2 Blade Server and Cisco UCS B230M2 Blade Server cannot use a vMedia policy as the policy is not supported on these blade servers.

Creating a CIMC vMedia Policy

Before You Begin

Make sure that you have access to the following:

- Remote vMedia Server
- vMedia Devices

Procedure

	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 # **create vmedia-policy** *policy-name*	Creates a vMedia policy with the specified policy name. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	UCS-A /org/vmedia-policy* # **create vmedia-mapping** *mapping -name*	Creates a vMedia policy sub-directory with the specified mapping name.
Step 4	UCS-A /org/vmedia-policy/vmedia-mapping # **set descr** *description*	(Optional) Provides a description for the vMedia policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 5	UCS-A /org/vmedia-policy/vmedia-mapping* # **set device type** *device-type*	Specifies the remote vMedia image type you wish to mount. Options are: • **CDD** - Scriptable vMedia CD. • **HDD** - Scriptable vMedia HDD.
Step 6	UCS-A /org/vmedia-policy/vmedia-mapping* # **set image-file** *image-file-name*	Specifies the type of remote vMedia image file name. Enter the full path to the backup configuration file. This field can contain the filename [with the file extension] only. **Note** Ensure that the full path to the file begins with "/" after the share name.
Step 7	UCS-A /org/vmedia-policy/vmedia-mapping* # **set image-path** *image-path*	Specifies the remote vMedia image path. Enter the full path to the remote vMedia configuration file.
Step 8	UCS-A /org/vmedia-policy/vmedia-mapping* # **set image-variable-name** {**none** | **service-profile-name**}	Specifies the name to be used for the image. Options are: • **none—**Enter the filename manually. • **service-profile-name—**Automatically uses the name of the service profile that the policy is associated with. **Note** If you specify the **image-variable-name** as the **service-profile-name**, do not rename the service profile. Renaming the service profile can result in vMedia mount failure.

	Command or Action	Purpose
Step 9	UCS-A /org/vmedia-policy/vmedia-mapping* # **set mount-protocol** *mount-protocol*	Specifies the remote vMedia mount protocol. Options are: • **CIFS** • **NFS** • **HTTP** • **HTTPS**
Step 10	UCS-A /org/vmedia-policy/vmedia-mapping* # **set auth-option** { **default** | **none** | **ntlm** | **ntlmi** | **ntlmssp** | **ntlmsspi** | **ntlmv2** | **ntlmv2i**}	Specifies the CIFS authentication options. This command is available only when you specify CIFS as the remote vMedia mount protocol. It is not available when you select any other remote vMedia mount protocol. The CIFS authentication options are: • **default—**NT LAN Manager Security Support Provider (NTLMSSP) protocol. Use this option only with Windows 2008 R2 and Windows 2012 R2. • **none—**No authentication is used. • **ntlm—**NT LAN Manager (NTLM) security protocol. Use this option only with Windows 2008 R2 and Windows 2012 R2. • **ntlmi—**NTLMi security protocol. Use this option only when you enable Digital Signing in the CIFS Windows server. • **ntlmssp—**NT LAN Manager Security Support Provider (NTLMSSP) protocol. Use this option only with Windows 2008 R2 and Windows 2012 R2. • **ntlmsspi—**Use this option only when you enable Digital Signing in the CIFS Windows server. • **ntlmv2—**NTLMv2 security protocol. Use this option only with Samba Linux. • **ntlmv2i—**NTLMv2i security protocol. Use this option only with Samba Linux.
Step 11	UCS-A /org/vmedia-policy/vmedia-mapping* # **set password**	Specifies the remote vMedia image password.
Step 12	UCS-A /org/vmedia-policy/vmedia-mapping* # **set remote-ip** *remote-ip*	Specifies the remote vMedia image IP address.
Step 13	UCS-A /org/vmedia-policy/vmedia-mapping* # **set user-id** *user-id*	Specifies the user id for mounting the vMedia device. Enter the username that Cisco UCS Manager should use to log in to the remote server.

	Command or Action	Purpose
		This field does not apply if the protocol is NFS. This field is optional if the protocol is HTTP.
Step 14	UCS-A /org/vmedia-policy/vmedia-mapping* # **commit-buffer**	Commits the transaction to the system configuration.

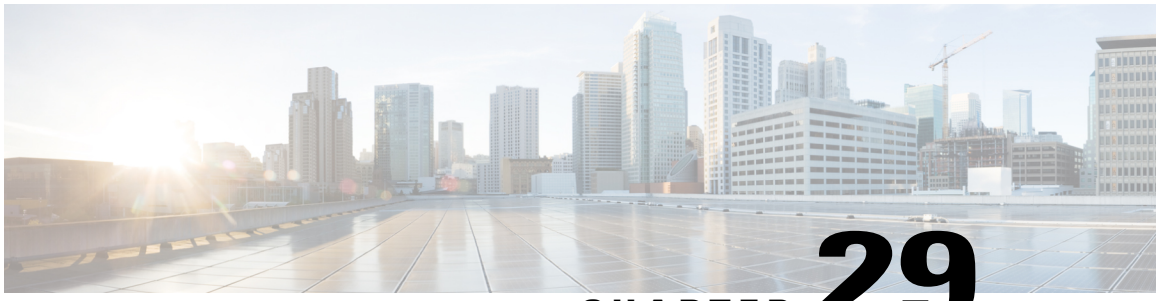
The following example creates a vMedia policy named vMediaPolicy2, selects remote vMedia device type, mount protocol, image location, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create vmedia-policy vmediapolicy2
UCS-A /org/vmedia-policy* # create vmedia-mapping map1
UCS-A /org/vmedia-policy/vmedia-mapping* # set descr vmedia-map
UCS-A /org/vmedia-policy/vmedia-mapping* # set device-type cdd
UCS-A /org/vmedia-policy/vmedia-mapping* # set image-file-name win2011.iso
UCS-A /org/vmedia-policy/vmedia-mapping* # set image-path cifs
UCS-A /org/vmedia-policy/vmedia-mapping* # set image-variable-name service-profile-name
UCS-A /org/vmedia-policy/vmedia-mapping* # set mount-protocol cifs
UCS-A /org/vmedia-policy/vmedia-mapping* # set auth-option default
UCS-A /org/vmedia-policy/vmedia-mapping* # set password Password:
UCS-A /org/vmedia-policy/vmedia-mapping* # set remote-ip 172.41.1.158
UCS-A /org/vmedia-policy/vmedia-mapping* # set user-id Adminstrator
UCS-A /org/vmedia-policy/vmedia-mapping* # commit-buffer
```

Note When vMedia policy is created the **Retry on Mount Fail** option is set to **Yes**. The following example changes the **Retry on Mount Fail** option to **No** .

```
UCS-A# scope org /
UCS-A /org # create vmedia-policy vmediapolicy2
UCS-A /org/vmedia-policy* # set retry-on-mount-fail No
UCS-A /org/vmedia-policy* # commit-buffer
```

Warning When you set the **Retry on Mount Fail** option to **No**, a warning message appears stating: **This will disable automatic retry of mount in case of any vMedia mount failure**.



CHAPTER 29

Configuring Server Boot

This chapter includes the following sections:

Boot Policy

The Cisco UCS Manager enables you to create a boot policy for blade servers, rack servers, and modular servers.

The Cisco UCS Manager boot policy overrides the boot order in the BIOS setup menu and determines the following:

- Selection of the boot device
- Location from which the server boots
- Order in which boot devices are invoked

For example, you can have associated servers boot from a local device, such as a local disk or CD-ROM (VMedia), or you can select a SAN boot or a LAN (PXE) boot.

You can either create a named boot policy to associate with one or more service profiles, or create a boot policy for a specific service profile. A boot policy must be included in a service profile, and that service profile must be associated with a server for it to take effect. If you do not include a boot policy in a service profile, Cisco UCS Manager applies the default boot policy.

Note Changes to a boot policy might be propagated to all servers created with an updating service profile template that includes that boot policy. Re-association of the service profile with the server to rewrite the boot order information in the BIOS is automatically triggered.

You can also specify the following for the boot policy:

- Local LUN name. The name specified is the logical name in the storage profile, not the deployed name. For modular servers, you can specify both a primary and secondary name. For other servers, specify only a primary name. Specifying a secondary name results in a configuration error.
- Specific JBOD disk number for booting from JBOD disks. This is not supported for the Modular servers.
- Any LUN for backward compatibility; however, we do not recommend this. Other devices must not have bootable images to ensure a successful boot.

UEFI Boot Mode

Unified Extensible Firmware Interface (UEFI) is a specification that defines a software interface between an operating system and platform firmware. Cisco UCS Manager uses UEFI to replace the BIOS firmware interfaces. This allows the BIOS to run in UEFI mode while still providing legacy support.

You can choose either legacy or UEFI boot mode when you create a boot policy. Legacy boot mode is supported for all Cisco UCS servers. UEFI boot mode is supported only on M3 and higher servers, and allows you to enable UEFI secure boot mode.

UEFI PXE boot is supported with all Cisco VIC adapters on Cisco UCS rack servers integrated with Cisco UCS Manager Release 2.2(4) and later releases. Beginning with Cisco UCS Manager Release 2.2(1), UEFI PXE boot is supported on all Cisco blade servers.

The following limitations apply to the UEFI boot mode:

- UEFI boot mode is not supported with the following combinations:
  - Gen-3 Emulex and QLogic adapters on Cisco UCS blade and rack servers integrated with Cisco UCS Manager.
  - iSCSI boot for all adapters on Cisco UCS rack servers integrated with Cisco UCS Manager.
- If you want to use UEFI boot mode with two iSCSI LUNs, you must manually specify a common iSCSI initiator name in the service profile that is applied to both underlying iSCSI eNICs rather than allowing Cisco UCS Manager to select the name from an IQN suffix pool. If you do not supply a common name, Cisco UCS Manager will not be able to detect the second iSCSI LUN.
- You cannot mix UEFI and legacy boot mode on the same server.

- The server will boot correctly in UEFI mode only if the boot devices configured in the boot policy have UEFI-aware operating systems installed. If a compatible OS is not present, the boot device is not displayed on the **Actual Boot Order** tab in the **Boot Order Details** area.
- In some corner cases, the UEFI boot may not succeed because the UEFI boot manager entry was not saved correctly in the BIOS NVRAM. You can use the UEFI shell to enter the UEFI boot manager entry manually. This situation could occur in the following situations:
  - If a blade server with UEFI boot mode enabled is disassociated from the service profile, and the blade is manually powered on using the **Equipment** tab or the front panel.
  - If a blade server with UEFI boot mode enabled is disassociated from the service profile, and a direct VIC firmware upgrade is attempted.
  - If a blade or rack server with UEFI boot mode enabled is booted off SAN LUN, and the service profile is migrated.

You can create UEFI boot parameters in Cisco UCS Manager. UEFI Boot Parameters, on page 571 provides more information.

UEFI Secure Boot

Cisco UCS Manager supports UEFI secure boot on Cisco UCS B-Series M3 and M4 Blade servers, Cisco UCS C-Series M3 and M4 Rack servers, and Cisco UCS S-Series M4 Rack servers. When UEFI secure boot is enabled, all executables, such as boot loaders and adapter drivers, are authenticated by the BIOS before they can be loaded. To be authenticated, the images must be signed by either the Cisco Certificate Authority (CA) or a Microsoft CA.

The following limitations apply to UEFI secure boot:

- UEFI boot mode must be enabled in the boot policy.
- The Cisco UCS Manager software and the BIOS firmware must be at Release 2.2 or greater.

Note UEFI boot mode is supported on Cisco UCS C-Series and S-Series rack servers beginning with Release 2.2(3a).

- User-generated encryption keys are not supported.
- UEFI secure boot can only be controlled by Cisco UCS Manager.
- If you want to downgrade to an earlier version of Cisco UCS Manager, and you have a server in secure boot mode, you must disassociate, then re-associate the server before downgrading. Otherwise, server discovery is not successful.

CIMC Secure Boot

With CIMC secure boot, only Cisco signed firmware images can be installed and run on the servers. When the CIMC is updated, the image is certified before the firmware is flashed. If certification fails, the firmware is not flashed. This prevents unauthorized access to the CIMC firmware.

Guidelines and Limitations for CIMC Secure Boot

- CIMC secure boot is supported on Cisco UCS M3 rack servers.

Note CIMC secure boot is enabled by default on the Cisco UCS C220 M4, C240 M4 rack servers, and is automatically enabled on the Cisco UCS C460 M4 rack server after upgrading to CIMC firmware release 2.2(3) or higher.

- After CIMC secure boot is enabled, you cannot disable it.
- After CIMC secure boot is enabled on a server, you cannot downgrade to a CIMC firmware image prior to 2.1(3).

Determining the CIMC Secure Boot Status

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *server-num*	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # **scope cimc**	Enters server CIMC mode.
Step 3	UCS-A /server/cimc # **show secure-boot**	Displays the CIMC secure boot status for the specified server. This can be one of the following: • **Unsupported—**CIMC secure boot is not supported on the server. • **Disabled—**CIMC secure boot is supported, but is disabled on the server. • **Enabling—**CIMC secure boot has been enabled, and the operation is in process. • **Enabled—**CIMC secure boot is enabled on the server.

The following example shows how to display the CIMC secure boot status:

```
UCS-A# scope server 1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # show secure-boot
Secure Boot: Disabled
UCS-A /chassis/server/cimc #
```

Enabling CIMC Secure Boot

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *server-num*	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # **scope cimc**	Enters server CIMC mode.
Step 3	UCS-A /server/cimc # **enable secure-boot**	Enables CIMC secure boot status for the specified server. CIMC secure boot is only supported on Cisco UCS M3 rack servers. **Note** Once enabled, CIMC secure boot cannot be disabled.
Step 4	UCS-A /server/cimc # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to enable CIMC secure boot and commit the transaction:

```
UCS-A# scope server 1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # enable secure-boot
Warning: When committed, CIMC Secure Boot and Installation Feature will be enabled for the
 server.
This is an irreversible operation!!

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

Creating a Boot Policy

You can also create a local boot policy that is restricted to a service profile or service profile template. However, Cisco recommends that you create a global boot policy that can be included in multiple service profiles or service profile templates.

Before You Begin

If you are creating a boot policy that boots the server from a SAN LUN and you require reliable SAN boot operations, you must first remove all local disks from servers associated with a service profile that includes the boot policy.

Note This does not apply for Cisco UCS M3 and M4 servers.

Procedure

	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 # **create boot-policy** *policy-name* [**purpose** {**operational** | **utility**}]	Creates a boot policy with the specified policy name, and enters organization boot policy mode. When you create the boot policy, specify the **operational** option. This ensures that the server boots from the operating system installed on the server. The **utility** options is reserved and should only be used if instructed to do so by a Cisco representative.
Step 3	UCS-A /org/boot-policy # **set descr** *description*	(Optional) Provides a description for the boot policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks do not appear in the description field of any **show** command output.
Step 4	UCS-A /org/boot-policy # **set reboot-on-update** {**no** | **yes**}	Specifies whether the servers using this boot policy are automatically rebooted after you make changes to the boot order.
Step 5	UCS-A /org/boot-policy # **set enforce-vnic-name** {**no** | **yes**}	If you choose **yes**, Cisco UCS Manager displays a configuration error and reports whether one or more of the vNICs, vHBAs, or iSCSI vNICs listed in the **Boot Order** table match the server configuration in the service profile. If you choose **no**, Cisco UCS Manager uses the vNICs, vHBAs, or iSCSI vNICs (as appropriate for the boot option) from the service profile.
Step 6	UCS-A /org/boot-policy # **set boot-mode** {**legacy** | **uefi**}	Specifies whether the servers using this boot policy are using UEFI or legacy boot mode.
Step 7	UCS-A /org/boot-policy # **commit-buffer**	Commits the transaction to the system configuration.
Step 8	UCS-A /org/boot-policy # **create boot-security**	Enters boot security mode for the specified boot policy.
Step 9	UCS-A /org/boot-policy/boot-security # **set secure-boot** {**no** | **yes**}	Specifies whether secure boot is enabled for the boot policy.
Step 10	UCS-A /org/boot-policy/boot-security # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a boot policy named boot-policy-LAN, specify that servers using this policy will not be automatically rebooted when the boot order is changed, set the UEFI boot mode, enable UEFI boot security, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create boot-policy boot-policy-LAN purpose operational
UCS-A /org/boot-policy* # set descr "Boot policy that boots from the LAN."
UCS-A /org/boot-policy* # set reboot-on-update no
UCS-A /org/boot-policy* # set boot-mode uefi
UCS-A /org/boot-policy* # commit-buffer
UCS-A /org/boot-policy # create boot-security
UCS-A /org/boot-policy/boot-security* # set secure-boot yes
UCS-A /org/boot-policy/boot-security* # commit-buffer
UCS-A /org/boot-policy/boot-security #
```

What to Do Next

Configure one or more of the following boot options for the boot policy and set their boot order:

- **LAN Boot —**Boots from a centralized provisioning server. It is frequently used to install operating systems on a server from that server.

  If you choose the LAN Boot option, continue to Configuring a LAN Boot for a Boot Policy, on page 562.

- **SAN Boot —**Boots from an operating system image on the SAN. You can specify a primary and a secondary SAN boot. If the primary boot fails, the server attempts to boot from the secondary.

  We recommend that you use a SAN boot policy, because it offers the most service profile mobility within the system. If you boot from the SAN, when you move a service profile from one server to another, the new server boots from exactly the same operating system image. Therefore, the new server appears to be exactly the same server to the network.

  If you choose the SAN Boot option, continue to Configuring a SAN Boot for a Boot Policy, on page 528.

- **Virtual Media Boot —**Mimics the insertion of a physical CD into a server. It is typically used to manually install operating systems on a server.

  If you choose the Virtual Media boot option, continue to Configuring a Virtual Media Boot for a Boot Policy, on page 567.

Tip If you configure a local disk and a SAN LUN for the boot order storage type and the operating system or logical volume manager (LVM) is configured incorrectly, the server might boot from the local disk rather than the SAN LUN.

For example, on a server with Red Hat Linux installed, where the LVM is configured with default LV names and the boot order is configured with a SAN LUN and a local disk, Linux reports that there are two LVs with the same name and boots from the LV with the lowest SCSI ID, which could be the local disk.

Include the boot policy in a service profile and template.

SAN Boot

You can configure a boot policy to boot one or more servers from an operating system image on the SAN. The boot policy can include a primary and a secondary SAN boot. If the primary boot fails, the server attempts to boot from the secondary.

Cisco recommends using a SAN boot, because it offers the most service profile mobility within the system. If you boot from the SAN when you move a service profile from one server to another, the new server boots from the same operating system image. Therefore, the new server appears as the same server to the network.

To use a SAN boot, ensure that the following is configured:

- The Cisco UCS domain must be able to communicate with the SAN storage device that hosts the operating system image.
- A boot target LUN (Logical Unit Number) on the device where the operating system image is located.

Note SAN boot is not supported on Gen-3 Emulex adapters on Cisco UCS blade and rack servers.

Configuring a SAN Boot for a Boot Policy

Tip If you configure a local disk and a SAN LUN for the boot order storage type and the operating system or logical volume manager (LVM) is configured incorrectly, the server might boot from the local disk rather than the SAN LUN.

For example, on a server with Red Hat Linux installed, where the LVM is configured with default LV names and the boot order is configured with a SAN LUN and a local disk, Linux reports that there are two LVs with the same name and boots from the LV with the lowest SCSI ID, which could be the local disk.

This procedure continues directly from Creating a Boot Policy, on page 525.

Before You Begin

Create a boot policy to contain the SAN boot configuration.

Note If you are creating a boot policy that boots the server from a SAN LUN and you require reliable SAN boot operations, we recommend that you first remove all local disks from servers associated with a service profile that includes the boot policy.

This does not apply for Cisco UCS M3 and M4 servers.

Beginning with Release 2.2, all SAN boot-related CLI commands have been moved to the SAN scope. Any existing scripts from previous releases that use SAN boot under the storage scope instead of **org/boot-policy/san** or **org/service-profile/boot-definition/san** should be updated.

Procedure

	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 san**	Creates a SAN boot for the boot policy and enters organization boot policy storage mode.
Step 4	UCS-A /org/boot-policy/san # **set order** *order_number*	Sets the boot order for the SAN boot. Enter an integer between 1 and 16.
Step 5	UCS-A /org/boot-policy/san # **create san-image** {**primary** | **secondary**}	Creates a SAN image location, and if the san-image option is specified, enters organization boot policy storage SAN image mode. When using the enhanced boot order on Cisco UCS M3 servers, or M4 servers, the boot order that you define is used. For standard boot mode using the terms "primary" or "secondary" do not imply a boot order. The effective order of boot devices within the same device class is determined by the PCIe bus scan order.
Step 6	UCS-A /org/boot-policy/ssn/san-image # **set vhba** *vhba-name*	Specifies the vHBA to be used for the SAN boot.
Step 7	UCS-A /org/boot-policy/san/san-image # **create path** {**primary** | **secondary**}	Creates a primary or secondary SAN boot path and enters organization boot policy SAN path mode. When using the enhanced boot order on Cisco UCS M3 servers, or M4 servers, the boot order that you define is used. For standard boot mode using the terms "primary" or "secondary" do not imply a boot order. The effective order of boot devices within the same device class is determined by the PCIe bus scan order.
Step 8	UCS-A /org/boot-policy/san/san-image/path # **set** {**lun** *lun-id* | **wwn** *wwn-num*}	Specifies the LUN or WWN to be used for the SAN path to the boot image.
Step 9	UCS-A /org/boot-policy/san/san-image/path # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to enter the boot policy named lab1-boot-policy, create a SAN boot for the policy, set the boot order to 1, create a primary SAN image, use a vHBA named vHBA2, create primary path using LUN 0, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy lab1-boot-policy
UCS-A /org/boot-policy # create san
UCS-A /org/boot-policy/san* # set order 1
UCS-A /org/boot-policy/san* # create san-image primary
UCS-A /org/boot-policy/san/san-image* # set vhba vHBA2
UCS-A /org/boot-policy/san/san-image* # create path primary
UCS-A /org/boot-policy/san/san-image/path* # set lun 0
UCS-A /org/boot-policy/san/san-image/path* # commit-buffer
UCS-A /org/boot-policy/san/san-image/path #
```

The following example shows how to create a SAN boot for the service profile SP_lab1, set the boot order to 1, create a primary SAN image, use a vHBA named vHBA2, create primary path using LUN 0, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile SP_lab1
UCS-A /org/service-profile # create boot-definition
UCS-A /org/service-profile/boot-definition* # create san
UCS-A /org/service-profile/boot-definition/san* # create san-image primary
UCS-A /org/service-profile/boot-definition/san/san-image* # set vhba vHBA2
UCS-A /org/service-profile/boot-definition/san/san-image* # create path primary
UCS-A /org/service-profile/boot-definition/san/san-image/path* # set lun 0
UCS-A /org/service-profile/boot-definition/san/san-image/path* # commit-buffer
UCS-A /org/service-profile/boot-definition/san/san-image/path #
```

What to Do Next

Include the boot policy in a service profile and template.

iSCSI Boot

iSCSI boot enables a server to boot its operating system from an iSCSI target machine located remotely over a network.

iSCSI boot is supported on the following Cisco UCS hardware:

- Cisco UCS blade servers that have the Cisco UCS M51KR-B Broadcom BCM57711 network adapter and use the default MAC address provided by Broadcom.
- Cisco UCS M81KR Virtual Interface Card
- Cisco UCS VIC-1240 Virtual Interface Card
- Cisco UCS VIC-1280 Virtual Interface Card
- Cisco UCS rack servers that have the Cisco UCS M61KR-B Broadcom BCM57712 network adapter.
- Cisco UCS P81E Virtual Interface Card
- Cisco UCS VIC 1225 Virtual Interface Cardon Cisco UCS rack servers

There are prerequisites that must be met before you configure iSCSI boot. For a list of these prerequisites, see iSCSI Boot Guidelines and Prerequisites, on page 531.

For a high-level procedure for implementing iSCSI boot, see Configuring iSCSI Boot, on page 534.

iSCSI Boot Process

Cisco UCS Manager uses the iSCSI vNIC and iSCSI boot information created for the service profile in the association process to program the adapter, located on the server. After the adapter is programmed, the server reboots with the latest service profile values. After the power on self-test (POST), the adapter attempts to initialize using these service profile values. If the adapter can use the values and log in to its specified target, the adapter initializes and posts an iSCSI Boot Firmware Table (iBFT) to the host memory and a valid bootable LUN to the system BIOS. The iBFT that is posted to the host memory contains the initiator and target configuration that is programmed on the primary iSCSI VNIC.

Note Previously, the host could see only one of the boot paths configured, depending on which path completed the LUN discovery first, and would boot from that path. Now, when there are two iSCSI boot vNICs configured, the host sees both of the boot paths. So for multipath configurations, a single IQN must be configured on both the boot vNICs. If there are different IQNs configured on the boot vNICs on a host, the host boots with the IQN that is configured on the boot vNIC with the lower PCI order.

The next step, which is the installation of the operating system (OS), requires an OS that is iBFT capable. During installation of the OS, the OS installer scans the host memory for the iBFT table and uses the information in the iBFT to discover the boot device and create an iSCSI path to the target LUN. Some OSs requires a NIC driver to complete this path. If this step is successful, the OS installer finds the iSCSI target LUN on which to install the OS.

Note The iBFT works at the OS installation software level and might not work with HBA mode (also known as TCP offload). Whether iBFT works with HBA mode depends on the OS capabilities during installation. Also, for a server that includes a Cisco UCS M51KR-B Broadcom BCM57711 adapter, the iBFT normally works at a maximum transmission unit (MTU) size of 1500, regardless of the MTU jumbo configuration. If the OS supports HBA mode, you might need to set HBA mode, dual-fabric support, and jumbo MTU size after the iSCSI installation process.

iSCSI Boot Guidelines and Prerequisites

These guidelines and prerequisites must be met before configuring iSCSI boot:

- After the iSCSI boot policies are created, a user with ls-compute privileges can include them in a service profile or service profile template. However, a user with only ls-compute privileges cannot create iSCSI boot policies.
- To set up iSCSI boot from a Windows 2008 server where the second vNIC (failover vNIC) must boot from an iSCSI LUN, consult Microsoft Knowledge Base Article 976042. Microsoft has a known issue where Windows might fail to boot from an iSCSI drive or cause a bugcheck error if the networking hardware is changed. To work around this issue, follow the resolution recommended by Microsoft.
- The storage array must be licensed for iSCSI boot and the array side LUN masking must be properly configured.

- Two IP addresses must be determined, one for each iSCSI initiator. If possible, the IP addresses should be on the same subnet as the storage array. The IP addresses are assigned statically or dynamically using the Dynamic Host Configuration Protocol (DHCP).
- You cannot configure boot parameters in the Global boot policy. Instead, after configuring boot parameters, include the boot policy in the appropriate service profile.
- The operating system (OS) must be iSCSI Boot Firmware Table (iBFT) compatible.
- For Cisco UCS M51KR-B Broadcom BCM57711 network adapters:
  - Servers that use iSCSI boot must contain the Cisco UCS M51KR-B Broadcom BCM57711 network adapter. For information on installing or replacing an adapter card, see the *Cisco UCS B250 Extended Memory Blade Server Installation and Service Note*. The service note is accessible from the *Cisco UCS B-Series Servers Documentation Roadmap* at http://www.cisco.com/go/unifiedcomputing/b-series-doc.
  - Set the MAC addresses on the iSCSI device.
  - If you are using the DHCP Vendor ID (Option 43), configure the MAC address of an iSCSI device in /etc/dhcpd.conf.
  - HBA mode (also known as TCP offload) and the boot to target setting are supported. However, only Windows OS supports HBA mode during installation.
  - Before installing the OS, disable the boot to target setting in the iSCSI adapter policy, then after installing the OS, re-enable the boot to target setting.

    Note Each time you change an adapter policy setting, the adapter reboots to apply the new setting.

  - When installing the OS on the iSCSI target, the iSCSI target must be ordered *before* the device where the OS image resides. For example, if you are installing the OS on the iSCSI target from a CD, the boot order should be the iSCSI target and then the CD.
  - After the server is iSCSI booted, do not modify the Initiator Name, Target name, LUN, iSCSI device IP, or Netmask/gateway using the Broadcom tool.
  - Do not interrupt the POST (power on self-test) process or the Cisco UCS M51KR-B Broadcom BCM57711 network adapter will fail to initialize.
- For Cisco UCS M81KR Virtual Interface Card and Cisco UCS VIC-1240 Virtual Interface Card:

  For Cisco UCS VIC-1240 Virtual Interface Card:
  - Do not set MAC addresses on the iSCSI device.
  - HBA mode and the boot to target setting are *not* supported.
  - When installing the OS on the iSCSI target, the iSCSI target must be ordered *after* the device where the OS image resides. For example, if you are installing the OS on the iSCSI target from a CD, the boot order should be the CD and then the iSCSI target.
  - If you are using the DHCP Vendor ID (Option 43), the MAC address of the overlay vNIC must be configured in /etc/dhcpd.conf.
  - After the server is iSCSI booted, do not modify the IP details of the overlay vNIC.

- The VMware ESX/ESXi operating system does not support storing a core dump file to an iSCSI boot target LUN. Dump files must be written to a local disk.

Initiator IQN Configuration

Cisco UCS uses the following rules to determine the initiator IQN for an adapter iSCSI vNIC at the time a service profile is associated with a physical server:

- An initiator IQN at the service profile level *and* at the iSCSI vNIC level cannot be used together in a service profile.
- If an initiator IQN is specified at the service profile level, all of the adaptor iSCSI vNICs are configured to use the same initiator IQN, except in the case of DHCP Option 43, where the initiator IQN is set to empty on the adapter iSCSI vNIC.
- When an initiator IQN is set at the iSCSI vNIC level, the initiator IQN at the service profile level is removed, if one is present.
- If there are two iSCSI vNIC in a service profile and only one of them has the initiator IQN set, the second one is configured with the default IQN pool. You can change this configuration later. The only exception is if DHCP Option 43 is configured. In this case, the initiator IQN on the second iSCSI vNIC is removed during service profile association.

Note If you change an iSCSI vNIC to use the DHCP Option 43 by setting the vendor ID, it does not remove the initiator IQN configured at the service profile level. The initiator IQN at the service profile level can still be used by another iSCSI vNIC which does not use the DHCP Option 43.

Enabling MPIO on Windows

You can enable (MPIO) to optimize connectivity with storage arrays.

Note If you change the networking hardware, Windows might fail to boot from an iSCSI drive. For more information, see Microsoft support Article ID: 976042.

Before You Begin

The server on which you enable the Microsoft Multipath I/O (MPIO) must have a Cisco VIC driver.

If there are multiple paths configured to the boot LUN, only one path should be enabled when the LUN is installed.

Procedure

Step 1 In the service profile associated with the server, configure the primary iSCSI vNIC.
For more information, see Creating an iSCSI vNIC in a Service Profile, on page 544.

Step 2 Using the primary iSCSI vNIC, install the Windows operating system on the iSCSI target LUN.

Step 3 After Windows installation completes, enable MPIO on the host.

Step 4 In the service profile associated with the server, add the secondary iSCSI vNIC to the boot policy. For more information, see Creating an iSCSI Adapter Policy, on page 535.

Configuring iSCSI Boot

When you configure an adapter or blade in Cisco UCS to iSCSI boot from a LUN target, complete all of the following steps.

Procedure

	Command or Action	Purpose
Step 1	Configure the iSCSI boot adapter policy.	(Optional) For more information, see Creating an iSCSI Adapter Policy, on page 535.
Step 2	Configure the authentication profiles for the initiator and target.	(Optional) For more information, see Creating an Authentication Profile, on page 537.
Step 3	To configure the iSCSI initiator to use an IP address from a pool of IP addresses, add a block of IP addresses to the iSCSI initiator pool.	(Optional) For more information, see Adding a Block of IP Addresses to the Initiator Pool, on page 539.
Step 4	Create a boot policy that can be used in any service profile. Alternatively, you can create a local boot policy only for the specific service policy. However, Cisco recommends that you create a boot policy that can be shared with multiple service profiles.	For more information about creating a boot policy that can be used in any service profile, see Creating an iSCSI Adapter Policy, on page 535.
Step 5	If you created a boot policy that can be used in any service profile, assign it to the service profile. Otherwise, proceed to the next step.	For more information, see Creating a Service Profile Template, on page 602.
Step 6	Configure an Ethernet vNIC in a service profile.	The Ethernet vNIC is used as the overlay vNIC for the iSCSI device. For more information, see Configuring a vNIC for a Service Profile, on page 610.
Step 7	Create an iSCSI vNIC in a service profile.	For more information, see Creating an iSCSI vNIC in a Service Profile, on page 544.

	Command or Action	Purpose
Step 8	Set the iSCSI initiator to boot using a static IP Address, an IP address from an IP pool, or DHCP.	See either Creating an iSCSI Initiator that Boots Using a Static IP Address, on page 546, Creating an iSCSI Initiator that Boots Using an IP Address from an IP Pool, on page 548, or Creating an iSCSI Initiator that Boots Using DHCP, on page 550.
Step 9	Create an iSCSI static or auto target.	For more information, see either Creating an iSCSI Static Target, on page 557 or Creating an iSCSI Auto Target, on page 560.
Step 10	Associate the service profile with a server.	For more information, see Associating a Service Profile with a Blade Server or Server Pool, on page 626.
Step 11	Verify the iSCSI boot operation.	For more information, see *Verifying iSCSI Boot*.
Step 12	Install the OS on the server.	For more information, see one of the following guides: • *Cisco UCS B-Series Blade Servers VMware Installation Guide* • *Cisco UCS B-Series Blade Servers Linux Installation Guide* • *Cisco UCS B-Series Blade Servers Windows Installation Guide*
Step 13	Boot the server.	

Creating an iSCSI Adapter Policy

Procedure

	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 # **create iscsi-policy** *policy-name*	Creates the iSCSI adapter policy.
Step 3	UCS-A /org/iscsi-policy # **set descr** *description*	(Optional) Provides a description for the iSCSI adapter policy.

	Command or Action	Purpose
Step 4	UCS-A /org/iscsi-policy # **set iscsi-protocol-item connection-timeout** *timeout-secs*	The number of seconds to wait until Cisco UCS assumes that the initial login has failed and the iSCSI adapter is unavailable. Enter an integer between 0 and 255. If you enter 0, Cisco UCS uses the value set in the adapter firmware (default: 15 seconds).
Step 5	UCS-A /org/iscsi-policy # **set iscsi-protocol-item dhcp-timeout** *timeout-secs*	The number of seconds to wait before the initiator assumes that the DHCP server is unavailable. Enter an integer between 60 and 300 (default: 60 seconds).
Step 6	UCS-A /org/iscsi-policy # **set iscsi-protocol-item lun-busy-retry-count** *num*	The number of times to retry the connection in case of a failure during iSCSI LUN discovery. Enter an integer between 0 and 60. If you enter 0, Cisco UCS uses the value set in the adapter firmware (default: 15 seconds).
Step 7	UCS-A /org/iscsi-policy # **set iscsi-protocol-item tcp-time-stamp** {**no** | **yes**}	Specifies whether to apply a TCP timestamp. With this setting, transmitted packets are given a time stamp of when the packet was sent so that the packet's round-trip time can be calculated, when needed. This setting applies only to Cisco UCS M51KR-B Broadcom BCM57711 adapters.
Step 8	UCS-A /org/iscsi-policy # **set iscsi-protocol-item hbamode** {**no** | **yes**}	Specifies whether to enable HBA mode. This option should only be enabled for servers with the Cisco UCS NIC M51KR-B adapter running the Windows operating system.
Step 9	UCS-A /org/iscsi-policy # **set iscsi-protocol-item boottotarget** {**no** | **yes**}	Specifies whether to boot from the iSCSI target. This option only applies to servers with the Cisco UCS NIC M51KR-B adapter. It should be disabled until you have installed an operating system on the server.
Step 10	UCS-A /org/iscsi-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create an iSCSI adapter policy called iscsiboot, set the connection timeout, DHCP timeout, and LUN busy retry count, apply a TCP timestamp, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create iscsi-policy iscsiboot
UCS-A /org/iscsi-policy* # set iscsi-protocol-item connection-timeout 60
UCS-A /org/iscsi-policy* # set iscsi-protocol-item dhcp-timeout 200
UCS-A /org/iscsi-policy* # set iscsi-protocol-item lun-busy-retry-count 5
UCS-A /org/iscsi-policy* # set iscsi-protocol-item tcp-time-stamp yes
UCS-A /org/iscsi-policy* # set iscsi-protocol-item hbamode yes
UCS-A /org/iscsi-policy* # set iscsi-protocol-item boottotarget yes
UCS-A /org/iscsi-policy* # commit-buffer
UCS-A /org/iscsi-policy #
```

What to Do Next

Include the adapter policy in a service profile and template.

Deleting an iSCSI Adapter Policy

Procedure

	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 # **delete iscsi-policy** *policy-name*	Deletes the iSCSI adapter policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI adapter policy named iscsi-adapter-pol and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete iscsi-policy iscsi-adapter-pol
UCS-A /org* # commit-buffer
UCS-A /org #
```

Creating an Authentication Profile

If you use authentication for iSCSI boot, you need to create an authentication profile for both the initiator and target.

Procedure

	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 # **create auth-profile** *profile-name*	Creates an authentication profile with the specified name. The name can be up to 16 alphanumeric characters.
Step 3	UCS-A /org/auth-profile* # **set user-id** *id-name*	Creates a log in for authentication.
Step 4	UCS-A /org/auth-profile* # **set password**	Creates a password for authentication.

	Command or Action	Purpose
Step 5	UCS-A /org/auth-profile* # **commit-buffer**	Commits the transaction to the system configuration.
Step 6	UCS-A /org/auth-profile* # **exit**	Exits the current mode.
Step 7	Repeat steps 2 through 6 to create an authentication profile for the target.	

The following example shows how to create an authentication profile for an initiator and target and commit the transaction:

```
UCS-A# scope org
UCS-A /org # create auth-profile InitAuth
UCS-A /org/auth-profile* # set user-id init
UCS-A /org/auth-profile* # set  password
Enter a password:
Confirm the password:
UCS-A /org/auth-profile* # commit-buffer
UCS-A /org/auth-profile # exit
UCS-A /org # create auth-profile TargetAuth
UCS-A /org/auth-profile* # set user-id target
UCS-A /org/auth-profile* # set  password
Enter a password:
Confirm the password:
UCS-A /org/auth-profile* # commit-buffer
UCS-A /org/auth-profile # exit
```

What to Do Next

Create an Ethernet vNIC to be used as the overlay vNIC for the iSCSI device, and then create an iSCSI vNIC.

Deleting an Authentication Profile

Procedure

	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 # **delete auth-profile** *auth-profile-name*	Deletes the specified authentication profile.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an authentication profile called iscsi-auth and commit the transaction:

```
UCS-A# scope org
UCS-A /org # delete auth-profile iscsi-auth
```

```
UCS-A /org* # commit-buffer
UCS-A /org #
```

Adding a Block of IP Addresses to the Initiator Pool

You can create a group of IP addresses to be used for iSCSI boot. Cisco UCS Manager reserves the block of IP addresses you specify.

The IP pool must not contain any IP addresses that were assigned as static IP addresses for a server or service profile.

Procedure

	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 ip-pool iscsi-initiator-pool**	Enters the mode to specify an iSCSI initiator pool.
Step 3	UCS-A /org/ip-pool # **set descr** *description*	(Optional) Provides a description for the IP pool. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/ip-pool # **set assignmentorder** {**default** | **sequential**}	This can be one of the following: • **default—**Cisco UCS Manager selects a random identity from the pool. • **sequential—**Cisco UCS Manager selects the lowest available identity from the pool.
Step 5	UCS-A /org/ip-pool# **create block** *from_ip_address to_ip_address default_gateway subnet_mask*	Creates a block of IP addresses for the iSCSI initiator.
Step 6	UCS-A/org/ip-pool/block# **show detail expand**	(Optional) Shows the block of IP addresses that you have created.
Step 7	UCS-A /org/ip-pool/block # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create an IP initiator pool for the iSCSI vNIC and commit the transaction:

```
UCS-A # scope org /
UCS-A /org # scope ip-pool iscsi-initiator-pool
UCS-A /org/ip-pool # create block 40.40.40.10 40.40.40.50 40.40.40.1 255.0.0.0
```

```
UCS-A /org/ip-pool/block # show detail expand
Block of IP Addresses:
    From: 40.40.40.10
    To: 40.40.40.50
    Default Gateway: 40.40.40.1
    Subnet Mask: 255.0.0.0
UCS-A /org/ip-pool/block # commit buffer
```

What to Do Next

Configure one or more service profiles or service profile templates to obtain the iSCSI initiator IP address from the iSCSI initiator IP pool.

Deleting a Block of IP Addresses from the Initiator Pool

Procedure

	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 ip-pool iscsi-initiator-pool**	Enters the mode to specify an iSCSI initiator pool.
Step 3	UCS-A /org/ip-pool# **delete block** *from_ip_address to_ip_address*	Deletes the specified block of IP addresses from the initiator pool.
Step 4	UCS-A/org/ip-pool/block# **show detail expand**	(Optional) Shows that the block of IP addresses has been deleted.
Step 5	UCS-A /org/ip-pool# **commit buffer**	Commits the transaction to the system configuration.

The following example shows how to delete a block of IP addresses from the initiator pool and commit the transaction:

```
UCS-A # scope org /
UCS-A /org # scope ip-pool iscsi-initiator-pool
UCS-A /org/ip-pool # delete block 40.40.40.10 40.40.40.50 40.40.40.1 255.0.0.0
UCS-A /org/ip-pool # show detail expand

IP Pool:
    Name: iscsi-initiator-pool
    Size: 0
    Assigned: 0
    Descr:
UCS-A /org/ip-pool # commit buffer
```

Creating an iSCSI Boot Policy

You can add up to two iSCSI vNICs per boot policy. One vNIC acts as the primary iSCSI boot source, and the other acts as the secondary iSCSI boot source.

Procedure

	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 # **create boot-policy** *policy-name* [**purpose** {**operational** | **utility**}]	Creates a boot policy with the specified policy name, and enters organization boot policy mode. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved. When you create the boot policy, specify the **operational** option. This ensures that the server boots from the operating system installed on the server. The **utility** options is reserved and should only be used if instructed to do so by a Cisco representative.
Step 3	UCS-A /org/boot-policy # **set descr** *description*	(Optional) Provides a description for the boot policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks do not appear in the description field of any **show** command output.
Step 4	UCS-A /org/boot-policy # **set enforce-vnic-name** {**no** | **yes**}	(Optional) If you choose **yes**, Cisco UCS Manager reports whether the device name specified in the boot policy matches what is specified in the service profile. If you choose **no**, Cisco UCS Manager uses any vNIC, vHBA, or iSCSI device from the service profile and does not report whether the device name specified in the boot policy matches what is specified in the service profile.
Step 5	UCS-A /org/boot-policy # **set reboot-on-update** {**no** | **yes**}	Specifies whether the servers using this boot policy are automatically rebooted after you make changes to the boot order. In the Cisco UCS Manager GUI, if the **Reboot on Boot Order Change** check box is checked for a boot policy, and if CD-ROM or Floppy is the last device in the boot order, deleting or adding the device does not directly affect the boot order and the server does not reboot.

	Command or Action	Purpose
Step 6	UCS-A /org/boot-policy # **create iscsi**	Adds an iSCSI boot to the boot policy.
Step 7	UCS-A /org/boot-policy/iscsi # **create path** {**primary** | **secondary**}	Specifies the primary and secondary paths that Cisco UCS Manager uses to reach the iSCSI target .With iSCSI boot, you set up two paths. Cisco UCS Manager uses the primary path first, and if that fails, then it uses the secondary path.
Step 8	UCS-A /org/boot-policy/iscsi/path # **create iscsivnicname** *iscsi-vnic-name*	Creates an iSCSI vNIC.
Step 9	UCS-A /org/boot-policy/iscsi/path # **exit**	Exits iSCSI path mode.
Step 10	UCS-A /org/boot-policy/iscsi/path # **set order** *order-num*	Specifies the order for the iSCSI boot in the boot order.
Step 11	Repeat steps 8-10 to create secondary iSCSI vNICs.	(Optional)
Step 12	UCS-A /org/boot-policy/iscsi # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create an iSCSI boot policy named iscsi-boot-policy-LAN, provide a description for the boot policy, specify that servers using this policy are not automatically rebooted when the boot order is changed, set the boot order for iSCSI boot to 2, create an iSCSI boot and associate it with a vNIC called iscsienic1, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create boot-policy iscsi-boot-policy-LAN purpose operational
UCS-A /org/boot-policy* # set descr "Boot policy that boots from iSCSI."
UCS-A /org/boot-policy* # set enforce-vnic-name yes
UCS-A /org/boot-policy* # set reboot-on-update no
UCS-A /org/boot-policy* # create iscsi
UCS-A /org/boot-policy/iscsi* # create path primary
UCS-A /org/boot-policy/iscsi/path* # set iscsivnicname iscsienic1
UCS-A /org/boot-policy/iscsi/path* # exit
UCS-A /org/boot-policy/iscsi* # set order 2
UCS-A /org/boot-policy/iscsi* # commit-buffer
UCS-A /org/boot-policy #
```

What to Do Next

Include the boot policy in a service profile and template.

After a server is associated with a service profile that includes this boot policy, you can verify the actual boot order in the **Boot Order Details** area on the **General** tab for the server.

Deleting iSCSI Devices from a Boot Policy

Procedure

	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** *boot-pol-name*	Enters boot policy organization mode for the specified boot policy.
Step 3	UCS-A /org/boot-policy # **delete iscsi**	Deletes the iSCSI boot from the boot policy.
Step 4	UCS-A /org/boot-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI boot from the boot policy named boot-policy-iscsi and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope boot-policy boot-policy-iscsi
UCS-A /org/boot-policy # delete iscsi
UCS-A /org/boot-policy* # commit-buffer
UCS-A /org/boot-policy #
```

Setting an Initiator IQN at the Service Profile Level

In a service profile, you can create an initiator with a specific IQN or one that is derived from a pool of IQNs.

Before You Begin

You cannot delete an IQN using the CLI.

To understand the initiator IQN configuration guidelines, see Initiator IQN Configuration, on page 533.

Procedure

	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 service profile organization mode for the service profile.

	Command or Action	Purpose
Step 3	UCS-A /org/service-profile# **set iscsi-identity** {**initiator name***initiator-name*|**initiator-pool-name***pool-name*}	Creates an initiator with the specified name. The name can be up to 16 alphanumeric characters.
Step 4	UCS-A /org/service-profile* # **commit buffer**	Commits the transaction to the system configuration.
Step 5	UCS-A /org/auth-profile* # **exit**	Exits the current mode.

The following example shows how to create a specific name for an iSCSI initiator and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # set iscsi-identity initiator-name manual:IQN
UCS-A /org/service-profile* # commit-buffer
```

Creating an iSCSI vNIC in a Service Profile

You can create an iSCSI vNIC in a service profile.

Before You Begin

You must have an Ethernet vNIC in a service profile to be used as the overlay vNIC for the iSCSI device.

Procedure

	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 service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # **create vnic-iscsi** *iscsi-vnic-name* .	Specifies the iSCSI vNIC name.
Step 4	UCS-A /org/service-profile/vnic-iscsi* # **set iscsi-adaptor-policy** *iscsi-adaptor-name*	(Optional) Specifies the iSCSI adapter policy that you have created for this iSCSI vNIC.
Step 5	UCS-A /org/service-profile/vnic-iscsi* # **set auth-name** *authentication-profile-name*	(Optional) Sets the authentication profile to be used by the iSCSI vNIC. The authentication profile must already exist for it to be set. For more information, see Creating an Authentication Profile, on page 537.

	Command or Action	Purpose
Step 6	UCS-A /org/service-profile/vnic-iscsi* # **set identity** { **dynamic-mac** {*dynamic-mac-address* | **derived** } | **mac-pool** *mac-pool-name* }	Specifies the MAC address for the iSCSI vNIC. **Note** The MAC address is only set for Cisco UCS NIC M51KR-B adapters.
Step 7	UCS-A /org/service-profile/vnic-iscsi* # **set iscsi-identity** {**initiator-name** *initiator-name* | **initiator-pool-name** *iqn-pool-name*}	Specifies the name of the iSCSI initiator or the name of an IQN pool from which the iSCSI initiator name will be provided. The iSCSI initiator name can be up to 223 characters.
Step 8	UCS-A /org/service-profile/vnic-iscsi* # **set overlay-vnic-name** *overlay-vnic-name*	Specifies the Ethernet vNIC that is used by the iSCSI device as the overlay vNIC. For more information, see Configuring a vNIC for a Service Profile, on page 610.
Step 9	UCS-A /org/service-profile/vnic-iscsi* # **create eth-if**	Creates an Ethernet interface for a VLAN assigned to the iSCSI vNIC.
Step 10	UCS-A /org/service-profile/vnic-iscsi/eth-if* # **set vlanname** *vlan-name*.	Specifies the VLAN name. The default VLAN is default. For the Cisco UCS M81KR Virtual Interface Card and the Cisco UCS VIC-1240 Virtual Interface Card, the VLAN that you specify must be the same as the native VLAN on the overlay vNIC. For the Cisco UCS M51KR-B Broadcom BCM57711 adapter, the VLAN that you specify can be any VLAN assigned to the overlay vNIC.
Step 11	UCS-A /org/service-profile/vnic-iscsi # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create an iSCSI vNIC called scsivnic1, add it to an existing service profile called accounting, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # create vnic-iscsi iSCSI1
UCS-A /org/service-profile/vnic-iscsi* # set iscsi-adaptor-policy iscsiboot
UCS-A /org/service-profile/vnic-iscsi* # set auth-name initauth
UCS-A /org/service-profile/vnic-iscsi* # set identity dynamic-mac derived
UCS-A /org/service-profile/vnic-iscsi* # set iscsi-identity initiator-name iSCSI1
UCS-A /org/service-profile/vnic-iscsi* # set overlay-vnic-name eth1
UCS-A /org/service-profile/vnic-iscsi* # create eth-if
UCS-A /org/service-profile/vnic-iscsi/eth-if* # set vlanname default
UCS-A /org/service-profile/vnic-iscsi/eth-if* # commit buffer
```

What to Do Next

Configure an iSCSI initiator to boot using a static IP address, an IP address from a configured IP pool, or DHCP.

Deleting an iSCSI vNIC from a Service Profile

Procedure

	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 service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # **delete vnic-iscsi** *iscsi-vnic-name*	Deletes the specified iSCSI vNIC from the specified service profile.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI vNIC called scsivnic1 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # delete vnic-iscsi scsivnic1
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Creating an iSCSI Initiator that Boots Using a Static IP Address

In a service profile, you can create an iSCSI initiator and configure it to boot using a static IP address.

Before You Begin

You have completed the following:

- Created iSCSI overlay vNICs in a service profile.
- Created an iSCSI vNIC in a service profile.

Procedure

	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*.

	Command or Action	Purpose
Step 2	UCS-A /org # **scope service-profile** *profile-name*	Enters service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # **scope vnic-iscsi** *iscsi-vnic-name*	Enters the configuration mode for the specified iSCSI vNIC.
Step 4	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # **create ip-if**	Creates an IP interface.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if* # **enter static-ip-params**	Specifies that you are entering static IP boot parameters.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # **set addr** *ip-address*	Specifies the static IP address.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # **set default-gw** *ip-address*	Specifies the default gateway IP address.
Step 8	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # **set primary-dns** *ip-address*	Specifies the primary DNS IP address.
Step 9	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # **set secondary-dns** *ip-address*	Specifies the secondary DNS IP address.
Step 10	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # **set subnet** *subnet-ip-address*	Specifies the subnet mask.
Step 11	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/eth-if/ip-if/static-ip-params* # **commit buffer**	Commits the transaction to the system configuration.

The following example shows how to configure the initiator to boot using a static IP address and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # enter static-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # set addr
10.104.105.193
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # set default-gw
10.104.105.1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # set primary-dns
 11.11.11.100
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # set secondary-dns
 11.11.11.100
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # set subnet
```

```
255.255.255.0
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # commit-buffer
```

What to Do Next

Create an iSCSI target.

Deleting the Static IP Address Boot Parameters from an iSCSI Initiator

In a service profile, you can delete the static IP address boot parameters from an iSCSI initiator.

Procedure

	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 service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # **scope vnic-iscsi** *iscsi-vnic-name*	Enters the configuration mode for the specified iSCSI vNIC.
Step 4	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # **scope ip-if**	Enters the configuration mode for an IP interface.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # **delete static-ip-params**	Deletes the static IP boot parameters from an initiator.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/static-ip-params* # **commit buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the static IP address boot parameters from the initiator and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # scope ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if # delete static-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # commit-buffer
```

Creating an iSCSI Initiator that Boots Using an IP Address from an IP Pool

In a service profile, you can create an iSCSI initiator and configure it to boot using an IP address from an IP pool that you have created.

Before You Begin

You have completed the following:

- Created an overlay vNIC in a service profile
- Created an iSCSI vNIC in a service profile.

Procedure

	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 service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # **scope iscsi-boot**	Enters the configuration mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # **scope vnic-iscsi** *iscsi-vnic-name*	Enters the configuration mode for the specified iSCSI vNIC.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi* # **scope ip-if**	Enters the configuration mode for the iSCSI Ethernet interface.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # **enter pooled-ip-params**	Specifies that the iSCSI initiator boot using one of the IP addresses from the previously created iSCSI initiator IP pool.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/pooled-ip-params* # **commit buffer**	Commits the transaction to the system configuration.

The following example shows how to create an iSCSI initiator and configure it to boot using an IP address from an IP pool:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # scope ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # enter pooled-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/pooled-ip-params* # commit buffer
```

What to Do Next

Create an iSCSI target.

Deleting the IP Pool Boot Parameter from an iSCSI Initiator

In a service profile, you can create an iSCSI initiator and configure it to boot using an IP address from an IP pool that you have created.

Procedure

	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 service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # **scope iscsi-boot**	Enters the configuration mode for configuring the iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot/ # **scope vnic-iscsi** *iscsi-vnic-name*	Enters the configuration mode for the specified iSCSI vNIC.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # **enter ip-if**	Enters the configuration mode for an IP interface.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # **delete pooled-ip-params**	Specifies that the iSCSI initiator does not use an IP address from an IP pool to boot.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/pooled-ip-params* # **commit buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the boot using an IP address from an IP poo parameter and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # enter ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # delete pooled-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/pooled-ip-params* # commit buffer
```

Creating an iSCSI Initiator that Boots Using DHCP

In a service profile, you can create an iSCSI initiator and configure it to boot using DHCP.

Before You Begin

You have completed the following:

- Created iSCSI overlay vNICs in a service profile.
- Created an iSCSI vNIC in a service profile.

Procedure

	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 service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # **scope iscsi-boot**	Enters the configuration mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # **scope vnic-iscsi** *iscsi-vnic-name*	Enters the configuration mode for the specified iSCSI vNIC.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # **create ip-if**	Creates an IP interface.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # **create dhcp-ip-params**	Specifies that you are setting the initiator to boot using DHCP.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/dhcp-ip-params* # **commit buffer**	Commits the transaction to the system configuration.

The following example shows how to configure the initiator to boot using DHCP and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # create dhcp-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/dhcp-ip-params* # commit-buffer
```

What to Do Next

Create an iSCSI target.

Deleting the DHCP Boot Parameter from an iSCSI Initiator

In a service profile, you can remove the DHCP boot parameter from an iSCSI initiator.

Procedure

	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 service profile organization mode for the service profile.
Step 3	UCS-A /org/service-profile # **scope iscsi-boot**	Enters the configuration mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # **scope vnic-iscsi** *iscsi-vnic-name*	Enters the configuration mode for the specified iSCSI vNIC.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # **enter ip-if**	Enters the configuration mode for an IP interface.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # **delete dhcp-ip-params**	Specifies that the initiator does not use DHCP to boot.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/dhcp-ip-params* # **commit buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the boot using DHCP parameter and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # enter ip-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if* # delete dhcp-ip-params
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ip-if/dhcp-ip-params* # commit-buffer
```

IQN Pools

An IQN pool is a collection of iSCSI Qualified Names (IQNs) for use as initiator identifiers by iSCSI vNICs in a Cisco UCS domain.

IQN pool members are of the form *prefix*:*suffix*:*number*, where you can specify the prefix, suffix, and a block (range) of numbers.

An IQN pool can contain more than one IQN block, with different number ranges and different suffixes, but sharing the same prefix.

Creating an IQN Pool

Note In most cases, the maximum IQN size (prefix + suffix + additional characters) is 223 characters. When using the Cisco UCS NIC M51KR-B adapter, you must limit the IQN size to 128 characters.

Procedure

	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 iqn-pool** *pool-name*	Creates an IQN pool with the specified pool name and enters organization IQN pool mode. This name can be between 1 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	UCS-A /org/iqn-pool # **set iqn-prefix** *prefix*	Specifies the prefix for the IQN block members. Unless limited by the adapter card, the prefix can contain up to 150 characters.
Step 4	UCS-A /org/iqn-pool # **set descr** *description*	(Optional) Provides a description for the IQN pool. Enter up to 256 characters. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 5	UCS-A /org/iqn-pool # **set assignmentorder** {**default** | **sequential**}	This can be one of the following: • **default—**Cisco UCS Manager selects a random identity from the pool. • **sequential—**Cisco UCS Manager selects the lowest available identity from the pool.
Step 6	UCS-A /org/iqn-pool # **create block** *suffix from to*	Creates a block (range) of IQNs, and enters organization IQN pool block mode. You must specify the base suffix, the starting suffix number, and the ending suffix number. The resulting IQN pool members are of the form *prefix***:***suffix***:***number*. The suffix can be up to 64 characters. **Note** An IQN pool can contain more than one IQN block. To create multiple blocks, enter multiple **create block** commands from organization IQN pool mode.
Step 7	UCS-A /org/iqn-pool/block # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create an IQN pool named pool4, provide a description for the pool, specify a prefix and a block of suffixes to be used for the pool, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create iqn-pool pool4
UCS-A /org/iqn-pool* # set iqn-prefix iqn.alpha.com
UCS-A /org/iqn-pool* # set descr "This is IQN pool 4"
UCS-A /org/iqn-pool* # create block beta 3 5
UCS-A /org/iqn-pool/block* # commit-buffer
UCS-A /org/iqn-pool/block #
```

What to Do Next

Include the IQN suffix pool in a service profile and template.

Adding a Block to an IQN Pool

Procedure

	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 iqn-pool** *pool-name*	Enters organization IQN pool mode for the specified pool.
Step 3	UCS-A /org/iqn-pool # **create block** *suffix from to*	Creates a block (range) of IQN suffixes, and enters organization IQN pool block mode. You must specify the base suffix, the starting suffix number, and the ending suffix number. The resulting IQN pool members are of the form *prefix*:*suffix*:*number*. **Note** An IQN pool can contain more than one IQN block. To create multiple blocks, enter multiple **create block** commands from organization IQN pool mode.
Step 4	UCS-A /org/iqn-pool/block # **commit-buffer**	Commits the transaction to the system configuration.
Step 5	UCS-A /org/iqn-pool/block # **exit**	(Optional) Returns to organization IQN pool mode.
Step 6	UCS-A /org/iqn-pool # **show block**	(Optional) Displays the blocks of suffixes.

This example shows how to add a block of IQN suffixes to an IQN pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope iqn-pool pool4
UCS-A /org/iqn-pool # create block beta 3 5
```

```
UCS-A /org/iqn-pool/block* # commit-buffer
UCS-A /org/iqn-pool/block # exit
UCS-A /org/iqn-pool # show block
Block of IQN Names:
    Suffix     From  To
    ---------- ----- --
    beta           3     5

UCS-A /org/iqn-pool #
```

Deleting a Block from an IQN Pool

If you delete an address block from a pool, Cisco UCS Manager does not reallocate any addresses in that block that were assigned to vNICs or vHBAs. All assigned addresses from a deleted block remain with the vNIC or vHBA to which they are assigned until one of the following occurs:

- The associated service profiles are deleted.
- The vNIC or vHBA to which the address is assigned is deleted.
- The vNIC or vHBA is assigned to a different pool.

Procedure

	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 iqn-pool** *pool-name*	Enters organization IQN pool mode for the specified pool.
Step 3	UCS-A /org/iqn-pool # **delete block** *suffix from to*	Deletes a block (range) of IQNs. You must specify the base suffix and the first and last numbers in the block to be deleted.
Step 4	UCS-A /org/iqn-pool # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to delete a block of suffixes from an IQN pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope iqn-pool pool4
UCS-A /org/iqn-pool # delete block beta 0 12
UCS-A /org/iqn-pool* # commit-buffer
UCS-A /org/iqn-pool #
```

Deleting an IQN Pool

If you delete a pool, Cisco UCS Manager does not reallocate any addresses from that pool that were assigned to vNICs or vHBAs. All assigned addresses from a deleted pool remain with the vNIC or vHBA to which they are assigned until one of the following occurs:

- The associated service profiles are deleted.
- The vNIC or vHBA to which the address is assigned is deleted.
- The vNIC or vHBA is assigned to a different pool.

Procedure

	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 iqn-pool** *pool-name*	Deletes the specified IQN pool.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the IQN pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete iqn-pool pool4
UCS-A /org* # commit-buffer
UCS-A /org #
```

Viewing IQN Pool Usage

Procedure

	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 iqn-pool** *pool-name*	Enters organization IQN pool mode for the specified pool.
Step 3	UCS-A /org/iqn-pool # **show pooled**	Displays the assignments of the IQN block members.

The following example shows how to display the assignments of suffixes in the IQN pool named pool4:

```
UCS-A# scope org /
UCS-A /org # scope iqn-pool pool4
UCS-A /org/iqn-pool # show pooled
Pooled:
    Name       Assigned Assigned To Dn
    ---------- -------- --------------
    beta:3     No
    beta:4     No
    beta:5     No
```

```
UCS-A /org/iqn-pool #
```

Creating an iSCSI Static Target

You can create a static target.

Before You Begin

You have already created an iSCSI vNIC.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organi enter the root organization mode, type / as the *or*
Step 2	UCS-A /org # **scope service-profile** *profile-name*	Enters service profile organization mode for the profile to which you want to add an iSCSI target
Step 3	UCS-A /org/service-profile # **scope iscsi-boot**	Enters the mode for configuring iSCSI boot para
Step 4	UCS-A /org/service-profile/iscsi-boot # **scope vnic-iscsi** *iscsi-vnic-name*	Enters the iSCSI vNIC mode for the specified vN
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # **create static-target-if** {**1** | **2**}	Creates a static target for the iSCSI vNIC and as priority level to it. Valid priority levels are 1 or 2.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # **set name** *name*	A regular expression that defines the iSCSI Qualif (IQN) or Extended Unique Identifier (EUI) name iSCSI target. You can enter any alphanumeric characters as we following special characters: • . (period) • : (colon) • - (dash) **Important** This name must be properly format standard IQN or EUI guidelines. The following examples show properly formatte target names: • iqn.2001-04.com.example • iqn.2001-04.com.example:storage:diskarrays-sn • iqn.2001-04.com.example:storage.tape1.sy • iqn.2001-04.com.example:storage.disk2.sy • eui.02004567A425678D

	Command or Action	Purpose
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # **set port** *port-num*	The port associated with the iSCSI target. Enter an integer between 1 and 65535. The default is
Step 8	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # **set auth-name** *auth-profile*	(Optional) If you need the target to authenticate itself and have an authentication profile, you need to specify the na authentication profile. The name of the associated iSCSI authentication pro
Step 9	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # **set ipaddress** *ipv4-address*	The IPv4 address assigned to the iSCSI target.
Step 10	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # **create lun**	Creates the LUN that corresponds to the location of interface.
Step 11	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # **set id** *id-number*	Specifies the target LUN id. Valid values are from 0 65535.
Step 12	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # **exit**	Exits the current configuration mode.
Step 13	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if # **exit**	Exits the current configuration mode.
Step 14	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # **commit-buffer**	Commits the transaction to the system configuration
Step 15	Repeat steps 5 through 14 to create a second static target.	(Optional)

The following example shows how to create two iSCSI static target interfaces and commit the transaction:

```
UCS-A # scope org test
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create static-target-if 1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set name statictarget1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set port 3260
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set auth-name
authprofile1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set ip-address
192.168.10.10
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # create lun
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # set id 1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create static-target-if 2
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set ipaddress
192.168.10.11
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set name statictarget2
```

```
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set port 3260
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # set auth-name
authprofile1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # create lun
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # set id 1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if/lun* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/static-target-if* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer
```

What to Do Next

To configure a second iSCSI device, repeat the steps for creating an iSCSI vNIC, initiator, and target.

Deleting an iSCSI Static Target

You can delete an iSCSI static target. However, you must have at least one iSCSI static target remaining after you delete one. Therefore, you must have two iSCSI static targets in order to delete one of them.

Note If you have two iSCSI targets and you delete the first priority target, the second priority target becomes the first priority target, although the Cisco UCS Manager still shows it as the second priority target.

Procedure

	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 service profile organization mode for the service profile to which you want to add an iSCSI target.
Step 3	UCS-A /org/service-profile # **scope iscsi-boot**	Enters the mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # **scope vnic-iscsi** *iscsi-vnic-name*	Enters the iSCSI vNIC mode for the specified vNIC name.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # **delete static-target-if**	Deletes the static target for the iSCSI vNIC.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI static target and commit the transaction:

```
UCS-A # scope org test
UCS-A /org # scope service-profile sample
UCS-A /org # scope iscsi-boot
```

```
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi trial
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # delete static-target-if 1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi #
```

Creating an iSCSI Auto Target

You can create an iSCSI auto target with or without the vendor IDs.

Before You Begin

These prerequisites must be met before creating iSCSI auto target:

- You have already created an iSCSI vNIC in a service profile.
- You have considered the prerequisites for the VIC that you are using. For more information, see iSCSI Boot Guidelines and Prerequisites, on page 531

Procedure

	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 service profile organization mode for the service profile that you want to add an iSCSI target interface to.
Step 3	UCS-A /org # **scope iscsi-boot** **Example:**	Enters the mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # **scope vnic-iscsi** *iscsi-vnic-name*	Enters iSCSI vNIC service profile organization mode for the specified vNIC name.
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/ # **create auto-target-if**	Creates an auto target for the iSCSI vNIC. If you plan to use an auto target without the vendor ID, you must configure an initiator name. For more information, see Creating an iSCSI vNIC in a Service Profile, on page 544.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/auto-target-if* # **set dhcp-vendor-id** *vendor-id*	(Optional) Sets a vendor ID for the auto target. The vendor ID can be up to 32 alphanumeric characters.
Step 7	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/auto-target-if* # **exit**	Exists the current configuration mode.

	Command or Action	Purpose
Step 8	UCS-A /org/service-profile/iscis-boot/vnic-iscsi # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create an iSCSI auto target *without* a vendor ID and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create auto-target-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/auto-target-if* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer
```

The following example shows how to create an iSCSI auto target *with* a vendor ID and commit the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # create auto-target-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/auto-target-if* # set dhcp-vendor-id
iSCSI_Vendor
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi/auto-target-if* # exit
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer
```

What to Do Next

To configure a second iSCSI device, repeat the steps for creating an iSCSI vNIC, initiator, and target.

Deleting an iSCSI Auto Target

You can delete an auto target only if you have a static target set.

Procedure

	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 the service profile mode for the service profile to which you want to add an iSCSI target.
Step 3	UCS-A /org/service-profile # **scope iscsi-boot**	Enters the mode for configuring iSCSI boot parameters.
Step 4	UCS-A /org/service-profile/iscsi-boot # **scope vnic-iscsi** *iscsi-vnic-name*	Enters the iSCSI vNIC mode for the specified vNIC name.

	Command or Action	Purpose
Step 5	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # **delete auto-target-if**	Deletes the auto target.
Step 6	UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI auto target and commit the transaction:

```
UCS-A # scope org test
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # scope iscsi-boot
UCS-A /org/service-profile/iscsi-boot # scope vnic-iscsi iSCSI1
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # delete auto-target-if
UCS-A /org/service-profile/iscsi-boot/vnic-iscsi # commit-buffer
```

Verifying iSCSI Boot

Use the KVM console to view the boot up messages as the adapter is booting. For information on how to access the KVM console, see the *Starting the KVM Console* chapter.

This step can only be performed using the Cisco UCS Manager GUI. For more information, see the *Starting the KVM Console* chapter in the *UCS Manager GUI Configuration Guide.*

- For the Cisco UCS M51KR-B Broadcom BCM57711, the following message appears: Logging in the 1st iSCSI Target…. Succeeded.
- For the Cisco UCS M81KR Virtual Interface Card, the following message appears: Option ROM installed successfully.

LAN Boot

You can configure a boot policy to boot one or more servers from a centralized provisioning server on the LAN. A LAN (or PXE) boot is frequently used to install operating systems on a server from that LAN server.

You can add more than one type of boot device to a LAN boot policy. For example, you could add a local disk or virtual media boot as a secondary boot device.

Configuring a LAN Boot for a Boot Policy

Before You Begin

Create a boot policy to contain the LAN boot configuration.

Procedure

	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 lan**	Creates a LAN boot for the boot policy and enters organization boot policy LAN mode.
Step 4	UCS-A /org/boot-policy/lan # **set order** {**1** | **2** | **3** | **4**}	Specifies the boot order for the LAN boot.
Step 5	UCS-A /org/boot-policy/lan # **create path** {**primary** | **secondary**}	Creates a primary or secondary LAN boot path and enters organization boot policy LAN path mode.
Step 6	UCS-A /org/boot-policy/lan/path # **set vnic** *vnic-name*	Specifies the vNIC to use for the LAN path to the boot image.
Step 7	UCS-A /org/boot-policy/lan/path # **commit-buffer**	Commits the transaction to the system configuration.

The following example enters the boot policy named lab2-boot-policy, creates a LAN boot for the policy, sets the boot order to 2, creates primary and secondary paths using the vNICs named vNIC1 and vNIC2 , and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy lab2-boot-policy
UCS-A /org/boot-policy* # create lan
UCS-A /org/boot-policy/lan* # set order 2
UCS-A /org/boot-policy/lan* # create path primary
UCS-A /org/boot-policy/lan/path* # set vnic vNIC1
UCS-A /org/boot-policy/lan/path* # exit
UCS-A /org/boot-policy/lan* # create path secondary
UCS-A /org/boot-policy/lan/path* # set vnic vNIC2
UCS-A /org/boot-policy/lan/path* # commit-buffer
UCS-A /org/boot-policy/lan/path #
```

What to Do Next

Include the boot policy in a service profile and template.

Local Devices Boot

Cisco UCS Manager allows you to boot from different local devices.

Note For Cisco UCS M3 and M4 blade and rack servers using enhanced boot order, you can select both top-level and second-level boot devices. For Cisco UCS M1 and M2 blade and rack servers using standard boot order, you can only select a top-level device.

Local Disk Boot

If a server has a local drive, you can configure a boot policy to boot the server from the top-level local disk device or from any of the following second-level devices:

- Local LUN—Enables boot from local disk or local LUN.
- Local JBOD—Enables boot from a bootable JBOD.
- SD card—Enables boot from SD card.
- Internal USB—Enables boot for internal USB.
- External USB—Enables boot from external USB.
- Embedded Local LUN—Enables boot from the embedded local LUN on the Cisco UCS 240 M4 server.
- Embedded Local Disk—Enables boot from the embedded local disk on the Cisco UCS C240 M4SX and the M4L servers.

Note Second-level devices are only available for Cisco UCS M3 and M4 blade and rack servers using enhanced boot order. For Cisco UCS M1 and M2 blade and rack servers using standard boot order, you can choose only the top-level **Add Local Disk**.

Virtual Media Boot

You can configure a boot policy to boot one or more servers from a virtual media device that is accessible from the server. A virtual media device mimics the insertion of a physical CD/DVD disk (read-only) or floppy disk (read-write) into a server. This type of server boot is typically used to manually install operating systems on a server.

Note Second-level devices are only available for Cisco UCS M3 and M4 blade and rack servers using enhanced boot order. For Cisco UCS M1 and M2 blade and rack servers using standard boot order, you can choose only the top-level **Add CD/DVD** or **Add Floppy**.

Remote Virtual Drive Boot

You can configure a boot policy to boot one or more servers from a remote virtual drive that is accessible from the server.

Configuring a Local Disk Boot for a Boot Policy

You can also create a local boot policy that is restricted to a service profile or service profile template. However, Cisco recommends that you create a global boot policy that can be included in multiple service profiles or service profile templates.

You can add more than one type of boot device to a boot policy. For example, you could add a virtual media boot as a secondary boot device.

Note Beginning with Release 2.2, if you want to add any top-level local storage device to the boot order, you must use **create local-any** after the **create local** command. If you have any policies from previous releases that contain a local storage device, they will be modified to use local-any during upgrade.

Procedure

	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-any** | **local-lun** | **sd-card** | **usb-extern** | **usb-intern** }	Specifies the type of local storage. This can be one of the following: • local-any—Any type of local storage device. This option can be used in either legacy or UEFI boot mode. **Note** Cisco UCS M1 and M2 blade and rack servers using standard boot order can only use local-any. • local-lun—A local hard disk drive. • sd-card—An SD card. • usb-extern—An external USB card. • usb-intern—An internal USB card. For Cisco UCS M3 and M4 blade and rack servers using enhanced boot order, you can select both top-level and second-level boot devices. For Cisco UCS M1 and M2 blade and rack

	Command or Action	Purpose
		servers using standard boot order, you can only select a top-level device.
Step 6	UCS-A /org/boot-policy/storage/local/*local-storage-device* # **set order** *order_number*	Sets the boot order for the specified local storage device. Enter an integer between 1 and 16. When using the enhanced boot order on Cisco UCS M3 servers, or M4 servers, the boot order that you define is used. For standard boot mode using the terms "primary" or "secondary" do not imply a boot order. The effective order of boot devices within the same device class is determined by the PCIe bus scan order.
Step 7	UCS-A /org/boot-policy/storage/local/*local-storage-device* # **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, set the boot order to 3, 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/sd-card* # set order 3
UCS-A /org/boot-policy/storage/local/sd-card* # commit-buffer
UCS-A /org/boot-policy/storage/local/sd-card #
```

The following example shows how to create a local SD card boot for the service profile SP_lab1, set the boot order to 3, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile SP_lab1
UCS-A /org/service-profile # create boot-definition
UCS-A /org/service-profile/boot-definition* # create storage
UCS-A /org/service-profile/boot-definition/storage* # create local
UCS-A /org/service-profile/boot-definition/storage/local* # create sd-card
UCS-A /org/service-profile/boot-definition/storage/local* # set order 3
UCS-A /org/service-profile/boot-definition/storage/local* # commit-buffer
UCS-A /org/service-profile/boot-definition/storage/local #
```

The following example shows how to create any top-level local device boot for the service profile SP_lab1, set the boot order to 3, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile SP_lab1
UCS-A /org/service-profile # create boot-definition
UCS-A /org/service-profile/boot-definition* # create storage
UCS-A /org/service-profile/boot-definition/storage* # create local
UCS-A /org/service-profile/boot-definition/storage/local* # create local-any
UCS-A /org/service-profile/boot-definition/storage/local/local-any* # set order 3
UCS-A /org/service-profile/boot-definition/storage/local/local-any* # commit-buffer
UCS-A /org/service-profile/boot-definition/storage/local/local-any #
```

What to Do Next

Include the boot policy in a service profile and template.

Configuring a Virtual Media Boot for a Boot Policy

Note Virtual Media requires the USB to be enabled. If you modify the BIOS settings that affect the USB functionality, you also affect the Virtual Media. Therefore, Cisco recommends that you leave the following USB BIOS defaults for best performance:

- Make Device Non Bootable—set to **disabled**
- USB Idle Power Optimizing Setting—set to **high-performance**

Before You Begin

Create a boot policy to contain the virtual media boot configuration.

Procedure

	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 virtual-media** {**read-only** | **read-only-local** | **read-only-remote** | **read-write** | **read-write-drive** | **read-write-local** | **read-write-remote**}	Creates the specified virtual media boot for the boot policy and enters organization boot policy virtual media mode. This can be one of the following: • read-only—Local or remote CD/DVD. This option can be used in either legacy or UEFI boot mode. • read-only-local—Local CD/DVD. • read-only-remote—Remote CD/DVD. • read-write—Local or remote floppy disk drive. This option can be used in either legacy or UEFI boot mode. • read-write-drive—Remote USB drive. • read-write-local—Local floppy disk drive. • read-write-remote—Remote floppy disk drive. **Note** For Cisco UCS M3 and M4 blade and rack servers using enhanced boot order, you can select both top-level and second-level boot devices. For Cisco UCS M1 and M2 blade and rack servers using standard boot order, you can only select a top-level device.

	Command or Action	Purpose
Step 4	UCS-A /org/boot-policy/virtual-media # **set order** *order_number*	Sets the boot order for the virtual-media boot. Enter an integer between 1 and 16.
Step 5	UCS-A /org/boot-policy/virtual-media # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to enter the boot policy named lab3-boot-policy, create a CD/DVD virtual media boot, set the boot order to 3, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy lab3-boot-policy
UCS-A /org/boot-policy* # create virtual-media read-only-local
UCS-A /org/boot-policy/virtual-media* # set order 3
UCS-A /org/boot-policy/virtual-media* # commit-buffer
```

What to Do Next

Include the boot policy in a service profile and template.

Creating a CIMC vMedia Boot Policy

You can also create a local boot policy that is restricted to a service profile or service profile template. However, Cisco recommends that you create a global boot policy that can be included in multiple service profiles or service profile templates.

Procedure

	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 # **create boot-policy** *policy-name*	Creates a boot policy with the specified policy name, and enters organization boot policy mode.
Step 3	UCS-A /org/boot-policy* # **create virtual-media** *?*	Displays a list of local and remote devices to your can access and boot.
Step 4	UCS-A /org/boot-policy* # **create virtual-media** {**access** | **vMediaMappingName**}	Displays a list of local and remote devices to your can access and boot.
Step 5	UCS-A /org/boot-policy* # **create virtual-media read-write-remote-drive vMediaMap0**}	Creates vMedia Boot Device configuration for specified vMedia.
Step 6	UCS-A /org/boot-policy/virtual-media* # **commit-buffer**	Commits the transaction to the system configuration.

	Command or Action	Purpose
Step 7	UCS-A /org/boot-policy/virtual-media* # **show detail expand**	Displays the following boot order. **Boot virtual media:** **Order:** 1 **Access:** Read Write Remote vMedia Drive **Name:** vmediaMap0

The following example creates a CIMC vMedia boot policy.

```
UCS-A# scope org /
UCS-A /org* # create boot-policy boot-policy vm-vmediamap-boot
UCS-A /org/boot-policy* # create virtual-media
```

Viewing a CIMC vMedia Mount

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis_id*/*blade_id*	Enters chassis server mode for the specified server.
Step 2	UCS-A# /chassis/server # **scope cimc**	Enters CIMC mode.
Step 3	UCS-A /chassis/server/cimc # **show vmedia-mapping-list detail expand**	Displays the vMedia mapping details.

The following example shows how to view a CIMC vMedia mount.

```
UCS-A# scope server 1/2
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # show vmedia-mapping-list detail expand

vMedia Mapping List:
vMedia Mapping:
Disk Id: 1
Mapping Name: cdd
Device Type: Cdd
Remote IP: 172.31.1.167
Image Path: cifs
Image File Name: ubunt-14.11-desktop-i386.iso
Mount Protocol: Cifs
Mount Status: Mounted
Error: None
Password:
User ID: Adminstrator

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

Configuring an EFI Shell Boot for a Boot Policy

You can create a boot policy with an EFI Shell as the boot device. Booting from an EFI Shell prevents loss of data and provides more options to script, debug, and control various booting scenarios. EFI Shell is supported as a boot device only in the **Uefi** boot mode.

Before You Begin

To configure EFI Shell as a boot device, ensure that the boot mode is set to **Uefi**.

Important In an EFI Shell boot policy, If you edit the boot mode to **Legacy**, Cisco UCS Manager removes the EFI Shell boot device and sets the boot policy to default.

Procedure

	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 boot-policy** *policy-name*	Creates a boot policy with the specified policy name, and enters organization boot policy mode.
Step 3	UCS-A /org/boot-policy* # **set boot-mode** {**legacy** | **uefi**}	Specifies whether the servers using this boot policy are using UEFI or legacy boot mode. **Note** To configure EFI Shell as a boot device, ensure that the boot mode is set to **Uefi**
Step 4	UCS-A /org/boot-policy* # **create efi-shell**	Creates an EFI Shell boot for the boot policy and enters organization boot policy mode.
Step 5	UCS-A /org/boot-policy/efi-shell* # **commit-buffer**	Commits the transaction to the system configuration.

```
UCS-A# scope org
UCS-A /org # create boot-policy efi_shell
UCS-A /org/boot-policy* # set boot-mode uefi
UCS-A /org/boot-policy* # create efi-shell
UCS-A /org/boot-policy/efi-shell* # commit-buffer
```

What to Do Next

Include the boot policy in a service profile and template.

Deleting a Boot Policy

Procedure

	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 # **delete boot-policy** *policy-name*	Deletes the specified boot policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the boot policy named boot-policy-LAN and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete boot-policy boot-policy-LAN
UCS-A /org* # commit-buffer
UCS-A /org #
```

UEFI Boot Parameters

UEFI boot mode for servers is dependent on information that is stored on the platform hardware. The boot entry, which contains information about the UEFI OS boot loader, is stored in the BIOS flash of the server. In Cisco UCS Manager releases earlier than Release 2.2(4), when a service profile is migrated from one server to another server, the boot loader information is not available on the destination server. Hence, the BIOS cannot load the boot loader information for the server to boot in UEFI boot mode.

Cisco UCSM Release 2.2(4) introduces UEFI boot parameters to provide the BIOS with information about the location of the UEFI OS boot loader on the destination server from where the BIOS loads it. Now, the server can use the boot loader information and boot in UEFI boot mode.

Guidelines and Limitations for UEFI Boot Parameters

- You can configure UEFI boot parameters only if the boot mode is UEFI.
- When you upgrade Cisco UCS Manager to Release 2.2(4), UEFI boot failure during service profile migration is not handled automatically. You must explicitly create the UEFI boot parameters in the target device to successfully boot to the UEFI-capable OS.
- UEFI boot parameters are supported on all M3 and higher servers that support second-level boot order.
- You can specify UEFI boot parameters for the following device types:
  - SAN LUN
  - ISCSI LUN
  - Local LUN

- UEFI boot parameters are specific to each operating system. You can specify UEFI boot parameters for the following operating systems:
  - VMware ESX
  - SuSE Linux
  - Microsoft Windows
  - Red Hat Enterprise Linux 7

Configuring UEFI Boot Parameters for a Local LUN

Before You Begin

Ensure that the boot mode for the local LUN is set to UEFI.

Procedure

	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 # **scope storage**	Enters organization boot policy storage mode for the boot policy.
Step 4	UCS-A /org/boot-policy/storage # **scope local**	Enters the boot policy local storage mode.
Step 5	UCS-A /org/boot-policy/storage/local/ # **scope** {**local-any** | **local-lun** | **sd-card** | **usb-extern** | **usb-intern** }	Specifies the type of local storage. This can be one of the following: • local-any—Any type of local storage device. This option

	Command or Action	Purpose
		can be used in either legacy or UEFI boot mode. **Note** Cisco UCS M1 and M2 blade and rack servers using standard boot order can only use local-any. • local-lun—A local hard disk drive. • sd-card—An SD card. • usb-extern—An external USB card. • usb-intern—An internal USB card. **Important** The only type of local storage for which you can configure UEFI boot parameters is **local-lun**.

	Command or Action	Purpose
Step 6	UCS-A /org/boot-policy/storage/local/local-lun # **scope local-lun-image-path** {**primary** | **secondary**}	Enters the image path for the local LUN.
Step 7	UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path # **create uefi-boot-param**	Creates UEFI boot parameters and enters UEFI boot parameter mode.
Step 8	UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # **set bootloader-name** *name*	Sets the name of the boot loader.
Step 9	UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # **set bootloader-path** *path*	Sets the path of the boot loader.
Step 10	UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # **set boot-description** "*description*"	Sets a description for the boot loader.
Step 11	UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create UEFI boot parameters for a local LUN, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy bp1
UCS-A /org/boot-policy* # scope storage
UCS-A /org/boot-policy/storage* # scope local
UCS-A /org/boot-policy/storage/local* # scope local-lun
UCS-A /org/boot-policy/storage/local/local-lun # scope local-lun-image-path primary
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path # create uefi-boot-param
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # set
 bootloader-name grub.efi
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # set
 bootloader-path EFI\redhat
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* # set
 boot-description "Red Hat Enterprise Linux"
UCS-A /org/boot-policy/storage/local/local-lun/local-lun-image-path/uefi-boot-param* #
commit-buffer
```

Configuring UEFI Boot Parameters for an iSCSI LUN

Before You Begin

Ensure that the boot mode for the iSCSI LUN is set to UEFI.

Procedure

	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 # **scope iscsi**	Enters organization boot policy iSCSI mode for the boot policy.
Step 4	UCS-A /org/boot-policy/iscsi # **scope path** {**primary** | **secondary**}	Enters the image path for the iSCSI LUN.
Step 5	UCS-A /org/boot-policy/iscsi/path # **create uefi-boot-param**	Creates UEFI boot parameters and enters UEFI boot parameter mode.
Step 6	UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # **set bootloader-name** *name*	Sets the name of the boot loader.
Step 7	UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # **set bootloader-path** *path*	Sets the path of the boot loader.
Step 8	UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # **set boot-description** "*description*"	Sets a description for the boot loader.
Step 9	UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create UEFI boot parameters for an iSCSI LUN, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy bp2
UCS-A /org/boot-policy* # scope iscsi
UCS-A /org/boot-policy/iscsi # scope path primary
UCS-A /org/boot-policy/iscsi/path # create uefi-boot-param
UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # set bootloader-name grub.efi
UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # set bootloader-path EFI\redhat
UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # set boot-description "Red Hat Enterprise
 Linux"
UCS-A /org/boot-policy/iscsi/path/uefi-boot-param* # commit-buffer
```

Configuring UEFI Boot Parameters for a SAN LUN

Before You Begin

Ensure that the boot mode for the SAN LUN is set to UEFI.

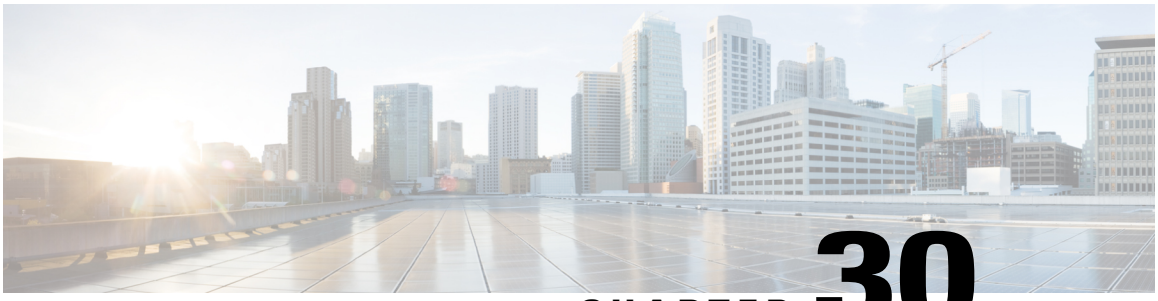
Procedure

	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 # **scope san**	Enters organization boot policy SAN mode for the boot policy.
Step 4	UCS-A /org/boot-policy/san # **scope san-image** {**primary** | **secondary**}	Enters the SAN image.
Step 5	UCS-A /org/boot-policy/san/san-image # **scope path** {**primary** | **secondary**}	Enters the image path for the SAN LUN.
Step 6	UCS-A /org/boot-policy/san/san-image/path # **create uefi-boot-param**	Creates UEFI boot parameters and enters UEFI boot parameter mode.
Step 7	UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # **set bootloader-name** *name*	Sets the name of the boot loader.
Step 8	UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # **set bootloader-path** *path*	Sets the path of the boot loader.
Step 9	UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # **set boot-description** "*description*"	Sets a description for the boot loader.
Step 10	UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create UEFI boot parameters for a SAN LUN, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope boot-policy bp3
UCS-A /org/boot-policy* # scope san
UCS-A /org/boot-policy/san # scope san-image primary
UCS-A /org/boot-policy/san/san-image # scope path primary
UCS-A /org/boot-policy/san/san-image/path # create uefi-boot-param
UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # set bootloader-name grub.efi
```

```
UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # set bootloader-path EFI\redhat
UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # set boot-description "Red Hat
 Enterprise Linux"
UCS-A /org/boot-policy/san/san-image/path/uefi-boot-param* # commit-buffer
```



CHAPTER 30

Deferring Deployment of Service Profile Updates

This chapter includes the following sections:

Service Profile Deferred Deployments

Some modifications to a service profile or to an updating service profile template can be disruptive and require a reboot of the server. You can, however, configure deferred deployment to control when those disruptive configuration changes are implemented. For example, you can choose to deploy the service profile changes immediately or have them deployed during a specified maintenance window. You can also choose whether or not a service profile deployment requires explicit user acknowledgment.

Deferred deployment is available for all configuration changes that occur through the association of a service profile with a server. These configuration changes can be prompted by a change to a service profile, to a policy that is included in a service profile, or to an updating service profile template. For example, you can defer the upgrade and activation of firmware through host firmware packages and management firmware packages, such as server BIOS, RAID controller, host HBA, and network adapters. However, you cannot defer the direct deployment of firmware images for components that do not use either of the firmware packages, such as Cisco UCS Manager, fabric interconnects, and I/O modules.

Deferred deployment is not available for the following actions which require the reboot of a server:

- Initial association of a service profile with a server
- Final disassociation of a service profile from a server, without associating the service profile with a different server
- Decommissioning a server
- Re-acknowledging a server
- Resetting a server

If you want to defer the deployment of service profile changes, you must configure one or more maintenance policies and configure each service profile with a maintenance policy. If you want to define the time period when the deployment should occur, you also need to create at least one schedule with one or more recurring occurrences or one time occurrences, and include that schedule in a maintenance policy.

Schedules for Deferred Deployments

A schedule contains a set of occurrences. These occurrences can be one time only or can recur at a specified time and day each week. The options defined in the occurrence, such as the duration of the occurrence or the maximum number of tasks to be run, determine whether a service profile change is deployed. For example, if a change cannot be deployed during a given maintenance window because the maximum duration or number of tasks was reached, that deployment is carried over to the next maintenance window.

Each schedule checks periodically to see whether the Cisco UCS domain entered one or more maintenance windows. If so, the schedule executes the deployments that are eligible according to the constraints specified in the maintenance policy.

A schedule contains one or more occurrences, which determine the maintenance windows associated with that schedule. An occurrence can be one of the following:

One Time Occurrence

One time occurrences define a single maintenance window. These windows continue until the maximum duration of the window or the maximum number of tasks that can be run in the window is reached.

Recurring Occurrence

Recurring occurrences define a series of maintenance windows. These windows continue until the maximum number of tasks or the end of the day specified in the occurrence was reached.

Maintenance Policy

A maintenance policy determines how Cisco UCS Manager reacts when a change that requires a server reboot is made to a service profile associated with a server or to an updating service profile bound to one or more service profiles.

The maintenance policy specifies how Cisco UCS Manager deploys the service profile changes. The deployment can occur in one of the following ways:

- Immediately
- When acknowledged by a user with administrator privileges
- Automatically at the time specified in a schedule
- On the next reboot or shutdown without waiting for the user acknowledgment or the timer scheduling option

Note If the **On Next Boot** option is enabled in a maintenance policy, and you downgrade from Cisco UCS Manager Release 3.1(1) or later releases to any release earlier than Cisco UCS Manager Release 2.2(8), firmware downgrade will fail. Disable **On Next Boot** from the maintenance policy to continue with the downgrade.

You can use the soft shutdown timer in the maintenance policy to configure the wait time for performing a hard shutdown. The soft shutdown timer is applicable when you reboot the server for the following:

- Reset the server using the **Gracefully Restart OS** option.
- Shut down the server with the **In case of graceful shutdown failure, a hard shutdown will be issued after X seconds** option.
- Modify a service profile that requires a server reboot.

If the maintenance policy is configured to deploy the change during a scheduled maintenance window, the policy must include a valid schedule. The schedule deploys the changes in the first available maintenance window.

Note A maintenance policy only prevents an immediate server reboot when a configuration change is made to an associated service profile. However, a maintenance policy does not prevent the following actions from taking place right away:

- Deleting an associated service profile from the system
- Disassociating a server profile from a server
- Directly installing a firmware upgrade without using a service policy
- Resetting the server

Pending Activities for Deferred Deployments

If you configure a deferred deployment in a Cisco UCS domain, Cisco UCS Manager enables you to view all pending activities. You can see activities that are waiting for user acknowledgement and those that are scheduled.

If a Cisco UCS domain has pending activities, Cisco UCS Manager GUI notifies users with admin privileges when they log in.

Cisco UCS Manager displays information about all pending activities, including the following:

- Name of the service profile to deploy and associate with a server
- Server affected by the deployment
- Disruption caused by the deployment
- Change performed by the deployment

Note You cannot specify the maintenance window in which a specific pending activity is applied to the server. The maintenance window depends upon how many activities are pending and which maintenance policy is assigned to the service profile. However, any user with admin privileges can manually initiate a pending activity and reboot the server immediately, whether it is waiting for user acknowledgment or for a maintenance window.

Guidelines and Limitations for Deferred Deployments

Cannot Undo All Changes to Service Profiles or Service Profile Templates

If you cancel a pending change, Cisco UCS Manager attempts to roll back the change without rebooting the server. However, for complex changes, Cisco UCS Manager may have to reboot the server a second time to roll back the change. For example, if you delete a vNIC, Cisco UCS Manager reboots the server according to the maintenance policy included in the service profile. You cannot cancel this reboot and change, even if you restore the original vNIC in the service profile. Instead, Cisco UCS Manager schedules a second deployment and reboot of the server.

Association of Service Profile Can Exceed Boundaries of Maintenance Window

After Cisco UCS Manager begins the association of the service profile, the scheduler and maintenance policy do not have any control over the procedure. If the service profile association does not complete within the allotted maintenance window, the process continues until it is completed. For example, this can occur if the association does not complete in time because of retried stages or other issues.

Cannot Specify Order of Pending Activities

Scheduled deployments run in parallel and independently. You cannot specify the order in which the deployments occur. You also cannot make the deployment of one service profile change dependent upon the completion of another.

Cannot Perform Partial Deployment of Pending Activity

Cisco UCS Manager applies all changes made to a service profile in the scheduled maintenance window. You cannot make several changes to a service profile at the same time and then have those changes be spread across several maintenance windows. When Cisco UCS Manager deploys the service profile changes, it updates the service profile to match the most recent configuration in the database.

Configuring Schedules

Creating a Schedule

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **create scheduler** *sched-name*	Creates a scheduler and enters scheduler mode.
Step 3	UCS-A /system/scheduler # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a scheduler called maintenancesched and commits the transaction:

```
UCS-A# scope system
UCS-A /system # create scheduler maintenancesched
UCS-A /system/scheduler* # commit-buffer
UCS-A /system/scheduler #
```

What to Do Next

Create a one time occurrence or recurring occurrence for the schedule.

Creating a One Time Occurrence for a Schedule

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope schedule** *sched-name*	Enters scheduler system mode.
Step 3	UCS-A /system/scheduler # **create occurrence one-time** *occurrence-name*	Creates a one-time occurrence.
Step 4	UCS-A /system/scheduler/one-time # **set date** *month day-of-month year hour minute*	Sets the date and time this occurrence should run.
Step 5	UCS-A /system/scheduler/one-time # **set concur-tasks** {**unlimited** | *max-num-concur-tasks*	(Optional) Sets the maximum number of tasks that can run concurrently during this occurrence. If the maximum number of tasks is reached, the scheduler waits for the amount of time set in the

	Command or Action	Purpose
		minimum interval property before scheduling new tasks.
Step 6	UCS-A /system/scheduler/one-time # **set max-duration** {**none** | *num-of-days num-of-hours num-of-minutes num-of-seconds*}	(Optional) Sets the maximum length of time that this schedule occurrence can run. Cisco UCS completes as many scheduled tasks as possible within the specified time.
Step 7	UCS-A /system/scheduler/one-time # **set min-interval** {**none** | *num-of-days num-of-hours num-of-minutes num-of-seconds*}	(Optional) Sets the minimum length of time that the system should wait before starting a new task.
Step 8	UCS-A /system/scheduler/one-time # **set proc-cap** {**unlimited** | *max-num-of-tasks*}	(Optional) Sets the maximum number of scheduled tasks that can be run during this occurrence.
Step 9	UCS-A /system/scheduler/one-time # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a one time occurrence called onetimemaint for a scheduler called maintsched, sets the maximum number of concurrent tasks to 5, sets the start date to April 1, 2011 at 11:00, and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope scheduler maintsched
UCS-A /system/scheduler # create occurrence one-time onetimemaint
UCS-A /system/scheduler/one-time* # set date apr 1 2011 11 00
UCS-A /system/scheduler/one-time* # set concur-tasks 5
UCS-A /system/scheduler/one-time* # commit-buffer
UCS-A /system/scheduler/one-time #
```

Creating a Recurring Occurrence for a Schedule

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope schedule** *sched-name*	Enters scheduler system mode.
Step 3	UCS-A /system/scheduler # **create occurrence recurring** *occurrence-name*	Creates a recurring occurrence.
Step 4	UCS-A /system/scheduler/recurring # **set day** {**even-day** | **every-day** | **friday** | **monday** | **never** | **odd-day** | **saturday** | **sunday** | **thursday** | **tuesday** | **wednesday**}	(Optional) Specifies the day on which Cisco UCS runs an occurrence of this schedule. By default, this property is set to never.

	Command or Action	Purpose
Step 5	UCS-A /system/scheduler/recurring # **set hour** *hour*	(Optional) Specifies the hour at which this occurrence starts. **Note** Cisco UCS ends all recurring occurrences on the same day in which they start, even if the maximum duration has not been reached. For example, if you specify a start time of 11 p.m. and a maximum duration of 3 hours, Cisco UCS starts the occurrence at 11 p.m. but ends it at 11:59 p.m. after only 59 minutes.
Step 6	UCS-A /system/scheduler/recurring # **set minute** *minute*	(Optional) Specifies the minute at which this occurrence starts.
Step 7	UCS-A /system/scheduler/recurring # **set concur-tasks** {**unlimited** | *max-num-concur-tasks*	(Optional) Sets the maximum number of tasks that can run concurrently during this occurrence. If the maximum number of tasks is reached, the scheduler waits for the amount of time set in the minimum interval property before scheduling new tasks.
Step 8	UCS-A /system/scheduler/recurring # **set max-duration** {**none** | *num-of-hours num-of-minutes num-of-seconds*}	(Optional) Sets the maximum length of time that this schedule occurrence can run. Cisco UCS completes as many scheduled tasks as possible within the specified time.
Step 9	UCS-A /system/scheduler/recurring # **set min-interval** {**none** | *num-of-days num-of-hours num-of-minutes num-of-seconds*}	(Optional) Sets the minimum length of time that the system should wait before starting a new task.
Step 10	UCS-A /system/scheduler/recurring # **set proc-cap** {**unlimited** | *max-num-of-tasks*}	(Optional) Sets the maximum number of scheduled tasks that can be run during this occurrence.
Step 11	UCS-A /system/scheduler/recurring # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a recurring occurrence called recurringmaint for a scheduler called maintsched, sets the maximum number of concurrent tasks to 5, sets the day this occurrence will run to even days, sets the time it will start to 11:05, and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope scheduler maintsched
UCS-A /system/scheduler # create occurrence recurring recurringmaint
UCS-A /system/scheduler/recurring* # set day even-day
UCS-A /system/scheduler/recurring* # set hour 11
UCS-A /system/scheduler/recurring* # set minute 5
UCS-A /system/scheduler/recurring* # set concur-tasks 5
UCS-A /system/scheduler/recurring* # commit-buffer
UCS-A /system/scheduler/recurring #
```

Deleting a One Time Occurrence from a Schedule

If this is the only occurrence in a schedule, that schedule is reconfigured with no occurrences. If the schedule is included in a maintenance policy and that policy is assigned to a service profile, any pending activities related to the server associated with the service profile cannot be deployed. You must add a one time occurrence or a recurring occurrence to the schedule to deploy the pending activity.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope scheduler** *sched-name*	Enters scheduler system mode.
Step 3	UCS-A /system/scheduler # **delete occurrence one-time** *occurrence-name*	Deletes the specified one-time occurrence.
Step 4	UCS-A /system/scheduler # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a one time occurrence called onetimemaint from scheduler maintsched and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope scheduler maintsched
UCS-A /system/scheduler # delete occurrence one-time onetimemaint
UCS-A /system/scheduler* # commit-buffer
UCS-A /system/scheduler #
```

Deleting a Recurring Occurrence from a Schedule

If this is the only occurrence in a schedule, that schedule is reconfigured with no occurrences. If the schedule is included in a maintenance policy and that policy is assigned to a service profile, any pending activities related to the server associated with the service profile cannot be deployed. You must add a one time occurrence or a recurring occurrence to the schedule to deploy the pending activity.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope scheduler** *sched-name*	Enters scheduler system mode.
Step 3	UCS-A /system/scheduler # **delete occurrence recurring** *occurrence-name*	Deletes the specified recurring occurrence.
Step 4	UCS-A /system/scheduler # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a recurring occurrence called onetimemaint from scheduler maintsched and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope scheduler maintsched
UCS-A /system/scheduler # delete occurrence recurring onetimemaint
UCS-A /system/scheduler* # commit-buffer
UCS-A /system/scheduler #
```

Deleting a Schedule

If this schedule is included in a maintenance policy, the policy is reconfigured with no schedule. If that policy is assigned to a service profile, any pending activities related to the server associated with the service profile cannot be deployed. You must add a schedule to the maintenance policy to deploy the pending activity.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **delete scheduler** *sched-name*	Deletes a scheduler and enters scheduler mode.
Step 3	UCS-A /system # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a scheduler called maintenancesched and commits the transaction:

```
UCS-A# scope system
UCS-A /system # delete scheduler maintenancesched
UCS-A /system* # commit-buffer
UCS-A /system #
```

Configuring Maintenance Policies

Creating a Maintenance Policy

Before You Begin

If you plan to configure this maintenance policy for deferred deployment, create a schedule.

Procedure

	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* .

	Command or Action	Purpose
Step 2	UCS-A /org # **create maint-policy** *policy-name*	Creates the specified maintenance policy and enters maintenance policy mode.
Step 3	UCS-A /org/maint-policy # **set reboot-policy** {**immediate** | **timer-automatic** | **user-ack**}	When a service profile is associated with a server, the server needs to be rebooted to complete the association. Specifying the reboot-policy command determines when the reboot occurs for all service profiles that include this maintenance policy. Possible values include: • **immediate**--The server reboots as soon as the change is made to the service profile. • **timer-automatic** --You select the schedule that specifies when maintenance operations can be applied to the server using the set scheduler command. Cisco UCS reboots the server and completes the service profile changes at the scheduled time. • **user-ack** --The user must explicitly acknowledge the changes by using the **apply pending-changes** command before changes are applied.
Step 4	UCS-A /org/maint-policy # **set on-next-boot**	(Optional) With the policy enabled, the host OS reboot, shutdown, reset or server reset, shutdown also triggers the associated FSM to apply the changes that are waiting for the **user-ack** or **timer-automatic** maintenance window.
Step 5	UCS-A /org/maint-policy # **set soft-shutdown-timer** { *150-seconds* | *300-seconds* | *600-seconds* \ | *never* }	Specifies the time in seconds for Cisco UCS Manager to wait after issuing a soft shutdown to allow servers to gracefully shut down and reboot within the specified time instead of issuing a hard shutdown after 150 seconds.
Step 6	UCS-A /org/maint-policy # **set scheduler** *scheduler-name*	(Optional) If the reboot-policy property is set to timer-automatic, you must select the schedule that specifies when maintenance operations can be applied to the server. Cisco UCS reboots the server and completes the service profile changes at the scheduled time.
Step 7	UCS-A /org/maint-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a maintenance policy called maintenance, sets the system to reboot immediately when a service profile is associated with a server, sets the soft shutdown timer to 300 seconds, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create maint-policy maintenance
UCS-A /org/maint-policy* # set reboot-policy immediate
UCS-A /org/maint-policy* # set soft-shutdown-timer 300-secs
UCS-A /org/maint-policy* # commit-buffer
```

```
UCS-A /org/maint-policy #
```

The following example enters a maintenance policy called maintenance, sets the system to reboot when you explicitly acknowledge changes made to the service profile, sets the on-next-boot option, sets the soft shutdown timer to 300 seconds, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # enter maint-policy maintenance
UCS-A /org/maint-policy* # set reboot-policy user-ack
UCS-A /org/maint-policy* # set on-next-boot
UCS-A /org/maint-policy* # set soft-shutdown-timer 300-secs
UCS-A /org/maint-policy* # commit-buffer
UCS-A /org/maint-policy #
```

Deleting a Maintenance Policy

Procedure

	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 # **delete maint-policy** *policy-name*	Deletes the specified maintenance policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a maintenance policy called maintenance and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete maint-policy maintenance
UCS-A /org/maint-policy* # commit-buffer
UCS-A /org/maint-policy #
```

Managing Pending Activities

Viewing Pending Activities

Procedure

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

	Command or Action	Purpose
Step 2	UCS-A /org # **scope service-profile** *profile-name*	Enters organization service profile mode for the specified service.
Step 3	UCS-A /org/service-profile # **show pending-changes** [**detail** | **expand**]	Displays details about pending-changes.

The following example shows how to display pending changes for a service profile called accounting:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # show pending-changes detail

Pending Changes:
    Scheduler:
    Changed by: admin
    Acked by:
    Mod. date: 2010-09-20T20:36:09.254
    State: Untriggered
    Admin State: Untriggered
    Pend. Changes: 0
    Pend. Disr.: 0
UCS-A /org/service-profile #
```

Deploying a Service Profile Change Waiting for User Acknowledgement

Cisco UCS Manager CLI cannot deploy all pending service profile changes (for multiple service profiles) waiting for user acknowledgement. To simultaneously deploy all pending service profile changes for multiple service profiles, use Cisco UCS Manager GUI.

Important You cannot stop Cisco UCS Manager from rebooting the affected server after you acknowledge a pending activity.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode. 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.
Step 3	UCS-A /org/service-profile # **apply pending-changes immediate**	Applies the pending changes immediately. Cisco UCS Manager immediately reboots the server affected by the pending activity.

The following example shows how to apply pending changes for a service profile called accounting:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # apply pending-changes immediate
UCS-A /org/service-profile #
```

Deploying a Scheduled Service Profile Change Immediately

Cisco UCS Manager CLI cannot deploy all scheduled service profile changes (for multiple service profiles) at the same time. To simultaneously deploy all scheduled service profile changes for multiple service profiles, use Cisco UCS Manager GUI.

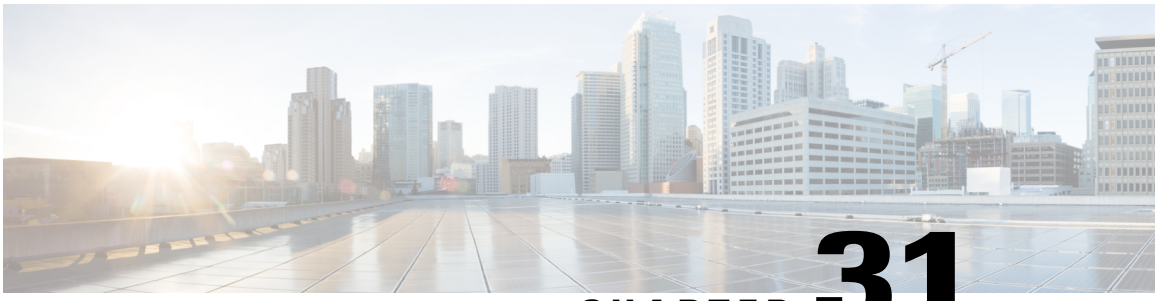
Important You cannot stop Cisco UCS Manager from rebooting the affected server after you acknowledge a pending activity.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode. 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.
Step 3	UCS-A /org/service-profile # **apply pending-changes immediate**	Applies the pending changes immediately. Cisco UCS Manager immediately reboots the server affected by the pending activity.

The following example shows how to apply pending changes for a service profile called accounting:

```
UCS-A# scope org /
UCS-A /org # scope service-profile accounting
UCS-A /org/service-profile # apply pending-changes immediate
UCS-A /org/service-profile #
```



CHAPTER 31

Service Profiles

This chapter includes the following sections:

Service Profiles that Override Server Identity

This type of service profile provides the maximum amount of flexibility and control. This profile allows you to override the identity values that are on the server at the time of association and use the resource pools and policies set up in Cisco UCS Manager to automate some administration tasks.

You can disassociate this service profile from one server, then associate it with another server. This re-association can be done either manually or through an automated server pool policy. The burned-in settings, such as UUID and MAC address on the new server are overwritten with the configuration in the service profile.

As a result, the change in the server is transparent to your network. You do not need to reconfigure any component or application on your network to begin using the new server.

This profile allows you to take advantage of and manage system resources through resource pools and policies, such as the following:

- Virtualized identity information, including pools of MAC addresses, WWN addresses, and UUIDs
- Ethernet and Fibre Channel adapter profile policies
- Firmware package policies
- Operating system boot order policies

Unless the service profile contains power management policies, a server pool qualification policy, or another policy that requires a specific hardware configuration, you can use the profile for any type of server in the Cisco UCS domain.

You can associate these service profiles with either a rack-mount server or a blade server. The ability to migrate the service profile depends upon whether you choose to restrict migration of the service profile.

Note If you choose not to restrict migration, Cisco UCS Manager does not perform any compatibility checks on the new server before migrating the existing service profile. If the hardware of both servers are not similar, the association might fail.

Service Profiles that Inherit Server Identity

This hardware-based service profile is the simplest to use and create. This profile uses the default values in the server and mimics the management of a rack-mounted server. It is tied to a specific server and cannot be moved or migrated to another server.

You do not need to create pools or configuration policies to use this service profile.

This service profile inherits and applies the identity and configuration information that is present at the time of association, such as the following:

- MAC addresses for the two NICs
- For a converged network adapter or a virtual interface card, the WWN addresses for the two HBAs
- BIOS versions
- Server UUID

Important The server identity and configuration information inherited through this service profile might not have the values burned into the server hardware at the manufacturer if those values were changed before this profile is associated with the server.

Guidelines and Recommendations for Service Profiles

In addition to any guidelines or recommendations that are specific to policies and pools included in service profiles and service profile templates, such as the local disk configuration policy, adhere to the following guidelines and recommendations that impact the ability to associate a service profile with a server:

Limit to the Number of vNICs that Can Be Configured on a Rack-Mount Server

You can configure up to 56 vNICs per supported adapter, such as the Cisco UCS P81E Virtual Interface Card (N2XX-ACPCI01), on any rack-mount server that is integrated with Cisco UCS Manager.

No Power Capping Support for Rack-Mount Servers

Power capping is not supported for rack servers. If you include a power control policy in a service profile that is associated with a rack-mount server, the policy is not implemented.

QoS Policy Guidelines for vNICs

You can only assign a QoS policy to a vNIC if the priority setting for that policy is not set to **fc**, which represents the Fibre Channel system class. You can configure the priority for the QoS policy with any other system class.

QoS Policy Guidelines for vHBAs

You can only assign a QoS policy to a vHBA if the priority setting for that policy is set to **fc**, which represents the Fibre Channel system class.

The Host Control setting for a QoS policy applies to vNICs only. It has no effect on a vHBA.

Inband Service Profiles

Configuring an Inband Service Profile

This procedure explains how to create an inband service profile.

Note All Cisco UCS M3 and M4 servers configured in Cisco UCS Manager GUI with an out-of-band configuration using the server CIMC from the **Equipment** tab, will automatically get an inband network (VLAN) and IPv4/IPv6 configuration as specified in the inband profile. Removing the network or IP pool name from the inband profile configuration will delete the inband configuration from the server, if the server inband configuration was derived from the inband profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope eth-uplink**	Enters the Ethernet uplink configuration mode.
Step 2	UCS-A /eth-uplink # **scope inband-profile**	Enters the inband profile configuration mode.
Step 3	UCS-A /eth-uplink/inband-profile # **set net-group-name***vlan-group-name*	Sets the network group name for the inband profile.
Step 4	UCS-A /eth-uplink/inband-profile # **set default-vlan-name***vlan-name*	Sets the default VLAN for the inband profile.
Step 5	UCS-A /eth-uplink/inband-profile # **set default-pool-name** *pool-name*	Sets the IP pool for the inband profile.
Step 6	UCS-A /eth-uplink/inband-profile # **commit-buffer**	Commits the transaction to the system configuration.

The example below creates the inband service profile inband-profile, sets the network group name to inband-vlan-group, sets the default VLAN to Inband_VLAN, sets the IP pool to inband_default, and commits the transaction:

```
UCS-A #scope eth-uplink
UCS-A /eth-uplink # scope inband-profile
UCS-A /eth-uplink/inband-profile # set net-group-name inband-vlan-group
UCS-A /eth-uplink/inband-profile* # set default-vlan-name Inband_VLAN
UCS-A /eth-uplink/inband-profile* # set pool-name inband_default
UCS-A /eth-uplink/inband-profile* # commit-buffer
UCS-A /eth-uplink/inband-profile #
```

Configuring an Inband Management Service Profile

This procedure explains how to configure an inband management service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** /	Enters the organization configuration mode.
Step 2	UCS-A /org # **create service-profile***sp-name*	Creates the service profile specified and enters service profile configuration mode.
Step 3	UCS-A /org/service-profile # **create mgmt-iface***in-band*	Creates the management interface specified and enters management interface configuration mode

	Command or Action	Purpose
Step 4	UCS-A /org/service-profile/mgmt-iface # **create mgmt-vlan**	Creates a management VLAN and enters the management VLAN configuration mode.
Step 5	UCS-A/org/service-profile/mgmt-iface/mgmt-vlan # set **network-name** *network-name*	Sets the management VLAN network name.
Step 6	UCS-A /org/service-profile/mgmt-iface/mgmt-vlan # **create ext-pooled-ip**	Creates an external IP pool and enters the IP pool configuration mode.
Step 7	UCS-A /org/service-profile/mgmt-iface/mgmt-vlan/ext-pooled-ip # **set name** *pool-name*	Sets the name of the external IPv4 pool.
Step 8	UCS-A /org/service-profile/mgmt-iface/mgmt-vlan/ext-pooled-ip # **exit**	Exits IPv4 pool configuration mode.
Step 9	UCS-A /org/service-profile/mgmt-iface/mgmt-vlan #**create ext-pooled-ip6**	Creates an external IPv6 pool and enters the IPv6 pool configuration mode.
Step 10	UCS-A /org/service-profile/mgmt-iface/mgmt-vlan/ext-pooled-ip6 # **set name** *pool-name*	Sets the name of the external IPv6 pool.
Step 11	UCS-A /org/service-profile/mgmt-iface/mgmt-vlan/ext-pooled-ip6 # **commit-buffer**	Commits the transaction to the system configuration.

The example below creates a service profile name inband_sp, configures a management interface named in-band, creates a management VLAN, sets the network name to Inband_VLAN, creates an external IPv4 pool and sets the name to inband_default, creates an external IP and an external IPv6 management pool, sets the name of both pools to inband_default, and commits the transaction:

```
UCS-A# scope org
UCS-A /org # create service-profile inband_sp
UCS-A /org/service-profile* # create mgmt-iface in-band
UCS-A /org/service-profile/mgmt-iface* # create mgmt-vlan
UCS-A /org/service-profile/mgmt-iface/mgmt-vlan* # set network-name Inband_VLAN
UCS-A /org/service-profile/mgmt-iface/mgmt-vlan* # create ext-pooled-ip
UCS-A /org/service-profile/mgmt-iface/mgmt-vlan/ext-pooled-ip* # set name inband_default
UCS-A /org/service-profile/mgmt-iface/mgmt-vlan/ext-pooled-ip* # exit
UCS-A /org/service-profile/mgmt-iface/mgmt-vlan* # create ext-pooled-ip6
UCS-A /org/service-profile/mgmt-iface/mgmt-vlan/ext-pooled-ip6* # set name inband_default
UCS-A /org/service-profile/mgmt-iface/mgmt-vlan/ext-pooled-ip6* # commit-buffer
UCS-A /org/service-profile/mgmt-iface/mgmt-vlan/ext-pooled-ip6 # exit
UCS-A /org/service-profile/mgmt-iface/mgmt-vlan # exit
UCS-A /org/service-profile/mgmt-iface # exit
```

What to Do Next

Associate the inband management interface service profile to a server.

Deleting the Inband Configuration from a Service Profile

This procedure explains how to delete the inband configuration from a service profile.

Note If an inband profile is configured in Cisco UCS Manager with a default VLAN name and a default pool name, the server CIMC will automatically get an inband configuration from the inband profile within one minute after deleting the configuration from the service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** /	Enters the organization configuration mode.
Step 2	UCS-A/org # **scope service-profile** *blade1*	Enters the organization profile configuration mode.
Step 3	UCS-A/org/service-profile # **delete mgmt-iface***in-band*	Deletes the specified service profile.
Step 4	UCS-A/org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example scopes to the service profile blade1, deletes the management interface in-band, and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope service-profile blade1
UCS-A /org/service-profile # delete mgmt-iface in-band
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile
```

Configuring Inband Management on the CIMC

This procedure explains how to configure inband management on a server CIMC to pooled IP.

Note Setting the inband management IP to static IP is similar to setting the inband management IP to pooled IP. The example below creates a management interface on chassis 1, server 1 named in-band, sets the IPv4 and IPv6 states to static, and commits the transaction. This example also creates a management VLAN, creates an external static IPv4, brings up the IPv4, creates an external static IPv6, brings up the IPv6, and commits the transaction:

```
UCS-A# scope server 1/1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # create mgmt-iface in-band
UCS-A /chassis/server/cimc/mgmt-iface* # set ipv4state static
UCS-A /chassis/server/cimc/mgmt-iface* # set ipv6state static
UCS-A /chassis/server/cimc/mgmt-iface* # commit-buffer
UCS-A /chassis/server/cimc/mgmt-iface # show detail

External Management Interface:
    Mode: In Band
    Ip V4 State: Static
    Ip V6 State: Static
    Is Derived from Inband Profile: No
UCS-A /chassis/server/cimc/mgmt-iface # set
  ipv4state  IpV4State
  ipv6state  IpV6State
  mode       Mode


UCS-A /chassis/server/cimc/mgmt-iface # create mgmt-vlan
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan* # create ext-static-ip
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-static-ip* # set addr x.x.x.1
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-static-ip* # set subnet 255.255.255.0
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-static-ip* # set default-gw x.x.x.254
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-static-ip* # commit-buffer
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-static-ip # up
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan # create ext-static-ip6
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-static-ip6* # set addr
xxxx:xxxx:xxxx:1::
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-static-ip6* # set default-gw
xxxx:xxxx:xxxx:1::0001
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-static-ip6* # set prefix 64
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-static-ip6* # commit-buffer
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-static-ip6 # up
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan # show detail expand

External Management Virtual LAN:
    Network Name:
    Id: 1

    External Management Static IP:
        IP Address: x.x.x.1
        Default Gateway: 10.193.1.254
        Subnet: 255.255.255.0
        Primary DNS IP: 0.0.0.0
        Secondary DNS IP: 0.0.0.0

    External Management Static IPv6:
        IP Address: xxxx:xxxx:xxxx:1::
        Default Gateway: xxxx:xxxx:xxxx:1::0001
        Prefix: 64
        Primary DNS IP: ::
        Secondary DNS IP: ::
```

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassi-numserver-num*	Enters chassis server mode for the specified server.
Step 2	UCS-A /chassis/server # **scope cimc**	Enters the CIMC configuration mode.
Step 3	UCS-A /chassis/server /chassis/server/cimc # **create mgmt-iface***in-band*	Creates the management interface specified and enters management interface configuration mode.
Step 4	UCS-A /chassis/server/cimc/mgmt-iface* # **set ipv4state pooled**	Sets IPv4 state to pooled.
Step 5	UCS-A /chassis/server/cimc/mgmt-iface *# **set ipv6state pooled**	Sets IPv6 state to pooled.
Step 6	UCS-A /chassis/server/cimc/mgmt-iface* # **create mgmt-vlan**	Creates a management VLAN and enters the management VLAN configuration mode.
Step 7	UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan* # **set network-name** *network-name*	Sets the management VLAN network name.
Step 8	UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan* # **create ext-pooled-ip**	Creates an external IPv4 pool and enters the IPv4 pool configuration mode.
Step 9	UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-pooled-ip* # **set name** *pool-name*	Sets the name of the external IPv4 pool.
Step 10	UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-pooled-ip* # **exit**	Exits IPv4 pool configuration mode.
Step 11	UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan* #**create ext-pooled-ip6**	Creates an external IPv6 pool and enters the IPv6 pool configuration mode.
Step 12	UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-pooled-ip6* # **set name** *pool-name*	Sets the name of the external IPv6 pool.
Step 13	UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-pooled-ip6* # **commit-buffer**	Commits the transaction to the system configuration.

The example below creates a management interface on chassis 1, server 1 named in-band, sets the IPv4 and IPv6 states to pooled, creates a management VLAN, sets the network name to Inband, creates an external

IPv4 pool, sets the name to inband_default. Creates an external IPv6 pool, sets the name to inband_default, and commits the transaction:

```
UCS-A# scope server 1/1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # create mgmt-iface in-band
UCS-A /chassis/server/cimc/mgmt-iface* # set ipv4state pooled
UCS-A /chassis/server/cimc/mgmt-iface* # set ipv6state pooled
UCS-A /chassis/server/cimc/mgmt-iface* # create mgmt-vlan
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan* # set network-name Inband
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan* # create ext-pooled-ip
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-pooled-ip* # set name Inband_default
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-pooled-ip* # exit
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan* # create ext-pooled-ip6
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-pooled-ip6* # set name Inband_default
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-pooled-ip6* # commit-buffer
UCS-A /chassis/server/cimc/mgmt-iface/mgmt-vlan/ext-pooled-ip6 #
```

Deleting the Inband Configuration from the CIMC

This procedure explains how to delete the inband configuration from a server CIMC.

Note If an inband profile is configured in Cisco UCS Manager with a default VLAN name and a default pool name, the server CIMC will automatically get an inband configuration from the inband profile within one minute after deleting the configuration from the service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassi-numserver-num*	Enters chassis server mode for the specified server.
Step 2	UCS-A /chassis/server # **scope cimc**	Enters the CIMC configuration mode.
Step 3	UCS-A /chassis/server /chassis/server/cimc # **delete mgmt-iface***in-band*	Deletes the specified service profile.
Step 4	UCS-A /chassis/server /chassis/server/cimc # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the deletes the management interface named in-band from chassis1, server 1, and commits the transaction:

```
UCS-A# scope server 1/1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # delete mgmt-iface in-band
UCS-A /chassis/server/cimc* # commit-buffer
UCS-A /chassis/server/cimc #
```

Initial and Existing Templates

With a service profile template, you can quickly create several service profiles with the same basic parameters, such as the number of vNICs and vHBAs, and with identity information drawn from the same pools.

Tip If you need only one service profile with similar values to an existing service profile, you can clone a service profile in the Cisco UCS Manager GUI.

For example, if you need several service profiles with similar values to configure servers to host database software, you can create a service profile template, either manually or from an existing service profile. You then use the template to create the service profiles.

Cisco UCS supports the following types of service profile templates:

Initial template

Service profiles created from an initial template inherit all the properties of the template. Service profiles created from an initial service profile template are bound to the template. However, changes to the initial template do not *automatically* propagate to the bound service profiles. If you want to propagate changes to bound service profiles, unbind and rebind the service profile to the initial template.

Updating template

Service profiles created from an updating template inherit all the properties of the template and remain connected to the template. Any changes to the template automatically update the service profiles created from the template.

Note Service profiles that are created from the initial template and normal service profiles fetch the lowest available IDs in the sequential pool when you pres**Reset**.

Service profiles created from updating template might attempt to retain the same ID when you press**Reset** even when lower IDs of sequential pool are free.

Creating a Service Profile Template

Procedure

	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 service-profile** *profile-name* {**initial-template** | **updating-template**}	Creates the specified service profile template and enters organization service profile mode.

	Command or Action	Purpose
		Enter a unique *profile-name* to identify this service profile template. This name can be between 2 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and this name must be unique across all service profiles and service profile templates within the same organization.
Step 3	UCS-A /org/service-profile # **set bios-policy** *policy-name*	Associates the specified BIOS policy with the service profile.
Step 4	UCS-A /org/service-profile # **set boot-policy** *policy-name*	Associates the specified boot policy with the service profile.
Step 5	UCS-A /org/service-profile # **set descr** *description*	(Optional) Provides a description for the service profile. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 6	UCS-A /org/service-profile # **set dynamic-vnic-conn-policy** *policy-name*	Associates the specified dynamic vNIC connection policy with the service profile.
Step 7	UCS-A /org/service-profile # **set ext-mgmt-ip-state** {**none** | **pooled**}	Specifies how the management IP address will be assigned to the service profile. You can set the management IP address policy using the following options: • None-- The service profile is not assigned an IP address. • Pooled-- The service profile is assigned an IP address from the management IP pool. **Note** Setting the management IP address to static for a service profile template will result in an error.
Step 8	UCS-A /org/service-profile # **set host-fw-policy** *policy-name*	Associates the specified host firmware policy with the service profile.
Step 9	UCS-A /org/service-profile # **set identity** {**dynamic-uuid** {*uuid* | **derived**} | **dynamic-wwnn** {*wwnn* | **derived**} | **uuid-pool** *pool-name* | **wwnn-pool** *pool-name*}	Specifies how the server acquires a UUID or WWNN. You can do one of the following: • Create a unique UUID in the form *nnnnnnnn-nnnn-nnnn-nnnnnnnnnnnn* . • Derive the UUID from the one burned into the hardware at manufacture.

	Command or Action	Purpose
		• Use a UUID pool. • Create a unique WWNN in the form *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh*. • Derive the WWNN from one burned into the hardware at manufacture. • Use a WWNN pool.
Step 10	UCS-A /org/service-profile # **set ipmi-access-profile** *profile-name*	Associates the specified IPMI access profile with the service profile.
Step 11	UCS-A /org/service-profile # **set lan-connectivity-policy-name** *policy-name*	Associates the specified LAN connectivity policy with the service profile. **Note** You cannot have a LAN connectivity policy and locally created vNICs in the same service profile. When you add a LAN connectivity policy to a service profile, any existing vNIC configuration is erased.
Step 12	UCS-A /org/service-profile # **set local-disk-policy** *policy-name*	Associates the specified local disk policy with the service profile.
Step 13	UCS-A /org/service-profile # **set maint-policy** *policy-name*	Associates the specified maintenance policy with the service profile.
Step 14	UCS-A /org/service-profile # **set mgmt-fw-policy** *policy-name*	Associates the specified management firmware policy with the service profile.
Step 15	UCS-A /org/service-profile # **set power-control-policy** *policy-name*	Associates the specified power control policy with the service profile.
Step 16	UCS-A /org/service-profile # **set san-connectivity-policy-name** *policy-name*	Associates the specified SAN connectivity policy with the service profile. **Note** You cannot have a SAN connectivity policy and locally created vHBAs in the same service profile. When you add a SAN connectivity policy to a service profile, any existing vHBA configuration is erased.
Step 17	UCS-A /org/service-profile # **set scrub-policy** *policy-name*	Associates the specified scrub policy with the service profile.
Step 18	UCS-A /org/service-profile # **set sol-policy** *policy-name*	Associates the specified serial over LAN policy with the service profile.
Step 19	UCS-A /org/service-profile # **set stats-policy** *policy-name*	Associates the specified statistics policy with the service profile.
Step 20	UCS-A /org/service-profile # **set user-label** *label-name*	Specifies the user label associated with the service profile.

	Command or Action	Purpose
Step 21	UCS-A /org/service-profile # **set vcon** {**1** | **2**} **selection** {**all** | **assigned-only** | **exclude-dynamic** | **exclude-unassigned**}	Specifies the selection preference for the specified vCon.
Step 22	UCS-A /org/service-profile # **set vcon-profile** *policy-name*	Associates the specified vNIC/vHBA placement profile with the service profile. **Note** You can either assign a vNIC/vHBA placement profile to the service profile, or set vCon selection preferences for the service profile, but you do not need to do both.
Step 23	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a service profile template and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create service-profile ServTemp2 updating-template
UCS-A /org/service-profile* # set bios-policy biospol1
UCS-A /org/service-profile* # set boot-policy bootpol32
UCS-A /org/service-profile* # set  descr "This is a service profile example."
UCS-A /org/service-profile* # set dynamic-vnic-conn-policy mydynvnicconnpolicy
UCS-A /org/service-profile* # set ext-mgmt-ip-state pooled
UCS-A /org/service-profile* # set host-fw-policy ipmi-user987
UCS-A /org/service-profile* # set identity dynamic-uuid derived
UCS-A /org/service-profile* # set ipmi-access-profile ipmiProf16
UCS-A /org/service-profile* # set local-disk-policy localdiskpol33
UCS-A /org/service-profile* # set maint-policy maintpol4
UCS-A /org/service-profile* # set mgmt-fw-policy mgmtfwpol75
UCS-A /org/service-profile* # set power-control-policy powcontrpol13
UCS-A /org/service-profile* # set scrub-policy scrubpol55
UCS-A /org/service-profile* # set sol-policy solpol2
UCS-A /org/service-profile* # set stats-policy statspol4
UCS-A /org/service-profile* # set user-label mylabel
UCS-A /org/service-profile* # vcon-policy myvconnpolicy
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

What to Do Next

- (Optional) Configure a boot definition for the service profile. Use this option only if you have not associated a boot policy with the service profile.
- Create a service profile instance from the service profile template.

Creating a Service Profile Instance from a Service Profile Template

Before You Begin

Verify that there is a service profile template from which to create a service profile instance.

Procedure

	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 # **create service-profile** *profile-name* **instance**	Creates the specified service profile instance and enters organization service profile mode. Enter a unique *profile-name* to identify this service profile template. This name can be between 2 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and this name must be unique across all service profiles and service profile templates within the same organization.
Step 3	UCS-A /org/service-profile # **set src-templ-name** *profile-name*	Specifies the source service profile template to apply to the service profile instance. All configuration settings from the service profile template will be applied to the service profile instance.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a service profile instance named ServProf34, applies the service profile template named ServTemp2, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create service-profile ServProf34 instance
UCS-A /org/service-profile* # set src-templ-name ServTemp2
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

What to Do Next

Associate the service profile to a server, rack server, or server pool.

Unbinding a Service Profile from a Service Profile Template

To unbind a service profile from a service profile template, bind the service profile to an empty value (quotes without space).

Procedure

	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 # **set src-templ-name ""**	Unbinds the service profile from the service profile template.
Step 4	UCS-A /org/service-profile* # **commit-buffer**	Commits the transaction to the system configuration.

The following example unbinds the service profile named ServiceProf1 and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope service-profile ServiceProf1
UCS-A /org/service-profile # set src-templ-name ""
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Creating a Hardware-Based Service Profile

Procedure

	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 # **create service-profile** *profile-name* **instance**	Creates the specified service profile instance and enters organization service profile mode. Enter a unique *profile-name* to identify this service profile. This name can be between 2 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and this name must be unique across all service profiles and service profile templates within the same organization.
Step 3	UCS-A /org/service-profile # **set bios-policy** *policy-name*	Associates the specified BIOS policy with the service profile.

	Command or Action	Purpose
Step 4	UCS-A /org/service-profile # **set boot-policy** *policy-name*	Associates the specified boot policy with the service profile.
Step 5	UCS-A /org/service-profile # **set descr** *description*	(Optional) Provides a description for the service profile. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 6	UCS-A /org/service-profile # **set dynamic-vnic-conn-policy** *policy-name*	Associates the specified dynamic vNIC connection policy with the service profile.
Step 7	UCS-A /org/service-profile # **set ext-mgmt-ip-state** {**none** | **pooled** | **static**}	Specifies how the management IP address will be assigned to the service profile. You can set the management IP address policy using the following options: • None-- The service profile is not assigned an IP address. • Pooled-- The service profile is assigned an IP address from the management IP pool. • Static-- The service profile is assigned the configured static IP address.
Step 8	UCS-A /org/service-profile # **set host-fw-policy** *ipmi-user-name*	Associates the specified host forwarding policy with the service profile.
Step 9	UCS-A /org/service-profile # **set identity** {**dynamic-uuid** {*uuid* | **derived**} | **dynamic-wwnn** {*wwnn* | **derived**} | **uuid-pool** *pool-name* | **wwnn-pool** *pool-name*}	Specifies how the server acquires a UUID or WWNN. You can do one of the following: • Create a unique UUID in the form *nnnnnnnn-nnnn-nnnn-nnnnnnnnnnnn*. • Derive the UUID from the one burned into the hardware at manufacture. • Use a UUID pool. • Create a unique WWNN in the form *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh* **:** *hh*. • Derive the WWNN from one burned into the hardware at manufacture. • Use a WWNN pool.

	Command or Action	Purpose
Step 10	UCS-A /org/service-profile # **set ipmi-access-profile** *profile-name*	Associates the specified IPMI access profile with the service profile.
Step 11	UCS-A /org/service-profile # **set local-disk-policy** *policy-name*	Associates the specified local disk policy with the service profile.
Step 12	UCS-A /org/service-profile # **set maint-policy** *policy-name*	Associates the specified maintenance policy with the service profile.
Step 13	UCS-A /org/service-profile # **set mgmt-fw-policy** *policy-name*	Associates the specified management forwarding policy with the service profile.
Step 14	UCS-A /org/service-profile # **set power-control-policy** *policy-name*	Associates the specified power control policy with the service profile.
Step 15	UCS-A /org/service-profile # **set scrub-policy** *policy-name*	Associates the specified scrub policy with the service profile.
Step 16	UCS-A /org/service-profile # **set sol-policy** *policy-name*	Associates the specified serial over LAN policy with the service profile.
Step 17	UCS-A /org/service-profile # **set stats-policy** *policy-name*	Associates the specified statistics policy with the service profile.
Step 18	UCS-A /org/service-profile # **set user-label** *label-name*	Specifies the user label associated with the service profile.
Step 19	UCS-A /org/service-profile # **set vcon** {**1** | **2**} **selection** {**all** | **assigned-only** | **exclude-dynamic** | **exclude-unassigned**}	Specifies the selection preference for the specified vCon.
Step 20	UCS-A /org/service-profile # **set vcon-policy** *policy-name*	Associates the specified vNIC/vHBA placement policy with the service profile. **Note** You can either assign a vNIC/vHBA placement profile to the service profile, or set vCon selection preferences for the service profile, but you do not need to do both.
Step 21	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a service profile instance and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create service-profile ServInst90 instance
UCS-A /org/service-profile* # set bios-policy biospol1
UCS-A /org/service-profile* # set boot-policy bootpol32
UCS-A /org/service-profile* # set  descr "This is a service profile example."
UCS-A /org/service-profile* # set ext-mgmt-ip-state pooled
UCS-A /org/service-profile* # set host-fw-policy ipmi-user987
UCS-A /org/service-profile* # set identity dynamic-uuid derived
UCS-A /org/service-profile* # set ipmi-access-profile ipmiProf16
UCS-A /org/service-profile* # set local-disk-policy localdiskpol33
```

```
UCS-A /org/service-profile* # set maint-policy maintpol4
UCS-A /org/service-profile* # set mgmt-fw-policy mgmtfwpol75
UCS-A /org/service-profile* # set power-control-policy powcontrpol13
UCS-A /org/service-profile* # set scrub-policy scrubpol55
UCS-A /org/service-profile* # set sol-policy solpol2
UCS-A /org/service-profile* # set stats-policy statspol4
UCS-A /org/service-profile* # set user-label mylabel
UCS-A /org/service-profile* # vcon-policy myvconnpolicy
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

What to Do Next

- (Optional) Configure a boot definition for the service profile. Use this option only if you have not associated a boot policy with the service profile.
- Associate the service profile with a blade server, server pool, or rack server.

Configuring a vNIC for a Service Profile

Procedure

	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** *profile-name*	Enters organization service profile mode for the specified service profile.
Step 3	UCS-A /org/service-profile # **create vnic** *vnic-name* [**eth-if** *eth-if-name*] [**fabric** {**a** | **b**}]	Creates a vNIC for the specified service profile and enters organization service profile vNIC mode.
Step 4	UCS-A /org/service-profile/vnic # **set adapter-policy** *policy-name*	Specifies the adapter policy to use for the vNIC.
Step 5	UCS-A /org/service-profile/vnic # **set fabric** {**a** | **a-b** | **b** | **b-a**}	Specifies the fabric to use for the vNIC. If you did not specify the fabric when creating the vNIC template in Step 3, you have the option to specify it with this command. If you want this vNIC to be able to access the second fabric interconnect if the default one is unavailable, choose **a-b** (A is the primary) or **b-a** (B is the primary) .

	Command or Action	Purpose
		Note Do not enable fabric failover for the vNIC under the following circumstances: • If the Cisco UCS domain is running in Ethernet Switch Mode. vNIC fabric failover is not supported in that mode. If all Ethernet uplinks on one fabric interconnect fail, the vNICs do not fail over to the other. • If you plan to associate this vNIC to a server with an adapter that does not support fabric failover, such as the Cisco UCS 82598KR-CI 10-Gigabit Ethernet Adapter. If you do so, Cisco UCS Manager generates a configuration fault when you associate the service profile with the server.
Step 6	UCS-A /org/service-profile/vnic # **set identity** {**dynamic-mac** {*mac-addr* | **derived**} | **mac-pool** *mac-pool-name*}	Specifies the identity (MAC address) for the vNIC. You can set the identity using one of the following options: • Create a unique MAC address in the form *nn* **:** *nn* **:** *nn* **:** *nn* **:** *nn* **:** *nn* . • Derive the MAC address from one burned into the hardware at manufacture. • Assign a MAC address from a MAC pool.
Step 7	UCS-A /org/service-profile/vnic # **set mtu** *size-num*	The maximum transmission unit, or packet size, that this vNIC accepts. Enter an integer between 1500 and 9216. **Note** If the vNIC has an associated QoS policy, the MTU specified here must be equal to or less than the MTU specified in the associated QoS system class. If this MTU value exceeds the MTU value in the QoS system class, packets might get dropped during data transmission.
Step 8	UCS-A /org/service-profile/vnic # **set nw-control-policy** *policy-name*	The network control policy the vNIC should use.
Step 9	UCS-A /org/service-profile/vnic # **set order** {*order-num* | **unspecified**}	Specifies the relative order for the vNIC.
Step 10	UCS-A /org/service-profile/vnic # **set pin-group** *group-name*	The LAN pin group the vNIC should use.
Step 11	UCS-A /org/service-profile/vnic # **set qos-policy** *policy-name*	The quality of service policy the vNIC should use.
Step 12	UCS-A /org/service-profile/vnic # **set stats-policy** *policy-name*	The statistics collection policy the vNIC should use.

	Command or Action	Purpose
Step 13	UCS-A /org/service-profile/vnic # **set template-name** *policy-name*	Specifies the dynamic vNIC connectivity policy to use for the vNIC.
Step 14	UCS-A /org/service-profile/vnic # **set vcon** {**1** | **2** | **3** | **4** | **any**}	Assigns the vNIC to the specified vCon. Use the **any** keyword to have Cisco UCS Manager automatically assign the vNIC.
Step 15	UCS-A /org/service-profile/vnic # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a vNIC for a service profile and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile ServInst90
UCS-A /org/service-profile* # create vnic vnic3 fabric a
UCS-A /org/service-profile/vnic* # set adapter-policy AdaptPol2
UCS-A /org/service-profile/vnic* # set fabric a-b
UCS-A /org/service-profile/vnic* # set identity mac-pool MacPool3
UCS-A /org/service-profile/vnic* # set mtu 8900
UCS-A /org/service-profile/vnic* # set nw-control-policy ncp5
UCS-A /org/service-profile/vnic* # set order 0
UCS-A /org/service-profile/vnic* # set pin-group EthPinGroup12
UCS-A /org/service-profile/vnic* # set qos-policy QosPol5
UCS-A /org/service-profile/vnic* # set stats-policy StatsPol2
UCS-A /org/service-profile/vnic* # set template-name VnicConnPol3
UCS-A /org/service-profile/vnic* # set set vcon any
UCS-A /org/service-profile/vnic* # commit-buffer
UCS-A /org/service-profile/vnic #
```

Creating vNIC Pairs on a Service Profile

Procedure

	Command or Action	Purpose
Step 1	UCS-A /org # **scope** *org-name* .	Enters the organization mode for the specified organization. To enter the root organization mode enter "org" as the org-name.
Step 2	UCS-A /org # **scope service-profile** *service profile name* .	Enters the name of the service profile where you want to create the vNIC pair.
Step 3	UCS-A /org # scope service-profile **create vnic** *eth0*.	Assigns a name to the vNIC for creating the redundancy pair.
Step 4	UCS-A /org/service-profile/vnic* # **set template-name***vNIC-primary* .	Specifies to use the Primary vNIC template that you can link to a Secondary vNIC template to create a vNIC pair at the service profile level.
Step 5	UCS-A /org/service-profile/vnic* # **exit** .	Exits the Primary vNIC template to use to create the vNIC pair.

	Command or Action	Purpose
		Note You can now create the peer vNIC to link to vNIC eth0. Ensure to commit the transaction after linking vNIC eth0 to vNIC eth1 to create the vNIC pair.
Step 6	UCS-A /org/service-profile # **create vnic** *eth1*.	Assigns a name to the vNIC for creating the peer vNIC to create the pair that you link to vNIC eth0.
Step 7	UCS-A /org/service-profile/vnic* **set template-name** *vNIC secondary* .	Specifies to use the Secondary vNIC template as the peer template to a Primary vNIC template to create a vNIC pair that you can use at the service profile level.
Step 8	UCS-A /org/service-profile/vnic* # **exit** .	Exits the Secondary vNIC template to use to create the vNIC pair.
Step 9	UCS-A /org/service-profile* # **commit-buffer** .	Commits the transaction to the system configuration.

The following example creates a vNIC redundancy pair from a service profile and commits the transaction:

```
UCS-A # scope org
UCS-A /org # scope service-profile test-sp
UCS-A /org/service-profile # create vNIC eth0
UCS-A /org/service-profile/vnic* # set template-name vNIC-primary
UCS-A /org/service-profile/vnic* # exit
UCS-A /org/service-profile* # create vNIC eth1
UCS-A /org/service-profile/vnic* # set template-name vNIC-secondary
UCS-A /org/service-profile/vnic* # exit
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Configuring a vHBA for a Service Profile

Procedure

	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.
Step 3	UCS-A /org/service-profile # **create vhba** *vhba-name* [**fabric** {**a** | **b**}] [**fc-if** *fc-if-name*]	Creates a vHBA for the specified service profile and enters organization service profile vHBA mode.
Step 4	UCS-A /org/service-profile/vhba # **set adapter-policy** *policy-name*	Specifies the adapter policy to use for the vHBA.

	Command or Action	Purpose
Step 5	UCS-A /org/service-profile/vhba # **set admin-vcon** {**1** | **2** | **any**}	Assigns the vHBA to one or all virtual network interface connections.
Step 6	UCS-A /org/service-profile/vhba # **set identity** {**dynamic-wwpn** {*wwpn* | **derived**} | **wwpn-pool** *wwn-pool-name*}	Specifies the WWPN for the vHBA. You can set the storage identity using one of the following options: • Create a unique WWPN in the form *hh*:*hh*:*hh*:*hh*:*hh*:*hh*:*hh*:*hh*. You can specify a WWPN in the range from 20:00:00:00:00:00:00:00 to 20:FF:FF:FF:FF:FF:FF:FF or from 50:00:00:00:00:00:00:00 to 5F:FF:FF:FF:FF:FF:FF:FF. If you want the WWPN to be compatible with Cisco MDS Fibre Channel switches, use the WWPN template **20:00:00:25:B5:XX:XX:XX**. • Derive the WWPN from one burned into the hardware at manufacture. • Assign a WWPN from a WWN pool.
Step 7	UCS-A /org/service-profile/vhba # **set max-field-size** *size-num*	Specifies the maximum size of the Fibre Channel frame payload (in bytes) that the vHBA supports.
Step 8	UCS-A /org/service-profile/vhba # **set order** {*order-num* | **unspecified**}	Specifies the PCI scan order for the vHBA.
Step 9	UCS-A /org/service-profile/vhba # **set pers-bind** {**disabled** | **enabled**}	Disables or enables persistent binding to Fibre Channel targets.
Step 10	UCS-A /org/service-profile/vhba # **set pin-group** *group-name*	Specifies the SAN pin group to use for the vHBA.
Step 11	UCS-A /org/service-profile/vhba # **set qos-policy** *policy-name*	Specifies the QoS policy to use for the vHBA.
Step 12	UCS-A /org/service-profile/vhba # **set stats-policy** *policy-name*	Specifies the statistics threshold policy to use for the vHBA.
Step 13	UCS-A /org/service-profile/vhba # **set template-name** *policy-name*	Specifies the vHBA template to use for the vHBA.
Step 14	UCS-A /org/service-profile/vhba # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a vHBA for a service profile and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile ServInst90
UCS-A /org/service-profile* # create vhba vhba3 fabric b
```

```
UCS-A /org/service-profile/vhba* # set adapter-policy AdaptPol2
UCS-A /org/service-profile/vhba* # set admin-vcon any
UCS-A /org/service-profile/vhba* # set identity wwpn-pool SanPool7
UCS-A /org/service-profile/vhba* # set max-field-size 2112
UCS-A /org/service-profile/vhba* # set order 0
UCS-A /org/service-profile/vhba* # set pers-bind enabled
UCS-A /org/service-profile/vhba* # set pin-group FcPinGroup12
UCS-A /org/service-profile/vhba* # set qos-policy QosPol5
UCS-A /org/service-profile/vhba* # set stats-policy StatsPol2
UCS-A /org/service-profile/vhba* # set template-name SanConnPol3
UCS-A /org/service-profile/vhba* # commit-buffer
UCS-A /org/service-profile/vhba #
```

Creating vHBA Pairs on a Service Profile

Procedure

	Command or Action	Purpose
Step 1	UCS-A/ org # **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 name of the service profile where you want to create the vHBA pair.
Step 3	UCS-A/ org # service-profile **create vhba** *fc0*.	Assigns a name to the vHBA for creating the redundancy pair.
Step 4	UCS-A /org/service-profile # **set template-name** *vhba primary* .	Specifies to use the Primary vHBA template that you can link to a Secondary vHBA template to create a vHBA pair at the service profile level.
Step 5	UCS-A /org/service-profile # **exit** .	Exits the Primary vHBA template to use to create the vHBA pair. **Note** You can now create the peer vHBA to link to vHBA fc0. Ensure to commit the transaction after linking vHBA fc0 to vHBA fc1 to create the vHBA pair.
Step 6	UCS-A /org/service-profile # **create vhba** *fc1*.	Assigns a name to the vHBA for creating the peer vHBA to create the pair that you link to vHBA fc0.
Step 7	UCS-A/ org # service-profile **set template-name** *vhba secondary* .	Specifies to use the Secondary vHBA template as the peer template to a Primary vHBA template to create a vHBA pair that you can use at the service profile level.
Step 8	UCS-A/ # org service profile **commit-buffer** .	Commits the transaction to the system configuration.

The following example creates a vHBA redundancy pair from a service profile and commits the transaction:

```
UCS-A/ # scope org
```

```
UCS-A /org # scope service-profile test-sp
UCS-A /org/service-profile # create vhba fc0
UCS-A /org/service-profile/vhba* # set template-name vhba-primary
UCS-A /org/service-profile/vhba* # exit
UCS-A /org/service-profile* # create vhba fc1
UCS-A /org/service-profile/vhba* # set template-name vhba-secondary
UCS-A /org/service-profile/vhba* # exit
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Configuring a Local Disk for a Service Profile

Procedure

	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 # **create local-disk-config**	Creates a local disk configuration for the service profile and enters organization service profile local disk configuration mode.
Step 4	UCS-A /org/service-profile/local-disk-config # **set descr** *description*	(Optional) Provides a description for the local disk configuration.
Step 5	UCS-A /org/service-profile/local-disk-config # **set mode** {**any-configuration** | **no-local-storage** | **no-raid** | **raid-0-striped** | **raid-1-mirrored** | **raid-5-striped-parity** | **raid-6-striped-dual-parity** | **raid-10-mirrored-and-striped**}	Specifies the mode for the local disk.
Step 6	UCS-A /org/service-profile/local-disk-config # **create partition**	Creates a partition for the local disk and enters organization service profile local disk configuration partition mode.
Step 7	UCS-A /org/service-profile/local-disk-config/partition # **set descr** *description*	(Optional) Provides a description for the partition.
Step 8	UCS-A /org/service-profile/local-disk-config/partition # **set size** {*size-num* | **unspecified**}	Specifies the partition size in MBytes.
Step 9	UCS-A /org/service-profile/local-disk-config/partition # **set type** {**ext2** | **ext3** | **fat32** | **none** | **ntfs** | **swap**}	Specifies the partition type.

	Command or Action	Purpose
Step 10	UCS-A /org/service-profile/local-disk-config/partition # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a local disk for a service profile and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile ServInst90
UCS-A /org/service-profile # scope boot-definition
UCS-A /org/service-profile # create local-disk-config
UCS-A /org/service-profile/local-disk-config* # set mode raid-1-mirrored
UCS-A /org/service-profile/local-disk-config* # create partition
UCS-A /org/service-profile/local-disk-config/partition* # set size 1000000
UCS-A /org/service-profile/local-disk-config/partition* # set type ntfs
UCS-A /org/service-profile/local-disk-config/partition* # commit-buffer
UCS-A /org/service-profile/local-disk-config/partition #
```

Configuring Serial over LAN for a Service Profile

Procedure

	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.
Step 3	UCS-A /org/service-profile # **create sol-config**	Creates a serial over LAN configuration for the service profile and enters organization service profile SoL configuration mode.
Step 4	UCS-A /org/service-profile/sol-config # {**disable** | **enable**}	Disables or enables the serial over LAN configuration for the service profile.
Step 5	UCS-A /org/service-profile/sol-config # **set descr** *description*	(Optional) Provides a description for the serial over LAN configuration.
Step 6	UCS-A /org/service-profile/sol-config # **set speed** {**115200** | **19200** | **38400** | **57600** | **9600**}	Specifies the serial baud rate.
Step 7	UCS-A /org/service-profile/sol-config # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures serial over LAN for the service profile named ServInst90 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile ServInst90
UCS-A /org/service-profile # create sol-config
UCS-A /org/service-profile/sol-config* # enable
UCS-A /org/service-profile/sol-config* # set descr "Sets serial over LAN to 9600 baud."
UCS-A /org/service-profile/sol-config* # set speed 9600
UCS-A /org/service-profile/sol-config* # commit-buffer
UCS-A /org/service-profile/sol-config #
```

Service Profile Boot Definition Configuration

Configuring a Boot Definition for a Service Profile

Procedure

	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 the specified service.
Step 3	UCS-A /org/service-profile # **create boot-definition**	Creates a boot definition for the service profile and enters organization service profile boot definition mode.
Step 4	UCS-A /org/service-profile/boot-definition # **set descr** *description*	(Optional) Provides a description for the boot definition.
Step 5	UCS-A /org/service-profile/boot-definition # **set reboot-on-update** {**no** | **yes**}	(Optional) Specifies whether to automatically reboot all servers that use this boot definition after changes are made to the boot order. By default, the reboot on update option is disabled.
Step 6	UCS-A /org/service-profile/boot-definition # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a boot definition for a service profile and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile ServInst90
UCS-A /org/service-profile* # create boot-definition
UCS-A /org/service-profile/boot-definition* # set descr "This boot definition reboots on
update."
UCS-A /org/service-profile/boot-definition* # set reboot-on-update yes
UCS-A /org/service-profile/boot-definition* # commit-buffer
UCS-A /org/service-profile/boot-definition #
```

What to Do Next

Configure one or more of the following boot options for the boot definition and set their boot order:

- **LAN Boot —**Boots from a centralized provisioning server. It is frequently used to install operating systems on a server from that server.

  If you choose the LAN Boot option, continue to Configuring a LAN Boot for a Service Profile Boot Definition , on page 619.

- **Storage Boot —** Boots from an operating system image on the SAN. You can specify a primary and a secondary SAN boot. If the primary boot fails, the server attempts to boot from the secondary.

  We recommend that you use a SAN boot, because it offers the most service profile mobility within the system. If you boot from the SAN, when you move a service profile from one server to another, the new server boots from exactly the same operating system image. Therefore, the new server appears to be exactly the same server to the network.

  If you choose the Storage Boot option, continue to Configuring a Storage Boot for a Service Profile Boot Definition , on page 620.

- **Virtual Media Boot —**Mimics the insertion of a physical CD into a server. It is typically used to manually install operating systems on a server.

  If you choose the Virtual Media boot option, continue to Configuring a Virtual Media Boot for a Service Profile Boot Definition , on page 622.

Configuring a LAN Boot for a Service Profile Boot Definition

Before You Begin

Configure a boot definition for a service profile.

Procedure

	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 # **scope boot-definition**	Enters organization service profile boot definition mode.
Step 4	UCS-A /org/service-profile/boot-definition # **create lan**	Creates a LAN boot for the service profile boot definition and enters service profile boot definition LAN mode.
Step 5	UCS-A /org/service-profile/boot-definition/lan # **set order** {**1** | **2** | **3** | **4**}	Specifies the boot order for the LAN boot.

	Command or Action	Purpose
Step 6	UCS-A /org/service-profile/boot-definition/lan # **create path** {**primary** | **secondary**}	Creates a primary or secondary LAN boot path and enters service profile boot definition LAN path mode.
Step 7	UCS-A /org/service-profile/boot-definition/lan/path # **set vnic** *vnic-name*	Specifies the vNIC to use for the LAN image path.
Step 8	UCS-A /org/service-profile/boot-definition/lan/path # **commit-buffer**	Commits the transaction to the system configuration.

The following example enters the service profile named ServInst90, creates a LAN boot for the service profile boot definition, sets the boot order to 2, creates a primary path, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile ServInst90
UCS-A /org/service-profile* # scope boot-definition
UCS-A /org/service-profile/boot-definition* # create lan
UCS-A /org/service-profile/boot-definition/lan* # set order 2
UCS-A /org/service-profile/boot-definition/lan* # create path primary
UCS-A /org/service-profile/boot-definition/lan/path* # set vnic vnic3
UCS-A /org/service-profile/boot-definition/lan/path* # commit-buffer
UCS-A /org/service-profile/boot-definition/lan/path #
```

Configuring a Storage Boot for a Service Profile Boot Definition

Before You Begin

Configure a boot definition for a service profile.

Procedure

	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.
Step 3	UCS-A /org/service-profile # **scope boot-definition**	Enters organization service profile boot definition mode.
Step 4	UCS-A /org/service-profile/boot-definition # **create storage**	Creates a storage boot for the service profile boot definition and enters service profile boot definition storage mode.

	Command or Action	Purpose
Step 5	UCS-A /org/service-profile/boot-definition/storage # **set order** {**1** | **2** | **3** | **4**}	Specifies the boot order for the storage boot.
Step 6	UCS-A /org/service-profile/boot-definition/storage # **create** {**local** | **san-image** {**primary** | **secondary**}}	Creates a local storage boot or a SAN image boot. If a SAN image boot is created, it enters service profile boot definition storage SAN image mode.
Step 7	UCS-A /org/service-profile/boot-definition/storage/san-image # **create path** {**primary** | **secondary**}	Creates a primary or secondary SAN image path and enters service profile boot definition storage SAN image path mode. When using the enhanced boot order on Cisco UCS M3 servers, or M4 servers, the boot order that you define is used. For standard boot mode using the terms "primary" or "secondary" do not imply a boot order. The effective order of boot devices within the same device class is determined by the PCIe bus scan order.
Step 8	UCS-A /org/service-profile/boot-definition/storage/san-image/path # **set lun** *lun-num*	Specifies the LUN used for the SAN image path.
Step 9	UCS-A /org/service-profile/boot-definition/storage/san-image/path # **set vhba** *vhba-name*	Specifies the vHBA used for the SAN image path.
Step 10	UCS-A /org/service-profile/boot-definition/storage/san-image/path # **set wwn** *wwn-num*	Specifies the WWN used for the SAN image path.
Step 11	UCS-A /org/service-profile/boot-definition/storage/san-image/path # **commit-buffer**	Commits the transaction to the system configuration.

The following example enters the service profile named ServInst90, creates a storage boot for the service profile boot definition, sets the boot order to 2, creates a primary path, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile ServInst90
UCS-A /org/service-profile* # scope boot-definition
UCS-A /org/service-profile/boot-definition* # create storage
UCS-A /org/service-profile/boot-definition/storage* # create san-image primary
UCS-A /org/service-profile/boot-definition/storage* # set order 2
UCS-A /org/service-profile/boot-definition/storage/san-image* # create path primary
UCS-A /org/service-profile/boot-definition/storage/san-image/path* # set lun 27512
UCS-A /org/service-profile/boot-definition/storage/san-image/path* # set vhba vhba3
UCS-A /org/service-profile/boot-definition/storage/san-image/path* # set wwn
20:00:00:00:20:00:00:23
```

```
UCS-A /org/service-profile/boot-definition/storage/san-image/path* # commit-buffer
UCS-A /org/service-profile/boot-definition/storage/san-image/path #
```

Configuring a Virtual Media Boot for a Service Profile Boot Definition

Before You Begin

Configure a boot definition for a service profile.

Procedure

	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.
Step 3	UCS-A /org/service-profile # **scope boot-definition**	Enters organization service profile boot definition mode.
Step 4	UCS-A /org/service-profile/boot-definition # **create virtual-media** {**read-only** | **read-write**}	Creates a read-only or read-write virtual media boot for the service profile boot definition and enters service profile boot definition virtual media mode.
Step 5	UCS-A /org/service-profile/boot-definition/virtual-media # **set order** {**1** | **2** | **3** | **4**}	Specifies the boot order for the virtual media boot.
Step 6	UCS-A /org/service-profile/boot-definition/virtual-media # **commit-buffer**	Commits the transaction to the system configuration.

The following example enters the service profile named ServInst90, creates a virtual media boot with read-only privileges for the service profile boot definition, sets the boot order to 3, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile ServInst90
UCS-A /org/service-profile* # scope boot-definition
UCS-A /org/service-profile/boot-definition* # create virtual-media read-only
UCS-A /org/service-profile/boot-definition/virtual-media* # set order 3
UCS-A /org/service-profile/boot-definition/virtual-media* # commit-buffer
UCS-A /org/service-profile/boot-definition/virtual-media #
```

Deleting a Boot Definition for a Service Profile

Procedure

	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 the specified service.
Step 3	UCS-A /org/service-profile # **delete boot-definition**	Deletes the boot definition for the service profile.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the boot definition for a service profile and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile ServInst90
UCS-A /org/service-profile # delete boot-definition
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Configuring Fibre Channel Zoning for a Service Profile

Configuring a vHBA Initiator Group with an Existing Storage Connection Policy

This procedure assumes that you want to use an existing global Fibre Channel storage connection policy. If you want to create a storage connection policy definition just for this service profile, see Configuring a vHBA Initiator Group with a local Storage Connection Policy Definition, on page 624.

For information about how to create a global Fibre Channel storage connection policy that is available to all service profiles, see Creating a Fibre Channel Storage Connection Policy, on page 377.

Procedure

	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.

	Command or Action	Purpose
Step 3	UCS-A /org/service-profile # **create initiator-group** *group-name*	Creates the specified initiator group for Fibre Channel zoning and enters service profile initiator group mode.
Step 4	UCS-A /org/service-profile/initiator-group # **create initiator** *vhba-name*	Creates the specified vHBA initiator in the initiator group. If desired, repeat this step to add a second vHBA initiator to the group.
Step 5	UCS-A /org/service-profile/initiator-group # **set storage-connection-policy** *policy-name*	Associates the specified storage connection policy with the service profile.
Step 6	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a vHBA initiator group named initGroupZone1 with two vHBA initiators for a service profile named ServInst90, includes an existing Fibre Channel storage connection policy, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope service-profile ServInst90
UCS-A /org/service-profile # create initiator-group initGroupZone1
UCS-A /org/service-profile/initiator-group* # create initiator vhba1
UCS-A /org/service-profile/initiator-group* # create initiator vhba2
UCS-A /org/service-profile/initiator-group* # set storage-connection-policy scpolicyZone1
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Configuring a vHBA Initiator Group with a local Storage Connection Policy Definition

This procedure assumes that you want to create a local Fibre Channel storage connection policy for a service profile. If you want to use an existing storage connection policy, see Configuring a vHBA Initiator Group with an Existing Storage Connection Policy, on page 623.

Procedure

	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* .

	Command or Action	Purpose
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 # **create initiator-group** *group-name*	Creates the specified initiator group for Fibre Channel zoning and enters service profile initiator group mode.
Step 4	UCS-A /org/service-profile/initiator-group # **create initiator** *vhba-name*	Creates the specified vHBA initiator in the vHBA initiator group. If desired, repeat this step to add a second vHBA initiator to the group.
Step 5	UCS-A /org/service-profile/initiator-group # **create storage-connection-def** *policy-name*	Creates the specified storage connection policy definition and enters storage connection definition mode.
Step 6	UCS-A /org/service-profile/initiator-group/storage-connection-def # **create storage-target** *wwpn*	Creates a storage target endpoint with the specified WWPN, and enters storage target mode.
Step 7	UCS-A /org/service-profile/initiator-group/storage-connection-def/storage-target # **set target-path** {**a** | **b**}	Specifies which fabric interconnect is used for communications with the target endpoint.
Step 8	UCS-A /org/service-profile/initiator-group/storage-connection-def/storage-target # **set target-vsan** *vsan*	Specifies which VSAN is used for communications with the target endpoint.
Step 9	UCS-A /org/service-profile/initiator-group # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a vHBA initiator group named initGroupZone1 with two vHBA initiators for a service profile named ServInst90, configures a local storage connection policy definition named scPolicyZone1, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile ServInst90
UCS-A /org/service-profile # create initiator-group initGroupZone1
UCS-A /org/service-profile/initiator-group* # create initiator vhba1
UCS-A /org/service-profile/initiator-group* # create initiator vhba2
UCS-A /org/service-profile/initiator-group* # create storage-connection-def scPolicyZone1
UCS-A /org/service-profile/initiator-group/storage-connection-def* # create storage-target
```

```
20:10:20:30:40:50:60:70
UCS-A /org/service-profile/initiator-group/storage-connection-def/storage-target* # set
target-path a
UCS-A /org/service-profile/initiator-group/storage-connection-def/storage-target* # set
target-vsan default
UCS-A /org/service-profile/initiator-group* # commit-buffer
UCS-A /org/service-profile/initiator-group #
```

Service Profiles and Service Profile Template Management

Associating a Service Profile with a Blade Server or Server Pool

Follow this procedure if you did not associate the service profile with a blade server or server pool when you created it, or to change the blade server or server pool with which a service profile is associated.

Procedure

	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 # **associate** {**server** *chassis-id* / *slot-id* | **server-pool** *pool-name qualifier*} [**restrict-migration**]	Associates the service profile with a single server, or to the specified server pool with the specified server pool policy qualifications. Adding the optional **restrict-migration** keyword prevents the service profile from being migrated to another server.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example associates the service profile named ServProf34 with the server in slot 4 of chassis 1 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile ServProf34
UCS-A /org/service-profile* # associate server 1/4
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Associating a Service Profile with a Rack Server

Follow this procedure if you did not associate the service profile with a rack server when you created it, or to change the rack server with which a service profile is associated.

Procedure

	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 # **associate server** *serv-id* [**restrict-migration**]	Associates the service profile with the specified rack server. Adding the optional the restrict-migration command prevents the service profile from being migrated to another server.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example associates the service profile named ServProf34 with the rack server 1 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile ServProf34
UCS-A /org/service-profile* # associate server 1
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Disassociating a Service Profile from a Server or Server Pool

This procedure covers disassociating a service profile from a blade server, rack server, or server pool.

Procedure

	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 # **disassociate**	Disassociates the service profile from the server or server pool.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example disassociates the service profile named ServProf34 from the server to which it was associated and commits the transaction:

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

Renaming a Service Profile

When you rename a service profile, the following occurs:

- Event logs and audit logs that reference the previous name for the service profile are retained under that name.
- A new audit record is created to log the rename operation.
- All records of faults against the service profile under its previous name are transferred to the new service profile name.

Note You cannot rename a service profile with pending changes.

Procedure

	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.
Step 3	UCS-A /org/service-profile # **rename-to** *new-profile-name*	Renames the specified service profile. When you enter this command, you are warned that you may lose all uncommitted changes in the CLI session. Type y to confirm that you want to continue. This name can be between 2 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and this name must be unique across all service profiles and service profile templates within the same organization.
Step 4	UCS-A /org/service-profile/ # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to change the name of a service profile from ServInst90 to ServZoned90 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope service-profile ServInst90
```

```
UCS-A /org/service-profile* # rename-to ServZoned90
Rename is a standalone operation. You may lose any uncommitted changes in this CLI session.
Do you want to continue? (yes/no): y
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Resetting the UUID Assigned to a Service Profile from a Pool in a Service Profile Template

If you change the UUID suffix pool assigned to an updating service profile template, Cisco UCS Manager does not change the UUID assigned to a service profile created with that template. If you want Cisco UCS Manager to assign a UUID from the newly assigned pool to the service profile, and therefore to the associated server, you must reset the UUID. You can only reset the UUID assigned to a service profile and its associated server under the following circumstances:

- The service profile was created from an updating service profile template and includes a UUID assigned from a UUID suffix pool.
- The UUID suffix pool name is specified in the service profile. For example, the pool name is not empty.
- The UUID value is not 0, and is therefore not derived from the server hardware.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the command mode for the organization for which you want to reset the UUID. If the system does not include multi-tenancy, type / as the *org-name* to enter the root organization.
Step 2	UCS-A /org # **scope service-profile** *profile-name*	Enters the service profile that requires the UUID for the associated server to be reset to a different UUID suffix pool.
Step 3	UCS-A /org/service-profile # **set identity dynamic-uuid derived**	Specifies that the service profile will obtain a UUID dynamically from a pool.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

This example resets the UUID of a service profile to a different UUID suffix pool:

```
UCS-A# scope org /
UCS-A /org # scope service-profile ServInst90
UCS-A /org/service-profile # set identity dynamic-uuid derived
UCS-A /org/service-profile* # commit-buffer
UCS-A /org/service-profile #
```

Resetting the MAC Address Assigned to a vNIC from a Pool in a Service Profile Template

If you change the MAC pool assigned to an updating service profile template, Cisco UCS Manager does not change the MAC address assigned to a service profile created with that template. If you want Cisco UCS Manager to assign a MAC address from the newly assigned pool to the service profile, and therefore to the associated server, you must reset the MAC address. You can only reset the MAC address assigned to a service profile and its associated server under the following circumstances:

- The service profile was created from an updating service profile template and includes a MAC address assigned from a MAC pool.
- The MAC pool name is specified in the service profile. For example, the pool name is not empty.
- The MAC address value is not 0, and is therefore not derived from the server hardware.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the command mode for the organization that contains the service profile for which you want to reset the MAC address. If the system does not include multi-tenancy, type / as the *org-name* to enter the root organization.
Step 2	UCS-A /org # **scope service-profile** *profile-name*	Enters the command mode for the service profile that requires the MAC address of the associated server to be reset to a different MAC address.
Step 3	UCS-A /org/service-profile # **scope vnic** *vnic-name*	Enters the command mode for the vNIC for which you want to reset the MAC address.
Step 4	UCS-A /org/service-profile/vnic # **set identity dynamic-mac derived**	Specifies that the vNIC will obtain a MAC address dynamically from a pool.
Step 5	UCS-A /org/service-profile/vnic # **commit-buffer**	Commits the transaction to the system configuration.

This example resets the MAC address of a vNIC in a service profile:

```
UCS-A# scope org /
UCS-A /org # scope service-profile ServInst90
UCS-A /org/service-profile #  scope vnic dynamic-prot-001
UCS-A /org/service-profile/vnic # set identity dynamic-mac derived
UCS-A /org/service-profile/vnic* # commit-buffer
UCS-A /org/service-profile/vnic #
```

Resetting the WWPN Assigned to a vHBA from a Pool in a Service Profile Template

If you change the WWPN pool assigned to an updating service profile template, Cisco UCS Manager does not change the WWPN assigned to a service profile created with that template. If you want Cisco UCS Manager to assign a WWPN from the newly assigned pool to the service profile, and therefore to the associated server, you must reset the WWPN. You can only reset the WWPN assigned to a service profile and its associated server under the following circumstances:

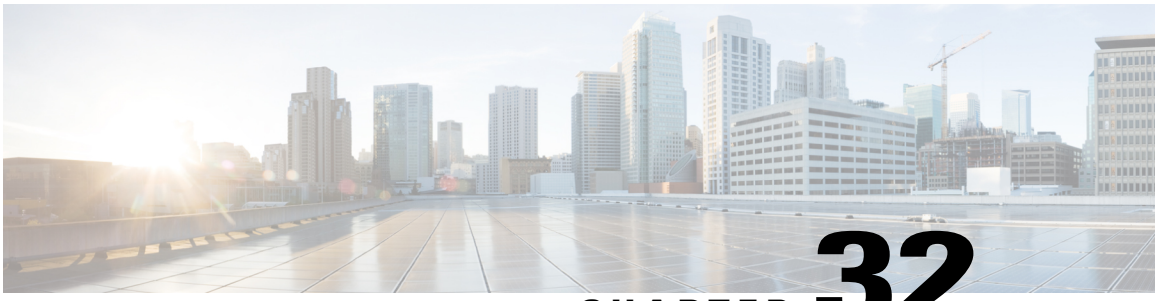
- The service profile was created from an updating service profile template and includes a WWPN assigned from a WWPN pool.
- The WWPN pool name is specified in the service profile. For example, the pool name is not empty.
- The WWPN value is not 0, and is therefore not derived from the server hardware.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the command mode for the organization that contains the service profile for which you want to reset the WWPN. If the system does not include multi-tenancy, type / as the *org-name* to enter the root organization.
Step 2	UCS-A /org # **scope service-profile** *profile-name*	Enters the service profile of the vHBA for which you want to reset the WWPN.
Step 3	UCS-A /org/service-profile # **scope vhba** *vhba-name*	Enters the command mode for vHBA for which you want to reset the WWPN.
Step 4	UCS-A /org/service-profile/vhba # **set identity dynamic-wwpn derived**	Specifies that the vHBA will obtain a WWPN dynamically from a pool.
Step 5	UCS-A /org/service-profile/vhba # **commit-buffer**	Commits the transaction to the system configuration.

This example resets the WWPN of a vHBA in a service profile:

```
UCS-A# scope org /
UCS-A /org # scope service-profile ServInst90
UCS-A /org/service-profile #  scope vhba vhba3
UCS-A /org/service-profile/vhba # set identity dynamic-wwpn derived
UCS-A /org/service-profile/vhba* # commit-buffer
UCS-A /org/service-profile/vhba #
```



CHAPTER 32

Configuring Storage Profiles

This part contains the following chapters:

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 638
- Redundant Virtual Drives with No Hot Spare Drives, on page 638
- Redundant Virtual Drives with Hot Spare Drives, on page 638
- Replacing Hot Spare Drives, on page 639
- Inserting Physical Drives into Unused Slots, on page 639

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 641

2 Automatically Configuring Disks in a Disk Group, on page 641 or Manually Configuring Disks in a Disk Group, on page 643

3 Configuring Virtual Drive Properties, on page 644

Setting the RAID Level

Procedure

	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 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 • 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** *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: • HDD • 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 for Cisco UCS C240, C220, C24, and C22 servers. For all other servers, the limit is 16 drives per server.. You

	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** *hot-spare-num*	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** *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 • glob-hot-spare: Global hot spare • normal
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.
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** *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
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

	Command or Action	Purpose
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
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.

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** *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	UCS-A /org* # **enter service-profile** *service-profile-name*	(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** *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	UCS-A /org # **scope service-profile** *service-profile-name*	(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 a Storage Profile PCH Controller Definition

You can create a PCH controller definition under a storage profile at the org level or at the service profile level.

Procedure

	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
		Note This task assumes the storage profile is at the org level. If the storage profile is at the service profile level, see the example below for the steps to scope to the storage profile definition under the service profile.
Step 2	UCS-A /org # **scope storage-profile** *storage-profile-name*	Enters storage-profile configuration mode for the selected storage profile.
Step 3	UCS-A /org/storage-profile # **create controller-def** *controller-definition-name*	Creates a PCH controller definition with the specified name and enters controller-definition configuration mode.
Step 4	UCS-A /org/storage-profile/controller-def* # **create controller-mode-config**	Creates a PCH controller configuration and enters controller-mode configuration mode.
Step 5	UCS-A /org/storage-profile/controller-def/controller-mode-config* # **set protect-config** {**yes**|**no**}	Specifies whether the server retains the configuration in the PCH controller even if the server is disassociated from the service profile.
Step 6	UCS-A /org/storage-profile/controller-def/controller-mode-config* # **set raid-mode** {**any-configuration** | **no-local-storage** | **no-raid** | **raid-0-striped** | **raid-1-mirrored** | **raid-5-striped-parity** | **raid-50--striped-parity-and-striped** | **raid-6-striped-dual-parity** | **raid-60-striped-dual-parity-and-striped** | **raid-10-mirrored-and-striped**}	Specifies the raid mode for the PCH controller.
Step 7	UCS-A /org/storage-profile/controller-def/controller-mode-config* # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to add a PCH controller definition called "raid1-controller" with raid mode set to RAID 1 Mirrored to the org-level storage profile named "storage-profile-A".

```
UCS-A# scope org /
UCS-A /org # scope storage-profile storage-profile-A
UCS-A /org/storage-profile # create controller-def raid1-controller
UCS-A /org/storage-profile/controller-def* # create controller-mode-config
UCS-A /org/storage-profile/controller-def/controller-mode-config* # set protect-config yes
UCS-A /org/storage-profile/controller-def/controller-mode-config* # set raid-mode
raid-1-mirrored
UCS-A /org/storage-profile/controller-def/controller-mode-config* # commit buffer
```

This example shows how to scope to the service profile called "Service-Profile1", create a storage profile, then create a PCH controller definition called "Raid60Ctrlr" within that storage profile. The controller definition has protection mode off and uses RAID 60 Striped Dual Parity and Striped.

```
UCS-A /org/service-profile # scope org /
UCS-A /org # scope service-profile Service-Profile1
UCS-A /org/service-profile # create storage-profile-def
UCS-A /org/service-profile/storage-profile-def* # create controller-def Raid60Ctrlr
UCS-A /org/service-profile/storage-profile-def/controller-def* # create controller-mode-config
UCS-A /org/service-profile/storage-profile-def/controller-def/controller-mode-config* # set
 protect-config no
UCS-A /org/service-profile/storage-profile-def/controller-def/controller-mode-config* # set
 raid-mode raid-60-striped-dual-parity-and-striped
UCS-A /org/service-profile/storage-profile-def/controller-def/controller-mode-config* #
commit-buffer
```

Deleting a Storage Profile PCH Controller Definition

Procedure

	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*. **Note** This task assumes the storage profile is at the org level. If the storage profile is at the service profile level, see the example below for the steps to scope to the storage profile definition under the service profile.
Step 2	UCS-A /org # **scope storage-profile** *storage-profile-name*	Enters storage-profile configuration mode for the selected storage profile.
Step 3	UCS-A /org/storage-profile # **delete controller-def** *controller-definition-name*	Deletes a PCH controller definition with the specified name.
Step 4	UCS-A /org/storage-profile* # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to delete a PCH controller definition called "raid1-controller" from the org-level storage profile named "storage-profile-A".

```
UCS-A# scope org
UCS-A /org # scope storage-profile storage-profile-A
UCS-A /org/storage-profile # delete controller-def raid1-controller
UCS-A /org/storage-profile* # commit-buffer
```

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** *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 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** *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.

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 sp1
```

```
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** *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* # **show local-lun**	(Optional) 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.

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** *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.

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**	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. • **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. • **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:

	Command or Action	Purpose
		• **Unknown** • **Optimal** • **Degraded** • **Inoperable** • **Partially Degraded**

```
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
```

Importing Foreign Configurations for a RAID Controller on a Blade Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** [*chassis-num*/*server-num* | *dynamic-uuid*]	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}	Enters RAID controller mode.

	Command or Action	Purpose
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** *server-id*	Enters server mode for the specified server.
Step 2	UCS-A /server # **scope raid-controller** *raid-contr-id* {**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** [*chassis-num*/*server-num* | *dynamic-uuid*]	Enters server mode for the specified server.

	Command or Action	Purpose
Step 2	UCS-A /chassis/server # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}	Enters RAID controller mode.
Step 3	UCS-A /chassis/server/raid-controller # **scope local-disk** *local-disk-id*	Enters local disk configuration mode.
Step 4	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**}	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.

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** *server-id*	Enters server mode for the specified server.
Step 2	UCS-A /server # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}	Enters RAID controller mode.
Step 3	UCS-A /server/raid-controller # **scope local-disk** *local-disk-id*	Enters local disk configuration mode.
Step 4	UCS-A /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**}	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. • **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** [*chassis-num*/*server-num* | *dynamic-uuid*]	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}	Enters RAID controller chassis mode.
Step 3	UCS-A /chassis/server/raid-controller # **delete virtual-drive id** *virtual-drive-id*	(Optional) Deletes the orphaned virtual drive with the specified virtual drive ID.
Step 4	UCS-A /chassis/server/raid-controller # **delete virtual-drive name** *virtual-drive-name*	(Optional) Deletes the orphaned virtual drive with the specified virtual drive name.
Step 5	UCS-A /chassis/server/raid-controller # **scope virtual-drive** *virtual-drive-id*	(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** *server-id*	Enters server mode for the specified server.
Step 2	UCS-A /server # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}	Enters RAID controller mode.
Step 3	UCS-A /server/raid-controller # **delete virtual-drive id** *virtual-drive-id*	(Optional) Deletes the orphaned virtual drive with the specified virtual drive ID.
Step 4	UCS-A /server/raid-controller # **delete virtual-drive name** *virtual-drive-name*	(Optional) Deletes the orphaned virtual drive with the specified virtual drive name.
Step 5	UCS-A /server/raid-controller # **scope virtual-drive** *virtual-drive-id*	(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** [*chassis-num*/*server-num* | *dynamic-uuid*]	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}	Enters RAID controller chassis mode.
Step 3	UCS-A /chassis/server/raid-controller # **scope virtual-drive** *virtual-drive-id*	Enters virtual drive mode for the specified virtual drive.
Step 4	UCS-A /chassis/server/raid-controller/virtual-drive # **set name** *virtual-drive-name*	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 vd1
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** *server-id*	Enters server mode for the specified server.
Step 2	UCS-A /server # **scope raid-controller** *raid-contr-id* {**sas** | **sata**}	Enters RAID controller mode.
Step 3	UCS-A /server/raid-controller # **scope virtual-drive** *virtual-drive-id*	Enters virtual drive mode for the specified virtual drive.
Step 4	UCS-A /server/raid-controller/virtual-drive # **set name** *virtual-drive-name*	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 vd1
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** *org-name*	Enters organization mode for the specified organization. To enter the

	Command or Action	Purpose
		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.
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/*local-storage-device* # **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 template.

Configuring the Boot Policy for a Local JBOD Disk

Procedure

	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-jbod**	Specifies a local JBOD disk as the local storage. JBOD is supported only on the following servers: • 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** *slot_number*	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 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 668
- Deploying and Undeploying a LUN, on page 669
- Renaming a Service Profile Referenced LUN, on page 670

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** *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.

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.

Procedure

	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.

	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** *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.

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
```

Viewing the Local Disk Locator LED State

Procedure

Step 1 UCS-A# **scope server** *id*
Enters server mode for the specified server.

Step 2 UCS-A/server # **scope local-disk** *id*
Enters the RAID controller for the specified local disk.

Step 3 UCS-A/server/local-disk # **show locator-led**
Shows the state of the disk locator LED.

The following example shows that the state of the local disk Locator LED is on:

```
USA-A# scope server 1
USA-A /server # scope local-disk 2
USA-A /serverlocal-disk # show locator-led
Locator LED:
    Equipment        Operational State
    ---------------- -----------------
    1/SAS-1/2        On
```

Turning On the Local Disk Locator LED

Procedure

Step 1 UCS-A# **scope server** *id*
Enters server mode for the specified server.

Step 2 UCS-A/server # **scope local-disk** *id*
Enters the RAID controller for the specified local disk.

Step 3 UCS-A /server/local-disk # **enable locator-led**
Turns on the disk locator LED.

Step 4 UCS-A/server/local-disk* # **commit-buffer**
Commits the command to the system configuration.

The following example displays how to turn on the local disk Locator LED:

```
UCS-A# scope server 1
UCS-A /server/raid-controller # scope local-disk 2
USA-A /server/raid-controller/local-disk # enable locator-led
USA-A /server/raid-controller/local-disk* # commit-buffer
```

Turning Off the Local Disk Locator LED

Procedure

Step 1 UCS-A# **scope server** *id*
Enters server mode for the specified server.

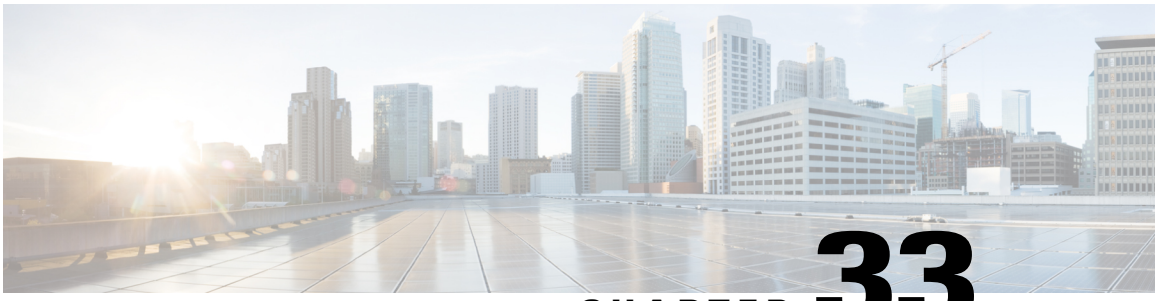
Step 2 UCS-A/server # **scope local-disk** *id*
Enters the RAID controller for the specified local disk.

Step 3 UCS-A/server/local-disk # **disable locator-led**
Turns off the disk locator LED.

Step 4 UCS-A/server/raid-controller/local-disk* # **commit-buffer**
Commits the command to the system configuration.

The following example displays how to disable the local disk Locator LED:

```
UCS-A# server 1
UCS-A /server # scope local-disk 2
USA-A /server/local-disk # disable locator-led
USA-A /server/local-disk* # commit-buffer
```



CHAPTER 33

Managing Power in Cisco UCS

This chapter includes the following sections:

Power Capping in Cisco UCS

You can control the maximum power consumption on a server through power capping, as well as manage the power allocation in the Cisco UCS Manager for the UCS B-Series Blade Servers, UCS Mini, and mixed UCS domains.

UCS Manager supports power capping on the following servers:

- UCS Mini 6324
- UCS 6300 Series Fabric Interconnects

You can use Policy Driven Chassis Group Power Cap, or Manual Blade Level Power Cap methods to allocate power that applies to all of the servers in a chassis.

Cisco UCS Manager provides the following power management policies to help you allocate power to your servers:

Power Management Policies	Description
Power Policy	Specifies the redundancy for power supplies in all chassis in a Cisco UCS domain.
Power Control Policies	Specifies the priority to calculate the initial power allocation for each blade in a chassis.
Global Power Allocation	Specifies the Policy Driven Chassis Group Power Cap or the Manual Blade Level Power Cap to apply to all servers in a chassis.
Global Power Profiling	Specifies how the power cap values of the servers are calculated. If it is enabled, the servers will be profiled during discovery through benchmarking. This policy applies when the Global Power Allocation Policy is set to Policy Driven Chassis Group Cap.

Viewing Power Measured for Blades

Procedure

	Command or Action	Purpose
Step 1	UCS-A# power-cap-mgmt # **show power-measured** .	Displays the measured power.

The following example lists the minimum and maximum power measured for blades.

```
UCS-A# show power-measured
Measured Power:
```

```
Device Id (W)  Minimum power (W) Maximum power (W) OperMethod
-------------- ----------------- ----------------- ----------
blade   1/1    168               252               Pnuos
blade   1/2    350               580               Static
blade   1/3    350               560               Static
blade   1/4    350               398               Static
blade   1/5    350               544               Static
blade   1/6    350               560               Static
blade   1/7    180               276               Pnuos
blade   1/8    350               544               Static
```

Rack Server Power Management

Power capping is not supported for rack servers.

Power Management Precautions

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.

Configuring the Power Policy

Power Policy for Cisco UCS Servers

The power policy is global and is inherited by all of the chassis' managed by the Cisco UCS Manager instance. You can add the power policy to a service profile to specify the redundancy for power supplies in all chassis' in the Cisco UCS domain. This policy is also known as the PSU policy.

For more information about power supply redundancy, see *Cisco UCS 5108 Server Chassis Hardware Installation Guide*.

Configuring the Power Policy

Procedure

	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 psu-policy**	Enters PSU policy mode.
Step 3	UCS-A /org/psu-policy # **set redundancy** {**grid** | **n-plus-1** | **non-redund**}	Specifies one of the following redundancy types: • **grid —**Two power sources are turned on, or the chassis requires greater than N+1 redundancy. If one source fails (which causes

	Command or Action	Purpose
		a loss of power to one or two PSUs), the surviving PSUs on the other power circuit continue to provide power to the chassis. • **n-plus-1 —**The total number of PSUs to satisfy non-redundancy, plus one additional PSU for redundancy, are turned on and equally share the power load for the chassis. If any additional PSUs are installed, Cisco UCS Manager sets them to a "turned-off" state. • **non-redund —**All installed power supplies (PSUs) are turned on and the load is evenly balanced. Only smaller configurations (requiring less than 2500W) can be powered by a single PSU. For more information about power redundancy, see the *Cisco UCS 5108 Server Chassis Installation Guide*.
Step 4	UCS-A /org/psu-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures the power policy to use grid redundancy and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope psu-policy
UCS-A /org/psu-policy # set redundancy grid
UCS-A /org/psu-policy* # commit-buffer
UCS-A /org/psu-policy #
```

Viewing and Modifying the Global Power Profiling Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A /power-cap-mgmt # **show profile-policy**	Displays the power profile policy.
Step 2	UCS-A /power-cap-mgmt # **set profile** {**no** | **yes**}	Set the profile policy.
Step 3	UCS-A /power-cap-mgmt* # **comm-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /power-cap-mgmt # **show profile-policy**	Displays whether the global power profiling policy is on. **Global Power Profiling Policy**: **Power Profiling** **Yes**

The following example show how to display the global power profiling policy

```
UCS-A /power-cap-mgmt # show profile-policy
Global Power Profiling Policy:
    Power Profiling
    ---------------
    No

UCS-A /power-cap-mgmt # set profile-policy
no   yes

UCS-A /power-cap-mgmt # set profile-policy yes
UCS-A /power-cap-mgmt* # comm-buffer
UCS-A /power-cap-mgmt # show profile-policy

Global Power Profiling Policy:
    Power Profiling
    ---------------
    Yes
```

Configuring the Global Power Allocation Policy

Global Power Allocation Policy

The Global Power Allocation Policy allows you to specify the Policy Driven Chassis Group Power Cap or Manual Blade-level Power Cap power allocation method applied to servers in a chassis.

Cisco recommends using the default Policy Driven Chassis Group Power Cap power allocation method.

Important Any change to the Manual Blade level Power Cap configuration results in the loss of any groups or configuration options set for the Policy Driven Chassis Group Power Cap.

Configuring the Global Power Allocation Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope power-cap-mgmt**	Enters power cap management mode.
Step 2	UCS-A /power-cap-mgmt # **set cap-policy** {**manual-blade-level-cap** | **policy-driven-chassis-group-cap**}	Sets the global cap policy to the specified power cap management mode. By default, the global cap policy is set to policy driven chassis group cap.
Step 3	UCS-A /power-cap-mgmt # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the global cap policy to manual blade power cap and commits the transaction:

```
UCS-A# scope power-cap-mgmt
UCS-A /power-cap-mgmt # set cap-policy manual-blade-level-cap
UCS-A /power-cap-mgmt* # commit-buffer
UCS-A /power-cap-mgmt #
```

Viewing the Power Cap Values for Servers

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope power-cap-mgmt**	Enters power cap management mode.
Step 2	UCS-A /power-cap-mgmt # **show power-measured**	Displays the minimum and maximum power cap values.

The following example shows how to display the minimum and maximum power cap values:

```
UCS-A# scope power-cap-mgmt
UCS-A /power-cap-mgmt # show power-measured

Measured Power:
    Device Id (W)  Minimum power (W) Maximum power (W) OperMethod
    -------------- ----------------- ----------------- ----------
    blade   1/1    234               353               Pnuos

UCS-A /power-cap-mgmt #
```

Configuring Policy-Driven Chassis Group Power Capping

Policy Driven Chassis Group Power Capping

When you select the Policy Driven Chassis Group Power Cap in the Global Cap Policy, Cisco UCS can maintain the over-subscription of servers without risking power failures. You can achieve over-subscription through a two-tier process. For example, at the chassis level, Cisco UCS divides the amount of power available among members of the power group, and at the blade level, the amount of power allotted to a chassis is divided among blades based on priority.

Each time a service profile is associated or disassociated, Cisco UCS Manager recalculates the power allotment for each blade server within the chassis. If necessary, power from lower-priority service profiles is redistributed to higher-priority service profiles.

UCS power groups cap power in less than one second to safely protect data center circuit breakers. A blade must stay at its cap for 20 seconds before the chassis power distribution is optimized. This is intentionally carried out over a slower timescale to prevent reacting to transient spikes in demand.

Note The system reserves enough power to boot a server in each slot, even if that slot is empty. This reserved power cannot be leveraged by servers requiring more power. Blades that fail to comply with the power cap are penalized.

Power Groups in UCS Manager

A power group is a set of chassis that all draw power from the same power distribution unit (PDU). In Cisco UCS Manager, you can create power groups that include one or more chassis, then set a peak power cap in AC watts for that power grouping.

Implementing power capping at the chassis level requires the following:

- IOM, CIMC, and BIOS version 1.4 or higher
- Two Power Supply Units (PSUs)

The peak power cap is a static value that represents the maximum power available to all blade servers within a given power group. If you add or remove a blade from a power group, but do not manually modify the peak power value, the power group adjusts the peak power cap to accommodate the basic power-on requirements of all blades within that power group.

A minimum of 890 AC watts should be set for each chassis. This converts to 800 watts of DC power, which is the minimum amount of power required to power an empty chassis. To associate a half-width blade, the group cap needs to be set to 1475 AC watts. For a full-width blade, it needs to be set to 2060 AC watts.

After a chassis is added to a power group, all service profile associated with the blades in the chassis become part of that power group. Similarly, if you add a new blade to a chassis, that blade inherently becomes part of the chassis' power group.

Note Creating a power group is not the same as creating a server pool. However, you can populate a server pool with members of the same power group by creating a power qualifier and adding it to server pool policy.

When a chassis is removed or deleted, the chassis gets removed from the power group.

UCS Manager supports explicit and implicit power groups.

- **Explicit**: You can create a power group, add chassis' and racks, and assign a budget for the group.
- **Implicit**: Ensures that the chassis is always protected by limiting the power consumption within safe limits. By default, all chassis that are not part of an explicit power group are assigned to the default group and the appropriate caps are placed. New chassis that connect to UCS Manager are added to the default power group until you move them to a different power group.

The following table describes the error messages you might encounter while assigning power budget and working with power groups.

Error Message	Cause	Recommended Action

`Insufficient budget for power group POWERGROUP_NAME`
and/or
`Chassis N cannot be capped as group cap is low. Please consider raising the cap.`
and/or
`Admin committed insufficient for power group GROUP_NAME, using previous value N`
and/or
`Power cap application failed for chassis N`
	One of these messages displays if you did not meet the minimum limit when assigning the power cap for a chassis, or the power requirement increased because of the addition of blades or change of power policies.	Increase the power cap limit to the **Minimum Power Cap for Allowing Operations (W)** value displayed on the **Power Group** page for the specified power group.

`Chassis N cannot be capped as the available PSU power is not enough for the chassis and the blades. Please correct the problem by checking input power or replace the PSU`	Displays when the power budget requirement for the chassis is more than the PSU power that is available.	Check the PSU input power and redundancy policy to ensure that enough power is available for the chassis. If a PSU failed, replace the PSU.
`Power cap application failed for server N`	Displays when the server is consuming more power than allocated and cannot be capped, or the server is powered on when no power is allocated.	Do not power on un-associated servers.
`P-State lowered as consumption hit power cap for server`	Displays when the server is capped to reduce the power consumption below the allocated power.	This is an information message. If a server should not be capped, in the service profile set the value of the power control policy **Power Capping** field to **no-cap**.
`Chassis N has a mix of high-line and low-line PSU input power sources.`	This fault is raised when a chassis has a mix of high-line and low-line PSU input sources connected.	This is an unsupported configuration. All PSUs must be connected to similar power sources.

Creating a Power Group

Before You Begin

Ensure that the global power allocation policy is set to Policy Driven Chassis Group Cap.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope power-cap-mgmt**	Enters power cap management mode.
Step 2	UCS-A /power-cap-mgmt # **create power-group** *power-group-name*	Creates a power group and enters power group mode.
Step 3	UCS-A /power-cap-mgmt/power-group # **set peak** {*peak-num* | **disabled** | **uninitialized**}	Specifies the maximum peak power (in watts) available to the power group.
Step 4	UCS-A /power-cap-mgmt/power-group # **create chassis** *chassis-id*	Adds the specified chassis to the power group and enters power group chassis mode.
Step 5	UCS-A /power-cap-mgmt/power-group/chassis # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a power group called powergroup1, specifies the maximum peak power for the power group (10000 watts), adds chassis 1 to the group, and commits the transaction:

```
UCS-A# scope power-cap-mgmt
UCS-A /power-cap-mgmt # create power-group powergroup1
UCS-A /power-cap-mgmt/power-group* # set peak 10000
UCS-A /power-cap-mgmt/power-group* # create chassis 1
UCS-A /power-cap-mgmt/power-group/chassis* # commit-buffer
UCS-A /power-cap-mgmt/power-group/chassis #
```

Deleting a Power Group

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope power-cap-mgmt**	Enters power cap management mode.
Step 2	UCS-A /power-cap-mgmt # **delete power-group** *power-group-name*	Deletes the specified power group.
Step 3	UCS-A /power-cap-mgmt/power-group/chassis # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a power group called powergroup1 and commits the transaction:

```
UCS-A# scope power-cap-mgmt
UCS-A /power-cap-mgmt # delete power-group powergroup1
UCS-A /power-cap-mgmt* # commit-buffer
UCS-A /power-cap-mgmt #
```

Power Control Policy

Cisco UCS uses the priority set in the power control policy along with the blade type and configuration to calculate the initial power allocation for each blade within a chassis. During normal operation, the active blades within a chassis can borrow power from idle blades within the same chassis. If all blades are active and reach the power cap, service profiles with higher priority power control policies take precedence over service profiles with lower priority power control policies.

Priority is ranked on a scale of 1-10, where 1 indicates the highest priority and 10 indicates lowest priority. The default priority is 5.

For mission-critical application a special priority called no-cap is also available. Setting the priority to no-cap prevents Cisco UCS from leveraging unused power from a particular server. With this setting, the server is allocated the maximum amount of power possible for that type of server.

Note You must include the power control policy in a service profile and that service profile must be associated with a server for it to take effect.

Creating a Power Control Policy

Procedure

	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 # **create power-control-policy** *power-control-pol-name*	Creates a power control policy and enters power control policy mode.
Step 3	UCS-A /org/power-control-policy # **set priority** {*priority-num* | **no-cap**}	Specifies the priority for the power control policy.
Step 4	UCS-A /org/power-control-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a power control policy called powerpolicy15, sets the priority at level 2, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create power-control-policy powerpolicy15
UCS-A /org/power-control policy* # set priority 2
```

```
UCS-A /org/power-control policy* # commit-buffer
UCS-A /org/power-control policy #
```

What to Do Next

Include the power control policy in a service profile.

Deleting a Power Control Policy

Procedure

	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 # **delete power-control-policy** *power-control-pol-name*	Deletes the specified power control policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes a power control policy called powerpolicy15 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete power-control-policy powerpolicy15
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Manual Blade-Level Power Capping

Manual Blade Level Power Cap

When manual blade-level power cap is configured in the global cap policy, you can set a power cap for each blade server in a Cisco UCS domain.

The following configuration options are available:

- **Watts—**You can specify the maximum amount of power that the server can consume at one time. This maximum can be any amount between 0 watts and 1100 watts.
- **Unbounded—**No power usage limitations are imposed on the server. The server can use as much power as it requires.

If the server encounters a spike in power usage that meets or exceeds the maximum configured for the server, Cisco UCS Manager does not disconnect or shut down the server. Instead, Cisco UCS Manager reduces the power that is made available to the server. This reduction can slow down the server, including a reduction in CPU speed.

Note If you configure the manual blade-level power cap using **Equipment** > **Policies** > **Global Policies** > **Global Power Allocation Policy**, the priority set in the Power Control Policy is no longer relevant.

Setting the Blade-Level Power Cap for a Server

Before You Begin

Ensure that the global power allocation policy is set to Manual Blade Level Cap.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-id* / *server-id*	Enters chassis server mode for the specified server.
Step 2	UCS-A /chassis/server # **set power-budget committed** {**unbounded** | *watts*}	Commits the server to one of the following power usage levels: • **unbounded —**Does not impose any power usage limitations on the server. • *watts* **—**Allows you to specify the upper level for power usage by the server. If you choose this setting, enter the maximum number of watts that the server can use. The range is 0 to 10000000 watts.
Step 3	UCS-A /chassis/server # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /chassis/server # **show power-budget**	(Optional) Displays the power usage level setting.

The following example limits the power usage for a server to 1000 watts and commits the transaction:

```
UCS-A# scope server 1/7
UCS-A /chassis/server # show power-budget

Budget:
   AdminCommitted (W)
   ------------------
   139

UCS-A /chassis/server # set power-budget committed unbounded
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server # show power-budget

Budget:
   AdminCommitted (W)
   ------------------
   Unbounded

UCS-A /chassis/server # set power-budget committed 1000
```

```
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server # show power-budget


Budget:
  AdminCommitted (W)
------------------
  1000
UCS-A /chassis/server #
```

Viewing the Blade-Level Power Cap

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-id* / *server-id*	Enters chassis server mode for the specified server.
Step 2	UCS-A /chassis/server # **show stats**	Displays the power usage statistics collected for the server.

The following example shows the server power usage:

```
UCS-A# scope server 2/4
UCS-A /chassis/server # show stats

Mb Power Stats:
    Time Collected: 2010-04-15T21:18:04.992
    Monitored Object: sys/chassis-1/blade-2/board
    Suspect: No
    Consumed Power (W): 118.285194
    Input Voltage (V): 11.948000
    Input Current (A): 9.900000
    Thresholded: Input Voltage Min

UCS-A /chassis/server #
```

Power Sync Policy

Cisco UCS Manager includes a global (default) power sync policy to address power synchronization issues between the associated service profiles and the servers. You can use the power sync policy to synchronize the power state when the desired power state of the service profile differs from the actual power state of the server. The power sync policy allows you to control when to synchronize the desired power state on the associated service profiles for M-series modular servers, rack-mount servers, and blade servers. The power sync policy does not affect other power-related policies.

The power sync policy applies to all the service profiles by default. You cannot delete the default power sync policy, but you can edit the default policy. You can create your own power sync policies and apply them to the service profiles. You can also create a power sync policy that is specific to a service profile and it always takes precedence over the default policy.

Cisco UCS Manager creates a fault on the associated service profile when the power sync policy referenced in the service profile does not exist. Cisco UCS Manager automatically clears the fault once you create a

power sync policy for the specified service profile or change the reference to an existing policy in the service profile.

Power Synchronization Behavior

Cisco UCS Manager synchronizes the power state only when the actual power state of the server is OFF. The current power synchronization behavior is based on the actual power state and the desired power state after shallow association occurs.
For example, the following events trigger shallow association:

- Fabric Interconnects(FI) and IOM disconnected.
- IOM reset
- FI power loss or reboot
- Chassis reacknowledgment
- Chassis power loss
- Service profile change

The following table describes the current power synchronization behavior:

Event	Desired Power State	Actual Power State Before Event	Actual Power State After Event
Shallow Association	ON	OFF	ON
Shallow Association	OFF	OFF	OFF
Shallow Association	ON	ON	ON
Shallow Association	OFF	ON	ON

Displaying the Global Power Sync Policy

Procedure

	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 power-sync-policy default**	Enters the global power sync policy mode.

	Command or Action	Purpose
Step 3	UCS-A /org/power/-sync-policy # **show** {**detail** | **expand** | **detail expand** }	Displays the global power sync policy information.

The following example displays the global (default) power sync policy:

```
UCS-A # scope org
UCS-A /org # scope power-sync-policy default-sync
UCS-A /org/power-sync-policy # show expand

Power Sync Policy:
    Name                 Power Sync Option
    -------------------- -----------------
    default              Default Sync

UCS-A /org/power-sync-policy # show detail expand

Power Sync Policy:
    Full Name: org-root/power-sync-default
    Name: default
    Description:
    Power Sync Option: Default Sync
    Policy Owner: Local

UCS-A /org/power-sync-policy #
```

Setting Global Policy Reference for a Service Profile

To refer the global power sync policy in a service profile, use the following commands in service profile mode:

Procedure

	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** *service-profile-name*	Enters the service profile mode for the specified service profile. The name of the service profile can be a minimum of two characters and a maximum up to 32 characters.
Step 3	UCS-A /org/service-profile # **set power-sync-policy default**	Specifies the global power sync policy that can be referenced in the service profile. You can also change the policy reference from the default to other power sync policies using this command.
Step 4	UCS-A /org/service-profile* # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets the reference to the global power sync policy for use in the service profile.

```
UCS-A # scope org
    UCS-A/org # scope service-profile spnew
    UCS-A/org/service-profile # set power-sync-policy default
    UCS-A/org/service-profile* # commit-buffer
```

Creating a Power Sync Policy

Procedure

	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 # **create power-sync-policy** *power-sync-pol-name*	Creates a power sync policy and enters power sync policy mode. The power sync policy name can be up to 16 characters.
Step 3	UCS-A /org/power-sync-policy* # **set descr** *optionall-description*	(Optional) Specifies the description of the power-sync-policy. You can also modify the description using the descr keyword.
Step 4	UCS-A /org/power-sync-policy* # **set sync-option** { **always-sync** | **default-sync** | **initial-only-sync** }	Specifies the power synchronization option to the physical server. You can also modify the power synchronization option using the sync-option keyword. This can be one of the following: • **Default Sync—**After the initial server association, any configuration change or management connectivity changes that you perform trigger a server reassociation. This option synchronizes the desired power state to the physical server if the physical server power state is off and the desired power state is on. This is the default behavior. • **Always Sync—**When the initial server association or the server reassociation occurs, this option always synchronizes the desired power state to the physical server even if the physical server power state is on and the desired power state is off. • **Initial Only Sync—**This option only synchronizes the power to a server when a service profile is associated to the server for the first time or when the server is re-commissioned. When you set this option, resetting the power state from the physical server side does not affect the desired power state on the service profile.
Step 5	UCS-A /org/power-sync-policy* # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a power sync policy called newSyncPolicy, sets the default sync-option, and commits the transaction to the system configuration:

```
UCS-A # scope org
UCS-A /org # create power-sync-policy newSyncPolicy
UCS-A /org/power-sync-policy* # set decsr newSyncPolicy
UCS-A /org/power-sync-policy* # set sync-option default-sync
UCS-A /org/power-sync-policy* # commit-buffer
UCS-A /org/power-sync-policy #
```

What to Do Next

Include the power sync policy in a service profile or in a service profile template.

Deleting a Power Sync Policy

Procedure

	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 # **delete power-sync-policy** *power-sync-pol-name*	Deletes the specified power sync policy.
Step 3	UCS-A /org # **commit buffer**	Commits the transaction to the system configuration.

The following example deletes the power sync policy called spnew and commits the transaction to the system:

```
UCS-A # scope org
UCS-A /org # delete power-sync-policy spnew
UCS-A /org # commit-buffer
```

Displaying All Power Sync Policies

Procedure

	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.

	Command or Action	Purpose
Step 2	UCS-A /org # **show power-sync-policy** {**detail** | **expand** | **detail expand** }	Displays the default, local, and other power sync policies.

The following example displays power sync policies that are defined:

```
UCS-A # scope org
UCS-A /org # show power-sync-policy expand
Power Sync Policy:
    Name                 Power Sync Option
    -------------------- -----------------
    default              Default Sync
    policy-1             Default Sync

UCS-A /org # show power-sync-policy detail expand
Power Sync Policy:
    Full Name: org-root/power-sync-default
    Name: default
    Description:
    Power Sync Option: Default Sync
    Policy Owner: Local

    Full Name: org-root/power-sync-policy-1
    Name: policy-1
    Description:
    Power Sync Option: Default Sync
    Policy Owner: Local

UCS-A /org #
```

Creating a Local Policy

To create a local power sync policy that you want to use by any service profile, create a power sync definition for the power sync policy.

Procedure

	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** *service-profile-name*	Enters the service profile mode for the specified service profile. The name of the service profile can be a minimum of two characters and a maximum up to 32 characters.
Step 3	UCS-A /org/service-profile # **create power-sync-definition**	Enters the power sync definition mode. You can create a power sync policy definition that you defined for the power sync policy.

	Command or Action	Purpose
Step 4	UCS-A /org/service-profile/power-sync-definition* # **set descr** *optional-description*	(Optional) Specifies the description of the power-sync-policy. You can also change the description using the descr keyword.
Step 5	UCS-A /org/service-profile/power-sync-definition* # **set sync-option** { **always-sync** | **default-sync** | **initial-only-sync** }	Specifies the power synchronization option to the physical server. You can also change the power synchronization option using the sync-option keyword.
Step 6	UCS-A /org/service-profile/power-sync-definition* # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates a local policy using the policy sync definition, sets the sync-option, and commits the transaction to the system configuration:

```
 UCS-A # scope org
 UCS-A/org # scope service-profile spnew
UCS-A/org/service-profile # create power-sync-definition
UCS-A/org/service-profile/power-sync-definition* # set decsr spnew
UCS-A/org/service-profile/power-sync-definition* # set sync-option default-sync
UCS-A/org/service-profile/power-sync-definition* #  commit-buffer
```

Showing a Local Policy

Procedure

	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** *service-profile-name*	Enters the service profile mode for the specified service profile. The name of the service profile can be a minimum of two characters and a maximum up to 32 characters.
Step 3	UCS-A /org/service-profile # **show power-sync-policy** {**detail** | **expand** | **detail expand** }	(Optional) Displays the local policy in the power-sync-policy mode.
Step 4	UCS-A /org/service-profile # **show power-sync-definition** {**detail** | **expand** | **detail expand** }	Displays the local policy for the specified service policy in the power-sync-definition mode. **Note** If you do not have a definition for the power sync policy, you can still use the command, but you cannot see anything displayed.

The following example displays the local policy in use by the service profile spnew:

```
 UCS-A # scope org
UCS-A/org # scope service-profile spnew
UCS-A/org/service-profile # show power-sync-definition expand

Power Sync Definition:
    Name                 Power Sync Option
    -------------------- -----------------
    spnew                Always Sync

UCS-A/org/service-profile # show power-sync-definition detail expand

Power Sync Definition:
    Full Name: org-root/ls-sp2/power-sync-def
    Name: spnew
    Description: optional description
    Power Sync Option: Always Sync
    Policy Owner: Local

UCS-A/org/service-profile #
```

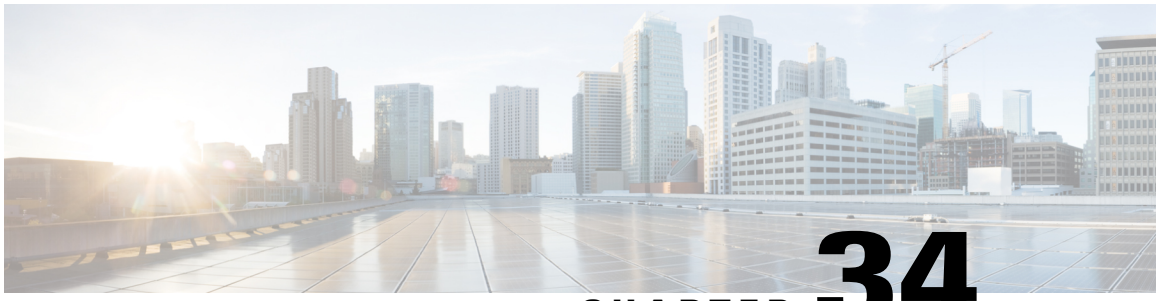
Deleting a Local Policy

Procedure

	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** *service-profile-name*	Enters the service profile mode for the specified service profile. The name of the service profile can be a minimum of two characters and a maximum up to 32 characters.
Step 3	UCS-A /org/service-profile # **delete power-sync-definition**	Enters the power sync definition mode. You can delete a power sync policy definition that you defined for the power sync policy.
Step 4	UCS-A /org/service-profile* # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the local policy in use by the service profile.

```
UCS-A # scope org
    UCS-A/org # scope service-profile spnew
    UCS-A/org/service-profile # delete power-sync-definition
    UCS-A/org/service-profile* # commit-buffer
```



CHAPTER 34

Managing Time Zones

This chapter includes the following sections:

Time Zones

Cisco UCS requires a domain-specific time zone setting and an NTP server to ensure the correct time display in Cisco UCS Manager. If you do not configure both of these settings in a Cisco UCS domain, the time does not display correctly.

Setting the Time Zone

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **set timezone**	At this point, you are prompted to enter a number corresponding to your continent, country, and time zone region. Enter the appropriate information at each prompt. When you have finished specifying the location information, you are prompted to confirm that the

	Command or Action	Purpose
		correct time zone information is being set. Enter **1** (yes) to confirm, or **2** (no) to cancel the operation.
Step 4	UCS-A /system/services # **commit-buffer**	Commits the transaction to the system configuration.
Step 5	UCS-A /system/services # **exit**	Enters system mode.
Step 6	UCS-A /system/services # **exit**	Enters EXEC mode.
Step 7	UCS-A /system/services # **show timezone**	Displays the configured timezone.

The following example configures the time zone to the Pacific time zone region, commits the transaction, and displays the configured time zone:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # set timezone
Please identify a location so that time zone rules can be set correctly.
Please select a continent or ocean.
1) Africa            4) Arctic Ocean     7) Australia       10) Pacific Ocean
2) Americas          5) Asia             8) Europe
3) Antarctica        6) Atlantic Ocean   9) Indian Ocean
#? Artic ocean
Please enter a number in range.
#? 2
Please select a country.
 1) Anguilla              18) Ecuador              35) Paraguay
 2) Antigua & Barbuda     19) El Salvador          36) Peru
 3) Argentina             20) French Guiana        37) Puerto Rico
 4) Aruba                 21) Greenland            38) St Kitts & Nevis
 5) Bahamas               22) Grenada              39) St Lucia
 6) Barbados              23) Guadeloupe           40) St Pierre & Miquelon
 7) Belize                24) Guatemala            41) St Vincent
 8) Bolivia               25) Guyana               42) Suriname
 9) Brazil                26) Haiti                43) Trinidad & Tobago
10) Canada                27) Honduras             44) Turks & Caicos Is
11) Cayman Islands        28) Jamaica              45) United States
12) Chile                 29) Martinique           46) Uruguay
13) Colombia              30) Mexico               47) Venezuela
14) Costa Rica            31) Montserrat           48) Virgin Islands (UK)
15) Cuba                  32) Netherlands Antilles 49) Virgin Islands (US)
16) Dominica              33) Nicaragua
17) Dominican Republic    34) Panama
#? 45
Please select one of the following time zone regions.
 1) Eastern Time
 2) Eastern Time - Michigan - most locations
 3) Eastern Time - Kentucky - Louisville area
 4) Eastern Time - Kentucky - Wayne County
 5) Eastern Standard Time - Indiana - most locations
 6) Eastern Standard Time - Indiana - Crawford County
 7) Eastern Standard Time - Indiana - Starke County
 8) Eastern Standard Time - Indiana - Switzerland County
 9) Central Time
10) Central Time - Michigan - Wisconsin border
11) Central Time - North Dakota - Oliver County
12) Mountain Time
13) Mountain Time - south Idaho & east Oregon
14) Mountain Time - Navajo
15) Mountain Standard Time - Arizona
16) Pacific Time
```

```
17) Alaska Time
18) Alaska Time - Alaska panhandle
19) Alaska Time - Alaska panhandle neck
20) Alaska Time - west Alaska
21) Aleutian Islands
22) Hawaii
#? 16

The following information has been given:

        United States
        Pacific Time

Therefore timezone 'America/Los_Angeles' will be set.
Local time is now:      Fri May 15 07:39:25 PDT 2009.
Universal Time is now:  Fri May 15 14:39:25 UTC 2009.
Is the above information OK?
1) Yes
2) No
#? 1
UCS-A /system/services* # commit-buffer
UCS-A /system/services # exit
UCS-A /system # exit
UCS-A# show timezone
Timezone: America/Los_Angeles (Pacific Time)
UCS-A#
```

Adding an NTP Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **create ntp-server** {*hostname* | *ip-addr*|*ip6-addr*}	Configures the system to use the NTP server with the specified hostname, IPv4 or IPv6 address.
Step 4	UCS-A /system/services # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures an NTP server with the IP address 192.168.200.101 and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # create ntp-server 192.168.200.101
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

The following example configures an NTP server with the IP address 4001::6 and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # create ntp-server 4001::6
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

Deleting an NTP Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **delete ntp-server** {*hostname* | *ip-addr*|*ip6-addr*}	Deletes the NTP server with the specified hostname, IPv4 or IPv6 address.

The following example deletes the NTP server with the IP address 192.168.200.101 and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # delete ntp-server 192.168.200.101
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

The following example deletes the NTP server with the IPv6 address 4001::6 and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # delete ntp-server 4001::6
UCS-A /system/services* # commit-buffer
UCS-A /system/services #
```

Setting the System Clock Manually

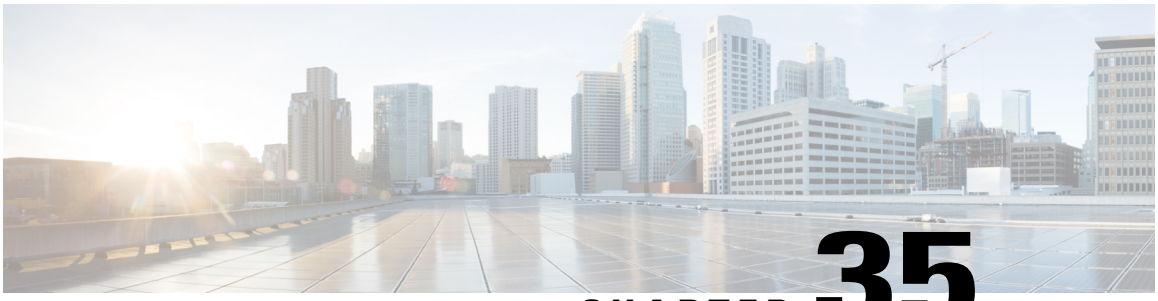
System clock modifications take effect immediately.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope services**	Enters system services mode.
Step 3	UCS-A /system/services # **set clock** *mon date year hour min sec*	Configures the system clock.

The following example configures the system clock and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope services
UCS-A /system/services # set clock apr 14 2010 15 27 00
UCS-A /system/services #
```



CHAPTER 35

Managing the Chassis

This chapter includes the following sections:

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

Perform the following procedure if you increase or decrease the number of links that connect the chassis to the fabric interconnect. Acknowledging the chassis ensures that Cisco UCS Manager is aware of the change in the number of links and that traffics flows along all available links.

After you enable or disable a port on a fabric interconnect, wait for at least 1 minute before you re-acknowledge the chassis. If you re-acknowledge the chassis too soon, the pinning of server traffic from the chassis might not get updated with the changes to the port that you enabled or disabled.

Procedure

	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.

The following example acknowledges chassis 2 and commits the transaction:

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

Decommissioning a Chassis

Procedure

	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.

The following example decommissions chassis 2 and commits the transaction:

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

```
Chassis:
    Chassis    Overall Status           Admin State
    ---------- ------------------------ -----------
             1 Operable                 Acknowledged
             2 Accessibility Problem    Decommission
UCS-A #
```

Removing a Chassis

Before You Begin

Physically remove the chassis before performing the following procedure.

Procedure

	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.

The following example removes chassis 2 and commits the transaction:

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

Recommissioning a Chassis

This procedure returns the chassis to the configuration and applies the chassis discovery policy to the chassis. After this procedure, you can access the chassis and any servers in it.

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

Procedure

	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, if you immediately run the **show chassis** command, you may not see any change in the Admin State of the chassis. It may take a while before the state of the chassis changes after it is recommissioned.

The following example recommissions a Cisco UCS 5108 chassis and commits the transaction:

```
UCS-A# show chassis

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

UCS-A# recommission chassis "Cisco Systems Inc" "N20-C6508" FOX1252GNNN
UCS-A* # commit-buffer
UCS-A #
```

Renumbering a Chassis

Note You cannot renumber a blade server through Cisco UCS Manager. The ID assigned to a blade server is determined by its physical slot in the chassis. To renumber a blade server, you must physically move the server to a different slot in the 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.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **show chassis inventory**	Displays information about your chassis.
Step 2	Verify that the chassis inventory does not include the following:	• The chassis you want to renumber • A chassis with the number you want to use

	Command or Action	Purpose
		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.

The following example decommissions two Cisco UCS chassis (chassis 8 and 9), switches their IDs, and commits the transaction:

```
UCS-A# show chassis inventory

    Chassis    PID       Vendor          Serial (SN) HW Revision
    ------- --------- ---------------   ----------- -------------
          1 N20-C6508 Cisco Systems Inc FOX1252GAAA 0
          2 N20-C6508 Cisco Systems Inc FOX1252GBBB 0
          3 N20-C6508 Cisco Systems Inc FOX1252GCCC 0
          4 N20-C6508 Cisco Systems Inc FOX1252GDDD 0
          5 N20-C6508 Cisco Systems Inc FOX1252GEEE 0
          6 N20-C6508 Cisco Systems Inc FOX1252GFFF 0
          7 N20-C6508 Cisco Systems Inc FOX1252GGGG 0
          8 N20-C6508 Cisco Systems Inc FOX1252GHHH 0
          9 N20-C6508 Cisco Systems Inc FOX1252GIII 0
         10 N20-C6508 Cisco Systems Inc FOX1252GJJJ 0
         11 N20-C6508 Cisco Systems Inc FOX1252GKKK 0
         12 N20-C6508 Cisco Systems Inc FOX1252GLLL 0
         13 N20-C6508 Cisco Systems Inc FOX1252GMMM 0
         14 N20-C6508 Cisco Systems Inc FOX1252GNNN 0

UCS-A# decommission chassis 8
UCS-A*# commit-buffer
UCS-A# decommission chassis 9
UCS-A*# commit-buffer
UCS-A# show chassis inventory

    Chassis    PID       Vendor          Serial (SN) HW Revision
    ------- --------- ---------------   ----------- -------------
          1 N20-C6508 Cisco Systems Inc FOX1252GAAA 0
          2 N20-C6508 Cisco Systems Inc FOX1252GBBB 0
          3 N20-C6508 Cisco Systems Inc FOX1252GCCC 0
          4 N20-C6508 Cisco Systems Inc FOX1252GDDD 0
          5 N20-C6508 Cisco Systems Inc FOX1252GEEE 0
          6 N20-C6508 Cisco Systems Inc FOX1252GFFF 0
          7 N20-C6508 Cisco Systems Inc FOX1252GGGG 0
         10 N20-C6508 Cisco Systems Inc FOX1252GJJJ 0
         11 N20-C6508 Cisco Systems Inc FOX1252GKKK 0
         12 N20-C6508 Cisco Systems Inc FOX1252GLLL 0
         13 N20-C6508 Cisco Systems Inc FOX1252GMMM 0
         14 N20-C6508 Cisco Systems Inc FOX1252GNNN 0

UCS-A# show chassis decommissioned

    Chassis    PID       Vendor          Serial (SN) HW Revision
    ------- --------- ---------------   ----------- -------------
```

```
          8 N20-C6508 Cisco Systems Inc FOX1252GHHH 0
          9 N20-C6508 Cisco Systems Inc FOX1252GIII 0

UCS-A# recommission chassis "Cisco Systems Inc" "N20-C6508" FOX1252GHHH 9
UCS-A* # commit-buffer
UCS-A# recommission chassis "Cisco Systems Inc" "N20-C6508" FOX1252GIII 8
UCS-A* # commit-buffer
UCS-A # show chassis inventory

    Chassis    PID       Vendor         Serial (SN) HW Revision
    ------- --------- ---------------   ----------- -------------
          1 N20-C6508 Cisco Systems Inc FOX1252GAAA 0
          2 N20-C6508 Cisco Systems Inc FOX1252GBBB 0
          3 N20-C6508 Cisco Systems Inc FOX1252GCCC 0
          4 N20-C6508 Cisco Systems Inc FOX1252GDDD 0
          5 N20-C6508 Cisco Systems Inc FOX1252GEEE 0
          6 N20-C6508 Cisco Systems Inc FOX1252GFFF 0
          7 N20-C6508 Cisco Systems Inc FOX1252GGGG 0
          8 N20-C6508 Cisco Systems Inc FOX1252GIII 0
          9 N20-C6508 Cisco Systems Inc FOX1252GHHH 0
         10 N20-C6508 Cisco Systems Inc FOX1252GJJJ 0
         11 N20-C6508 Cisco Systems Inc FOX1252GKKK 0
         12 N20-C6508 Cisco Systems Inc FOX1252GLLL 0
         13 N20-C6508 Cisco Systems Inc FOX1252GMMM 0
         14 N20-C6508 Cisco Systems Inc FOX1252GNNN 0
```

Toggling the Locator LED

Turning On the Locator LED for a Chassis

Procedure

	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.

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

Procedure

	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.

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 #
```

NVMe PCIe SSD Inventory

Cisco UCS Manager GUIdiscovers, identifies, and displays the inventory of Non-Volatile Memory Express (NVMe) Peripheral Component Interconnect Express (PCIe) SSD storage devices. You can view the health of the storage devices in the server. NVMe with PCIe SSD storage devices reduce latency, increased input/output operations per second (IOPS), and lower power consumption compared to SAS or SATA SSDs.

Viewing NVMe PCIe Local Disk Inventory Details

Procedure

	Command or Action	Purpose

Step 1	**Example:**
	`Local Disk 2:`
	`Product Name:`
	`PID:`
	`VID:`
	`Vendor: HGST`
	`Model: HUSPR3216ADP301`
	`Vendor Description:`
	`Serial: STM0001AE009`
	`HW Rev: 0`
	`Block Size: 512`
	`Blocks: 3125627568`
	`Operability: Operable`
	`Oper Qualifier Reason: N/A`
	`Presence: Equipped`
	`Size: 1526185`
	`Device Type: SSD`
	`Thermal: N/A`	

Viewing NVMe PCIe SSD RAID Controller Inventory Details

Procedure

	Command or Action	Purpose

Step 1	**Example:**
	`RAID Controller 7:`
	`Type: NVME`
	`Vendor: HGST`
	`Model: HUSPR3216ADP301`
	`Serial: STM0001AE009`
	`HW Revision: NVME`
	`PCI Addr: 131:00.0`
	`Raid Support: No`
	`OOB Interface Supported: No`	

CHAPTER 36

Managing Blade Servers

This chapter includes the following sections:

Blade Server Management

You can manage and monitor all blade servers in a Cisco UCS domain through Cisco UCS Manager. You can perform some blade server management tasks, such as changes to the power state, from the server and service profile.

The remaining management tasks can only be performed on the server.

The power supply units go into power save mode when a chassis has two blades or less. When a third blade is added to the chassis and is fully discovered, the power supply units return to regular mode.

If a blade server slot in a chassis is empty, Cisco UCS Manager provides information, errors, and faults for that slot. You can also re-acknowledge the slot to resolve server mismatch errors and to have Cisco UCS Manager rediscover the blade server in the slot.

Cisco UCS B460 M4 Blade Server Management

The Cisco UCS B460 M4 blade server consists of two full-width Cisco UCS B260 blade servers that are connected by a Cisco UCS scalability connector. Each individual blade server is called a node and can be either the master or slave node.

Because each Cisco UCS B460 M4 blade server has two different nodes, you should note the following:

- The master node is always the node in the highest numbered slots.
- Whenever the Cisco UCS B460 blade server is referred to in Cisco UCS Manager, the reference is to the master slot number.
- If you remove the Cisco UCS scalability connector from the Cisco UCS B460 M4 blade server, the **Physical Display** area in the Cisco UCS Manager GUI displays **Needs Resolution** on both master node slots and both slave node slots.
- The health LED displays both the individual health of the master and slave node, and the combined health of both nodes together. The combined health LED always displays the status of the node with the worst health. Any health LED alarms are shown individually.
- In the Cisco UCS Manager GUI, you can turn on and off the locator LEDs for either the master or the slave node. In the Cisco UCS Manager CLI, you can turn on and off the locator LEDs individually, or both locator LEDs at the same time.
- Power capping on the Cisco UCS B460 M4 blade server is applied at the server level. Each node is capped at one half of the total value.
- Updating firmware updates both the master and slave node at the same time. You cannot update the firmware on an individual node.
- Local disk configuration is supported only on the master node.
- The Cisco UCS B460 blade server does not distinguish between the SEL logs that are generated by either the master or the slave node. The logs are displayed on the same page and are differentiated by the slot number.
- On the Cisco UCS Manager GUI **Storage** tab, the **Local Disk Configuration Policy** and **Actual Disk Configurations** areas display only the data for the Cisco UCS B460 blade server master node. No fields are displayed for the slave node.

Upgrading to a Cisco UCS B460 M4 Blade Server

If you have a Cisco UCS B260 M4 blade server, you can purchase an upgrade kit to convert to a Cisco UCS B460 M4 blade server. For more information, see the appropriate *Cisco UCS Hardware Installation Guide*.

Before You Begin

You must have two Cisco UCS B260 M4 blade servers and a Cisco UCS scalability connector.

Procedure

Step 1 Verify that the existing Cisco UCS B260 M4 blade server is not associated with a service profile.

Step 2 Insert the second Cisco UCS B260 M4 blade server into the chassis either above or below the first blade server.

Note If the second blade server does not have a Cisco UCS scalability terminator, use the terminator from the first blade server.

Step 3 Decommission both Cisco UCS B260 M4 blade servers.

Step 4 Synchronize the firmware.
Use the **Firmware Auto Sync Server** policy in Cisco UCS Manager to automatically update the new server. For more information, see the appropriate *Cisco UCS B-Series Firmware Management Guide*.

Step 5 Replace the Cisco UCS scalability terminators with the Cisco UCS scalability connector.
The presence of the slots changes to mismatch, but discovery is not triggered.

Step 6 Reacknowledge the new Cisco UCS B460 M4 blade server.

Guidelines for Removing and Decommissioning Blade Servers

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

Decommissioning a Blade Server

If you want to temporarily decommission a physically present and connected blade server, you can temporarily remove it from the configuration. A portion of the server's information is retained by Cisco UCS Manager for future use, in case the blade server is recommissioned.

Removing a Blade Server

Removing is performed when you physically remove a blade server from the Cisco UCS Manager by disconnecting it from the chassis. You cannot remove a blade server from Cisco UCS Manager if it is physically present and connected to a chassis. After the physical removal of the blade server is completed, the configuration for that blade server can be removed in Cisco UCS Manager.

During removal, active links to the blade server are disabled, all entries from databases are removed, and the server is automatically removed from any server pools that it was assigned to during discovery.

Note Only servers added to a server pool automatically during discovery are removed automatically. Servers that were manually added to a server pool must be removed manually.

To add a removed blade server back to the configuration, it must be reconnected, then rediscovered. When a server is reintroduced to Cisco UCS Manager, it is treated as a new server and is subject to the deep discovery process. For this reason, it is possible for Cisco UCS Manager to assign the server a new ID that might be different from the ID that it held before.

Recommendations for Avoiding Unexpected Server Power Changes

If a server is not associated with a service profile, you can use any available means to change the server power state, including the physical Power or Reset buttons on the server.

If a server is associated with, or assigned to, a service profile, you should only use the following methods to change the server power state:

- In Cisco UCS Manager GUI, go to the **General** tab for the server or the service profile associated with the server and select **Boot Server** or **Shutdown Server** from the **Actions** area.
- In Cisco UCS Manager CLI, scope to the server or the service profile associated with the server and use the **power up** or **power down** commands.

Important Do *not* use any of the following options on an associated server that is currently powered off:

- **Reset** in the GUI
- **cycle cycle-immediate** or **reset hard-reset-immediate** in the CLI
- The physical Power or Reset buttons on the server

If you reset, cycle, or use the physical power buttons on a server that is currently powered off, the server's actual power state might become out of sync with the desired power state setting in the service profile. If the communication between the server and Cisco UCS Manager is disrupted or if the service profile configuration changes, Cisco UCS Manager might apply the desired power state from the service profile to the server, causing an unexpected power change.

Power synchronization issues can lead to an unexpected server restart, as shown below:

Desired Power State in Service Profile	Current Server Power State	Server Power State After Communication Is Disrupted
Up	Powered Off	Powered On

Desired Power State in Service Profile	Current Server Power State	Server Power State After Communication Is Disrupted
Down	Powered On	Powered On **Note** Running servers are not shut down regardless of the desired power state in the service profile.

Booting a Blade Server

Before You Begin

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

Procedure

	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 blade server associated with the service profile.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example boots the blade 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 Blade 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.

Note When a blade server that is associated with a service profile is shut down, the VIF down alert F0283 and F0479 are automatically suppressed.

Before You Begin

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

Procedure

	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 blade server associated with the service profile.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example shuts down the blade 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 Blade Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *server-num*	Enters chassis server mode for the specified blade server.
Step 2	UCS-A /chassis/server # **cycle** {**cycle-immediate** | **cycle-wait**}	Power cycles the blade server. Use the **cycle-immediate** keyword to immediately begin power cycling the blade server; use the **cycle-wait** keyword to schedule the power cycle to begin after all pending management operations have completed.

	Command or Action	Purpose
Step 3	UCS-A# **commit-buffer**	Commits the transaction to the system configuration.

The following example immediately power cycles blade server 4 in chassis 2 and commits the transaction:

```
UCS-A#  scope server 2/4
UCS-A /chassis/server # cycle cycle-immediate
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Performing a Hard Reset on a Blade 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.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *server-num*	Enters chassis server mode for the specified server.
Step 2	UCS-A /chassis/server # **reset** {**hard-reset-immediate** | **hard-reset-wait**}	Performs a hard reset of the blade 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 /server # **commit-buffer**	Commits the transaction to the system configuration.

The following example performs an immediate hard reset of blade server 4 in chassis 2 and commits the transaction:

```
UCS-A#  scope server 2/4
UCS-A /chassis/server # reset hard-reset-immediate
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Resetting a Blade Server to Factory Default Settings

You can now reset a blade server to its factory settings. By default, the factory reset operation does not affect storage drives and flexflash drives. This is to prevent any loss of data. However, you can choose to reset these devices to a known state as well.

Important Resetting storage devices will result in loss of data.

Perform the following procedure to reset the server to factory default settings.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** [*chassis-num*/*server-num* | *dynamic-uuid*]	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # **reset factory-default** [**delete-flexflash-storage** | **delete-storage** [**create-initial-storage-volumes**]]	Resets server settings to factory default using the following command options: • **factory-default—**Resets the server to factory defaults without deleting storage • **delete-flexflash-storage—**Resets the server to factory defaults and deletes flexflash storage • **delete-storage—**Resets the server to factory defaults and deletes all storage • **create-initial-storage-volumes—**Resets the server to factory defaults, deletes all storage, sets all disks to their initial state **Important** Do not use the **create-initial-storage-volumes** command option if you want to use storage profiles. Creating initial volumes when you are using storage profiles may result in configuration errors.
Step 3	UCS-A /chassis/server* # **commit-buffer**	Commits any pending transactions.

The following example resets the server settings to factory default without deleting storage, and commits the transaction:

```
UCS-A# scope server 2/4
UCS-A /chassis/server # reset factory-default
UCS-A /chassis/server* # commit-buffer
```

The following example resets the server settings to factory default, deletes flexflash storage, and commits the transaction:

```
UCS-A# scope server 2/4
UCS-A /chassis/server # reset factory-default delete-flexflash-storage

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

The following example resets the server settings to factory default, deletes all storage, and commits the transaction:

```
UCS-A# scope server 2/4
UCS-A /chassis/server # reset factory-default delete-storage
UCS-A /chassis/server* # commit-buffer
```

The following example resets the server settings to factory default, deletes all storage, sets all disks to their initial state, and commits the transaction:

```
UCS-A# scope server 2/4
UCS-A /chassis/server # reset factory-default delete-storage create-initial-storage-volumes
UCS-A /chassis/server* # commit-buffer
```

Acknowledging a Blade 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.

Procedure

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

The following example acknowledges server 4 in chassis 2 and commits the transaction:

```
UCS-A#  acknowledge server 2/4
UCS-A* # commit-buffer
UCS-A #
```

Removing a Blade Server from a Chassis

Before You Begin

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

Procedure

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

The following example removes blade server 4 in chassis 2 and commits the transaction:

```
UCS-A#  remove server 2/4
UCS-A* # commit-buffer
UCS-A #
```

What to Do Next

If you physically re-install the blade server, you must re-acknowledge the slot for the Cisco UCS Manager to rediscover the server.

For more information, see Acknowledging a Blade Server, on page 713.

Decommissioning a Blade Server

Procedure

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

The following example decommissions blade server 4 in chassis 2 and commits the transaction:

```
UCS-A# decommission server 2/4
UCS-A* # commit-buffer
UCS-A #
```

Turning On the Locator LED for a Blade Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *server-num*	Enters chassis server mode for the specified chassis.
Step 2	UCS-A /chassis/server # **enable locator-led** [**multi-master** | **multi-slave**]	Turns on the blade server locator LED. For the Cisco UCS B460 M4 blade server, you can add the following keywords: • **multi-master—**Turns on the LED for the master node only. • **multi-slave—**Turns on the LED for the slave node only.
Step 3	UCS-A /chassis/server # **commit-buffer**	Commits the transaction to the system configuration.

The following example turns on the locator LED for blade server 4 in chassis 2 and commits the transaction:

```
UCS-A# scope server 2/4
UCS-A /chassis/server # enable  locator-led
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

The following example turns on the locator LED for blade server 7 in chassis 2 and commits the transaction:

```
UCS-A# scope chassis 2/7
UCS-A /chassis/server # enable locator-led multi-master
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Turning Off the Locator LED for a Blade Server

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-num* / *server-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis/server # **disable locator-led** [**multi-master** | **multi-slave**]	Turns off the blade server locator LED. For the Cisco UCS B460 M4 blade server, you can add the following keywords: • **multi-master—**Turns off the LED for the master node only.

	Command or Action	Purpose
		• **multi-slave—**Turns off the LED for the slave node only.
Step 3	UCS-A /chassis/server # **commit-buffer**	Commits the transaction to the system configuration.

The following example turns off the locator LED for blade server 4 in chassis 2 and commits the transaction:

```
UCS-A# scope chassis 2/4
UCS-A /chassis/server # disable locator-led
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

The following example turns off the locator LED for blade server 7 in chassis 2 and commits the transaction:

```
UCS-A# scope chassis 2/7
UCS-A /chassis/server # disable locator-led multi-master
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Resetting the CMOS for a Blade 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.

Procedure

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

The following example resets the CMOS for blade server 4 in chassis 2 and commits the transaction:

```
UCS-A# scope server 2/4
UCS-A /chassis/server # reset-cmos
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Resetting the CIMC for a Blade 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 with the running version of the firmware for that 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.

Procedure

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

The following example resets the CIMC for blade server 4 in chassis 2 and commits the transaction:

```
UCS-A# scope server 2/4
UCS-A /chassis/server # scope CIMC
UCS-A /chassis/server/cimc # reset
UCS-A /chassis/server/cimc* # commit-buffer
UCS-A /chassis/server/cimc #
```

Clearing TPM for a Blade Server

You can clear TPM only on Cisco UCS M4 blade and rack-mount 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.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** [*chassis-num*/*server-num* | *dynamic-uuid*]	Enters server mode for the specified server.
Step 2	UCS-A# /chassis/server # **scope tpm** *tpm-ID*	Enters org TPM mode for the specified TPM.
Step 3	UCS-A# /chassis/server/tpm # **set adminaction clear-config**	Specifies that the TPM is to be cleared.

	Command or Action	Purpose
Step 4	UCS-A# /chassis/server/tpm # **commit-buffer**	Commits the transaction to the system configuration.

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

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

Recovering the Corrupt BIOS on a Blade Server

On rare occasions, an issue with a blade server may require you to recover the corrupted BIOS. This procedure is not part of the normal maintenance of a server. After you recover the BIOS, the blade server boots with the running version of the firmware for that server.

Before You Begin

Important Remove all attached or mapped USB storage from a server before you attempt to recover the corrupt BIOS on that server. If an external USB drive is attached or mapped from vMedia to the server, BIOS recovery fails.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-id* / *server-id*	Enters chassis server mode for the specified blade server in the specified chassis.
Step 2	UCS-A /chassis/server # **recover-bios** *version*	Loads and activates the specified BIOS version.
Step 3	UCS-A /chassis/server # **commit-buffer**	Commits the transaction.

The following example shows how to recover the BIOS:

```
UCS-A# scope server 1/7
UCS-A /chassis/server # recover-bios S5500.0044.0.3.1.010620101125
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Issuing an NMI from a Blade 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.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** [*chassis-num*/*server-num* | *dynamic-uuid*]	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # **diagnostic-interrupt**	
Step 3	UCS-A /chassis/server* # **commit-buffer**	Commits any pending transactions.

The following example sends an NMI from server 4 in chassis 2 and commits the transaction:

```
UCS-A# scope server 2/4
UCS-A /chassis/server # diagnostic-interrupt
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Health LED Alarms

The blade health LED is located on the front of each Cisco UCS B-Series blade server. Cisco UCS Manager allows you to view the sensor faults that cause the blade 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.
Sensor Name column	The name of the sensor that triggered the alarm.

Viewing Health LED Status

Procedure

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

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

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

CHAPTER 37

Managing Rack-Mount Servers

This chapter includes the following sections:

Rack-Mount Server Management

You can manage and monitor all rack-mount servers that are integrated with a Cisco UCS domain through Cisco UCS Manager. All management and monitoring features are supported for rack-mount servers except power capping. Some rack-mount server management tasks, such as changes to the power state, can be performed from both the server and service profile. The remaining management tasks can only be performed on the server.

Cisco UCS Manager provides information, errors, and faults for each rack-mount server that it has discovered.

Tip For information on how to integrate a supported Cisco UCS rack-mount server with Cisco UCS Manager, see the Cisco UCS C-series server integration guide or Cisco UCS S-series server integration guide for your Cisco UCS Manager release.

Guidelines for Removing and Decommissioning Rack-Mount Servers

Consider the following guidelines when deciding whether to remove or decommission a rack-mount server using Cisco UCS Manager:

Decommissioning a Rack-Mount server

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

Removing a Rack-Mount server

Removing is performed when you physically remove the server from the system by disconnecting the rack-mount server from the fabric extender. You cannot remove a rack-mount server from Cisco UCS Manager if it is physically present and connected to the fabric extender. Once the rack-mount server is disconnected, the configuration for that rack-mount server can be removed in Cisco UCS Manager.

During removal, management interfaces are disconnected, all entries from databases are removed, and the server is automatically removed from any server pools that it was assigned to during discovery.

Note Only those servers added to a server pool automatically during discovery will be removed automatically. Servers that have been manually added to a server pool have to be removed manually.

If you need to add a removed rack-mount server back to the configuration, it must be reconnected and then rediscovered. When a server is reintroduced to Cisco UCS Manager it is treated like a new server and is subject to the deep discovery process. For this reason, it's possible that Cisco UCS Manager will assign the server a new ID that may be different from the ID that it held before.

Recommendations for Avoiding Unexpected Server Power Changes

If a server is not associated with a service profile, you can use any available means to change the server power state, including the physical Power or Reset buttons on the server.

If a server is associated with, or assigned to, a service profile, you should only use the following methods to change the server power state:

- In Cisco UCS Manager GUI, go to the **General** tab for the server or the service profile associated with the server and select **Boot Server** or **Shutdown Server** from the **Actions** area.
- In Cisco UCS Manager CLI, scope to the server or the service profile associated with the server and use the **power up** or **power down** commands.

Important Do *not* use any of the following options on an associated server that is currently powered off:

- **Reset** in the GUI
- **cycle cycle-immediate** or **reset hard-reset-immediate** in the CLI
- The physical Power or Reset buttons on the server

If you reset, cycle, or use the physical power buttons on a server that is currently powered off, the server's actual power state might become out of sync with the desired power state setting in the service profile. If the communication between the server and Cisco UCS Manager is disrupted or if the service profile configuration changes, Cisco UCS Manager might apply the desired power state from the service profile to the server, causing an unexpected power change.

Power synchronization issues can lead to an unexpected server restart, as shown below:

Desired Power State in Service Profile	Current Server Power State	Server Power State After Communication Is Disrupted
Up	Powered Off	Powered On
Down	Powered On	Powered On **Note** Running servers are not shut down regardless of the desired power state in the service profile.

Booting a Rack-Mount Server

Before You Begin

Associate a service profile with a rack-mount server.

Procedure

	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 rack-mount server associated with the service profile.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example boots the rack-mount 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 Rack-Mount 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 rack-mount server.

Procedure

	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 rack-mount server associated with the service profile.
Step 4	UCS-A /org/service-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example shuts down the rack-mount 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 Rack-Mount Server

Procedure

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

The following example immediately power cycles rack-mount server 2 and commits the transaction:

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

Performing a Hard Reset on a Rack-Mount 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.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *server-num*	Enters server mode for the specified rack-mount server.
Step 2	UCS-A /server # **reset** {**hard-reset-immediate** | **hard-reset-wait**}	Performs a hard reset of the rack-mount server. Use the **hard-reset-immediate** keyword to immediately begin hard resetting the rack-mount 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 /server # **commit-buffer**	Commits the transaction to the system configuration.

The following example performs an immediate hard reset of rack-mount server 2 and commits the transaction:

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

Acknowledging a Rack-Mount 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.

Procedure

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

The following example acknowledges rack-mount server 2 and commits the transaction:

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

Decommissioning a Rack-Mount Server

Procedure

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

The following example decommissions rack-mount server 2 and commits the transaction:

```
UCS-A# decommission server 2
UCS-A* # commit-buffer
UCS-A #
```

Renumbering a Rack-Mount Server

Before You Begin

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

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **show server inventory**	Displays information about your servers.
Step 2	Verify that the server inventory does not include the following:	• The rack-mount server you want to renumber • A rack-mount server with the number you want to use If either of these rack-mount servers are listed in the server inventory, decommission those servers. You must wait until the decommission FSM is complete and the rack-mount servers are not listed in the server inventory before continuing. This might take several minutes. To see which servers have been decommissioned, issue the **show server decommissioned** command.
Step 3	UCS-A# **recommission server** *vendor-name model-name serial-numnew-id*	Recommissions and renumbers the specified rack-mount server.
Step 4	UCS-A# **commit-buffer**	Commits the transaction to the system configuration.

The following example decommissions a rack-mount server with ID 2, changes the ID to 3, recommissions that server, and commits the transaction:

```
UCS-A# show server inventory

Server  Equipped PID Equipped VID Equipped Serial (SN) Slot Status      Ackd Memory (MB)
Ackd Cores
------- ------------ ------------ -------------------- ---------------- ----------------
----------
1/1     UCSB-B200-M3 V01          FCH1532718P          Equipped         131072
16
1/2     UCSB-B200-M3 V01          FCH153271DF          Equipped         131072
16
1/3     UCSB-B200-M3 V01          FCH153271DL          Equipped         114688
16
1/4     UCSB-B200-M3 V01                               Empty
1/5                                                    Empty
1/6                                                    Empty
1/7     N20-B6730-1  V01          JAF1432CFDH          Equipped         65536
16
1/8                                                    Empty
1       R200-1120402W V01          QCI1414A02J          N/A              49152
12
2       R210-2121605W V01          QCI1442AHFX          N/A              24576            8
4       UCSC-BSE-SFF-C200 V01      QCI1514A0J7          N/A              8192             8

UCS-A# decommission server 2
UCS-A*# commit-buffer
UCS-A# show server decommissioned

Vendor            Model         Serial (SN) Server
----------------- ----------    ----------- ------
Cisco Systems Inc R210-2121605W QCI1442AHFX 2

UCS-A# recommission chassis "Cisco Systems Inc" "R210-2121605W" QCI1442AHFX 3
UCS-A* # commit-buffer
UCS-A # show server inventory

Server  Equipped PID Equipped VID Equipped Serial (SN) Slot Status      Ackd Memory (MB)
Ackd Cores
------- ------------ ------------ -------------------- ---------------- ----------------
----------
1/1     UCSB-B200-M3 V01          FCH1532718P          Equipped         131072
16
1/2     UCSB-B200-M3 V01          FCH153271DF          Equipped         131072
16
1/3     UCSB-B200-M3 V01          FCH153271DL          Equipped         114688
16
1/4     UCSB-B200-M3 V01                               Empty
1/5                                                    Empty
1/6                                                    Empty
1/7     N20-B6730-1  V01          JAF1432CFDH          Equipped         65536
16
1/8                                                    Empty
1       R200-1120402W V01          QCI1414A02J          N/A              49152
12
3       R210-2121605W V01          QCI1442AHFX          N/A              24576            8
4       UCSC-BSE-SFF-C200 V01      QCI1514A0J7          N/A              8192             8
```

Removing a Rack-Mount Server

Before You Begin

Physically disconnect the CIMC LOM cables that connect the rack-mount server to the fabric extender before performing the following procedure. For high availability setups, remove both cables.

Procedure

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

The following example removes rack-mount server 4 and commits the transaction:

```
UCS-A#  remove server 4
UCS-A* # commit-buffer
UCS-A #
```

What to Do Next

If you physically reconnect the rack-mount server, you must re-acknowledge it for the Cisco UCS Manager to rediscover the server.

For more information, see Acknowledging a Rack-Mount Server, on page 726.

Turning On the Locator LED for a Rack-Mount Server

Procedure

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

The following example turns on the locator LED for rack-mount server 2 and commits the transaction:

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

Turning Off the Locator LED for a Rack-Mount Server

Procedure

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

The following example turns off the locator LED for rack-mount server 2 and commits the transaction:

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

Resetting the CMOS for a Rack-Mount 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.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *server-num*	Enters server mode for the rack-mount server.
Step 2	UCS-A /server # **reset-cmos**	Resets the CMOS for the rack-mount server.
Step 3	UCS-A /server # **commit-buffer**	Commits the transaction to the system configuration.

The following example resets the CMOS for rack-mount server 2 and commits the transaction:

```
UCS-A# scope server 2
UCS-A /server # reset-cmos
UCS-A /server* # commit-buffer
UCS-A /server #
```

Resetting the CIMC for a Rack-Mount 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 with the running version of the firmware for that server.

Procedure

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

The following example resets the CIMC for rack-mount server 2 and commits the transaction:

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

Clearing TPM for a Rack-Mount Server

You can clear TPM only on Cisco UCS M4 blade and rack-mount 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.
- If the BIOS policy is set to default, you must reacknowledge the server before clearing TPM.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *server-num*	Enters server mode for the rack-mount server.

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

The following example shows how to clear TPM for a rack-mount server:

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

Recovering the Corrupt BIOS on a Rack-Mount Server

On rare occasions, an issue with a rack-mount server may require you to recover the corrupted BIOS. This procedure is not part of the normal maintenance of a rack-mount server. After you recover the BIOS, the rack-mount server boots with the running version of the firmware for that server.

Before You Begin

Important Remove all attached or mapped USB storage from a server before you attempt to recover the corrupt BIOS on that server. If an external USB drive is attached or mapped from vMedia to the server, BIOS recovery fails.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** *server-id*	Enters server mode for the specified rack-mount server.
Step 2	UCS-A /server # **recover-bios** *version*	Loads and activates the specified BIOS version.
Step 3	UCS-A /server # **commit-buffer**	Commits the transaction.

The following example shows how to recover the BIOS:

```
UCS-A# scope server 1
UCS-A /server # recover-bios S5500.0044.0.3.1.010620101125
```

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

Showing the Status for a Rack-Mount Server

Procedure

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

The following example shows the status for all servers in the Cisco UCS domain. The servers numbered 1 and 2 do not have a slot listed in the table because they are rack-mount servers.

```
Server Slot  Status    Availability  Overall Status  Discovery
------------------------------------------------------------------
1/1          Equipped  Unavailable   Ok              Complete
1/2          Equipped  Unavailable   Ok              Complete
1/3          Equipped  Unavailable   Ok              Complete
1/4          Empty     Unavailable   Ok              Complete
1/5          Equipped  Unavailable   Ok              Complete
1/6          Equipped  Unavailable   Ok              Complete
1/7          Empty     Unavailable   Ok              Complete
1/8          Empty     Unavailable   Ok              Complete
1            Equipped  Unavailable   Ok              Complete
2            Equipped  Unavailable   Ok              Complete
```

Issuing an NMI from a Rack-Mount Server

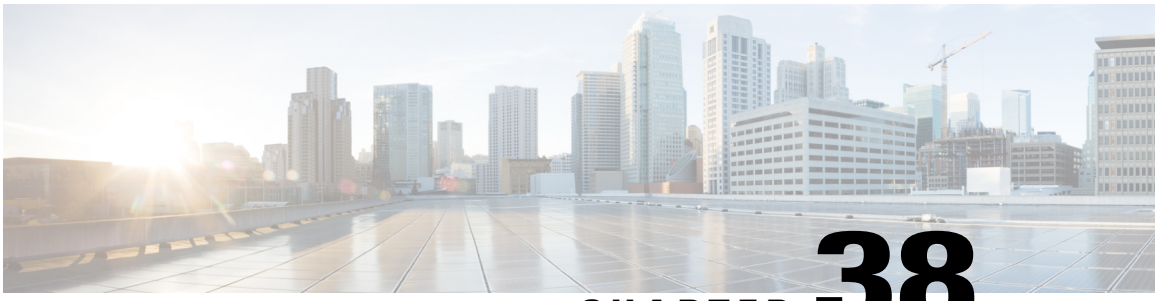
Perform the following procedure if the system remains unresponsive and you need Cisco UCS Manager to issue a Non-Maskable Interrupt (NMI) to the BIOS or operating system from the CIMC. This action creates a core dump or stack trace, depending on the operating system installed on the server.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope server** [*chassis-num*/*server-num* | *dynamic-uuid*]	Enters server mode for the specified server.
Step 2	UCS-A /chassis/server # **diagnostic-interrupt**	
Step 3	UCS-A /chassis/server* # **commit-buffer**	Commits any pending transactions.

The following example sends an NMI from server 4 in chassis 2 and commits the transaction:

```
UCS-A# scope server 2/4
UCS-A /chassis/server # diagnostic-interrupt
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```



CHAPTER 38

CIMC Session Management

This chapter includes the following sections:

CIMC Session Management

You can view and close any KVM, vMedia, and SOL sessions in Cisco UCS Manager. If you have administrator privileges, you can discontinue the KVM, vMedia, and SoL sessions of any user. . Cisco Integrated Management Controller (CIMC) provides session information to Cisco UCS Manager. When Cisco UCS Manager gets an event from CIMC, it updates its session table and displays the information to all users.

The session information consists of the following information:

- Name—The name of the user who launched the session.
- Session ID—The ID associated with the session. The format of the session ID for blades is [unique identifier] _ [chassis id] _ [Blade id]. The format of the session ID for racks is [unique identifier] _ 0 _ [Rack id].
- Type of session—KVM, vMedia, or SoL.
- Privilege level of the user—Read-Write, Read Only, or Granted.
- Administrative state—Active or Inactive. The value is active if the session is active. The value is inactive if the session terminate command has been issued but the session has not been terminated. This situation occurs when FSM of the server is in progress with another operation or when the connectivity to CIMC is lost.
- Source Address—The IP address of the computer from which the session was opened.
- Service Profile—The service profile associated with the session. The service profile attribute value for a CIMC session is displayed only if the session is opened on an IP address that is provided from the service profile.
- Server—The name of the server associated with the session.
- Login time—The date and time the session started.
- Last Update Time—The last time the session information was updated by CIMC.

A new session is generally added when a user connects to KVM, vMedia, or SOL. A Pnuos vMedia session will be displayed in the session table during the server discovery with the user name __vmediausr__.

The CIMC session data is available under the **CIMC Sessions** tab in Cisco UCS Manager GUI. Any CIMC session terminated by the user is audit logged with proper details.

Note To perform the GUI and CLI tasks that are described in this guide, a CIMC image version of 2.1(2a) or above is required for the session management support for the blade servers. The latest CIMC image version of 1.5(1l) and above is required for the rack-servers.

Viewing the CIMC Sessions Opened by the Local Users

Follow this task to view all the CIMC sessions opened by the local users or the CIMC sessions opened by a specific local user.

Note Viewing CIMC sessions of a specific server or a service-profile option is not present in CLI. It is available in GUI.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope security**	Enters security configuration mode.
Step 2	UCS-A /security # **show cimc-sessions local**	Displays all CIMC sessions opened by the local users.
Step 3	UCS-A /security # **show cimc-sessions local** *user-name*	Displays all CIMC sessions opened by a specific local user.

The following examples show how to view:

- All CIMC sessions opened by local users
- CIMC session opened by a specific local user
- Details of the CIMC session opened by a specific local user.

```
All sessions opened by local users:
UCS-A # scope security
UCS-A /security # show cimc-sessions local

Session ID   Type     User      Source Addr     Admin State
----------   ------ --------- --------------- ------------
42_1_1       Kvm      admin     10.106.22.117     Active
4_1_5        Kvm      admin     10.106.22.117     Active
5_1_5        Vmedia   admin     10.106.22.117     Active

Session opened by a specific local user:
UCS-A /security # show cimc-sessions local admin
```

```
Session ID   Type     User      Source Addr     Admin State
----------   ------ --------- --------------- ------------
42_1_1       Kvm      admin     10.106.22.117     Active

Details of session opened by a specific local user:
UCS-A /security # show cimc-sessions local admin detail
Session ID 42_1_1
   Type: Kvm
   User: admin
   Source Addr: 10.106.22.117
   Login Time: 2013-06-28T06:09:53.000
   Last Updated Time: 2013-06-28T06:21:52.000
   Admin State: Active
   Priv: RW
   Server: sys/chassis-1/blade-1
   Service Profile:
```

Viewing the CIMC Sessions Opened by the Remote Users

Follow this task to view all the CIMC sessions opened by the remote users or the CIMC sessions opened by a specific remote user.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope security**	Enters security configuration mode.
Step 2	UCS-A /security # **show cimc-sessions remote**	Displays all CIMC sessions opened by the remote users.
Step 3	UCS-A /security # **show cimc-sessions remote** *user-name*	Displays all CIMC sessions opened by a specific remote user.

The following examples show how to view:

- All CIMC sessions opened by remote users
- CIMC session opened by a specific remote user
- Details of the CIMC session opened by a specific remote user.

```
All sessions opened by remote users:
UCS-A # scope security
UCS-A /security # show cimc-sessions remote

Session ID   Type     User            Source Addr      Admin State
----------   ------ ---------        ---------------   ------------
43_1_1       Kvm     administrator     10.106.22.117     Active
6_1_5        Kvm     test-remote       10.106.22.117     Active
7_1_5        Vmedia  test-remote       10.106.22.117     Active

Session opened by a specific remote user:
UCS-A /security # show cimc-sessions remote administrator

Session ID   Type     User            Source Addr      Admin State
----------   ------ ---------        ---------------   ------------
43_1_1       Kvm     administrator     10.106.22.117     Active
```

```
Details of session opened by a specific remote user:
UCS-A /security # show cimc-sessions remote administrator detail
Session ID 43_1_1
   Type: Kvm
   User: administrator
   Source Addr: 10.106.22.117
   Login Time: 2013-06-28T06:09:53.000
   Last Updated Time: 2013-06-28T06:21:52.000
   Admin State: Active
   Priv: RW
   Server: sys/chassis-1/blade-1
   Service Profile:
```

Viewing the CIMC Sessions Opened by an IPMI User

To view the CIMC sessions opened by an IPMI user, complete the following steps:

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope org** *org-name*	Enters the root organization mode.
Step 2	UCS-A /org # **scope ipmi-access-profile** *profile-name*	Enters the IPMI access profile name.
Step 3	UCS-A /org/ipmi-access-profile # **scope ipmi-user** *user-name*	Enters an IPMI user name.
Step 4	UCS-A /org/ipmi-access-profile/ipmi-user # **show cimc-sessions**	Displays all CIMC sessions opened by the specified IPMI User.

The following example shows how to view all the CIMC sessions opened by an IPMI user:

```
UCS-A # scope org Finance
UCS-A /org* # scope ipmi-access-profile ReadOnly
UCS-A /org/ipmi-access-profile* # scope ipmi-user alice
UCS-A /org/ipmi-access-profile/ipmi-user # show cimc-sessions

Session ID    Type    User        Source Addr       Admin State
----------   ------  ---------   ---------------   --------------
45_1_1        sol     alice       10.106.22.117     Active
```

Clearing the CIMC Sessions of a Server

This task shows how to clear all CIMC sessions opened on a server. You can also clear the CIMC sessions on a server based on the session type and the user name.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope security**	Enters security configuration mode.
Step 2	UCS-A /security # **terminate cimc-sessions server** *chassis-id/blade-id*	Clears the CIMC sessions on a specific blade server of a chassis.
Step 3	UCS-A /security # **terminate cimc-sessions server** *Rack-server-id*	Clears the CIMC sessions on a specific rack server.
Step 4	UCS-A /security # **terminate cimc-sessions server** *server-id* **type** *session-type*	Clears the CIMC sessions of a specific type on a server.
Step 5	UCS-A /security # **terminate cimc-sessions server** *server-id* **user-name** *user-name*	Clears the CIMC sessions of a specific user on a server.

The first example shows how to clear all CIMC sessions on a server. The second example shows how to clear the CIMC sessions of a specific type on a server. The third example shows how to clear the CIMC sessions of a specific user on a server:

```
UCS-A /security # scope security
UCS-A /security # terminate cimc-sessions server 2/1
This will close KVM sessions. Are you sure? (yes/no):yes
UCS-A /security

UCS-A # scope security
UCS-A /security # terminate cimc-sessions server 2/1 type kvm
This will close KVM sessions. Are you sure? (yes/no):yes

UCS-A # scope security
UCS-A /security # terminate cimc-sessions server 2/1 user-name test-user
This will close KVM sessions. Are you sure? (yes/no):yes
```

Clearing All CIMC Sessions Opened by a Local User

This task shows how to clear the sessions opened by a local user.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope security**	Enters security configuration mode.
Step 2	UCS-A /security # **terminate cimc-sessions local-user** *user-name*	Clears all CIMC sessions opened by a local user.
Step 3	UCS-A /security # **terminate cimc-sessions local-user** *user-name* **type** {**kvm** | **vmedia sol** | **all**}	Clears all CIMC sessions of specific session type opened by a local user.

The following example shows how to clear the CIMC sessions opened by a local user:

```
UCS-A /security# scope security
UCS-A /security# terminate cimc-sessions local-user testuser
This will close cimc sessions. Are you sure? (yes/no):yes
UCS-A /security#
```

Clearing All CIMC Sessions Opened by a Remote User

This task shows how to clear CIMC sessions opened by a remote user.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope security**	Enters security configuration mode.
Step 2	UCS-A /security # **terminate cimc-sessions remote-user** *user-name*	Clears all CIMC sessions opened by a remote user.
Step 3	UCS-A /security # **terminate cimc-sessions remote-user** *user-name* **type** {**kvm** | **vmedia sol** | **all**}	Clears all CIMC sessions of specific session type opened by a remote user.

The following example shows how to clear all CIMC sessions opened by a remote user:

```
UCS-A /security# scope security
UCS-A /security# terminate cimc-sessions remote-user testuser
This will close cimc sessions. Are you sure? (yes/no):yes
UCS-A /security#
```

Clearing a Specific CIMC Session Opened by a Local User

This task shows how to clear a specific CIMC session opened by a local user.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope security**	Enters security configuration mode.
Step 2	UCS-A /security # **scope local-user** *user-name*	Enters local user mode.
Step 3	UCS-A /security/local user # **terminate cimc-session** *session-id*	Clears the chosen CIMC session.
Step 4	UCS-A /security/local user* # **commit-buffer**	Commits the transaction.

The following example shows how to clear a specific CIMC session opened by a local user and commits the transaction:

```
UCS-A /security# scope security
UCS-A /security# scope local-user admin
UCS-A /security/local user # terminate cimc-session 6_1_2
UCS-A /security/local user*# commit-buffer
UCS-A /security/local user#
```

Clearing a Specific CIMC Session Opened by a Remote User

This task shows how to clear a specific CIMC session opened by a remote user.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope security**	Enters security configuration mode.
Step 2	UCS-A /security # **scope remote -user** *user-name*	Enters remote user mode.
Step 3	UCS-A /security/remote user # **terminate cimc-session** *session-id*	Clears the chosen CIMC session.
Step 4	UCS-A /security/remote user* # **commit-buffer**	Commits the transaction.

The following example shows how to clear a specific CIMC session opened by a remote user and commits the transaction:

```
UCS-A /security# scope security
UCS-A /security# scope remote-user admin
UCS-A /security/remote user # terminate cimc-session 6_1_3
UCS-A /security/remote user*# commit-buffer
UCS-A /security/remote user#
```

Clearing a CIMC Session Opened by an IPMI User

To clear a CIMC session opened by an IPMI user, complete the following steps:

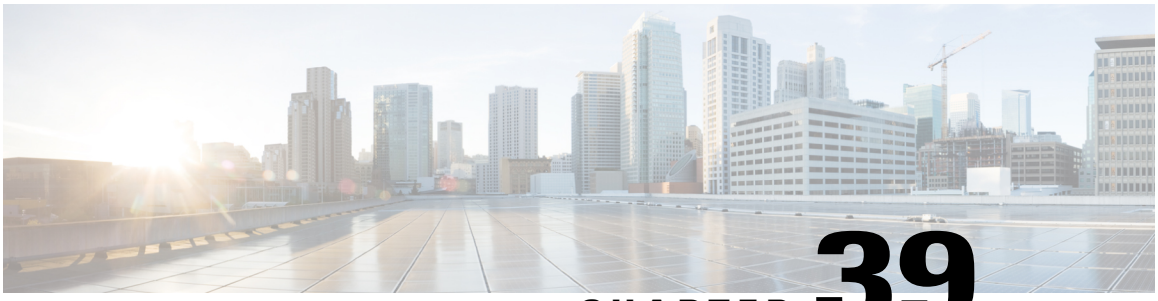
Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope org** *org-name*	Enters the root organization mode.
Step 2	UCS-A /org # **scope ipmi-access-profile** *profile-name*	Enters the IPMI access profile name.
Step 3	UCS-A /org/ipmi-access-profile # **scope ipmi-user** *user-name*	Enters the IPMI user.

	Command or Action	Purpose
Step 4	UCS-A /org/ipmi-access-profile/ipmi-user # **terminate cimc-sessions** *session-id*	Terminates a specific CIMC session opened by an IPMI user.
Step 5	UCS-A /org/ipmi-access-profile/ipmi-user * **commit-buffer**	Commits the changes.

The following example displays how to clear a specific CIMC session opened by an IPMI user and commits the changes:

```
UCS-A # scope org Finance
UCS-A /org* # scope ipmi-access-profile ReadOnly
UCS-A /org/ipmi-access-profile* # scope ipmi-user alice
UCS-A /org/ipmi-access-profile/ipmi-user # terminate cimc-sessions 5_1_2
UCS-A /org/ipmi-access-profile/ipmi-user* # commit-buffer
```



CHAPTER 39

Managing the I/O Modules

This chapter includes the following sections:

I/O Module Management in Cisco UCS Manager GUI

You can manage and monitor all I/O modules in a Cisco UCS domain through Cisco UCS Manager GUI.

Acknowledging an IO Module

Cisco UCS Manager Release 2.2(4) introduces the ability to acknowledge a specific IO module in a chassis.

Note This operation rebuilds the network connectivity between the IO module and the Fabrics to which it is connected.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **acknowledge iom** {**1** | **2**}	Acknowledges the specified IOM in the chassis.
Step 3	UCS-A /chassis* # **commit-buffer**	Commits the transaction to the system configuration.

The following example acknowledges IO Module 1 and commits the transaction:

```
UCS-A# scope chassis 1
UCS-A /chassis # acknowledge iom 1
UCS-A /chassis* # commit-buffer
UCS-A /chassis #
```

Resetting the I/O Module

Procedure

	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 iom** {**a b**}	Enters chassis IOM mode for the specified IOM.
Step 3	UCS-A /chassis/iom # **reset**	Resets the IOM.
Step 4	UCS-A /chassis/iom # **commit-buffer**	Commits the transaction to the system configuration.

The following example resets the IOM on fabric A and commits the transaction:

```
UCS-A# scope chassis 1
UCS-A /chassis # scope iom a
UCS-A /chassis/iom # reset
UCS-A /chassis/iom* # commit-buffer
UCS-A /chassis/iom #
```

Resetting an I/O Module from a Peer I/O Module

Sometimes, I/O module upgrades can result in failures or I/O modules can become unreachable from Cisco UCS Manager due to memory leaks. You can now reboot an I/O module that is unreachable through its peer I/O module.

Resetting the I/O module restores the I/O module to factory default settings, deletes all cache files and temporary files, but retains the size-limited OBFL file.

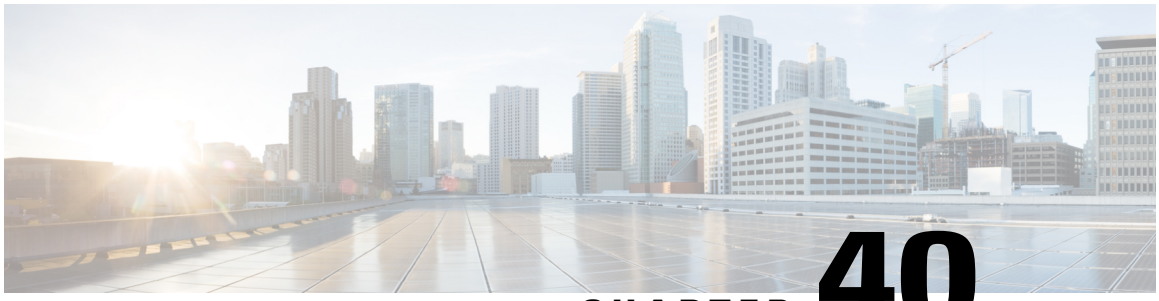
Procedure

	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 iom** {*a b*}	Enters chassis IOM mode for the specified IOM.

	Command or Action	Purpose
		Specify the peer IOM of the IOM that you want to reset.
Step 3	UCS-A /chassis/iom # **reset-peer**	Resets the peer IOM of the specified IOM.
Step 4	UCS-A /chassis/iom* # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to reset IOM b from IOM a:

```
UCS-A# scope chassis 1
UCS-A /chassis # scope iom a
UCS-A /chassis/iom # reset-peer
UCS-A /chassis/iom* # commit-buffer
```



CHAPTER 40

Backing Up and Restoring the Configuration

This chapter includes the following sections:

Backup Operations in UCS

When you perform a backup through Cisco UCS Manager, you take a snapshot of all or part of the system configuration and export the file to a location on your network. You cannot use Cisco UCS Manager to back up data on the servers.

You can perform a backup while the system is up and running. The backup operation only saves information from the management plane. It does not have any impact on the server or network traffic.

Backup Types

You can perform one or more of the following types of backups in Cisco UCS Manager and Cisco UCS Central:

- **Full state—**A binary file that includes a snapshot of the entire system. You can use the file generated from this backup to restore the system during disaster recovery. This file can restore or rebuild the configuration on the original fabric interconnect, or recreate the configuration on a different fabric interconnect. You cannot use this file for an import.

Note You can only use a full state backup file to restore a system that is running the same version as the system from which the backup file was exported.

- **All configuration—**An XML file that includes all system and logical configuration settings. You can use the file generated from this backup to import these configuration settings to the original fabric interconnect or to a different fabric interconnect. You cannot use this file for a system restore. This file does not include passwords for locally authenticated users.
- **System configuration—**An XML file that includes all system configuration settings such as usernames, roles, and locales. You can use the file generated from this backup to import these configuration settings to the original fabric interconnect or to a different fabric interconnect. You cannot use this file for a system restore.
- **Logical configuration—**An XML file that includes all logical configuration settings such as service profiles, VLANs, VSANs, pools, and policies. You can use the file generated from this backup to import these configuration settings to the original fabric interconnect or to a different fabric interconnect. You cannot use this file for a system restore.

Considerations and Recommendations for Backup Operations

Before you create a backup operation, consider the following:

Backup Locations

The backup location is the destination or folder on the network where you want Cisco UCS Manager to export the backup file. You can maintain only one backup operation for each location where you plan to save a backup file.

Potential to Overwrite Backup Files

If you rerun a backup operation without changing the filename, Cisco UCS Manager overwrites the existing file on the server. To avoid overwriting existing backup files, change the filename in the backup operation or copy the existing file to another location.

Multiple Types of Backups

You can run and export more than one type of backup to the same location. Change the backup type before you rerun the backup operation. We recommend that you change the filename for easier identification and to avoid overwriting the existing backup file.

Scheduled Backups

You can create a backup operation in advance and leave the admin state disabled, until you are ready to run the backup. Cisco UCS Manager does not run the backup operation, save, or export the configuration file until you set the admin state of the backup operation to enabled.

Incremental Backups

You cannot perform incremental backups.

Encryption of Full State Backups

Full state backups are encrypted so that passwords and other sensitive information are not exported as clear text.

FSM Tasks for Backup Policy and Configuration Export Policy

When configuring both **Backup Policy** and **Config Export Policy** on the **Policy Backup & Export** tab and using the same hostname for both policies, Cisco UCS Manager will create only one **Backup Operation** in the **Backup Configuration** page to run both tasks. Each policy run will not have a separate FSM task.

To see a separate FSM task for each policy, you can create a hostname alias in your DNS server to point to the same FTP/TFTP/SCP/SFTP server. Then you can use one hostname for the **Backup Policy** and another hostname for the **Config Export Policy**.

Scheduled Backups

You can configure policies in Cisco UCS to schedule the following types of backups:

- Full state
- All configuration

You cannot schedule any other type of backup.

Full State Backup Policy

The full state backup policy allows you to schedule regular full state backups of a snapshot of the entire system. You can choose whether to configure the full state backup to occur on a daily, weekly, or biweekly basis.

Cisco UCS Manager maintains a maximum number of backup files on the remote server. The maxfiles parameter is used when Cisco UCS Manager is registered with Cisco UCS Central. The maxfiles parameter is user configurable on Cisco UCS Central and controls the number of backup files stored on Cisco UCS Central.

If Cisco UCS Manager is not registered with Cisco UCS Central, and the user is storing backup files on a remote backup server, the backup files are not managed by Cisco UCS Manager. The remote machine server administrator must monitor the disk usage and rotate the backup files to create space for new backup files.

All Configuration Export Policy

The all configuration backup policy allows you to schedule a regular backup and export of all system and logical configuration settings. This backup does not include passwords for locally authenticated users. You can choose whether to configure the all configuration backup to occur on a daily, weekly, or bi-weekly basis.

Cisco UCS maintains a maximum number of backup files on the remote server. When that number is exceeded, Cisco UCS overwrites the oldest backup file.

Import Configuration

You can import any configuration file that was exported from Cisco UCS. The file does not need to have been exported from the same Cisco UCS.

Note You cannot import configuration from a higher release to a lower release.

The import function is available for all configuration, system configuration, and logical configuration files. You can perform an import while the system is up and running. An import operation modifies information on the management plane only. Some modifications caused by an import operation, such as a change to a vNIC assigned to a server, can cause a server reboot or other operations that disrupt traffic.

You cannot schedule an import operation. You can, however, create an import operation in advance and leave the admin state disabled until you are ready to run the import.Cisco UCS will not run the import operation on the configuration file until you set the admin state to enabled.

You can maintain only one import operation for each location where you saved a configuration backup file.

Import Methods

You can use one of the following methods to import and update a system configuration through Cisco UCS:

- **Merge—**The information in the imported configuration file is compared with the existing configuration information. If there are conflicts, the import operation overwrites the information on the Cisco UCS domain with the information in the import configuration file.
- **Replace—**The current configuration information is replaced with the information in the imported configuration file one object at a time.

System Restore

You can use the restore function for disaster recovery.

You can restore a system configuration from any full state backup file that was exported from Cisco UCS. The file does not need to have been exported from Cisco UCS on the system that you are restoring. When restoring using a backup file that was exported from a different system, we recommend that you use a system with the same or similar system configuration and hardware, including fabric interconnects, servers, adapters, and I/O module or FEX connectivity. Mismatched hardware and system configuration can lead to the restored system not fully functioning. If there is a mismatch between the I/O module links or servers on the two systems, acknowledge the chassis and servers after the restore operation.

The restore function is only available for a full state backup file. You cannot import a full state backup file. You perform a restore through the initial system setup. For more information, see the appropriate *Cisco UCS Central Installation and Upgrade Guide*.

Note You can only use a full state backup file to restore a system that is running the same version as the system from which the backup file was exported.

Required User Role for Backup and Import Operations

You must have a user account that includes the admin role to create and run backup and import operations.

Configuring Backup Operations

Creating a Backup Operation

Before You Begin

Obtain the backup server IPv4 or IPv6 address and authentication credentials.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **create backup** *URL backup-type* {**disabled** | **enabled**}	Creates a backup operation. Specify the *URL* for the backup file using one of the following syntax: • **ftp:**// *username@hostname* / *path* • **scp:**// *username@hostname* / *path* • **sftp:**// *username@hostname* / *path* • **tftp:**// *hostname* **:** *port-num* / *path* The *backup-type* argument can be one of the following values: • **all-configuration —**Backs up the server-, fabric-, and system-related configuration • **logical-configuration —**Backs up the fabric- and service profile-related configuration • **system-configuration —**Backs up the system-related configuration • **full-state —**Backs up the full state for disaster recovery

	Command or Action	Purpose
		Note • Full-state backup files cannot be imported using an import operation. They are used only to restore the configuration for a fabric interconnect. • You can only use a full state backup file to restore a system that is running the same version as the system from which the backup file was exported. You can save multiple backup operations, but only one operation for each hostname is saved. If you use the **enable** keyword, the backup operation automatically runs as soon as you enter the **commit-buffer** command. If you use the **disable** keyword, the backup operation will not run until it is enabled. When enabling a backup operation, you must specify the hostname you used when creating the backup operation.
Step 3	UCS-A /system # **commit-buffer**	Commits the transaction.

The following example shows how to create a disabled all-configuration backup operation for hostname host35 and commit the transaction:

```
UCS-A# scope system
UCS-A /system* # create backup scp://user@host35/backups/all-config9.bak all-configuration
 disabled
Password:
UCS-A /system* # commit-buffer
UCS-A /system #
```

Running a Backup Operation

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope backup** *hostname*	Enters system backup mode for the specified hostname.
Step 3	UCS-A /system/backup # **enable**	Enables the backup operation. **Note** For backup operations using FTP, SCP, SFTP, you are prompted for the password. Enter the password before committing the transaction.
Step 4	UCS-A /system/backup # **commit-buffer**	Commits the transaction.

The following example enables a backup operation named host35, enters the password for the SCP protocol, and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope backup host35
UCS-A /system/backup # enable
Password:
UCS-A /system/backup* # commit-buffer
UCS-A /system/backup #
```

Modifying a Backup Operation

You can modify a backup operation to save a file of another backup type to that location or to change the filename and avoid overwriting previous backup files.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope backup** *hostname*	Enters system backup mode for the specified hostname.
Step 3	UCS-A /system/backup # **disable**	(Optional) Disables an enabled backup operation so that it does not automatically run when the transaction is committed.
Step 4	UCS-A /system/backup # **enable**	(Optional) Automatically runs the backup operation as soon as you commit the transaction.
Step 5	UCS-A /system/backup # **set descr** *description*	(Optional) Provides a description for the backup operation. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any show command output.
Step 6	UCS-A /system/backup # **set protocol** {**ftp** | **scp** | **sftp** | **tftp**}	(Optional) Specifies the protocol to use when communicating with the remote server.
Step 7	UCS-A /system/backup # **set remote-file** *filename*	(Optional) Specifies the name of the configuration file that is being backed up.
Step 8	UCS-A /system/backup # **set type** *backup-type*	(Optional) Specifies the type of backup file to be made. The *backup-type* argument can be one of the following values:

	Command or Action	Purpose
		• **all-configuration —**Backs up the server, fabric, and system related configuration • **logical-configuration —**Backs up the fabric and service profile related configuration • **system-configuration —**Backs up the system related configuration • **full-state —**Backs up the full state for disaster recovery **Note** • Full-state backup files cannot be imported using an import operation. They are used only to restore the configuration for a fabric interconnect. • You can only use a full state backup file to restore a system that is running the same version as the system from which the backup file was exported.
Step 9	UCS-A /system/backup # **set preserve-pooled-values** {**no** | **yes**}	(Optional) Specifies whether pool-derived identity values or physical device user labels, such as vHBA WWPN, vNIC MAC, WWNN, and UUID, will be saved with the backup.
Step 10	UCS-A /system/backup # **set user** *username*	(Optional) Specifies the username the system should use to log in to the remote server. This step does not apply if the TFTP protocol is used.
Step 11	UCS-A /system/backup # **set password**	(Optional) After you press Enter, you are prompted to enter the password. Specifies the password for the remote server username. This step does not apply if the TFTP protocol is used.
Step 12	UCS-A /system/backup # **commit-buffer**	Commits the transaction.

The following example adds a description and changes the protocol, username, and password for the host35 backup operation and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope backup host35
UCS-A /system/backup # set descr "This is a backup operation for host35."
UCS-A /system/backup* # set protocol sftp
UCS-A /system/backup* # set user UserName32
UCS-A /system/backup* # set password
Password:
UCS-A /system/backup* # set preserve-pooled-values no
UCS-A /system/backup* # commit-buffer
UCS-A /system #
```

Deleting a Backup Operation

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **delete backup** *hostname*	Deletes the backup operation for the specified hostname.
Step 3	UCS-A /system # **commit-buffer**	Commits the transaction.

The following example deletes a backup operation for the host35 hostname and commits the transaction:

```
UCS-A# scope system
UCS-A /system # delete backup host35
UCS-A /system* # commit-buffer
UCS-A /system #
```

Configuring Scheduled Backups

Configuring the Full State Backup Policy

Before You Begin

Obtain the backup server IPv4 or IPv6 address and authentication credentials.

Procedure

	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 backup-policy default**	Enters the all configuration export policy mode.
Step 3	UCS-A /org/backup-policy # **set hostname** {*hostname* | *ip-addr* | *ip6-addr*}	Specifies the hostname, IPv4 or IPv6 address of the location where the backup policy is stored. This can be a server, storage array, local drive, or any read/write media that the fabric interconnect can access through the network.

	Command or Action	Purpose
		Note If you use a hostname rather than an IPv4 or IPv6 address, you must configure a DNS server. If the Cisco UCS domain is not registered with Cisco UCS Central or DNS management is set to **local**, configure a DNS server in Cisco UCS Manager. If the Cisco UCS domain is registered with Cisco UCS Central and DNS management is set to **global**, configure a DNS server in Cisco UCS Central.
Step 4	UCS-A /org/backup-policy # **set protocol** {**ftp** | **scp** | **sftp** | **tftp**}	Specifies the protocol to use when communicating with the remote server.
Step 5	UCS-A /org/backup-policy # **set user** *username*	Specifies the username the system should use to log in to the remote server. This step does not apply if the TFTP protocol is used.
Step 6	UCS-A /system/backup-policy # **set password**	After you press Enter, you are prompted to enter the password. Specifies the password for the remote server username. This step does not apply if the TFTP protocol is used.
Step 7	UCS-A /system/backup-policy # **set remote-file** *filename*	Specifies the full path to the backup file. This field can contain the filename as well as the path. If you omit the filename, the backup procedure assigns a name to the file.
Step 8	UCS-A /system/backup-policy # **set adminstate** {**disabled** | **enabled**}	Specifies the admin state for the policy. This can be one of the following: • **enabled—**Cisco UCS Manager exports the backup file using the schedule specified in the **Schedule** field. • **disabled—**Cisco UCS Manager does not export the file.
Step 9	UCS-A /system/backup-policy # **set schedule** {**daily** | **weekly** | **bi-weekly**}	Specifies the frequency with which Cisco UCS Manager exports the backup file.
Step 10	UCS-A /system/backup-policy # **set descr** *description*	Specifies a description for the backup policy. Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote).
Step 11	UCS-A /backup-policy # **commit-buffer**	Commits the transaction.

The following example shows how to configure the full state backup policy for a weekly backup and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope backup-policy default
UCS-A /org/backup-policy # set hostname host35
```

```
UCS-A /org/backup-policy* # set protocol scp
UCS-A /org/backup-policy* # set user UserName32
UCS-A /backup-policy* # set password
Password:
UCS-A /backup-policy* # set remote-file /backups/full-state1.bak
UCS-A /backup-policy* # set adminstate enabled
UCS-A /backup-policy* # set schedule weekly
UCS-A /backup-policy* # set descr "This is a full state weekly backup."
UCS-A /backup-policy* # commit-buffer
UCS-A /backup-policy #
```

Configuring the All Configuration Export Policy

Before You Begin

Obtain the backup server IPv4 or IPv6 address and authentication credentials.

Procedure

	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 cfg-export-policy default**	Enters the all configuration export policy mode.
Step 3	UCS-A /org/cfg-export-policy # **set hostname** {*hostname* | *ip-addr* | *ip6-addr*}	Specifies the hostname, IPv4 or IPv6 address of the location where the configuration file is stored. This can be a server, storage array, local drive, or any read/write media that the fabric interconnect can access through the network. **Note** If you use a hostname rather than an IPv4 or IPv6 address, you must configure a DNS server. If the Cisco UCS domain is not registered with Cisco UCS Central or DNS management is set to **local**, configure a DNS server in Cisco UCS Manager. If the Cisco UCS domain is registered with Cisco UCS Central and DNS management is set to **global**, configure a DNS server in Cisco UCS Central.
Step 4	UCS-A /org/cfg-export-policy # **set protocol** {**ftp** | **scp** | **sftp** | **tftp**}	Specifies the protocol to use when communicating with the remote server.
Step 5	UCS-A /org/cfg-export-policy # **set user** *username*	Specifies the username the system should use to log in to the remote server. This step does not apply if the TFTP protocol is used.
Step 6	UCS-A /system/cfg-export-policy # **set password**	After you press Enter, you are prompted to enter the password. Specifies the password for the remote server username. This step does not apply if the TFTP protocol is used.

	Command or Action	Purpose
Step 7	UCS-A /system/cfg-export-policy # **set remote-file** *filename*	Specifies the full path to the exported configuration file. This field can contain the filename as well as the path. If you omit the filename, the backup procedure assigns a name to the file.
Step 8	UCS-A /system/cfg-export-policy # **set adminstate** {**disabled** | **enabled**}	Specifies the admin state for the policy. This can be one of the following: • **enabled—**Cisco UCS Manager exports the configuration information using the schedule specified in the **Schedule** field. • **disabled—**Cisco UCS Manager does not export the information.
Step 9	UCS-A /system/cfg-export-policy # **set schedule** {**daily** | **weekly** | **bi-weekly**}	Specifies the frequency with which Cisco UCS Manager exports the configuration information.
Step 10	UCS-A /system/cfg-export-policy # **set descr** *description*	Specifies a description for the configuration export policy. Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote).
Step 11	UCS-A /cfg-export-policy # **commit-buffer**	Commits the transaction.

The following example shows how to configure the all configuration export policy for a weekly backup and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope cfg-export-policy default
UCS-A /org/cfg-export-policy # set hostname host35
UCS-A /org/cfg-export-policy* # set protocol scp
UCS-A /org/cfg-export-policy* # set user UserName32
UCS-A /cfg-export-policy* # set password
Password:
UCS-A /cfg-export-policy* # set remote-file /backups/all-config9.bak
UCS-A /cfg-export-policy* # set adminstate enabled
UCS-A /cfg-export-policy* # set schedule weekly
UCS-A /cfg-export-policy* # set descr "This is an all configuration backup."
UCS-A /cfg-export-policy* # commit-buffer
UCS-A /cfg-export-policy #
```

Configuring Backup/Export Configuration Reminders

Procedure

	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 backup-exp-policy**	Enters the backup/export configuration policy mode.
Step 3	UCS-A /org/backup-exp-policy # **show**	Displays the existing backup/export configuration policy.
Step 4	UCS-A /org/backup-exp-policy # **set adminstate** {**disable** | **enable**}	Specifies the admin state for the policy. This can be one of the following: • **enable**—Cisco UCS Manager raises a fault if a backup is not taken during the specified time period. • **disable**—Cisco UCS Manager does not raise a fault if a backup is not taken during the specified time period.
Step 5	UCS-A /org/backup-exp-policy # **set frequency** *Number_of_Days*	Specifies the number of days before you are reminded to take a backup. Enter an integer between 1 and 365. The default value is 30 days.
Step 6	UCS-A /org/backup-exp-policy # **commit-buffer**	Commits the transaction.

The following example shows how to view the current backup/export config policy, change the frequency of the reminders, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope backup-exp-policy
UCS-A /org/backup-exp-policy # set frequency 5
UCS-A /org/backup-exp-policy* # commit-buffer
UCS-A /org/backup-exp-policy #
```

Configuring Import Operations

Creating an Import Operation

You cannot import a Full State backup file. You can import any of the following configuration files:

- All configuration

- System configuration
- Logical configuration

Before You Begin

Collect the following information to import a configuration file:

- Backup server IP address and authentication credentials
- Fully-qualified name of a backup file

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **create import-config** *URL* {**disabled** | **enabled**} {**merge** | **replace**}	Creates an import operation. Specify the URL for the file being imported using one of the following syntax: • **ftp:**// *username***@***hostname* / *path* • **scp:**// *username*@*hostname* / *path* • **sftp:**// *username*@*hostname* / *path* • **tftp:**// *hostname* **:** *port-num* / *path* You can save multiple import operations, but only one operation for each hostname is saved. If you use the **enable** keyword, the import operation automatically runs as soon as you enter the **commit-buffer** command. If you use the **disable** keyword, the import operation will not run until it is enabled. When enabling an import operation, you must specify the hostname you used when creating the import operation. If you use the **merge** keyword, the configuration information is merged with the existing information. If there are conflicts, the system replaces the information on the current system with the information in the import configuration file. If you use the **replace** keyword, the system takes each object in the import configuration file and overwrites the corresponding object in the current configuration.
Step 3	UCS-A /system/import-config# **set descr** *description*	(Optional) Provides a description for the import operation. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any show command output.
Step 4	UCS-A /system/import-config # **commit-buffer**	Commits the transaction.

The following example creates a disabled import operation for hostname host35 that replaces the existing configuration and commits the transaction:

```
UCS-A# scope system
UCS-A /system* # create import-config scp://user@host35/backups/all-config9.bak disabled
replace
Password:
UCS-A /system/import-config* # commit-buffer
UCS-A /system/import-config #
```

Running an Import Operation

You cannot import a Full State backup file. You can import any of the following configuration files:

- All configuration
- System configuration
- Logical configuration

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **scope import-config** *hostname*	Enters system backup mode for the specified hostname.
Step 3	UCS-A /system/import-config # **enable**	Enables the import operation.
Step 4	UCS-A /system/import-config # **commit-buffer**	Commits the transaction.

The following example enables an import operation for the host35 hostname and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope import-config host35
UCS-A /system/import-config # enable
UCS-A /system/import-config* # commit-buffer
UCS-A /system/import-config #
```

Modifying an Import Operation

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.

	Command or Action	Purpose
Step 2	UCS-A /system # **scope import-config** *hostname*	Enters system import configuration mode for the specified hostname.
Step 3	UCS-A /system/import-config # **disable**	(Optional) Disables an enabled import operation so that it does not automatically run when the transaction is committed.
Step 4	UCS-A /system/import-config # **enable**	(Optional) Automatically runs the import operation as soon as you commit the transaction.
Step 5	UCS-A /system/import-config # **set action** {**merge** | **replace**}	(Optional) Specifies one of the following action types to use for the import operation: • **Merge —**The configuration information is merged with the existing information. If there are conflicts, the system replaces the information on the current system with the information in the import configuration file. • **Replace —**The system takes each object in the import configuration file and overwrites the corresponding object in the current configuration.
Step 6	UCS-A /system/import-config # **set descr** *description*	(Optional) Provides a description for the import operation. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any show command output.
Step 7	UCS-A /system/import-config # **set password**	(Optional) After you press Enter, you are prompted to enter the password. Specifies the password for the remote server username. This step does not apply if the TFTP protocol is used. **Note** Cisco UCS Manager does not store this password. Therefore, you do not need to enter this password unless you intend to enable and run the import operation immediately.
Step 8	UCS-A /system/import-config # **set protocol** {**ftp** | **scp** | **sftp** | **tftp**}	(Optional) Specifies the protocol to use when communicating with the remote server.
Step 9	UCS-A /system/import-config # **set remote-file** *filename*	(Optional) Specifies the name of the configuration file that is being imported.

	Command or Action	Purpose
Step 10	UCS-A /system/import-config # **set user** *username*	(Optional) Specifies the username the system should use to log in to the remote server. This step does not apply if the TFTP protocol is used.
Step 11	UCS-A /system/import-config # **commit-buffer**	Commits the transaction.

The following example adds a description, changes the password, protocol and username for the host35 import operation, and commits the transaction:

```
UCS-A# scope system
UCS-A /system # scope import-config host35
UCS-A /system/import-config # set descr "This is an import operation for host35."
UCS-A /system/import-config* # set password
Password:
UCS-A /system/import-config* # set protocol sftp
UCS-A /system/import-config* # set user jforlenz32
UCS-A /system/import-config* # commit-buffer
UCS-A /system/import-config #
```

Deleting an Import Operation

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope system**	Enters system mode.
Step 2	UCS-A /system # **delete import-config** *hostname*	Deletes the import operation for the specified hostname.
Step 3	UCS-A /system # **commit-buffer**	Commits the transaction.

The following example deletes the import operation for the host35 hostname and commits the transaction:

```
UCS-A# scope system
UCS-A /system # delete import-config host35
UCS-A /system* # commit-buffer
UCS-A /system #
```

Restoring the Configuration for a Fabric Interconnect

It is recommended that you use a full state backup file to restore a system that is running the same version as the system from which the backup file was exported. You can also use a full state backup to restore a system if they have the same release train. For example, you can use a full state backup taken from a system running Release 2.1(3a) to restore a system running Release 2.1(3f).

To avoid issues with VSAN or VLAN configuration, a backup should be restored on the fabric interconnect that was the primary fabric interconnect at the time of backup.

Before You Begin

Collect the following information to restore the system configuration:

- Fabric interconnect management port IPv4 address and subnet mask, or IPv6 address and prefix
- Default gateway IPv4 or IPv6 address
- Backup server IPv4 or IPv6 address and authentication credentials
- Fully-qualified name of a Full State backup file

Note You must have access to a Full State configuration file to perform a system restore. You cannot perform a system restore with any other type of configuration or backup file.

Procedure

Step 1 Connect to the console port.

Step 2 If the fabric interconnect is off, power on the fabric interconnect.
You will see the power on self-test message as the fabric interconnect boots.

Step 3 At the installation method prompt, enter **console** .

Step 4 Enter **restore** to restore the configuration from a full-state backup.

Step 5 Enter **y** to confirm that you want to restore from a full-state backup.

Step 6 Enter the IP address for the management port on the fabric interconnect.

Step 7 Enter the subnet mask for the management port on the fabric interconnect.

Step 8 Enter the IP address for the default gateway.

Step 9 Enter one of the following protocols to use when retrieving the backup configuration file:

- **scp**
- **ftp**
- **tftp**
- **sftp**

Step 10 Enter the IP address of the backup server.

Step 11 Enter the full path and filename of the Full State backup file.

Note You can only use a full state backup file to restore a system that is running the same version as the system from which the backup file was exported.

Step 12 Enter the username and password to access the backup server.
The fabric interconnect logs in to the backup server, retrieves a copy of the specified Full State backup file, and restores the system configuration. For a cluster configuration, you do not need to restore the secondary fabric interconnect. As soon as the secondary fabric interconnect reboots, Cisco UCS synchronizes the configuration with the primary fabric interconnect.

The following example restores a system configuration from the Backup.bak file, which was retrieved from the 20.10.20.10 backup server using FTP:

```
  Enter the configuration method. (console/gui) ? console

  Enter the setup mode; setup newly or restore from backup. (setup/restore) ? restore

  NOTE:
    To configure Fabric interconnect using a backup file on a remote server,
    you will need to setup management interface.
    The management interface will be re-configured (if necessary),
    based on information stored in the backup file.


  Continue to restore this Fabric interconnect from a backup file (yes/no) ? yes

  Physical Switch Mgmt0 IPv4 address : 192.168.10.10

  Physical Switch Mgmt0 IPv4 netmask : 255.255.255.0

  IPv4 address of the default gateway : 192.168.10.1

  Enter the protocol to get backup file (scp/ftp/tftp/sftp) ?  scp
  Enter the IP address of backup server: 20.10.20.10
  Enter fully qualified backup file name: Backup.bak
  Enter user ID: user
  Enter password:
    Retrieved backup configuration file.
  Configuration file - Ok


Cisco UCS 6100 Series Fabric Interconnect
UCS-A login:
```

Erasing the Configuration

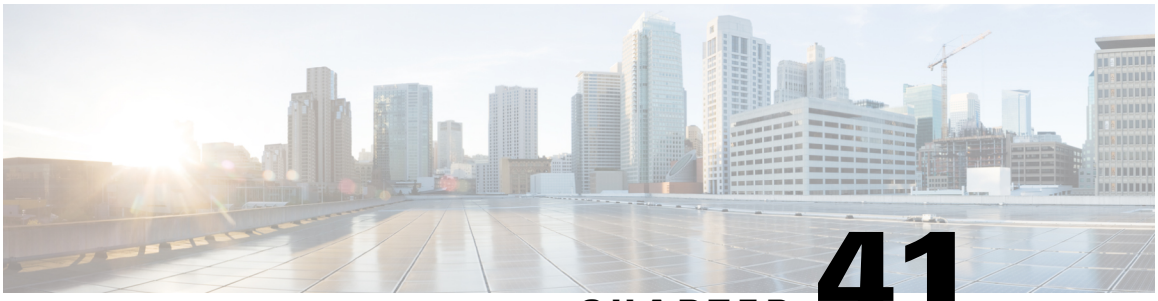
Caution You should erase the configuration only when it is necessary. Erasing the configuration completely removes the configuration and reboots the system in an unconfigured state. You must then either restore the configuration from a backup file or perform an initial system setup.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **connect local-mgmt**	Enters the local management CLI.
Step 2	UCS-A(local-mgmt)# **erase configuration**	Erases the configuration. You are prompted to confirm that you want to erase the configuration. Entering **yes** erases the configuration and reboots the system in an unconfigured state.

The following example erases the configuration:

```
UCS-A# connect local-mgmt
UCS-A(local-mgmt)# erase configuration
All UCS configurations will be erased and system will reboot. Are you sure? (yes/no): yes
```



CHAPTER 41

Recovering a Lost Password

This chapter includes the following sections:

Password Recovery for the Admin Account

The admin account is the system administrator or superuser account. If an administrator loses the password to this account, you can have a serious security issue. The procedure to recover the password for the admin account requires you to power cycle all fabric interconnects and will lead to a temporary data transmission outage.

When you recover the password for the admin account, you actually change the password for that account. You cannot retrieve the original password for that account.

You can reset the password for all other local accounts through Cisco UCS Manager. However, you must log in to Cisco UCS Manager with an account that includes aaa or admin privileges.

Caution For Cisco UCS Mini, this procedure requires you to pull all the fabric interconnects in a Cisco UCS domain out of their chassis slots. As a result, all data transmission in the Cisco UCS domain is stopped until you slide the fabric interconnects back into their chassis slots.

For other Cisco UCS configurations, this procedure requires you to power down all fabric interconnects. As a result, all data transmission in the Cisco UCS domain is stopped until you restart the fabric interconnects.

Determining the Leadership Role of a Fabric Interconnect

Note To determine the role of the fabric interconnects in a cluster when the admin password is lost, open the Cisco UCS Manager GUI from the IP addresses of both fabric interconnects. The subordinate fabric interconnect fails with the following message:

```
UCSM GUI is not available on secondary node.
```

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **show cluster state**	Displays the operational state and leadership role for both fabric interconnects in a cluster.

The following example displays the leadership role for both fabric interconnects in a cluster, where fabric interconnect A has the primary role and fabric interconnect B has the subordinate role:

```
UCS-A# show cluster state
Cluster Id: 0x4432f72a371511de-0xb97c000de1b1ada4

A: UP, PRIMARY
B: UP, SUBORDINATE

HA READY
```

Recovering the Admin Account Password in a Standalone Configuration

This procedure will help you to recover the password that you set for the admin account when you performed an initial system setup on the fabric interconnect. The admin account is the system administrator or superuser account.

Before You Begin

1. Physically connect the console port on the fabric interconnect to a computer terminal or console server
2. Determine the running versions of the following firmware:
   - The firmware kernel version on the fabric interconnect
   - The firmware system version

Tip To find this information, you can log in with any user account on the Cisco UCS domain.

Procedure

Step 1 Connect to the console port.

Step 2 Power cycle the fabric interconnect:

a) For Cisco UCS Mini, pull the fabric interconnect out of its chassis slot. For all other configurations, turn off the power to the fabric interconnect.

b) For Cisco UCS Mini, slide the fabric interconnect back into its chassis slot. For all other configurations, turn on the power to the fabric interconnect.

Step 3 In the console, press one of the following key combinations as it boots to get the `loader` prompt:

- Ctrl+l
- Ctrl+Shift+r

You may need to press the selected key combination multiple times before your screen displays the `loader` prompt.

Step 4 Boot the kernel firmware version on the fabric interconnect.

```
loader >
boot /installables/switch/
kernel_firmware_version
```

Example:

```
loader > boot /installables/switch/ucs-6100-k9-kickstart.4.1.3.N2.1.0.11.gbin

loader > boot /installables/switch/ucs-mini-k9-kickstart.5.0.3.N2.3.01a.bin
```

Step 5 Enter config terminal mode.

```
Fabric(boot)#
config terminal
```

Step 6 Reset the admin password.

```
Fabric(boot)(config)#
admin-password
password
```

Choose a strong password that includes at least one capital letter and one number. The password cannot be blank.

The new password displays in clear text mode.

Step 7 Exit config terminal mode and return to the boot prompt.

Step 8 Boot the system firmware version on the fabric interconnect.

```
Fabric(boot)#
load /installables/switch/
system_firmware_version
```

Example:

```
Fabric(boot)# load /installables/switch/ucs-6100-k9-system.4.1.3.N2.1.0.211.bin

Fabric(boot)# load /installables/switch/ucs-mini-k9-system.5.0.3.N2.3.01a.bin
```

Step 9 After the system image loads, log in to Cisco UCS Manager.

Recovering the Admin Account Password in a Cluster Configuration

This procedure will help you to recover the password that you set for the admin account when you performed an initial system setup on the fabric interconnects. The admin account is the system administrator or superuser account.

Before You Begin

1. Physically connect a console port on one of the fabric interconnects to a computer terminal or console server
2. Obtain the following information:
   - The firmware kernel version on the fabric interconnect
   - The firmware system version
   - Which fabric interconnect has the primary leadership role and which is the subordinate

Tip To find this information, you can log in with any user account on the Cisco UCS domain.

Procedure

Step 1 Connect to the console port.

Step 2 For the subordinate fabric interconnect:

a) For Cisco UCS Mini, pull the fabric interconnect out of its chassis slot. For all other configurations, turn off the power to the fabric interconnect.

b) For Cisco UCS Mini, slide the fabric interconnect back into its chassis slot. For all other configurations, turn on the power to the fabric interconnect.

c) In the console, press one of the following key combinations as it boots to get the `loader` prompt:

   - Ctrl+l
   - Ctrl+Shift+r

   You may need to press the selected key combination multiple times before your screen displays the `loader` prompt.

Step 3 Power cycle the primary fabric interconnect:

a) For Cisco UCS Mini, pull the fabric interconnect out of its chassis slot. For all other configurations, turn off the power to the fabric interconnect.

b) For Cisco UCS Mini, slide the fabric interconnect back into its chassis slot. For all other configurations, turn on the power to the fabric interconnect.

Step 4 In the console, press one of the following key combinations as it boots to get the `loader` prompt:

- Ctrl+l
- Ctrl+Shift+r

You may need to press the selected key combination multiple times before your screen displays the `loader` prompt.

Step 5 Boot the kernel firmware version on the primary fabric interconnect.

```
loader > boot /installables/switch/
kernel_firmware_version
```

Example:

```
loader > boot /installables/switch/ucs-6100-k9-kickstart.4.1.3.N2.1.0.11.gbin

loader > boot /installables/switch/ucs-mini-k9-kickstart.5.0.3.N2.3.01a.bin
```

Step 6 Enter config terminal mode.

```
Fabric(boot)# config terminal
```

Step 7 Reset the admin password.

```
Fabric(boot)(config)# admin-password password
```

Choose a strong password that includes at least one capital letter and one number. The password cannot be blank.

The new password displays in clear text mode.

Step 8 Exit config terminal mode and return to the boot prompt.

Step 9 Boot the system firmware version on the primary fabric interconnect.

```
Fabric(boot)# load /installables/switch/
system_firmware_version
```

Example:

```
Fabric(boot)# load /installables/switch/ucs-6100-k9-system.4.1.3.N2.1.0.211.bin

Fabric(boot)# load /installables/switch/ucs-mini-k9-system.5.0.3.N2.3.01a.bin
```

Step 10 After the system image loads, log in to Cisco UCS Manager.

Step 11 In the console for the subordinate fabric interconnect, do the following to bring it up:

a) Boot the kernel firmware version on the subordinate fabric interconnect.

```
loader > boot /installables/switch/
kernel_firmware_version
```

b) Boot the system firmware version on the subordinate fabric interconnect.

```
Fabric(boot)# load /installables/switch/
system_firmware_version
```