Cisco UCS S3260 Server Integration with Cisco UCS Manager Using the CLI, Release 3.2

First Published: 2017-08-18

Last Modified: 2018-03-21

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

CONTENTS

Preface

Audience

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

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

Conventions

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

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

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

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

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

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

Warning IMPORTANT SAFETY INSTRUCTIONS

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

SAVE THESE INSTRUCTIONS

Related Cisco UCS Documentation

Documentation Roadmaps

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

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

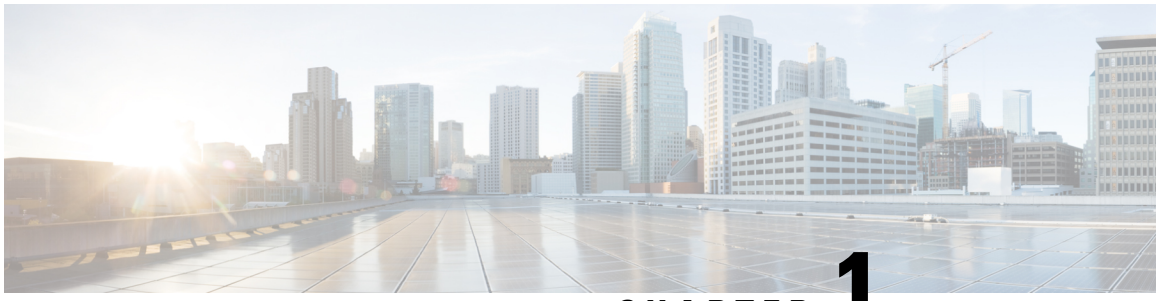
For information on supported firmware versions and supported UCS Manager versions for the rack servers that are integrated with the UCS Manager for management, refer to Release Bundle Contents for Cisco UCS Software.

Other Documentation Resources

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

Documentation Feedback

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



CHAPTER 1

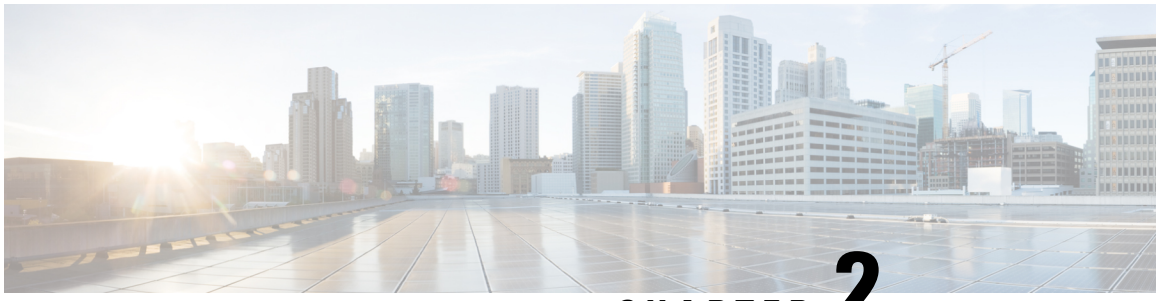
New and Changed Information

New and Changed Information for This Release

The following table provides an overview of the significant changes to this guide for this current release. The table does not provide an exhaustive list of all changes made to this guide or of all new features in this release.

Table 1: New Features and Changed Behavior in Cisco UCS Manager, 3.2(3a)

Feature	Description	Where Documented
6G-12G Mixed Mode	6G-12G Mixed Mode is available for SAS Expander Configuration Policy.	Creating SAS Expander Configuration Policy, on page 60
Single Path support for UCS S3260 Dual Pass Through (UCS-S3260-DHBA) controller	In Cisco UCS S3260 M5 servers, Cisco UCS Manager enables single path access to disks by configuring a single DiskPort per disk slot.	Disk Zoning Policies, on page 54
Extended support for Cisco UCS S3260 M5 Server	Cisco UCS Manager support for the UCS S3260 M5 rack-mount server.	About the Cisco UCS S3260 System, on page 3



CHAPTER 2

About the Cisco UCS S3260 System

The Cisco UCS S3260 is a dense storage rack server with dual server nodes, optimized for large data sets used in environments such as Big data, cloud, object storage, and content delivery. It belongs to the Cisco UCS S-Series rack-mount servers product family.

Beginning with Cisco UCS Manager Release 3.1(3), Cisco UCS C3260/C3X60 is renamed to Cisco UCS S3260. You may still see certain components in the system labeled as C3260/C3X60. For this release, the terms S3260 and C3260/C3X60 are used interchangeably. Both, S3260 and C3260/C3X60, refer to the same hardware component.

Cisco UCS Manager Release 3.2(3) introduces Cisco UCS S3260 M5 server. Cisco UCS S3260 M5 server integrates with Cisco UCS Manager the same way Cisco UCS S3260 does. The information and procedures in this document can be used for both Cisco UCS S3260 M4 and Cisco UCS S3260 M5 servers.

The Cisco UCS S3260 system is designed to operate in a standalone environment and as part of the Cisco Unified Computing System with Cisco UCS Manager integration. It assumes almost the same characteristics of its predecessor, Cisco UCS C3160, but with the following additional features:

- System IO Controllers (SIOC) with Cisco VIC 1300 Series Embedded Chip supporting dual-port 40Gbps
- Support of up to two server modules
- Capability to operate in a standalone mode and with Cisco UCS Manager
- Individual hard disk drives (HDD) can be assigned to either server in the dedicated or shared mode

In addition, one of the server slots in the Cisco UCS S3260 system can be utilized by a storage expansion module for an additional four 3.5" drives. The server modules can also accommodate two solid state drives (SSD) for internal storage dedicated to that module. The chassis supports Serial Attached SCSI (SAS) expanders that can be configured to assign the 3.5" drives to individual server modules.

Beginning with release 3.1(3), Cisco UCS S3260 system supports the following:

- Server SIOC Connectivity functionality
- Second RAID controller in the optional I/O expander module
- Dual HBA Controller

Note If a Cisco UCS S3260 system has Dual HBA Controller then you cannot downgrade Cisco UCS Manager to any release earlier than 3.1(3).

In a Cisco UCS S3260 system, both servers should have either dual RAID controllers or dual HBA controllers. Mixing the controller types is not supported.

Cisco UCS S3260 system supports Server SIOC Connectivity functionality. Using this functionality, you can configure the data path through both the primary and auxiliary SIOCs when the chassis has single server and dual SIOCs set up. For more details, see Server SIOC Connectivity Functionality, on page 11.

Cisco UCS S3260 system supports Second RAID controller in the optional I/O expander module that attaches to the top of the server node. You cannot downgrade Cisco UCS Manager, BMC, CMC, and BIOS to any release earlier than 3.1(3) depending on the number of disk zoned to the controllers :

Controller Configuration	Is Downgrade Possible?
Two controllers in the server (one in optional I/O expander) or one controller in the server (in optional I/O expander) and at least one disk is zoned to the controller in the optional I/O expander.	No
Two controllers in the server (one in optional I/O expander) or one controller in the server (in optional I/O expander) and at least one disk is pre-provisioned to controller in the optional I/O expander.	No
Two controllers in the server (one in optional I/O expander) or one controller in the server (in any slot) and disk are not zoned or pre-prvosioned to the controller in optional I/O expander.	Yes

License Requirement

ETH_PORT_ACTIVATION_PKG (for 6200 FI series), 40G_ETH_PORT_ACTIVATION_PKG (for 6300 FI - 6332), 10G_PORT_ACTIVATION_PKG (for 6300 FI - 6332-16UP), licenses are used when S3260 system is connected to FI as appliance (appliance port) or Cisco UCS Manager managed node (server port).

For more information on license requirement, refer *Server License Management* chapter in *Cisco UCS Manager Server Management Guide*.

How to Use This Guide

Cisco UCS S3260 systems managed through Cisco UCS Manager support most of the features that are supported by other S-Series Rack Servers managed through Cisco UCS Manager. Cisco UCS S3260 systems also introduce some new features and management capabilities to Cisco UCS Manager. These features and management capabilities are detailed in the following chapters of this guide:

- Overview—Provides detailed information about the architecture of the Cisco UCS S3260 system and its connectivity when managed through Cisco UCS Manager.

- Migration to Cisco UCS Manager-Managed Cisco UCS S3260—Describes the steps required to migrate either a standalone Cisco UCS C3160 or a standalone Cisco UCS S3260 server to a Cisco UCS Manager-managed Cisco UCS S3260 server.
- System Related Policies—Describes the chassis discovery policy and chassis connectivity policy that are applicable to Cisco UCS S3260 systems.
- Chassis Profiles—Provides detailed information about Chassis Profiles and Chassis Profile Templates, which can now be used to define the storage, firmware and maintenance characteristics of a Cisco UCS S3260 chassis.
- Storage Management—Describes the new storage components in a Cisco UCS S3260 system, and how to manage them.
- Firmware Management—Provides detailed information about Chassis Firmware Packages and the endpoints of Cisco UCS S3260 on which firmware can be updated manually.
- Chassis Management—Provides detailed information about the management of the Cisco UCS S3260 chassis.
- Server Management—Provides detailed information about the management of the Cisco UCS S3260 Server Node.
- SIOC Management—Provides detailed information about the management of the System Input/Output controllers (SIOCs) that are part of a Cisco UCS S3260 chassis.

All features and configuration tasks that are supported by Cisco UCS Manager Release 3.1 and later releases are described in the configuration guides that are listed in the following table. These guides must be used with this quick reference guide for Cisco UCS S3260 systems.

Guide	Description
Cisco UCS Manager Getting Started Guide	Discusses Cisco UCS architecture and Day 0 operations, including Cisco UCS Manager initial configuration, and configuration best practices.
Cisco UCS Manager Administration Guide	Discusses password management, role-based access configuration, remote authentication, communication services, CIMC session management, organizations, backup and restore, scheduling options, BIOS tokens and deferred deployments.
Cisco UCS Manager Infrastructure Management Guide	Discusses physical and virtual infrastructure components used and managed by Cisco UCS Manager.
Cisco UCS Manager Firmware Management Guide	Discusses downloading and managing firmware, upgrading through Auto Install, upgrading through service profiles, directly upgrading at endpoints using firmware auto sync, managing the capability catalog, deployment scenarios, and troubleshooting.
Cisco UCS Manager Server Management Guide	Discusses the new licenses, registering Cisco UCS domains with Cisco UCS Central, power capping, server boot, server profiles and server-related policies.

Guide	Description
Cisco UCS Manager Storage Management Guide	Discusses all aspects of storage management such as SAN and VSAN in Cisco UCS Manager.
Cisco UCS Manager Network Management Guide	Discusses all aspects of network management such as LAN and VLAN connectivity in Cisco UCS Manager.
Cisco UCS Manager System Monitoring Guide	Discusses all aspects of system and health monitoring including system statistics in Cisco UCS Manager.
Cisco UCS S3260 Server Integration with Cisco UCS Manager	Discusses all aspects of management of UCS S-Series servers that are managed through Cisco UCS Manager.

The Release Notes for Cisco UCS Manager, Release 3.2 has detailed information about new features, resolved caveats, open caveats, and workarounds for Cisco UCS Manager, Release 3.2

Cisco UCS S3260 System Architectural Overview

Architectural Overview

Figure 1: Cisco UCS S3260 System Overall Architecture Diagram

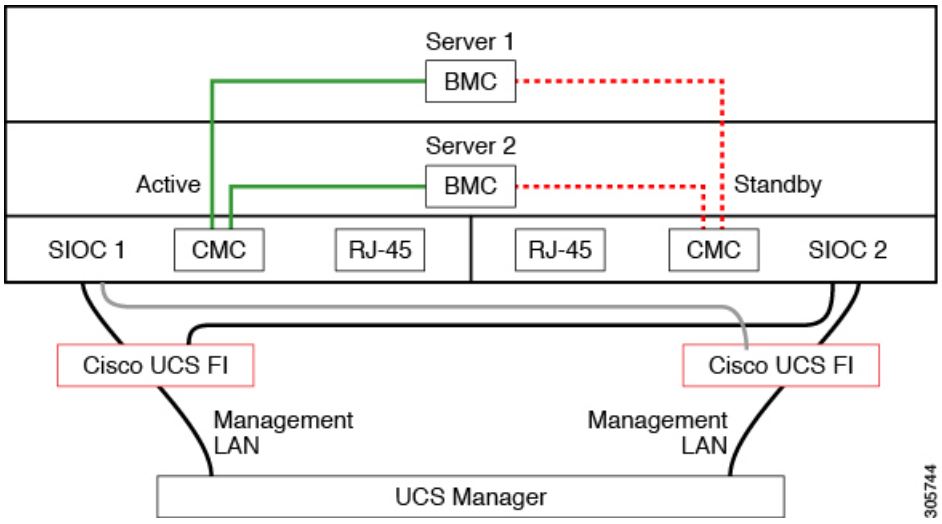
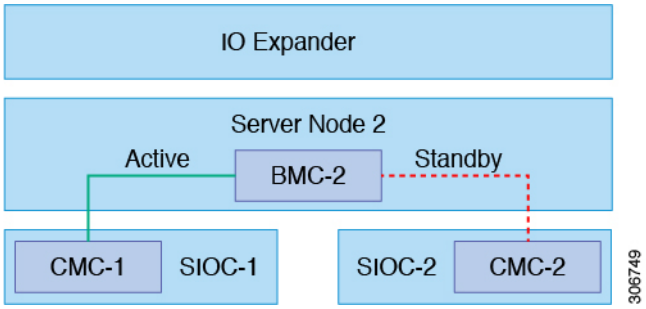


Figure 2: Cisco UCS S3260 System (with Single node Dual SIOC) Overall Architecture Diagram



The system uses a chassis management controller (CMC) to manage the server nodes. Each system I/O controller (SIOC) module contains an onboard CMC. If you have two SIOCs, the two CMCs act in an active/standby organization. The CMC in the SIOC that you log into with the Cisco IMC interface becomes the active CMC and it allows you to manage the BMCs in both server nodes.

All user interfaces run only on the active CMC. Configuration changes are automatically synchronized between the active and the standby CMCs.

When you power-cycle the system, the CMC in SIOC 1 is the active CMC by default. The active CMC will fail over to the standby CMC when any of the following conditions occur:

- The active CMC is rebooted or fails.
- The SIOC with active CMC is removed.
- Network connectivity is lost on the active CMC.

Building Blocks and Connectivity

The Cisco UCS S3260 chassis has a modular architecture consisting of the following modules:

Base Chassis

Contains four power supplies, eight fans, and a rail kit.

Server Node

One or two server nodes, each with two CPUs, 128, 256, or 512 GB of DIMM memory, and a RAID card in pass-through mode or a RAID card with a 1 GB or 4 GB cache.

System I/O Controller (SIOC) (Release 3.2(3) and earlier)

One or two System I/O Controllers, each of which includes a 1300-series VIC. The Cisco UCS S3260 SIOC has an integrated Cisco UCS VIC 1300 Series chip onboard, so there is no removable adapter card.

Optional Drive Expansion Node

Choice of either 4 x 4 TB drives (total capacity: 16TB), 4 x 6 TB drives (total capacity: 24 TB), or 4 x 10 TB drives (total capacity: 40TB).

Solid-State Boot Drives

Up to two SSDs per server node.

Cisco UCS Fabric Connectivity

The Cisco UCS S3260 chassis can be connected in one of the following ways:

- Direct connection to the fabric interconnects.
- Connectivity using FEX.

Direct Connection to Cisco UCS Fabric Interconnects

1. **Cisco UCS 6200 Series Fabric Interconnects**: The SIOCs can be connected directly to the 6248 FI ports. The SIOC uplink can be connected to an FI port in one of two ways:
   - 10G connectivity to a single FI port using a QSA cable
   - 4*10G port channel connectivity to 4 FI ports using a break-out cable
2. **Cisco UCS 6300 Series Fabric Interconnects**: The SIOC uplink can be connected directly to a 6300 Series FI port through a single 40G connection.

Connectivity using FEX

N2348UPQ and 2232 FEX: The SIOCs can be connected directly to the FEX ports through a single 10G connection using a QSA connector.

Deployment Options

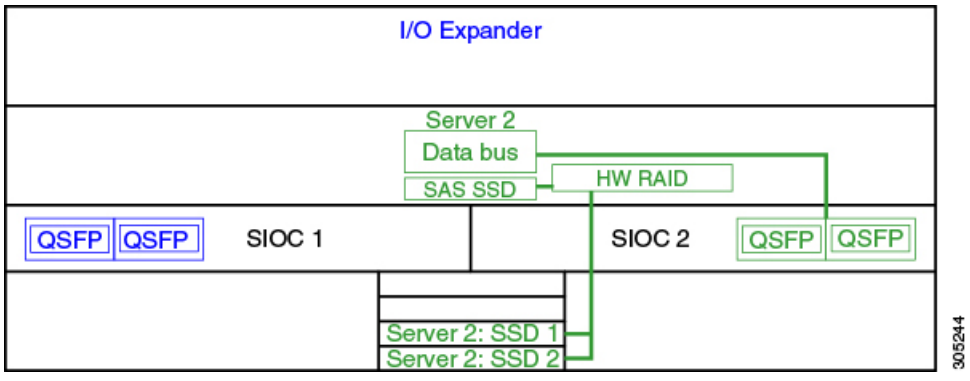
The following sections describe the three main deployment options for Cisco UCS S3260 systems—single and dual server connectivity.

Single Server Connectivity

The following illustration shows the associations for a single-server system:

- The data bus in server node 2 connects through SIOC 2.
- Server 2 SSDs 1 and 2 can be controlled by a RAID controller card in server node 2.

Figure 3: Single Server with I/O Expander



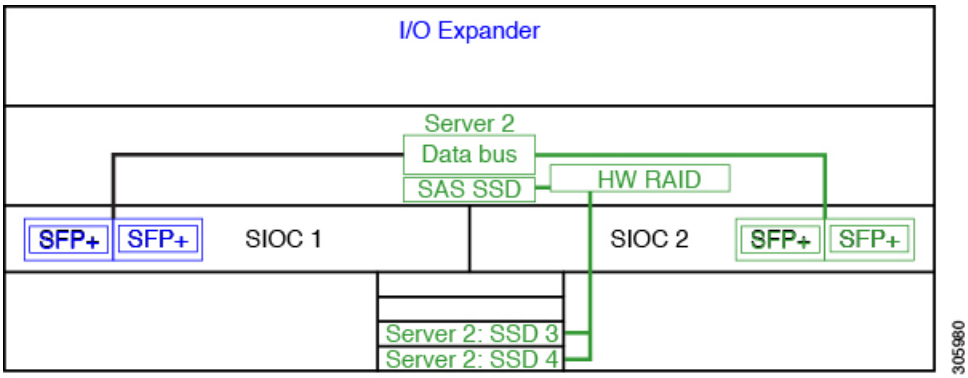
Single Server Connectivity (with Server SIOC Connectivity Functionality)

When a Cisco UCS S3260 system has single server and dual SIOCs, users can enable the Server SIOC Connectivity functionality. The following illustration shows the associations for a single-server system with Server SIOC Connectivity functionality enabled:

- The data bus in server node 2 connects through both the primary and auxiliary SIOCs.

Note Primary SIOC for server 1 is SIOC 1 and for Server 2 is SIOC 2. Auxiliary SIOC for server 1 is SIOC 2 and for server 2 is SIOC 1.

Figure 4: Single Server Single SIOC with Server SIOC Connectivity Functionality

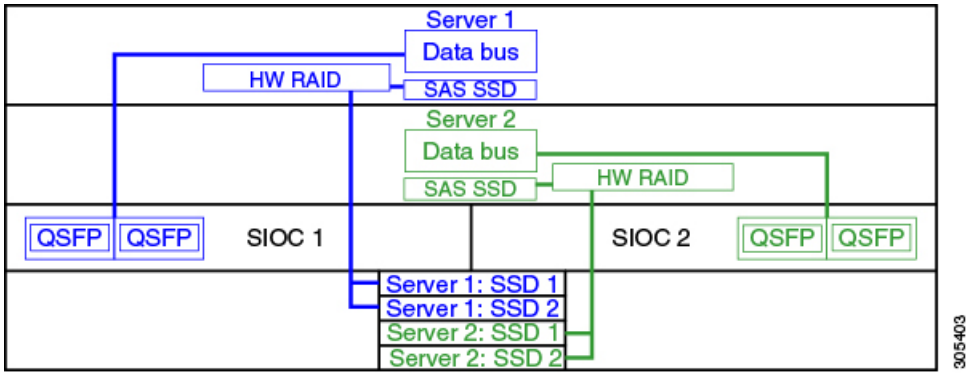


Dual Server Connectivity

In this mode of deployment, each server slot contains an independent server blade. The redundant server nodes along with the various components such as SAS SSDs provide high availability.

The following illustration shows a dual server system. For Cisco UCS C3X60 M3 server nodes, the PCH controller for Server 1 controls SSD1 and SSD2, and the PCH controller for Server 2 controls SSD3 and SSD4. For Cisco UCS C3X60 M4 server nodes, the RAID controller card on the servers controls the respective SSDs.

Figure 5: Dual Server System



Important For detailed information on storage controller considerations for a Cisco UCS S3260 system such as storage controllers supported for the various server nodes and the associated service notes, please refer to the "Storage Controllers" section in the *Cisco UCS S3260 Storage Server Installation and Service Guide.*

Management Through Cisco UCS Manager

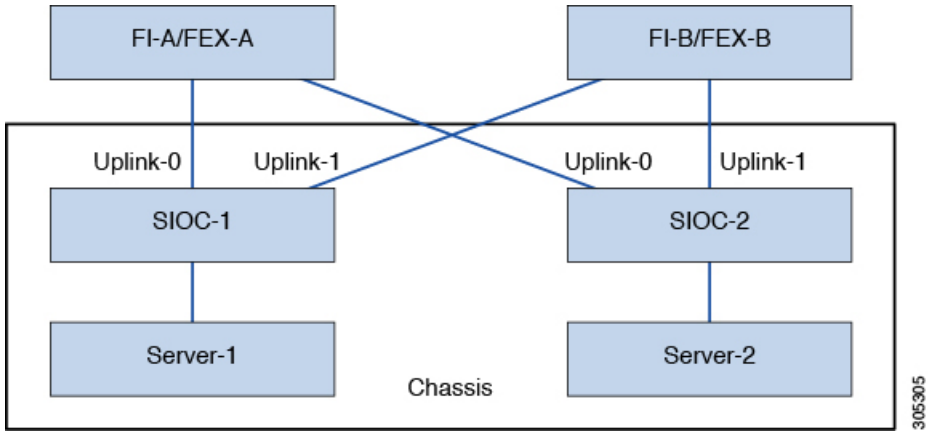
The Cisco UCS S3260 system can operate in either standalone mode or can be managed by Cisco UCS Manager.

Note UCS mini 6324 does not support Cisco UCS Manager integration of Cisco UCS S3260.

Dual Server Connectivity

The following illustration describes the connectivity for a Cisco UCS S3260 system with dual servers managed by Cisco UCS Manager:

Figure 6: Cisco UCS S3260 System with Cisco UCS Manager

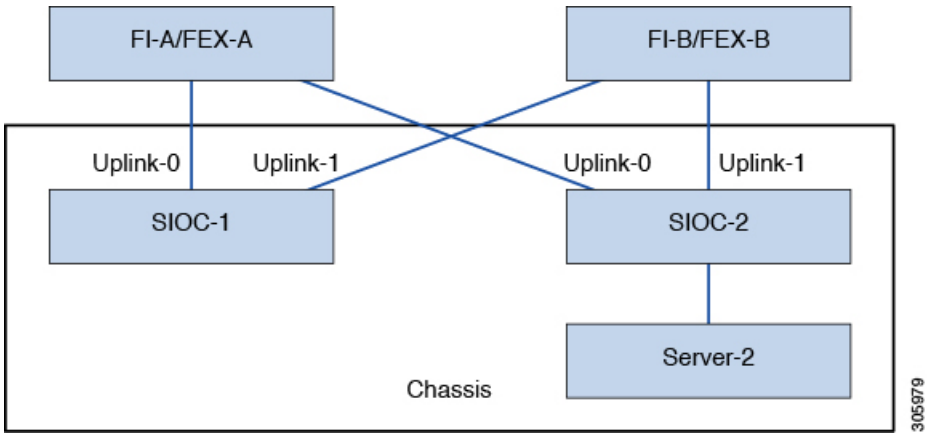


The 40G ports on the SIOCs can be connected to either a fabric interconnects or a FEX module. On each SIOC, one port can be connected to the primary fabric interconnect/FEX and the other port can be connected to the subordinate fabric interconnect/FEX. Traffic from each SIOC can reach fabric interconnects and FEXs.

Single Server and Single SIOC Connectivity

The following illustration describes the connectivity for a Cisco UCS S3260 system with single server and single SIOC managed by Cisco UCS Manager without Server SIOC Connectivity functionality :

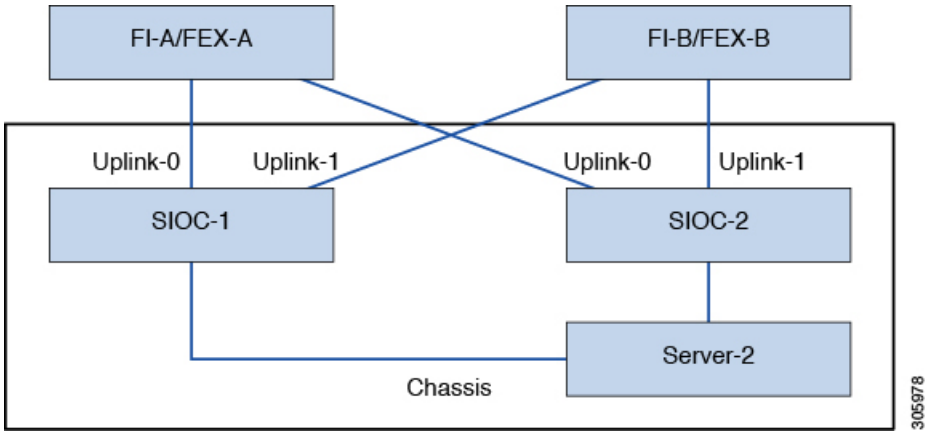
Figure 7: Cisco UCS S3260 System with Cisco UCS Manager (Single Server Single SIOC without Server SIOC Connectivity Functionality)



Single Server and Dual SIOC Connectivity (with Server SIOC Connectivity Functionality)

The following illustration describes the connectivity for a Cisco UCS S3260 system with single server and dual SIOCs managed by Cisco UCS Manager with the Server SIOC Connectivity functionality:

Figure 8: Cisco UCS S3260 System with Cisco UCS Manager (Single Server Single SIOC with Server SIOC Connectivity Feature)



Server SIOC Connectivity Functionality

Beginning with release 3.1(3), Cisco UCS S3260 system supports Server SIOC Connectivity functionality. Using this functionality, you can configure the data path through both the primary and auxiliary SIOCs when the chassis has single server and dual SIOCs set up.

Note Primary SIOC for server 1 is SIOC 1 and for Server 2 is SIOC 2. Auxiliary SIOC for server 1 is SIOC 2 and for server 2 is SIOC 1.

You can configure Server SIOC Connectivity functionality through chassis profile using **Compute Conn Policy** by selecting **single-server-dual-sioc** option.

Prerequisites for Server SIOC Connectivity Functionality

This functionality works only under the following conditions:

- Cisco UCS S3260 system is running release 3.1(3) or later.
- Associated BIOS, BMC and CMC firmware are running 3.1(3) or later.
- Chassis has single server and dual SIOCs.

Workflow - Cisco UCS Manager Upgrade

After Cisco UCS Manager is upgraded to release 3.1(3) or later release, chassis discovery is triggered and UCSM gets the operational state of Server SIOC Connectivity feature. User can now enable the feature using the **single-server-dual-sioc** option available for **Compute Conn Policy** under chassis profile.

Note Any change to Compute Connection Policy settings raises a pending-event. Chassis profile association starts automatically only after you acknowledge the pending-event.

In GUI, once **Compute Conn Policy** property is set to **single-server-dual-sioc**, then Cisco UCS Manager displays a message, warning that this operation causes server reboot. After acknowledging the message, chassis association is triggered. When Server SIOC Connectivity configuration is successfully deployed, Cisco UCS Manager automatically triggers server deep discovery.

In CLI, once **Compute Conn Policy** property is set to **single-server-dual-sioc**, run the **apply pending-changes immediate** command to start association.

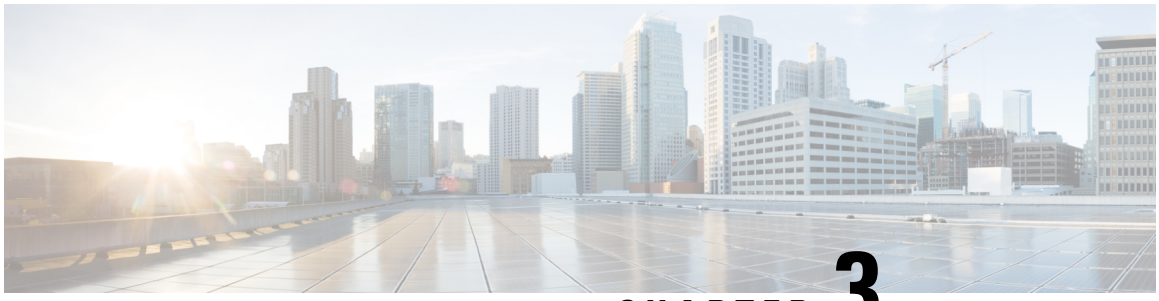
Once **Compute Conn Policy** is set to **single-server-dual-sioc**, you cannot downgrade Cisco UCS Manager to any release earlier than 3.1(3). Similarly, Cisco UCS Manager prevents BMC, CMC, and BIOS downgrade to any release earlier than 3.1(3).

Conditions Impacting the Functionality when single-server-dual-sioc Option is Enabled

- Server Replacement - When the server is replaced, the blade slot mismatch is detected. When you acknowledge the slot, server deep discovery is triggered followed by service profile association. When service profile association is triggered, then there can be the following two situations:
  1. When BIOS/BMC firmware is specified in the host firmware. If the BIOS/BMC firmware support Single Server Dual SIOC connectivity, then the service profile association process continues. If the BIOS/BMC firmware do not support Single Server Dual SIOC connectivity, then the association raises a config-issue.
  2. When BIOS/BMC firmware is not specified in the host firmware. Cisco UCS Manager checks if the running BIOS/BMC version support Single Server Dual SIOC connectivity. If the feature is not supported, then a config-issue is raised.
- SIOC Replacement - If the replaced SIOC is running 3.1(3) or later, then a user acknowledgment message is displayed when one of the SIOC is seated. Once you acknowledge SIOC action, then Cisco UCS Manager establishes the connectivity between the FI and the SIOC. In addition to that Cisco UCS Manager re-acknowledges the server that has the data path connectivity through this SIOC. The VNICs configured for the server are also re-acknowledged. See SIOC Removal or Replacement, on page 107 for more information.

If the replaced SIOC is running an earlier firmware version, then Cisco UCS Manager automatically changes the **Server SIOC Connectivity** operational state to **single-server-single-sioc**. You may update the firmware of the replaced SIOC by re-triggering chassis profile association.

- SIOC Removal - When any SIOC is removed, Cisco UCS Manager marks the SIOC and the corresponding adapter unit created under the server as missing.
- Adding Server in Chassis - When a new server is added in the chassis with this functionality enabled, then server discovery fails.
- Chassis/Server Disassociation - Server SIOC Connectivity functionality is not disabled if a server or chassis is disassociated.



CHAPTER 3

Migration to UCSM-Managed Cisco UCS S3260

This chapter includes the following sections:

Migration to UCSM Managed Cisco UCS S3260

Note Direct migration of Cisco UCS C3160 to UCSM managed Cisco UCS S3260 is not supported. First migrate standalone Cisco UCS C3160 to standalone Cisco UCS S3260 and then to UCSM managed Cisco UCS S3260.

Migrating Standalone Cisco UCS C3160 to UCSM Managed Cisco UCS S3260

To migrate standalone Cisco UCS C3160 to UCSM Managed Cisco UCS S3260:

1. Standalone Cisco UCS C3160 to Standalone Cisco UCS Cisco UCS S3260
2. Standalone Cisco UCS Cisco UCS S3260 to UCSM managed Cisco UCS S3260
3. Configure Server Ports Using Cisco UCS Manager

Migrating Standalone Cisco UCS Cisco UCS S3260 to UCSM Managed Cisco UCS S3260

To migrate standalone Cisco UCS Cisco UCS S3260 to UCSM Managed Cisco UCS S3260:

1. Standalone Cisco UCS Cisco UCS S3260 to UCSM managed Cisco UCS S3260
2. Configure Server Ports Using Cisco UCS Manager

Migrating Standalone Cisco UCS C3160 Server to Cisco UCS S3260 Server

To migrate Cisco UCS C3160 server to Cisco UCS S3260 Server, see Upgrading to Cisco UCS S3260 System With C3X60 M4 Server Nodes.

Migrating Standalone 3260 to UCSM Managed 3260

Prerequisites for Migrating Standalone Cisco UCS S3260 to UCSM Managed Cisco UCS S3260

Ensure that the following conditions are met before starting the migrating procedure:

- For M4 server, if the system is running an earlier version, download and run the Cisco UCS S3260 Host Upgrade Utility (HUU) for release 2.0(9) or later. Use the Host Upgrade Utility User Guide for release 2.0(9) or later for instructions on downloading and using the utility, see the instructions in the guide for your release: HUU Guides.
- For M5 server, download and run the Cisco UCS S3260 Host Upgrade Utility (HUU) for release 3.2(3) or later. Use the Host Upgrade Utility User Guide for release 3.2(3) or later for instructions on downloading and using the utility, see the instructions in the guide for your release: HUU Guides.
- Up to five IP addresses, either configured in your DHCP server or manually entered for static IP addresses. See System IP Addresses, on page 20 for more information.
- It is recommended to make a note of the existing system configurations before migrating to UCSM managed Cisco UCS S3260. These configurations can include the following:
  - Server UUID
  - Storage configuration
  - Network configuration
  - Boot policy
  - Number of vNICs
  - vNIC placements
  - MAC addresses
  - MTU

  You can create these configurations again using Cisco UCS Manager after the migration.
- If the system boot volume is created out of chassis HDD, then perform Booting From Chassis HDD, on page 17.

Booting From Chassis HDD

Before you begin

Before migrating to UCSM Managed Cisco UCS S3260, perform this procedure only if the system boot volume is created out of chassis HDD.

Procedure

Step 1 Associate the chassis with a chassis profile in which **Disk Zoning Policy** is set to **Preserve Config**.

For more information, see *Creating a Chassis Profile with the Wizard* for GUI procedure or *Creating a Chassis Profile* for CLI procedure.

Step 2 Within the service profile for the server, create a storage profile with a LUN using the **Prepare Claim Local LUN** option.

Note the name of the LUN. For more information on storage profiles, see http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/ucs-manager/GUI-User-Guides/Storage-Mgmt/3-1/b_UCSM_GUI_Storage_Management_Guide_3_1/b_UCSM_GUI_Storage_Management_Guide_3_1_chapter_010000.html#d1049e1627a1635.

Step 3 After associating the service profile, go to the storage profile on the service profile and select **Reclaim orphaned LUN** and choose the LUN (LUN on the chassis HDDs) for reclaim.

Step 4 In the boot policy, define the local LUN with the same name as noted in step 2.

What to do next

Proceed to Migrating from Standalone Cisco UCS S3260 to UCSM Managed Cisco UCS S3260, on page 17.

Migrating from Standalone Cisco UCS S3260 to UCSM Managed Cisco UCS S3260

Important If the system is running an earlier version than 2.0(13), perform the following procedure to migrate standalone Cisco UCS S3260 to UCSM managed Cisco UCS S3260.

Procedure

Step 1 Use the Cisco UCS S3260 HUU for 2.0(13) to upgrade the entire system to Cisco IMC release 2.0(13). Run the HUU for all the server nodes in Cisco UCS S3260 system.

Step 2 Shut down and remove power from the entire chassis. You must disconnect all power cords to completely remove power.

Step 3 Connect a keyboard and monitor to the system:

1. Connect a KVM cable (Cisco PID N20-BKVM) to the external KVM connector on the server node at the rear of the system.
2. Connect a VGA monitor and a USB keyboard to the connectors on the KVM cable.

Step 4 Connect power cords and then power on the system. Watch for the prompt to press F8.

Step 5 When prompted, press **F8** to enter the Cisco IMC Configuration Utility.

Step 6 Configure the networking properties for your desired IP addresses, NIC mode, and NIC redundancy.

1. Be aware of the Cisco UCS S3260 system requirement to set as many as five IP addresses. See System IP Addresses, on page 20 for more information. At this point in the procedure, the system requires three addresses:
   - One management IP address
   - One CMC address for the each SIOC
   - One BMC address for each server nodes

   Note If you use a DHCP server, the addresses are defined by the DHCP server. If you disable DHCP, you must set your own static management IP addresses and network settings.
2. Make networking settings using the Cisco IMC Configuration Utility, which you opened by pressing F8 during boot. See *Setting Up the System Using the Cisco IMC Configuration Utility* at http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/s/hw/S3260/install/S3260.html.
3. If you want to set static IP addresses for the CMC and BMC controllers, you will be directed to use the Cisco IMC management interface. See *Setting Static CMC and BMC Internal IP Addresses* at http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/s/hw/S3260/install/S3260.html.

Step 7 Upgrade the remaining system components to Cisco IMC release 2.0(13) or later by using the Cisco UCS Host Upgrade Utility.

Use the Host Upgrade Utility User Guide for release 2.0(13) or later for instructions on downloading and using the utility: HUU Guides

After the upgrade completes, Cisco UCS S3260 system is ready for UCSM integration.

Step 8 Watch for the prompt to press F8 and when prompted, press **F8** to enter the Cisco IMC Configuration Utility.

Step 9 Refer Resetting Cisco IMC to Factory Defaults to set the server to factory defaults or perform the following steps to use the utility to set the server controller to factory defaults:

1. Press **F1** to go to the second screen of the utility.
2. Select the **Chassis Controller Configuration** option under Factory Default. **Do not** select **Server Controller Configuration**.
3. Press **F10** to save changes and reboot the BMC.
4. Wait for the reboot to complete (approximately five minutes).

Step 10 Connect the 40G ports on the SIOCs to either a fabric Interconnect or a FEX module.

On each SIOC, one port is attached to FI-A/FEX-A and the other port is attached to FI-B/FEX-B. Traffic from each SIOC can reach both FIs/FEXs. Refer Cisco UCS S3260 System Architectural Overview, on page 6.

What to do next

Configure the server ports. Refer Configuring Server Ports Using Cisco UCS Manager, on page 20.

Migrating from Standalone Cisco UCS S3260 to UCSM Managed Cisco UCS S3260 [2.0(13) or later version]

Important If the system is already running 2.0(13) or later version, perform the following procedure to migrate standalone Cisco UCS S3260 to UCSM managed Cisco UCS S3260.

Procedure

Step 1 Shut down and remove power from the entire chassis. You must disconnect all power cords to completely remove power.

Step 2 Connect a keyboard and monitor to the system:

1. Connect a KVM cable (Cisco PID N20-BKVM) to the external KVM connector on the server node at the rear of the system.
2. Connect a VGA monitor and a USB keyboard to the connectors on the KVM cable.

Step 3 Connect power cords and then power on the system. Watch for the prompt to press F8.

Step 4 When prompted, press **F8** to enter the Cisco IMC Configuration Utility.

Step 5 Refer Resetting Cisco IMC to Factory Defaults to set the server to factory defaults or perform the following steps to use the utility to set the server controller to factory defaults:

1. Press **F1** to go to the second screen of the utility.
2. Select the **Chassis Controller Configuration** option under Factory Default. **Do not** select **Server Controller Configuration**.
3. Press **F10** to save changes and reboot the BMC.
4. Wait for the reboot to complete (approximately five minutes).

Step 6 Connect the 40G ports on the SIOCs to either a fabric Interconnect or a FEX module.

On each SIOC, one port is attached to FI-A/FEX-A and the other port is attached to FI-B/FEX-B. Traffic from each SIOC can reach both FIs/FEXs. Refer Cisco UCS S3260 System Architectural Overview, on page 6.

What to do next

Configure the server ports. Refer Configuring Server Ports Using Cisco UCS Manager, on page 20.

System IP Addresses

A Cisco UCS S3260 system can have up to five IP addresses:

Note All controllers present in the system must have IP addresses assigned in order to communicate with each other. All IP addresses can be assigned by your DHCP server, or you can assign static IP addresses.

- Management IP—This is the overall system virtual IP address. You log into this address when you access the system's Cisco IMC interface through your LAN connection to the active chassis management controller in SIOC 1 or SIOC 2.
- SIOC 1 CMC IP—This is the internal address for the chassis management controller (CMC) in SIOC 1. This address can be assigned by your DHCP server or you can set a static address by using the Cisco IMC interface.
- SIOC 2 CMC IP—This is the internal address for the CMC in SIOC 2 (if installed). This address can be assigned by your DHCP server or you can set a static address by using the Cisco IMC interface.
- Server 1 BMC IP—This is the internal address for the board management controller (BMC) in server node 1. This address can be assigned by your DHCP server or you can set a static address by using the Cisco IMC interface.
- Server 2 BMC IP—This is the internal address for the BMC in server node 2 (if installed). This address can be assigned by your DHCP server or you can set a static address by using the Cisco IMC interface.

Configuring Server Ports Using Cisco UCS Manager

Perform the following procedure to configure Ethernet ports as server ports. After the ports are configured as server ports, the migration process is complete.

Note If the Ethernet ports connected to standalone Cisco UCS S3260 were already configured as appliance ports, then re-configure them as server ports.

Procedure

Step 1 In the **Navigation** pane, click **Equipment**.

Step 2 Expand **Equipment** > **Fabric Interconnects** > ***Fabric_Interconnect_Name*** > **Fixed Module** > **Ethernet Ports**.

Step 3 Click a port under the **Ethernet Ports** node.

Step 4 In the **Work** pane, click the **General** tab.

Step 5 In the **Actions** area, click **Reconfigure**.

Step 6 From the drop-down list, choose **Configure as Server Port**.

What to do next

This completes the migration process. Cisco UCS Manager automatically discovers the S3260 Chassis. Chassis Discovery Policy, on page 23 provides more information about chassis discovery. Server Management, on page 93 provides more information how to use the Cisco UCS Manager to manage the server.

You can recreate configurations that existed in the standalone Cisco UCS S3260 by using Cisco UCS Manager. Cisco UCS Manager configuration guides listed under How to Use This Guide, on page 4 provide detailed information.

Migrating from UCSM Managed Cisco UCS S3260 M4 to UCSM Managed Cisco UCS S3260 M5

Before you begin

Ensure that the following conditions are met before starting the migrating procedure:

- If the system is running an earlier version, download and run the Cisco UCS S3260 Host Upgrade Utility (HUU) for release 3.2(3) or later. Use the Host Upgrade Utility User Guide for release 3.2(3) or later for instructions on downloading and using the utility, see the instructions in the guide for your release: HUU Guides.

Procedure

Step 1 Decommission the Cisco UCS S3260 M4 server from the chassis.

Refer Decommissioning a Server, on page 100.

Step 2 Shut down and remove power from the entire system. You must disconnect all power cords to completely remove power.

Step 3 Remove the existing M4 server node and replace it with a new M5 server node in either server bay.

Step 4 Connect power cords and then power on the system.

What to do next

You must re-acknowledge the slot for Cisco UCS Manager to rediscover the server.

Migrating from UCSM Managed Cisco UCS S3260 to Standalone Cisco UCS S3260

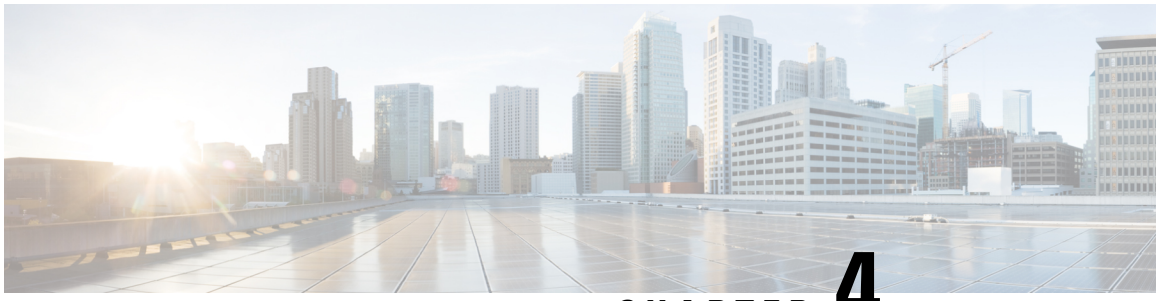
Procedure

Step 1 Decommission the chassis. See *Decommissioning a Chassis* procedure in *Chassis Management* chapter.

Step 2 Disconnect the SIOC cables.

What to do next

This completes the migration process. The chassis can now be used in standalone mode. For further configuration, see the Configuration guides for Cisco UCS S3260 system to manage the server. Cisco UCS S3260 system configuration guides are located at: http://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-c-series-integrated-management-controller/products-installation-and-configuration-guides-list.html.



CHAPTER 4

Equipment Related Policies

Chassis Discovery Policy

The chassis discovery policy determines how the system reacts when you add a new Cisco UCS S3260 chassis or an existing standalone Cisco UCS S3260 chassis to a Cisco UCS system. Cisco UCS S3260 System Architectural Overview, on page 6 describes the connectivity for a Cisco UCS S3260 system managed by Cisco UCS Manager. Cisco UCS Manager uses the settings in the chassis discovery policy to determine whether to group links from the system I/O controllers (SIOCs) to the fabric interconnects in fabric port channels.

To add a previously standalone Cisco UCS S3260 chassis to a Cisco UCS system, you must first configure it to factory default. You can then connect both SIOCs on the chassis to both fabric interconnects. After you connect the SIOCs on the chassis to the fabric interconnects, and mark the ports as server ports, chassis discovery begins.

Server Discovery

Cisco UCS Manager automatically discovers the Cisco UCS S3260 server nodes after the Cisco UCS S3260 chassis is discovered.

Note Server discovery fails if the SIOC corresponding to the server is not present.

Link Grouping

When you connect a Cisco UCS S3260 chassis through a FEX to a Cisco UCS 6200 Series fabric interconnect or directly to a Cisco UCS 6300 Series fabric interconnect, Cisco UCS Manager ignores the **Port Channel** preference and the SIOCs operate in the non-port channel mode.

When you connect a Cisco UCS S3260 chassis directly to a Cisco UCS 6200 Series fabric interconnect, you can use link grouping to determine whether all of the links from the SIOC to the fabric interconnect are grouped into a fabric port channel during chassis discovery. If the link grouping preference is set to **Port Channel**, all of the links from the SIOC to the fabric interconnect are grouped in a fabric port channel. If set to **None**, links from the SIOC are pinned to the fabric interconnect.

Set the link grouping preference to **Port Channel** if the Cisco UCS S3260 chassis is connected to a Cisco UCS 6200 Series fabric interconnect through a 4x10G breakout cable. If this is not done, chassis discovery will not complete.

Set the link grouping preference to **None** if the Cisco UCS S3260 chassis is connected to a fabric interconnect through a single 10G cable.

After changing the **Link Group Preference** value in the Cisco UCS Manager GUI, Decommission and then Recommission the Cisco UCS S3260 chassis for the change to take effect.

In the Cisco UCS domain, if there are other chassis operating in **Port Channel** mode, do the following:

1. Discover the chassis in the Cisco UCS system with the **Link Group Preference** set to **Port Channel**
2. Change the link aggregation preference for the Cisco UCS S3260 chassis through Chassis Connectivity Policy
3. Decommission the chassis
4. Recommission the chassis

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	(Optional) UCS-A /org/chassis-disc-policy # **set descr** *description*	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 4	UCS-A /org/chassis-disc-policy # **set link-aggregation-pref** {**none** | **port-channel**}	Specifies whether the links from the SIOCs or FEXes to the fabric interconnects are grouped into a port channel. Link aggregation can be one of the following: • **none**—links from the SIOC or FEX are pinned to the fabric interconnect.

	Command or Action	Purpose
		• **port-channel**—links from the SIOCs to the fabric interconnects are grouped into a port channel
Step 5	UCS-A /org/chassis-disc-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example scopes to the default chassis discovery policy, 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 descr "This is an example chassis discovery policy."
UCS-A /org/chassis-disc-policy* # set link-aggregation-pref port-channel
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.

Chassis Connectivity Policy

The chassis connectivity policy determines 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.

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.

The following table displays the various connectivity and media type support for FC/FCoE.

Table 2: Connectivity and Media Type Support for FC/FCoE

FI	Connectivity	Supported	Port-Channel	FC/FCoE
Cisco UCS 6454 Fabric Interconnect	Direct Connect– 10 GB – QSA	Yes Not using Breakout	[1]SIOC (Non Port-Channel)	Yes

FI	Connectivity	Supported	Port-Channel	FC/FCoE
Cisco UCS 6454 Fabric Interconnect	Direct Connect– 25G SFP 28	Yes Not using Breakout	SIOC - No	Yes
Cisco UCS 6454 Fabric Interconnect	2232 FEX - 10 GB - QSA	Yes	SIOC (Non Port-Channel)	Yes
63xx	Direct Connect 40 GB	Yes	SIOC (Non Port-Channel)	Yes
63xx	Direct Connect– 10 GB – QSA	Yes Not using Breakout	SIOC - No	No
63xx	Direct Connect 4x10 GB – Breakout	No	N/A	N/A
63xx	Direct Connect 2x10 GB – Reverse Breakout	Yes	SIOC - No	No
63xx	2348UPQ FEX - 10 GB – QSA	Yes	SIOC (Non Port-Channel)	Yes
63xx	2348UPQ FEX - 4x10 GB - Breakout	No	N/A	N/A
62xx	Direct Connect – 4x10 GB - Breakout Cable	Yes	SIOC (Port-Channel)	Yes
62xx	Direct Connect – 10 GB – QSA	Yes	SIOC (Non Port-Channel)	Yes
62xx	2232 FEX - 10 GB - QSA	Yes	SIOC (Non Port-Channel)	Yes
62xx	2232xx - 4x10 GB - Breakout Cable	No	N/A	N/A
6324	Direct Connect - 40 GB	No	N/A	N/A
6324	Direct Connect - 10 GB – QSA	No	N/A	N/A

[1] SIOC without PCIe slots used in release 3.2 and earlier

Configuring a Chassis Connectivity Policy

Caution Changing the connectivity mode for a chassis will require decommissioning and recommissioning the chassis for the change to take effect.

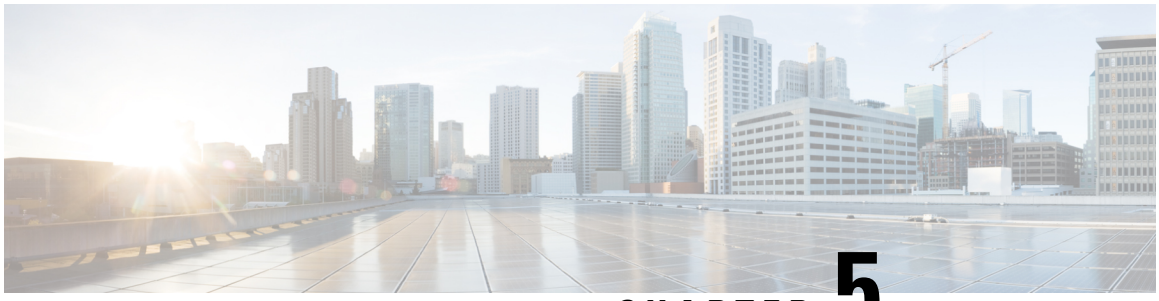
Procedure

	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 how the links from the SIOCs to the fabric interconnects are grouped. This can be one of the following: • **none**—No links are grouped in a port channel. • **port-channel**—All links from an SIOC to a fabric interconnect are grouped in a port channel. • **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.

Example

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 #
```



CHAPTER 5

Chassis Profiles

Chassis Profiles in Cisco UCS Manager

A chassis profile defines the storage, firmware, and maintenance characteristics of a chassis. You can create a chassis profile for the Cisco UCS S3260 chassis. When a chassis profile is associated to a chassis, Cisco UCS Manager automatically configures the chassis to match the configuration specified in the chassis profile.

Important At any given time, each S3260 chassis can be associated with only one chassis profile. Similarly, each chassis profile can be associated with only one S3260 chassis at a time.

A chassis profile includes the following information:

- **Chassis definition**—Defines the specific chassis to which the profile is assigned.
- **Maintenance policy**—Includes the maintenance policy to be applied to the profile.
- **Firmware specifications**—Defines the chassis firmware package that can be applied to a chassis through this profile.
- **Disk zoning policy**—Includes the zoning policy to be applied to the storage disks.
- **Compute Connection policy** — Defines the data path between the primary, auxiliary SIOC, and server.

Guidelines and Recommendations for Chassis Profiles

In addition to any guidelines or recommendations that are specific to the policies included in chassis profiles and chassis profile templates, such as the disk zoning policy, adhere to the following guidelines and recommendations that impact the ability to associate a chassis profile with a chassis:

- Each S3260 chassis can be associated with only one chassis profile. Similarly, each chassis profile can be associated with only one S3260 chassis at a time.
- Chassis profiles are supported only on the Cisco UCS S3260 chassis. The Cisco UCS 5108 blade server chassis does not support chassis profiles and cannot be associated to a chassis profile.
- For Cisco UCS S3260 chassis, S-Series server software bundles earlier than Cisco UCS Manager Release 3.1(2) are not supported.

Creating a Chassis 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 chassis-profile** *profile-name* **instance**	Creates the specified chassis profile instance and enters organization chassis profile mode. Enter a unique *profile-name* to identify this chassis profile. 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	(Optional) UCS-A /org/chassis-profile* # **set descr** *description*	Provides a description for the chassis 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.

	Command or Action	Purpose
Step 4	(Optional) UCS-A /org/chassis-profile* # **set chassisfwpolicyname** *chassis-firmware-policy-name*	Associates the specified chassis firmware policy with the chassis profile.
Step 5	(Optional) UCS-A /org/chassis-profile* # **set chassis-profile-maint-policy** *policy-name*	Associates the specified chassis maintenance policy with the chassis profile.
Step 6	(Optional) UCS-A /org/chassis-profile* # **set user-label** *label-name*	Specifies the user label associated with the chassis profile.
Step 7	(Optional) UCS-A /org/chassis-profile* # **set src-templ-name** *source-chassis-profile-template-name*	Binds the specified chassis profile template with the chassis profile.
Step 8	(Optional) UCS-A /org/chassis-profile* # **set disk-zoning-policy** *disk-zoning-policy-name*	Associates the specified disk zoning policy with the chassis profile.
Step 9	(Optional) UCS-A /org/chassis-profile* # **set compute-conn-policy** *compute-conn-policy-name*	
Step 10	(Optional) UCS-A /org/chassis-profile* # **set sas-expander-configuration-policy** *sas-expander-configuration-policy-name*	Associates the specified SAS expander configuration policy with the chassis profile.
Step 11	UCS-A /org/chassis-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a chassis profile instance and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create chassis-profile ChassisProfile1 instance
UCS-A /org/chassis-profile* # set  descr "This is a chassis profile example."
UCS-A /org/chassis-profile* # set chassis-profile-maint-policy chassismaintpol4
UCS-A /org/chassis-profile* # set user-label mycplabel
UCS-A /org/chassis-profile* # set chassisfwpolicyname cfp1
UCS-A /org/chassis-profile* # set src-templ-name chassispt1
UCS-A /org/chassis-profile* # set disk-zoning-policy dzp1
UCS-A /org/chassis-profile* # set compute-conn-policy ccp1
UCS-A /org/chassis-profile* # set sas-expander-configuration-policy secp1
UCS-A /org/chassis-profile* # commit-buffer
UCS-A /org/chassis-profile #
```

What to do next

Associate the chassis profile with a S3260 chassis.

Renaming a Chassis Profile

When you rename a chassis profile, the following occurs:

- Event logs and audit logs that reference the previous name for the chassis profile are retained under that name.
- A new audit record is created to log the rename operation.
- All records of faults against the chassis profile under its previous name are transferred to the new chassis profile name.

Note You cannot rename a chassis 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 chassis-profile** *profile-name*	Enters organization chassis profile mode for the specified chassis profile.
Step 3	UCS-A /org/chassis-profile # **rename-to** *new-profile-name*	Renames the specified chassis profile. 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 enter this command, you are warned that this is a standalone operation and that you may lose all uncommitted changes in this CLI session. Type **yes** to confirm that you want to continue.

Example

This example shows how to change the name of a chassis profile from CP5 to CP10 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # scope chassis-profile CP5
UCS-A /org/chassis-profile # rename-to CP10
Rename is a standalone operation. You may lose any uncommitted changes in this CLI session.
Do you want to continue? (yes/no): yes
```

```
The managed object in the current mode no longer exists. Changing to mode: /org
UCS-A /org #
```

Deleting a Chassis Profile

This procedure explains how to delete a chassis 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 chassis-profile** *profile-name*	Deletes the specified chassis profile.
Step 3	UCS-A /org* # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to delete a chassis profile ChasInst90 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org delete chassis-profile ChasInst90
UCS-A /org* # commit-buffer
UCS-A /org #
```

Chassis Profile Association

Associating a Chassis Profile with a Chassis

Follow this procedure if you did not associate the chassis profile with a chassis when you created it, or to change the chassis with which a chassis 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 chassis-profile** *profile-name*	Enters organization chassis profile mode for the specified chassis profile.

	Command or Action	Purpose
Step 3	UCS-A /org/chassis-profile # **associate chassis** *chassis-id* [**restrict-migration**]	Associates the chassis profile with a single chassis. Adding the optional **restrict-migration** keyword prevents the chassis profile from being migrated to another chassis.
Step 4	UCS-A /org/chassis-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example associates the chassis profile named ChassisProf1 with chassis 1, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope chassis-profile ChassisProf1
UCS-A /org/chassis-profile # associate chassis 1
UCS-A /org/chassis-profile* # commit-buffer
UCS-A /org/chassis-profile #
```

Disassociating a Chassis Profile from a Chassis

This procedure covers disassociating a chassis profile from a chassis.

Note When a chassis is disassociated from a chassis profile, effects of disk zoning policy will be still be persistent in the chassis.

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 chassis-profile** *profile-name*	Enters organization chassis profile mode for the specified chassis profile.
Step 3	UCS-A /org/chassis-profile # **disassociate**	Disassociates the chassis profile from the chassis.
Step 4	UCS-A /org/chassis-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example disassociates the chassis profile named ChassisProf1 from the chassis to which it was associated and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # scope chassis-profile ChassisProf1
UCS-A /org/chassis-profile # disassociate
UCS-A /org/chassis-profile* # commit-buffer
UCS-A /org/chassis-profile #
```

Chassis Profile Template

Creating a Chassis 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 chassis-profile** *profile-name* {**initial-template** | **updating-template**}	Creates the specified chassis profile template and enters organization chassis profile mode. Enter a unique *profile-name* to identify this chassis profile template. 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. Chassis profile template types are: • **initial-template**—Instances will not automatically update if this template is updated. • **updating-template**—Instances will automatically update if this template is updated.
Step 3	(Optional) UCS-A /org/chassis-profile* # **set descr** *description*	Provides a description for the chassis profile template.

	Command or Action	Purpose
		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/chassis-profile* # **set chassisfwpolicyname** *chassis-firmware-policy-name*	Associates the specified chassis firmware policy with the chassis profile template.
Step 5	UCS-A /org/chassis-profile* # **set chassis-profile-maint-policy** *policy-name*	Associates the specified chassis maintenance policy with the chassis profile template.
Step 6	UCS-A /org/chassis-profile* # **set user-label** *label-name*	Specifies the user label associated with the chassis profile template.
Step 7	UCS-A /org/chassis-profile* # **set src-templ-name** *source-chassis-profile-template-name*	Binds the specified chassis profile template with the chassis profile.
Step 8	UCS-A /org/chassis-profile* # **set disk-zoning-policy** *disk-zoning-policy-name*	Associates the specified disk zoning policy with the chassis profile template.
Step 9	UCS-A /org/chassis-profile* # **set compute-conn-policy** *compute-conn-policy-name*	Associates the specified compute conn policy with the chassis profile template.
Step 10	UCS-A /org/chassis-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a chassis profile template and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create chassis-profile ChassisProTemp1 updating-template
UCS-A /org/chassis-profile* # set  descr "This is a chassis profile template example."
UCS-A /org/chassis-profile* # set chassis-profile-maint-policy chassismaintpol2
UCS-A /org/chassis-profile* # set user-label mycptlabel
UCS-A /org/chassis-profile* # set chassisfwpolicyname cptf1
UCS-A /org/chassis-profile* # set src-templ-name chassispt1
UCS-A /org/chassis-profile* # set disk-zoning-policy dzp1
UCS-A /org/chassis-profile* # set compute-conn-policy ccp1
UCS-A /org/chassis-profile* # commit-buffer
UCS-A /org/chassis-profile #
```

What to do next

Create a chassis profile instance from the chassis profile template.

Creating a Chassis Profile Instance from a Chassis Profile Template

Before you begin

Verify that there is a chassis profile template from which to create a chassis profile instance.

Procedure

	Command or Action	Purpose
Step 1	UCSC(resource-mgr)# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 2	UCSC(resource-mgr) /org # **create chassis-profile** *profile-name* **instance**	Creates the specified chassis profile instance and enters organization chassis profile mode. Enter a unique *profile-name* to identify this chassis profile. 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	UCSC(resource-mgr) /org/chassis-profile* # **set src-templ-name** *profile-name*	Specifies the source chassis profile template to apply to the chassis profile instance. All configuration settings from the chassis profile template will be applied to the chassis profile instance.
Step 4	UCSC(resource-mgr) /org/chassis-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a chassis profile instance named ChassisProf02, applies the chassis profile template named ChassisProfTemp2, and commits the transaction:

```
UCSC(resource-mgr)# scope org /
UCSC(resource-mgr) /org* # create chassis-profile ChassisProf02 instance
UCSC(resource-mgr) /org/chassis-profile* # set src-templ-name ChassisProfTemp2
UCSC(resource-mgr) /org/chassis-profile* # commit-buffer
UCSC(resource-mgr) /org/chassis-profile #
```

What to do next

Associate the chassis profile to a chassis.

Binding a Chassis Profile to a Chassis Profile Template

You can bind a chassis profile to a chassis profile template. When you bind the chassis profile to a template, Cisco UCS Manager configures the chassis profile with the values defined in the chassis profile template. If the existing chassis profile configuration does not match the template, Cisco UCS Manager reconfigures the chassis profile. You can only change the configuration of a bound chassis profile through the associated 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 chassis-profile** *profile-name*	Enters organization chassis profile mode for the specified chassis profile.
Step 3	UCS-A /org/chassis-profile # **set src-templ-name***chassis-profile-template-name*	Binds the chassis profile to the specified chassis profile template.
Step 4	UCS-A /org/chassis-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example binds the chassis profile named ChassisProf1 to ChassisProfileTemplate1 and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope chassis-profile ChassisProf1
UCS-A /org/chassis-profile # set src-templ-name ChassisProfileTemplate1
UCS-A /org/chassis-profile* # commit-buffer
UCS-A /org/chassis-profile #
```

Unbinding a Chassis Profile from a Chassis Profile Template

To unbind a chassis profile from a chassis profile template, bind the chassis profile to an empty value (quotes without space).

Procedure

	Command or Action	Purpose
Step 1	UCSC# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 2	UCSC /org # **scope chassis-profile** *profile-name*	Enters organization chassis profile mode for the specified chassis profile.

	Command or Action	Purpose
Step 3	UCSC /org/chassis-profile # **set src-templ-name ""**	Unbinds the chassis profile from the chassis profile template.
Step 4	UCSC /org/chassis-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example unbinds the chassis profile named ChassisProf1 and commits the transaction:

```
UCSC# scope org
UCSC /org # scope chassis-profile ChassisProf1
UCSC /org/chassis-profile # set src-templ-name ""
UCSC /org/chassis-profile* # commit-buffer
UCSC /org/chassis-profile #
```

Maintenance Policy

Creating a Chassis Profile 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 # **create chassis-profile-maint-policy** *policy-name*	Creates the specified maintenance policy and enters maintenance policy mode.
Step 3	UCS-A /org/chassis-profile-maint-policy* # **set reboot-policy user-ack**	When a policy is associated with a chassis, the chassis needs to be re-acknowledged to complete the association. The user must explicitly acknowledge the changes by using the **apply pending-changes** command before changes are applied.
Step 4	(Optional) UCS-A /org/chassis-profile-maint-policy* # **set descr** *description*	A description of the policy. Cisco recommends including information about where and when to use the policy.
Step 5	(Optional) UCS-A /org/chassis-profile-maint-policy* # **set policy-owner global | local**	Specifies the owner for the maintenance policy. • global - The ownership of the global policy remains with Cisco UCS Central, and you cannot make any changes to the policy ownership using Cisco UCS Manager. You

	Command or Action	Purpose
		can associate global policies with chassis in one or more registered Cisco UCS domains. • local - You can only associate chassis in the same domain local policies.
Step 6	UCS-A /org/maint-policy #* **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a maintenance policy called maintenance, and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # create chassis-profile-maint-policy maintenance
UCS-A /org/chassis-profile-maint-policy* # set reboot-policy user-ack
UCS-A /org/chassis-profile-maint-policy* # commit-buffer
UCS-A /org/maint-policy #
```

Configuring the Maintenance Policy for a Chassis Profile/Chassis 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 chassis-profile** *profile-name*|*template-name*	Enters organization chassis profile/chassis profile template mode for the specified chassis profile//chassis profile template.
Step 3	UCS-A /org/chassis-profile # **set chassis-profile-maint-policy** *maintenance-policy-name*	Associates the specified maintenance policy with the chassis profile//chassis profile template. Use an existing maintenance policy name or enter a new policy.
Step 4	UCS-A /org/chassis-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to associate a maintenance policy with a chassis profile and commit the transaction:

```
UCS-A# scope org
UCS-A /org # scope chassis-profile ChassisProfile1
UCS-A /org/chassis-profile # set chassis-profile-maint-policy default
UCS-A /org/chassis-profile* # commit-buffer
UCS-A /org/chassis-profile #
```

Compute Connection Policy

Compute Connection Policy is used to store the user configuration of the server SIOC connectivity. The configuration settings are done using the property **Server SIOC Connectivity**, which can be set to:

- **single-server-single-sioc** (default) - The data path is configured through one SIOC when the chassis has single server and single SIOC or dual server and dual SIOCs.
- **single-server-dual-sioc** - When enabled, you can configure the data path through both the primary and auxiliary SIOCs when the chassis has single server and dual SIOCs. See Server SIOC Connectivity Functionality, on page 11 for more details.

Creating Compute Conn Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters the root organization mode.
Step 2	UCS-A /org # **create compute-conn-policy** *compute-conn-policy-name*	Creates the specified compute conn policy.
Step 3	(Optional) UCS-A /org/compute-conn-policy* # **set descr** *description*	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/compute-conn-policy* # **set server-sioc-connectivity {single-server-dual-sioc | single-server-single-sioc**	• **single-server-single-sioc (default)** - The data path is configured through one SIOC when the chassis has single server and single SIOC or dual server and dual SIOCs. • **single-server-dual-sioc** - When enabled, you can configure the data path through both the primary and auxiliary SIOCs when the chassis has single server and dual

	Command or Action	Purpose
		SIOCs. See Server SIOC Connectivity Functionality, on page 11 for more details.
Step 5	UCS-A /org/compute-conn-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

```
UCS-A# scope org
UCS-A /org # create compute-conn-policy ccptest
UCS-A /org/compute-conn-policy* # set descr "This is an example policy."
UCS-A /org/compute-conn-policy* # set server-sioc-connectivity single-server-dual-sioc
UCS-A /org/compute-conn-policy* # commit-buffer
UCS-A /org/compute-conn-policy #
```

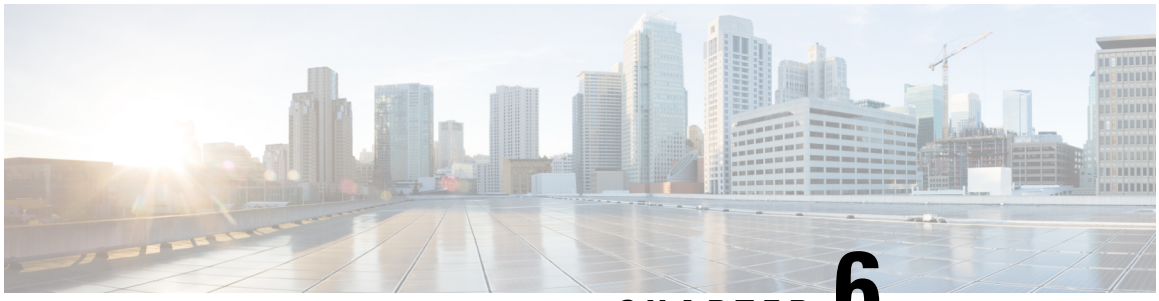
Associating a Compute Conn Policy to Chassis Profile

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters the root organization mode.
Step 2	UCS-A /org# **scope chassis_profile** *chassis-profile-name*	Enters the chassis profile mode.
Step 3	UCS-A /org/chassis-profile# **set compute-conn-policy** *compute-conn-policy-name*	Associates the specified compute conn policy to the chassis profile.
Step 4	UCS-A /org/chassis-profile# **commit-buffer**	Commits the transaction to the system configuration.
Step 5	UCS-A /org/chassis-profile# **apply pending-changes immediate**	Acknowledges the pending changes and triggers profile association. **Note** Any change to Compute Connection Policy settings raises a pending-event. Chassis profile association starts automatically only after you acknowledge the pending-event.
Step 6	UCS-A /org/chassis-profile# **commit-buffer**	Commits the transaction to the system configuration.

Example

```
UCS-A# scope org
UCS-A /org # scope chassis-profile chassisprofile1
UCS-A /org/chassis-profile # set compute-conn-policy compconpolicy1
UCS-A /org/chassis-profile* # commit-buffer
UCS-A /org/chassis-profile # apply pending-changes immediate
UCS-A /org/chassis-profile* # commit-buffer
```



CHAPTER 6

Cisco UCS S3260 System Storage Management

Storage Server Features and Components Overview

Storage Server Features

The following table summarizes the Cisco UCS S3260 system features:

Table 3: Cisco UCS S3260 System Features

Feature	Description
Chassis	Four rack unit (4RU) chassis
Processors	• Cisco UCS S3260 M3 server nodes: Two Intel Xeon E5-2600 v2 Series processors inside each server node. • Cisco UCS S3260 M4 server nodes: Two Intel Xeon E5-2600 v4 Series processors inside each server node. • Cisco UCS S3260 M5 server nodes: Two Intel Skylake 2S-EP processors inside each server node.
Memory	Up to 16 DIMMs inside each server node.
Multi-bit error protection	This system supports multi-bit error protection.

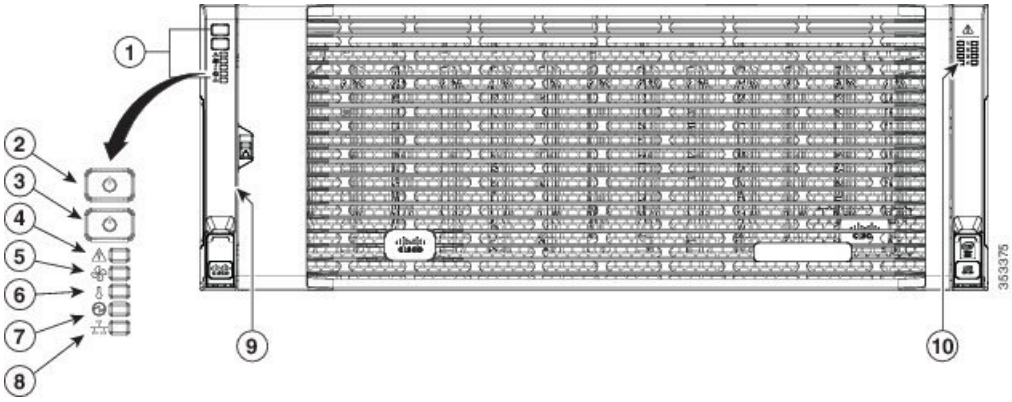
Feature	Description
Storage	The system has the following storage options: • Up to 56 top-loading 3.5-inch drives • Up to four 3.5-inch, rear-loading drives in the optional drive expander module • Up to four 2.5-inch, rear-loading SAS solid state drives (SSDs) • One 2.5-inch, NVMe drive inside the server node **Note** This is applicable for S3260 M4 servers only. • Two 7 mm NVMe drive inside the server node **Note** This is applicable for S3260 M5 servers only. • Two 15 mm NVMe drive supported for IO Expander
Disk Management	The system supports up to two storage controllers: • One dedicated mezzanine-style socket for a Cisco storage controller card inside each server node
RAID Backup	The supercap power module (SCPM) mounts to the RAID controller card.
PCIe I/O	The optional I/O expander provides two 8x Gen 3 PCIe expansion slots. Release 3.2(3) and later supports the following for S3260 M5 servers: • Intel X550 dual-port 10GBase-T • Qlogic QLE2692 dual-port 16G Fiber Channel HBA • N2XX-AIPCI01 Intel X520 Dual Port 10Gb SFP+ Adapter

Feature	Description
Network and Management I/O	The system can have one or two system I/O controllers (SIOCs). These provide rear-panel management and data connectivity. • Two SFP+ 40 Gb ports each SIOC. • One 10/100/1000 Ethernet dedicated management port on each SIOC. The server nodes each have one rear-panel KVM connector that can be used with a KVM cable, which provides two USB, one VGA DB-15, and one serial DB-9 connector.
Power	Two or four power supplies, 1050 W each (hot-swappable and redundant as 2+2).
Cooling	Four internal fan modules that pull front-to-rear cooling, hot-swappable. Each fan module contains two fans. In addition, there is one fan in each power supply.

Front Panel Features

The following image shows the front panel features for the Cisco UCS S3260 system:

Figure 9: Front Panel Features



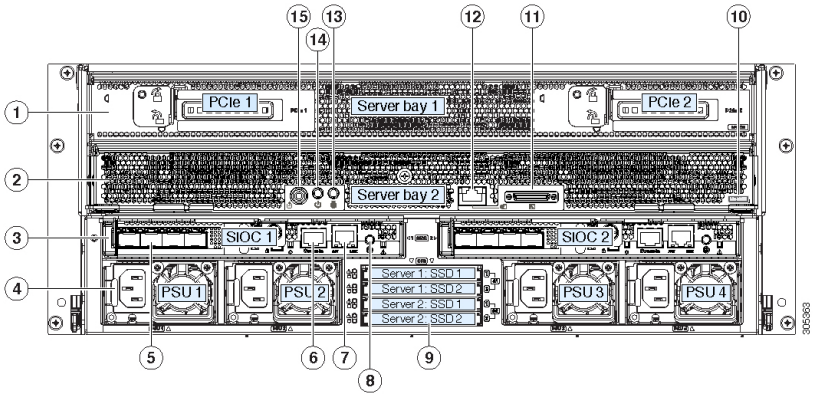
1	Operations panel	6	Temperature status LED
2	System Power button/LED	7	Power supply status LED
3	System unit identification button/LED	8	Network link activity LED

4	System status LED	9	Pull-out asset tag (not visible under front bezel)
5	Fan status LED	10	Internal-drive status LEDs

Rear Panel Features

The following image shows the rear panel features for the Cisco UCS S3260 system:

Figure 10: Front Panel Features



Disk Slots

1	Server bay 1 • (Optional) I/O expander, as shown (with Cisco UCS S3260 M4 and M5 server node only) • (Optional) server node • (Optional) drive expansion module	8	Not used at this time

2	Server bay 2 • (Optional) server node (Cisco UCS S3260 M4 and M5 shown) (Optional) drive expansion module	9	Not used at this time
3	System I/O controller (SIOC) • SIOC 1 is required if you have a server node in server bay 1 • SIOC 2 is required if you have server node in server bay 2	10	Solid state drive bays (up to four 2.5-inch SAS SSDs) • SSDs in bays 1 and 2 require a server node in server bay 1 • SSDs in bays 3 and 4 require a server node in server bay 2
4	Power supplies (four, redundant as 2+2)	11	Cisco UCS S3260 M4 server node label (M4 SVRN) **Note** This label identifies a Cisco UCS S3260 M4 and M5 server node. The Cisco UCS S3260 M3 server node does not have a label.
5	40-Gb SFP+ ports (two on each SIOC)	12	KVM console connector (one each server node). Used with a KVM cable that provides two USB, one VGA, and one serial connector
6	Chassis Management Controller (CMS) Debug Firmware Utility port (one each SIOC)	13	Server node unit identification button/LED

7	10/100/1000 dedicated management port, RJ-45 connector (one each SIOC)	14	Server node power button
		15	Server node reset button (resets chipset in the server node

Storage Server Components

Server Nodes

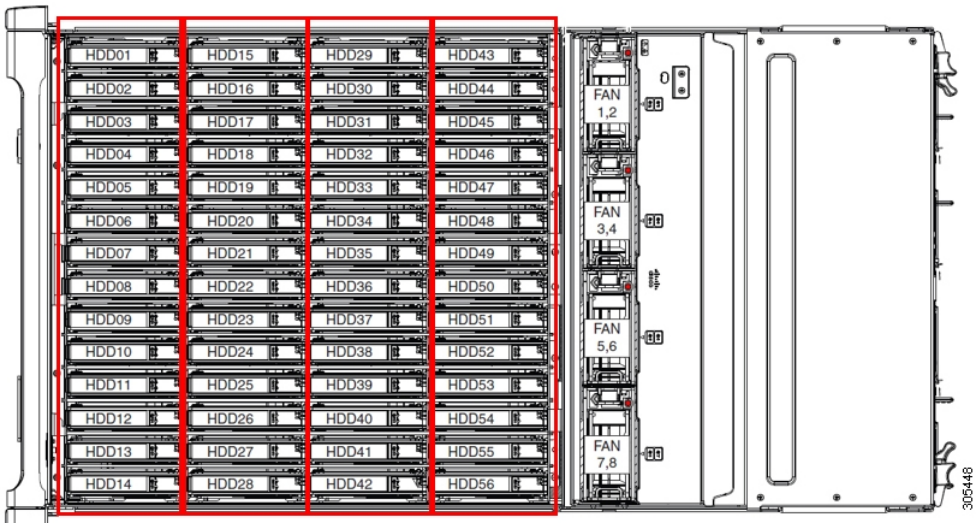
The Cisco UCS S3260 system consists of one or two server nodes, each with two CPUs, DIMM memory of 128, 256, or 512 GB, and a RAID card up to 4 GB cache or a pass-through controller. The server nodes can be one of the following:

- Cisco UCS S3260 M3 Server Node
- Cisco UCS S3260 M4 Server Node—This node might include an optional I/O expander module that attaches to the top of the server node.
- Cisco UCS S3260 M5 Server Node—This node might include an optional I/O expander module that attaches to the top of the server node.

Disk Slots

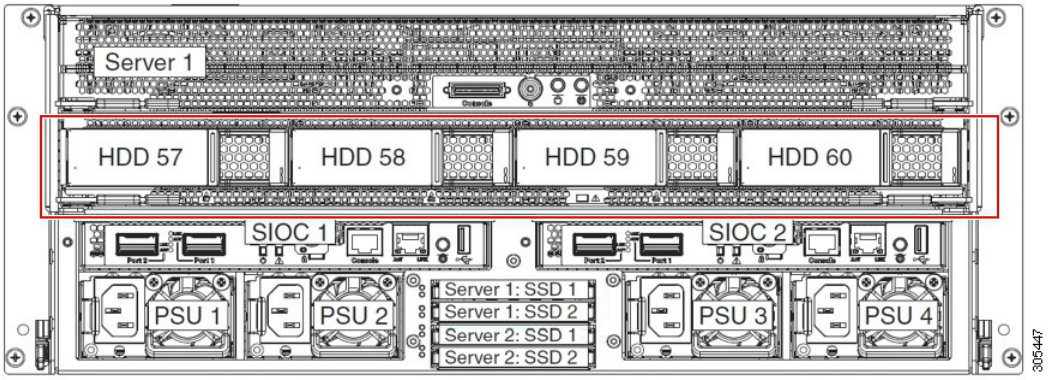
The Cisco UCS S3260 chassis has 4 rows of 14 disk slots on the HDD motherboard and 4 additional disk slots on the HDD expansion tray. The following image shows the disk arrangement for the 56 top-accessible, hot swappable 3.5-inch 6 TB or 4 TB 7200 rpm NL-SAS HDD drives. A disk slot has two SAS ports and each is connected a SAS expander in the chassis.

Figure 11: Cisco UCS S3260 Top View



The following image shows the Cisco UCS S3260 chassis with the 4 additional disk slots on the HDD expansion tray.

Figure 12: Cisco UCS 3260 with the HDD expansion tray (Rear View)



If you have two server nodes with two SIOCs, you will have the following functionality:

1. The top server node works with the left SIOC (Server Slot1 with SIOC1).
2. The bottom server works with the right SIOC (Sever Slot 2 with SIOC2).

If you have one server node with two SIOCs, you can enable Server SIOC Connectivity functionality. Beginning with release 3.1(3), Cisco UCS S3260 system supports Server SIOC Connectivity functionality. Using this functionality, you can configure the data path through both the primary and auxiliary SIOCs when the chassis has single server and dual SIOCs set up.

SAS Expanders

The Cisco UCS S3260 system has two SAS expanders that run in redundant mode and connect the disks at the chassis level to storage controllers on the servers. The SAS expanders provide two paths between a storage controller, and hence enable high availability. They provide the following functionality:

- Manage the pool of hard drives.
- Disk zone configuration of the hard drives to storage controllers on the servers.

Beginning with release 3.2(3a), Cisco UCS Manager can enable single path access to disk by configuring single DiskPort per disk slot. This ensures that the server discovers only a single device and avoid a multi-path configuration.

The following table describes how the ports in each SAS expander are connected to the disks based on the type of deployment.

Port range	Connectivity
1-56	Top accessible disks
57-60	Disks in the HDD expansion tray.

Note The number of SAS uplinks between storage controller and SAS expander can vary based on the type of controller equipped in the server.

Storage Enclosures

A Cisco UCS S3260 system has the following types of storage enclosures:

Chassis Level Storage Enclosures

- **HDD motherboard enclosure**—The 56 dual port disk slots in the chassis comprise the HDD motherboard enclosure.
- **HDD expansion tray**—The 4 additional dual disk slots in the Cisco UCS S3260 system comprise the HDD expansion tray.

Note The HDD expansion tray is a field replaceable unit (FRU). The disks will remain unassigned upon insertion, and can be assigned to storage controllers. For detailed steps on how to perform disk zoning, see Disk Zoning Policies, on page 54

Server level Storage Enclosures

Server level storage enclosures are pre-assigned dedicated enclosures to the server. These can be one of the following:

- **Rear Boot SSD enclosure**—This enclosure contains two 2.5 inch disk slots on the rear panel of the Cisco UCS S3260 system. Each server has two dedicated disk slots. These disk slots support SATA SSDs.
- **Server board NVMe enclosure**—This enclosure contains one PCIe NVMe controller.

Note In the Cisco UCS S3260 system, even though disks can be physically present on the two types of enclosures described above, from the host OS all the disks are viewed as part of one SCSI enclosure. They are connected to SAS expanders that are configured to run as single SES enclosure.

Storage Controllers

Mezzanine Storage Controllers

The following table lists the storage controller type, firmware type, modes, sharing and OOB support for the various storage controllers.

Table 4:

Storage Controller Type	Firmware type	Modes	Sharing	OOB Support
UCSC-S3X60-R1GB	Mega RAID	HW RAID, JBOD	No	Yes
UCS-C3K-M4RAID	Mega RAID	HW RAID, JBOD	No	Yes
UCSC-S3X60-HBA	Initiator Target	Pass through	Yes	Yes
UCS-S3260-DHBA	Initiator Target	Pass through	Yes	Yes
UCS-S3260-DRAID	Mega RAID	HW RAID, JBOD	No	Yes

Other storage controllers

SW RAID Controller—The servers in the Cisco UCS S3260 system support two dedicated internal SSDs embedded into the PCIe riser that is connected to the SW RAID Controller. This controller is supported on the Cisco C3000 M3 servers.

NVMe Controller—This controller is used by servers in the Cisco UCS S3260 system for inventory and firmware updates of NVMe disks.

For more details about the storage controllers supported in the various server nodes, see the related service note:

- Cisco UCS C3X60 M3 Server Node For Cisco UCS S3260 Storage Server Service Note
- Cisco UCS C3X60 M4 Server Node For Cisco UCS S3260 Storage Server Service Note
- Cisco UCS S3260 M5 Server Node For Cisco UCS S3260 Storage Server Service Note

Cisco UCS S3260 Storage Management Operations

The following table summarizes the various storage management operations that you can perform with the Cisco UCS Manager integrated Cisco UCS S3260 system.

Operation	Description	See:
Disk Sharing for High Availability	The SAS expanders in the Cisco UCS S3260 system can manage the pool of drives at the chassis level. To share disks for high availability, perform the following: 1. Creating disk zoning policies. 2. Creating disk slots and assigning ownership. 3. Associating disks to chassis profile.	"Disk Zoning Policies" section in this guide.
Storage Profiles, Disk Groups and Disk Group Configuration Policies	You can utilize Cisco UCS Manager's Storage Profile and Disk Group Policies for defining storage disks, disk allocation and management in the Cisco UCS S3260 system.	"Storage Profiles" section in the *Cisco UCS Manager Storage Management Guide, Release 3.2*.
Storage Enclosure Operations	You can swap the HDD expansion tray with a server, or remove the tray if it was previously inserted.	"Removing Chassis Level Storage Enclosures" section in this guide.

Disk Sharing for High Availability

Disk Zoning Policies

You can assign disk drives to the server nodes using disk zoning. Disk zoning can be performed on the controllers in the same server or on the controllers on different servers. Disk ownership can be one of the following:

Unassigned

Unassigned disks are those not visible to the server nodes.

Dedicated

If this option is selected, you will need to set the values for the **Server**, **Controller**, **Drive Path**, and **Slot Range** for the disk slot.

Note A disk is visible only to the assigned controller.

Beginning with release 3.2(3a), Cisco UCS Manager can enable single path access to disk by configuring single DiskPort per disk slot for Cisco UCS S3260 M5 and higher servers. Setting single path configuration ensures that the server discovers the disk drive only through a single drive path chosen in the configuration. Single path access is supported only for **Cisco UCS S3260 Dual Pass Through Controller** (UCS-S3260-DHBA)

Once single path access is enabled, you cannot downgrade to any release earlier than 3.2(3a). To downgrade, disable this feature and assign all the disk slots to both the disk ports by configuring disk path of the disk slots to **Path Both** in disk zoning policy.

Shared

Shared disks are those assigned to more than one controller. They are specifically used when the servers are running in a cluster configuration, and each server has its storage controllers in HBA mode.

Note Shared mode cannot be used under certain conditions when dual HBA controllers are used.

Chassis Global Hot Spare

If this option is selected, you will need to set the value for the **Slot Range** for the disk.

Important Disk migration and claiming orphan LUNs: To migrate a disk zoned to a server (Server 1) to another server (Server 2), you must mark the virtual drive (LUN) as transport ready or perform a hide virtual drive operation. You can then change the disk zoning policy assigned for that disk. For more information on virtual drive management, see the *Disk Groups and Disk Configuration Policies* section of the Cisco UCS Manager Storage Management Guide.

Creating a Disk Zoning 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 disk-zoning-policy** *diskzoning policy-name*	Creates a disk zoning policy name with the specified disk zoning policy name.
Step 3	UCS-A /org/disk-zoning-policy* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates the dzp1 disk zoning policy:

```
UCS-A# scope org
UCS-A /org # create disk-zoning-policy dzp1
UCS-A /org/disk-zoning-policy*# commit-buffer
UCS-A /org/disk-zoning-policy#
```

Creating Disk Slots and Assigning Ownership

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/ # **disk-zoning-policy** *disk-zoning-policy-name*	Enters the disk zoning policy.
Step 3	UCS-A org/disk-zoning-policy # **create disk-slot** *slot-id*	Creates disk slot with the specified slot number.
Step 4	UCS-A org/disk-zoning-policy/disk-slot* # **set ownership** *ownership-type {chassis-global-host-spare|dedicated|shared|unassigned}*	Specifies the disk ownership to be one of the following: • **chassis-global-hot-spare**—Chassis Global Hot Spare • **dedicated**—Dedicated Beginning with release 3.2(3a), Cisco UCS Manager can enable single path access to disk by configuring single DiskPort per

	Command or Action	Purpose
		disk slot. This ensures that the server discovers only a single device and avoid a multi-path configuration. Drive Path options are: • **path-both (Default)** - Drive path is zoned to both the SAS expanders. • **path-0** - Drive path is zoned to SAS expander 1. • **path-1** - Drive path is zoned to SAS expander 2. Use the following command to set the drivepath: `set drivepath drivepath{path-0/path-1/path-both}` • **shared**—Shared **Note** Shared mode cannot be used under certain conditions when dual HBA controllers are used. To view the conditions for **Shared** mode for Dual HBA controller, see Table 5: Limitations for Shared Mode for Dual HBA Controller, on page 56. • **unassigned**—Unassigned
Step 5	UCS-A org/disk-zoning-policy/disk-slot* # **create controller-ref** *server-id* **sas** *controller-id*	Creates controller reference for the specified server slot.
Step 6	UCS-A org/disk-zoning-policy/disk-slot # **commit-buffer**	Commits the transaction.

Table 5: Limitations for Shared Mode for Dual HBA Controller

Server	HDD Tray	Controller	Shared mode Support
Cisco UCS S3260	No	Dual HBA	Not Supported
Cisco UCS S3260	HDD Tray	Dual HBA	Not Supported
Pre-Provisioned	HDD Tray	Dual HBA	Not Supported

Example

The following example creates disk slot 1, sets the ownership as shared, creates a controller reference for the server slot 1, and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope disk-zoning-policy test
UCS-A /org/disk-zoning-policy* # create disk-slot 1
UCS-A /org/disk-zoning-policy/disk-slot* # set ownership shared
UCS-A /org/disk-zoning-policy/disk-slot* # create controller-ref 1 sas 1
UCS-A /org/disk-zoning-policy/disk-slot* # create controller-ref 2 sas 1
UCS-A /org/disk-zoning-policy/disk-slot* #commit-buffer
UCS-A /org/disk-zoning-policy/disk-slot #
```

Associating Disk Zoning Policies to Chassis 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 chassis-profile** *chassis-profile-name*	Creates a chassis profile with the specified name.
Step 3	UCS-A org/chassis-profile* # **set disk-zoning-policy** *disk-zoning-policy*	Sets the specified disk-zoning-policy.
Step 4	UCS-A org/chassis-profile* # **commit-buffer**	Commits the transaction.
Step 5	UCS-A org/chassis-profile # **associate chassis** *chassis-id*	Associates the disks in the disk zoning policy to the chassis with the specified chassis number.

Example

The following example creates the ch1 chassis profile, sets the disk zoning policy all56shared, commits the transaction and associates the disk in the all56shared policy with chassis 3:

```
UCS-A# scope org
UCS-A /org # create chassis-profile ch1
UCS-A /org/chassis-profile* # set disk-zoning-policy all56shared
UCS-A /org/chassis-profile* # commit-buffer
UCS-A /org/chassis-profile # associate chassis 3
UCS-A /org/fw-chassis-pack/pack-image #
```

Disk Migration

Before you can migrate a disk zoned from one server to another, you must mark the virtual drive(LUN) as transport ready or perform a hide virtual drive operation. This will ensure that all references from the service profile have been removed prior to disk migration. For more information on virtual drives, please refer to the "virtual drives" section in the Cisco UCS Manager Storage Management Guide, Release 3.2.

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 virtual-drive-container** *virtual-drive-container-num*	Enters the virtual drive container with the specified number.
Step 3	UCS-A /chassis/virtual-drive-container# **scope virtual-drive** *virtual-drive--num*	Enters the virtual drive for the specified virtual drive container.
Step 4	UCS-A /chassis/virtual-drive-container/virtual-drive# **scope virtual-drive** *virtual-drive--num* **set admin-state** *admin-state*	Specifies one of the following admin states for the virtual drive: • **clear-transport-ready** — Sets the state of the virtual drive to no longer be transport ready. • **delete** — Deletes the virtual drive. • **hide**— Choose this option for the safe migration of the virtual drive from one server to another. **Note** All virtual drives on a disk group must be marked as hidden before migrating or unassigning the disks from a server node. • **transport-ready** — Choose this option for the safe migration of the virtual drive from one server to another. **Note** When a virtual drive is marked as transport ready, the storage controller will disable all IO operations on the drive. In addition, after zoning the virtual drive and importing the foreign configuration, the virtual drive will be operational.

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

Example

The following example sets the state of the virtual drive 1001 in the virtual drive container 1 to transport ready:

```
UCS-A# scope chassis
UCS-A /chassis# scope virtual-drive-container 1
UCS-A /chassis/virtual-drive-container# scope virtual-drive 1001
UCS-A /chassis/virtual-drive-container/virtual-drive# set admin-state transport-ready
UCS-A /chassis/virtual-drive-container/virtual-drive# commit-buffer
```

Storage Enclosure Operations

Removing Chassis Level Storage Enclosures

You can remove the storage enclosure corresponding to HDD expansion tray in Cisco UCS Manager after it is physically removed. You cannot remove server level or any other chassis level storage enclosures.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-id*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **remove storage-enclosure** *storage-enclosure-name*	Removes the chassis level storage enclosure with the specified name.

Example

The following example removes storage enclosure 25 from chassis 2:

```
UCS-A# scope chassis 2
UCS-A /chassis# remove storage-enclosure 25
UCS-A /chassis#
```

SAS Expander Configuration Policy

Creating SAS Expander 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 sas-expander-configuration-policy** *sas-expander-configuration-policy-name*	Creates a SAS expander configuration policy with the specified policy name.
Step 3	(Optional) UCS-A /org/sas-expander-configuration-policy* # **set descr** *description*	Provides a description for the policy.
Step 4	(Optional) UCS-A /org/sas-expander-configuration-policy* # **set 6g-12g-mixed-mode** *disabled|enabled|no-change*	**Note** Enabling or disabling 6G-12G Mixed Mode causes system reboot. • **Disabled**—Connection Management is disabled in this policy and the Sas Expander uses only 6G speeds even if 12G is available. • **Enabled**—Connection Management is enabled in this policy and it intelligently shifts between 6G and 12 G speeds based on availability. • **No Change** (Default) —Pre-existing configuration is retained.
Step 5	UCS-A /org/sas-expander-configuration-policy* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates the secp1 SAS expander configuration policy:

```
UCS-A# scope org
UCS-A /org # create sas-expander-configuration-policy secp1
UCS-A /org/sas-expander-configuration-policy*# set 6g-12g-mixed-mode enabled
UCS-A /org/sas-expander-configuration-policy*# commit-buffer
UCS-A /org/sas-expander-configuration-policy#
```

Deleting a SAS Expander Configuration Policy

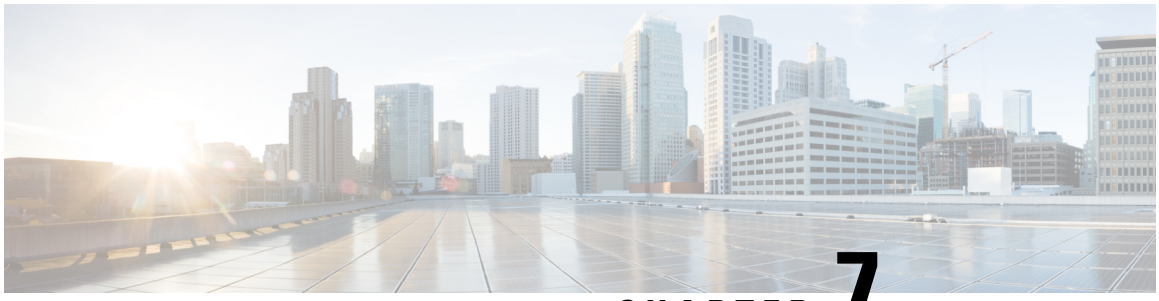
Procedure

	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 sas-expander-configuration-policy** *sas-expander-configuration-policy-name*	Deletes a SAS expander configuration policy with the specified policy name.
Step 3	UCS-A /org* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes the secp1 SAS expander configuration policy:

```
UCS-A# scope org
UCS-A /org # delete create sas-expander-configuration-policy secp1
UCS-A /org*# commit-buffer
UCS-A /org/#
```



CHAPTER 7

Firmware Management

Firmware Management for Cisco UCS S3260 Systems

Cisco UCS uses firmware obtained from and certified by Cisco to support the endpoints in a Cisco UCS domain. Each endpoint is a component in the Cisco UCS domain that requires firmware to function.

Cisco UCS Manager Firmware Management Guide, Release 3.2 provides detailed information about the complete firmware management process. Additionally, beginning with Cisco UCS Manager Release 3.1(2), you can upgrade the firmware of Cisco UCS S3260 chassis components by defining a chassis firmware policy and including it in the chassis profile associated with a Cisco UCS S3260 chassis.

You can upgrade a Cisco UCS domain with a S3260 chassis and servers through Cisco UCS Manager in the following ways:

- Upgrade infrastructure components through Auto Install—You can upgrade the infrastructure components, such as the Cisco UCS Manager software and the fabric interconnects, in a single step by using Auto Install. *Cisco UCS Manager Firmware Management Guide, Release 3.2* provides detailed information about the Auto Install process.
- Upgrade chassis through one of the following:
  - Upgrade chassis components through Auto Install—Beginning with Cisco UCS Manager Release 3.2(3), you can upgrade the firmware of Cisco UCS S3260 chassis components in a single step by using Auto Install.
  - Upgrade chassis through chassis firmware packages in chassis profiles—This option enables you to upgrade all chassis endpoints in a single step. The chassis endpoints that you can upgrade through a chassis firmware package are:
    - Chassis Adapter
    - Chassis Management Controller
    - Chassis Board Controller
    - Local Disk

Note You can upgrade local disks in the chassis through a chassis firmware package. Upgrade the local disks in a server through a host firmware package.

- SAS Expander

- Upgrade servers through firmware packages in service profiles—This option enables you to upgrade all server endpoints in a single step, reducing the amount of disruption caused by a server reboot. You can combine this option with the deferred deployment of service profile updates to ensure that server reboots occur during scheduled maintenance windows. The server endpoints that you can upgrade through a host firmware package are:

  Cisco UCS Manager Firmware Management Guide, Release 3.2 provides detailed information about upgrading server endpoints through host firmware packages.

You can also directly upgrade the firmware at each infrastructure, chassis, and server endpoint. This option enables you to upgrade many infrastructure, chassis, and server endpoints directly, including the fabric interconnects, SAS expanders, CMCs, chassis adapters, storage controllers, and board controllers. However, direct upgrade is not available for all endpoints, including the storage controller, HBA firmware, HBA option ROM and local disk.

This chapter explains the following newly introduced firmware management capabilities for the Cisco UCS S3260 system:

- Upgrading firmware through chassis firmware packages in chassis profiles
- Directly upgrading firmware on Cisco UCS S3260 chassis and server endpoints

Firmware Upgrades through Chassis Firmware Packages in Chassis Profiles

Cisco UCS Manager Release 3.1(2) introduces support for chassis profiles and chassis firmware packages on Cisco UCS S3260 chassis. You can upgrade the firmware of Cisco UCS S3260 chassis endpoints by defining a chassis firmware package and including it in the chassis profile associated with a chassis. You cannot manually upgrade the firmware of a chassis that is associated with a chassis profile.

Note If any chassis component is in the failed state, chassis profile association fails. Cisco recommends bringing the chassis component back up before continuing with chassis profile association. To continue association without bringing the chassis component back up, exclude the component before association.

You cannot upgrade the firmware on a server through chassis profiles. Upgrade the firmware on servers through service profiles.

Servers in a chassis are automatically powered down before the chassis upgrade process begins.

Chassis Firmware Package

This policy enables you to specify a set of firmware versions that make up the chassis firmware package (also known as the chassis firmware pack). The chassis firmware package includes the following firmware for chassis endpoints:

- **Chassis Adapter**
- **Chassis Management Controller**
- **Chassis Board Controller**
- **Local Disk**

Note **Local Disk** is excluded by default from the chassis firmware package.

- **SAS Expander**

Tip You can include more than one type of firmware in the same chassis firmware package. For example, a chassis firmware package can include both board controller firmware and chassis adapter firmware for two different models of adapters. However, you can only have one firmware version with the same type, vendor, and model number. The system recognizes which firmware version is required for an endpoint and ignores all other firmware versions.

You can also exclude firmware of specific components from a chassis firmware package either when creating a new chassis firmware package or when modifying an existing chassis firmware package. For example, if you do not want to upgrade the board controller firmware through the chassis firmware package, you can exclude board controller firmware from the list of firmware package components.

Important Each chassis firmware package is associated with one list of excluded components.

The chassis firmware package is pushed to all chassis associated with chassis profiles that include this policy.

This policy ensures that the chassis firmware is identical on all chassis associated with chassis profiles that use the same policy. Therefore, if you move the chassis profile from one chassis to another, the firmware versions are maintained. Also, if you change the firmware version for an endpoint in the chassis firmware package, new versions are applied to all the affected chassis profiles immediately.

For a chassis firmware package to take effect, include this policy in a chassis profile, and associate that chassis profile with a chassis.

This policy is not dependent upon any other policies. Ensure that the appropriate firmware has been downloaded to the fabric interconnect. If the firmware image is not available when Cisco UCS Manager is associating a chassis with a chassis profile, Cisco UCS Manager ignores the firmware upgrade and completes the association.

Stages of a Firmware Upgrade through Chassis Firmware Packages in Chassis Profiles

You can use the chassis firmware package policies in chassis profiles to upgrade chassis firmware.

Caution If you modify a chassis firmware package by adding an endpoint or changing firmware versions for an existing endpoint, Cisco UCS Manager upgrades the endpoints after you acknowledge the change by clicking **Pending Activities**. This process disrupts data traffic to and from the chassis.

New Chassis Profile

For a new chassis profile, this upgrade takes place over the following stages:

Chassis Firmware Package Policy Creation

During this stage, you create the chassis firmware package.

Chassis Profile Association

During this stage, you include the chassis firmware package in a chassis profile, and then associate the chassis profile with a chassis. The system pushes the selected firmware versions to the endpoints. The chassis must be reacknowledged to ensure that the endpoints are running the versions specified in the firmware package.

Existing Chassis Profile

For chassis profiles that are associated with a chassis, Cisco UCS Manager upgrades the firmware after you acknowledge the change by clicking **Pending Activities**.

Effect of Updates to Firmware Packages in Chassis Profiles

To update firmware through a chassis firmware package in a chassis profile, you need to update the firmware in the package. What happens after you save the changes to a firmware package depends upon how the Cisco UCS domain is configured.

The following table describes the most common option for upgrading chassis with a firmware package in a chassis profile.

Chassis Profile	Maintenance Policy	Upgrade Actions
The chassis firmware package is included in one or more chassis profiles, and each chassis profile is associated with one chassis. OR The chassis firmware package is included in an updating chassis profile template, and the chassis profile created from that template is associated with one chassis.	Configured for user acknowledgment	The following occurs when you update the chassis firmware package: 1. Cisco UCS asks you to confirm your change and advises that a user-acknowledgement of the chassis is required. 2. Click the flashing **Pending Activities** button to select the chassis you want to reacknowledge, and apply the new firmware. 3. Cisco UCS verifies the model numbers and vendor against all chassis associated with chassis profiles that include this policy. If the model numbers and vendor match a firmware version in the policy, Cisco UCS reacknowledges the chassis and updates the firmware. A manual reacknowledgment of the chassis does not cause Cisco UCS to apply the chassis firmware package, nor does it cancel the pending activities. You must acknowledge or cancel the pending activity through the **Pending Activities** button.

Creating or Updating a Chassis Firmware Package

Tip You can include more than one type of firmware in the same chassis firmware package. For example, a chassis firmware package can include both board controller firmware and chassis adapter firmware for two different models of adapters. However, you can only have one firmware version with the same type, vendor, and model number. The system recognizes which firmware version is required for an endpoint and ignores all other firmware versions.

You can also exclude firmware of specific components from a chassis firmware package either when creating a new chassis firmware package or when modifying an existing chassis firmware package.

Important Each chassis firmware package is associated with one list of excluded components, which is common across all firmware packages.

Before you begin

Ensure that the appropriate firmware was downloaded to the fabric interconnect.

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 fw-chassis-pack** *pack-name*	Creates a chassis firmware package with the specified package name and enters organization firmware chassis package mode.
Step 3	(Optional) UCS-A org/fw-chassis-pack* # **set chassispack-vers** *version-num*	Specifies the package image version number. Changing this number triggers firmware updates on all components using the firmware through a chassis profile. Use this step only when updating a chassis firmware package, not when creating a package.
Step 4	(Optional) UCS-A org/fw-chassis-pack* # **set servicepack-vers** *servicepack-version-num*	Specifies the service pack version number. You cannot directly upgrade to a service pack without selecting a base chassis pack. The images from the service pack will take precedence over the images from Chassis Package.
Step 5	UCS-A org/fw-chassis-pack* # **create exclude-chassis-component** {**chassis-adaptor** | **chassis-board-controller** | **chassis-management-controller** | **local-disk** | **sas-expander**}	Excludes the specified component from the chassis firmware package. **Note** local-disk is excluded from the chassis firmware package by default.
Step 6	Required: UCS-A org/fw-chassis-pack* # **delete exclude-chassis-component** {**chassis-adapter** | **chassis-board-controller** | **chassis-management-controller** | **local-disk** | **sas-expander**}	Includes the specified component from the chassis firmware package.
Step 7	UCS-A org/fw-chassis-pack* # **commit-buffer**	Commits the transaction.

Example

The following example creates the cp1 chassis firmware package, includes the local disk component, and commits the transaction:

```
UCS-A# scope org
UCS-A /org # create fw-chassis-pack cp1
UCS-A /org/fw-chassis-pack* # delete exclude-chassis-component local-disk
UCS-A /org/fw-chassis-pack/exclude-chassis-component* # commit-buffer
UCS-A /org/fw-chassis-pack/exclude-chassis-component #
```

The following example excludes the chassis board controller component from the cp1 chassis firmware package, and commits the transaction:

```
UCS-A# scope org
UCS-A /org # enter fw-chassis-pack cp1
UCS-A /org/fw-chassis-pack* # create exclude-chassis-component chassis-board
-controller
UCS-A /org/fw-chassis-pack/exclude-chassis-component* # commit-buffer
UCS-A /org/fw-chassis-pack/exclude-chassis-component #
```

The following example adds a service pack to the cp1 chassis firmware package, and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope fw-chassis-pack cp1
UCS-A /org/fw-chassis-pack # set servicepack-vers 3.1(3)SP1
UCS-A /org/fw-chassis-pack* # commit-buffer
UCS-A /org/fw-chassis-pack #
```

Upgrading a UCS Domain with Cisco UCS S3260 Servers

Before you begin

- Ensure that all the servers nodes are shut down.
- Ensure that the UCS domain has an assigned chassis policy that references a chassis firmware package policy and a chassis maintenance policy.

Procedure

Step 1 Upgrade infrastructure firmware through Auto Install. See Upgrading the Infrastructure Firmware with Auto Install.

Step 2 Update the chassis firmware package policy.

- If you are using the default chassis firmware package policy, update the **default** chassis firmware package policy with the new package version. See Creating or Updating a Chassis Firmware Package, on page 67.
- You can create a new chassis firmware package policy using the new chassis package version, and configure the existing or assigned chassis profile (accept any UserAck). See Creating or Updating a Chassis Firmware Package, on page 67 to create a new chassis firmware package policy.

This process may take 1-2 hours. You can monitor the status in the chassis FSM tab.

Step 3 Update the host firmware. See Upgrading the Server Firmware with Auto Install.

Note Updating the host firmware is possible only through Cisco UCS Manager GUI.

Direct Firmware Upgrade on S3260 Chassis and Server Endpoints

The following sections provide detailed information about upgrading S3260 Chassis and Server endpoints.

S3260 Chassis Endpoints

To trigger firmware upgrade on S3260 Chassis components, use the following order:

1. Update CMC 1 firmware
2. Update CMC 2 firmware
3. Update Chassis Adapter 1 firmware
4. Update Chassis Adapter 2 firmware
5. Update SAS Expander 1 firmware
6. Update SAS Expander 2 firmware
7. Activate SAS Expander 1 firmware
8. Activate SAS Expander 2 firmware
9. Activate CMC 1 firmware
10. Activate CMC 2 firmware
11. Activate Chassis Adapter 1 firmware
12. Activate Chassis Adapter 2 firmware
13. Activate Chassis Board Controller

Note You cannot manually update the firmware for local disk in a chassis. The local disk firmware is updated when you explicitly include it in a chassis firmware package.

Cisco UCS S3260 Server Node Endpoints

To trigger firmware upgrade on server endpoints, use the following order:

1. Update CIMC
2. Activate CIMC
3. Update BIOS
4. Activate BIOS
5. Activate Board Controller
6. Activate Storage Controller

While upgrading firmware, Cisco recommends that you use the following order:

1. Upgrade infrastructure—Cisco UCS Manager software and the fabric interconnects
2. Upgrade chassis and server endpoints

While downgrading firmware, Cisco recommends that you use the following order:

1. Downgrade chassis and server endpoints
2. Downgrade infrastructure—Cisco UCS Manager software and the fabric interconnects

Direct Firmware Upgrade on Chassis Endpoints

Updating and Activating the CMC Firmware on a Chassis

Caution Do not remove the hardware that contains the endpoint or perform any maintenance on it until the update process completes. If the hardware is removed or otherwise unavailable due to maintenance, the firmware update fails. This failure might corrupt the backup partition. You cannot update the firmware on an endpoint with a corrupted backup partition.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-id*	Enters chassis mode.
Step 2	UCS-A /chassis # **scope sioc** {**1** | **2**}	Enters the specified SIOC.
Step 3	UCS-A /chassis/sioc # **scope cmc**	Enters chassis CMC mode.
Step 4	UCS-A /chassis/sioc/cmc # **update firmware** *version-num*	Updates the selected firmware version on the CMC in the chassis.
Step 5	(Optional) UCS-A /chassis/sioc/cmc* # **commit-buffer**	Commits the transaction. Use this step only if you intend to use the **show update status** command in Step 5 to verify that the firmware update completed successfully before activating the firmware in Step 6. You can skip this step and commit the **update firmware** and **activate firmware** commands in the same transaction; however, if the firmware update does not complete successfully, the firmware activation does not start. Cisco UCS Manager copies the selected firmware image to the backup memory partition and verifies that image is not corrupt.

	Command or Action	Purpose
		The image remains as the backup version until you explicitly activate it.
Step 6	(Optional) UCS-A /chassis/sioc/cmc # **show update status**	Displays the status of the firmware update. Use this step only if you want to verify that the firmware update completed successfully. The firmware update is complete when the update status is Ready. The CLI does not automatically refresh, so you may have to enter the **show update status** command multiple times until the task state changes from Updating to Ready. Continue to Step 6 when the update status is Ready.
Step 7	UCS-A /chassis/sioc/cmc # **activate firmware** *version-num*	Activates the selected firmware version on the CMC in the server.
Step 8	UCS-A /chassis/sioc/cmc* # **commit-buffer**	Commits the transaction.
Step 9	(Optional) UCS-A /chassis/sioc/cmc # **show activate status**	Displays the status of the firmware activation. Use this step only if you want to verify that the firmware activation completed successfully. The CLI does not automatically refresh, so you may have to enter the **show activate status** command multiple times until the task state changes from Activating to Ready.
Step 10	(Optional) CS-A /chassis/sioc/cmc # **show firmware**	Displays the running firmware version, the Update status and the Activate status.

Example

The following example updates and activates the CMC firmware to version 2.0(8.13) in the same transaction, without verifying that the firmware update and firmware activation completed successfully:

```
UCS-A# scope chassis 2
UCS-A# /chassis # scope sioc 1
UCS-A# /chassis/sioc # scope cmc
UCS-A# /chassis/sioc/cmc # update firmware 2.0(8.13)
UCS-A# /chassis/sioc/cmc* # activate firmware 2.0(8.13)
UCS-A# /chassis/sioc/cmc* # commit-buffer
UCS-A# /chassis/sioc/cmc # show firmware
CMC:
    Running-Vers: 2.0(8.13)
    Package Vers: 3.1(2.222)C
    Update-Status: Ready
    Activate-Status: Ready
```

The following example updates the CMC firmware to version 2.0(8.13), verifies that the firmware update completed successfully before starting the firmware activation, activates the CMC firmware, and verifies that the firmware activation completed successfully:

```
UCS-A# scope chassis 2
UCS-A# /chassis # scope sioc 1
UCS-A# /chassis/sioc # scope cmc
UCS-A# /chassis/sioc/cmc # update firmware 2.0(8.13)
UCS-A# /chassis/sioc/cmc* # commit-buffer
UCS-A# /chassis/sioc/cmc # show update status
Status: Ready
UCS-A# /chassis/sioc/cmc # activate firmware 2.0(8.13)
UCS-A# /chassis/sioc/cmc* # commit-buffer
UCS-A# /chassis/sioc/cmc # show activate status
Status: Ready
UCS-A# /chassis/sioc/cmc # show firmware
CMC:
    Running-Vers: 2.0(8.13)
    Package Vers: 3.1(0.344)M
    Update-Status: Ready
    Activate-Status: Ready
```

Updating and Activating the Chassis Adapter Firmware on a Chassis

Updating and activating the chassis adapter firmware affects all servers in a chassis.

Before you begin

Gracefully power down the servers.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-id*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **scope sioc** {**1** | **2**}	Enters the specified SIOC.
Step 3	UCS-A /chassis/sioc # **scope adapter**	Enters chassis adapter mode.
Step 4	UCS-A /chassis/sioc/adapter # **show image**	Displays the available software images for the chassis adapter.
Step 5	UCS-A /chassis/sioc/adapter # **update firmware** *version-num*	Updates the selected firmware version on the chassis adapter.
Step 6	(Optional) UCS-A /chassis/sioc/adapter* # **commit-buffer**	Commits the transaction. Use this step only if you intend to use the **show update status** command in Step 6 to verify that the firmware update completed successfully before activating the firmware in Step 7. You can skip this step and commit the **update firmware** and **activate firmware**

	Command or Action	Purpose
		commands in the same transaction; however, if the firmware update does not complete successfully, the firmware activation does not start.
Step 7	(Optional) UCS-A /chassis/sioc/adapter # **show update status**	Displays the status of the firmware update. Use this step only if you want to verify that the firmware update completed successfully. The firmware update is complete when the update status is **Ready**. The CLI does not automatically refresh, so you may have to enter the **show update status** command multiple times until the task state changes from **Updating** to **Ready**. Continue to Step 7 when the update status is **Ready**.
Step 8	UCS-A /chassis/sioc/adapter # **activate firmware** *version-num*	Activates the selected firmware version on the chassis adapter.
Step 9	UCS-A /chassis/sioc/adapter* # **commit-buffer**	Commits the transaction.
Step 10	UCS-A /chassis/sioc/adapter # **show activate status**	Displays the status of the firmware activation. Use this step only if you want to verify that the firmware activation completed successfully. The CLI does not automatically refresh, so you may have to enter the **show activate status** command multiple times until the task state changes from **Activating** to **Ready**.
Step 11	(Optional) UCS-A /chassis/sioc/adapter # **show firmware**	Displays the running firmware version, the **Update** status and the **Activate** status.

Example

The following example updates and activates the chassis adapter firmware in the same transaction, without verifying that the firmware update and firmware activation completed successfully:

```
UCS-A# scope chassis 1
UCS-A# /chassis # scope sioc 2
UCS-A# /chassis/sioc # scope adapter
UCS-A# /chassis/sioc/adapter # show image
Name                                       Type                Version
------------------------------------------ ------------------- -------
ucs-2200.3.1.2.222.gbin                    Chassis Adaptor     3.1(2b)
ucs-2200.3.1.300.102.gbin                  Chassis Adaptor     3.1(300.102)
ucs-m83-8p40-vic.4.1.1.58.gbin             Chassis Adaptor     4.1(1.58)
ucs-pcie-c40q-03.4.1.1.58.gbin             Chassis Adaptor     4.1(1.58)
```

```
UCS-A# /chassis/sioc/adapter # update firmware 3.1(2b)
UCS-A# /chassis/sioc/adapter* # activate firmware 3.1(2b)
UCS-A# /chassis/sioc/adapter* # commit-buffer
UCS-A# /chassis/sioc/adapter # show firmware
Adapter:
    Running-Vers: 3.1(2b)
    Package-Vers:
    Update-Status: Ready
    Activate-Status: Ready
```

The following example updates the chassis adapter firmware, verifies that the firmware update completed successfully before starting the firmware activation, activates the chassis adapter firmware, and verifies that the firmware activation completed successfully:

```
UCS-A# scope chassis 1
UCS-A# /chassis # scope sioc 2
UCS-A# /chassis/sioc # scope adapter
UCS-A# /chassis/sioc/adapter # show image
Name                                       Type                Version
------------------------------------------ ------------------- -------
ucs-2200.3.1.2.222.gbin                    Chassis Adaptor     3.1(2b)
ucs-2200.3.1.300.102.gbin                  Chassis Adaptor     3.1(300.102)
ucs-m83-8p40-vic.4.1.1.58.gbin             Chassis Adaptor     4.1(1.58)
ucs-pcie-c40q-03.4.1.1.58.gbin             Chassis Adaptor     4.1(1.58)

UCS-A# /chassis/sioc/adapter # update firmware 3.1(2b)
UCS-A# /chassis/sioc/adapter* # commit-buffer
UCS-A# /chassis/sioc/adapter # show update status
Status: Ready
UCS-A# /chassis/sioc/adapter # activate firmware 3.1(2b)
UCS-A# /chassis/sioc/adapter* # commit-buffer
UCS-A# /chassis/sioc/adapter # show activate status
Status: Ready
UCS-A# /chassis/sioc/adapter # show firmware
Adapter:
    Running-Vers: 3.1(2b)
    Package-Vers:
    Update-Status: Ready
    Activate-Status: Ready
```

Updating and Activating the SAS Expander Firmware on a Chassis

Caution Do not remove the hardware that contains the endpoint or perform any maintenance on it until the update process completes. If the hardware is removed or otherwise unavailable due to maintenance, the firmware update fails. This failure might corrupt the backup partition. You cannot update the firmware on an endpoint with a corrupted backup partition.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-id*	Enters chassis mode.
Step 2	UCS-A /chassis # **scope sas-expander** *sas-id*	Enters chassis SAS expander mode for the specified SAS expander.
Step 3	UCS-A /chassis/sas-expander # **update firmware** *version-num*	Updates the selected firmware version on the specified SAS expander in the chassis.
Step 4	(Optional) UCS-A /chassis/sas-expander* # **commit-buffer**	Commits the transaction. Use this step only if you intend to use the **show update status** command in Step 5 to verify that the firmware update completed successfully before activating the firmware in Step 6. You can skip this step and commit the **update firmware** and **activate firmware** commands in the same transaction; however, if the firmware update does not complete successfully, the firmware activation does not start. Cisco UCS Manager copies the selected firmware image to the backup memory partition and verifies that image is not corrupt. The image remains as the backup version until you explicitly activate it.
Step 5	(Optional) UCS-A /chassis/sas-expander # **show update status**	Displays the status of the firmware update. Use this step only if you want to verify that the firmware update completed successfully. The firmware update is complete when the update status is **Ready**. The CLI does not automatically refresh, so you may have to enter the **show update status** command multiple times until the task state changes from **Updating** to **Ready**. Continue to Step 6 when the update status is **Ready**.
Step 6	UCS-A /chassis/sas-expander # **activate firmware** *version-num*	Activates the selected firmware version on the specified SAS expander in the chassis.
Step 7	UCS-A /chassis/sas-expander* # **commit-buffer**	Commits the transaction.
Step 8	(Optional) UCS-A /chassis/sas-expander # **show activate status**	Displays the status of the firmware activation. Use this step only if you want to verify that the firmware activation completed successfully. The CLI does not automatically refresh, so you may have to enter the **show activate status** command multiple times until the task state changes from Activating to Ready.

	Command or Action	Purpose
Step 9	(Optional) UCS-A /chassis/sas-expander # **show firmware**	Displays the running firmware version, the Update status and the Activate status.

Example

The following example updates and activates the SAS expander firmware to version 3.1(2b) in the same transaction, without verifying that the firmware update and firmware activation completed successfully:

```
UCS-A# scope chassis 2
UCS-A# /chassis # scope sas-expander
UCS-A# /chassis/sas-expander # update firmware 3.1(2b)
UCS-A# /chassis/sas-expander* # activate firmware 3.1(2b)
UCS-A# /chassis/sas-expander* # commit-buffer
UCS-A# /chassis/sas-expander # show firmware
Running-Vers    Package-Vers    Activate-Status
--------------- --------------- ---------------
3.1(2b)                         Ready
```

The following example updates the SAS expander firmware to version 3.1(2b), verifies that the firmware update completed successfully before starting the firmware activation, activates the SAS expander firmware, and verifies that the firmware activation completed successfully:

```
UCS-A# scope chassis 2
UCS-A# /chassis # scope sas-expander
UCS-A# /chassis/sas-expander # update firmware 3.1(2b)
UCS-A# /chassis/sas-expander* # commit-buffer
UCS-A# /chassis/sas-expander # show update status
Status: Ready
UCS-A# /chassis/sas-expander # activate firmware 3.1(2b)
UCS-A# /chassis/sas-expander* # commit-buffer
UCS-A# /chassis/sas-expander # show activate status
Status: Ready
UCS-A# /chassis/sas-expander # show firmware
    Running-Vers: 3.1(2b)
    Package Vers: 3.1(2b)
    Update-Status: Ready
    Activate-Status: Ready
```

Activating the Board Controller Firmware on a Chassis

Note Cisco UCS Manager does not support activation of board controller firmware to earlier versions.

Before you begin

Gracefully power down the servers.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-id*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **scope sioc** {1 | 2}	Enters the specified SIOC in the chassis.
Step 3	UCS-A /chassis/sioc #**scope boardcontroller**	Enters board controller mode for the chassis.
Step 4	UCS-A /chassis/sioc/boardcontroller # **activate firmware** *version-num*	Activates the selected firmware version on the board controller in the chassis.
Step 5	UCS-A /chassis/sioc/boardcontroller* # **commit-buffer**	Commits the transaction to the system configuration.
Step 6	UCS-A /chassis/sioc/boardcontroller # **show firmware**	Displays the running firmware version and the activate status.

Example

The following example shows how to activate the board controller firmware on a chassis:

```
UCS-A# scope chassis 1
UCS-A /chassis # scope sioc 1
UCS-A /chassis/sioc # scope boardcontroller
UCS-A /chassis/sioc/boardcontroller # activate firmware 3.1.211
Warning: When committed, this command will soft shutdown the servers and may power cycle
the chassis while activating the board controller.
Associated servers power state will be restored after chassis power cycle.
UCS-A# /chassis/sioc/boardcontroller* # commit-buffer
UCS-A /chassis/sioc/boardcontroller # show firmware
Board Controller:
    Running-Vers: NA
    Package-Vers: 3.1(2b)C
    Activate-Status: Ready
UCS-A /chassis/boardcontroller* #
```

Direct Firmware Upgrade on Server Endpoints

Updating and Activating the CIMC Firmware on a Cisco UCS S3260 Storage Server

The activation of firmware for a CIMC does not disrupt data traffic. However, it will interrupt all KVM sessions and disconnect any vMedia attached to the server.

Caution Do not remove the hardware that contains the endpoint or perform any maintenance on it until the update process completes. If the hardware is removed or otherwise unavailable due to maintenance, the firmware update fails. This failure might corrupt the backup partition. You cannot update the firmware on an endpoint with a corrupted backup partition.

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 cimc**	Enters chassis server CIMC mode.
Step 3	UCS-A /chassis/server/cimc # **show image**	Displays the available software images for the adapter.
Step 4	UCS-A /chassis/server/cimc # **update firmware** *version-num*	Updates the selected firmware version on the CIMC in the server.
Step 5	(Optional) UCS-A /chassis/server/cimc* # **commit-buffer**	Commits the transaction. Use this step only if you intend to use the **show firmware** command in Step 6 to verify that the firmware update completed successfully before activating the firmware in Step 7. You can skip this step and commit the **update-firmware** and **activate-firmware** commands in the same transaction; however, if the firmware update does not complete successfully, the firmware activation does not start. Cisco UCS Manager copies the selected firmware image to the backup memory partition and verifies that image is not corrupt. The image remains as the backup version until you explicitly activate it.
Step 6	(Optional) UCS-A /chassis/server/cimc # **show firmware**	Displays the status of the firmware update. Use this step only if you want to verify that the firmware update completed successfully. The firmware update is complete when the update status is Ready. The CLI does not automatically refresh, so you may have to enter the **show firmware** command multiple times until the task state changes from Updating to Ready. Continue to Step 7 when the update status is Ready.
Step 7	UCS-A /chassis/server/cimc # **activate firmware** *version-num*	Activates the selected firmware version on the CIMC in the server.

	Command or Action	Purpose
Step 8	UCS-A /chassis/server/cimc* # **commit-buffer**	Commits the transaction.
Step 9	(Optional) UCS-A /chassis/server/cimc # **show firmware**	Displays the status of the firmware activation. Use this step only if you want to verify that the firmware activation completed successfully. The CLI does not automatically refresh, so you may have to enter the **show firmware** command multiple times until the task state changes from Activating to Ready.

Example

The following example updates and activates the CIMC firmware in the same transaction, without verifying that the firmware update and firmware activation completed successfully:

```
UCS-A# scope server 3/1
UCS-A# /chassis/server # scope cimc
UCS-A# /chassis/server/cimc # show image
Name                                       Type               Version
------------------------------------------ ------------------ -------
ucs-b200-m1-k9-cimc.3.1.20.26.gbin         CIMC               3.1(20.26)
ucs-b200-m3-k9-cimc.3.1.20.26.gbin         CIMC               3.1(20.26)
ucs-b200-m4-k9-cimc.3.1.20.26.gbin         CIMC               3.1(20.26)
ucs-b22-m3-k9-cimc.3.1.20.26.gbin          CIMC               3.1(20.26)
ucs-b230-m2-k9-cimc.3.1.20.26.gbin         CIMC               3.1(20.26)
ucs-b250-m1-k9-cimc.3.1.20.26.gbin         CIMC               3.1(20.26)
ucs-b420-m3-k9-cimc.3.1.20.26.gbin         CIMC               3.1(20.26)
ucs-b420-m4-k9-cimc.3.1.20.26.gbin         CIMC               3.1(20.26)
ucs-b440-m2-k9-cimc.3.1.20.26.gbin         CIMC               3.1(20.26)
ucs-c22-k9-cimc.2.0.12.73.gbin             CIMC               2.0(12.73)
ucs-c220-k9-cimc.2.0.12.73.gbin            CIMC               2.0(12.73)
ucs-c220-m4-k9-cimc.2.0.12.73.gbin         CIMC               2.0(12.73)
ucs-c240-k9-cimc.2.0.12.73.gbin            CIMC               2.0(12.73)
ucs-c240-m4-k9-cimc.2.0.12.73.gbin         CIMC               2.0(12.73)
ucs-c3260-m3-k9-cimc.2.0.12.73.gbin        CIMC               2.0(12.73)
ucs-c3260-m4-k9-cimc.2.0.12.73.gbin        CIMC               2.0(12.73)
ucs-c460-m4-k9-cimc.2.0.12.73.gbin         CIMC               2.0(12.73)
ucs-EXM4-1-k9-cimc.3.1.20.26.gbin          CIMC               3.1(20.26)
ucs-EXM4-2-k9-cimc.3.1.20.26.gbin          CIMC               3.1(20.26)


...

UCS-A# /chassis/server/cimc # update firmware 2.0(12.73)
UCS-A# /chassis/server/cimc* # activate firmware 2.0(12.73)
UCS-A# /chassis/server/cimc* # commit-buffer
UCS-A# /chassis/server/cimc #
```

The following example updates the CIMC firmware, verifies that the firmware update completed successfully before starting the firmware activation, activates the CIMC firmware, and verifies that the firmware activation completed successfully:

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

```
UCS-A# /chassis/server # scope cimc
UCS-A# /chassis/server/cimc # show image
Name                                       Type                Version
------------------------------------------ ------------------- -------
ucs-b200-m1-k9-cimc.3.1.20.26.gbin         CIMC                3.1(20.26)
ucs-b200-m3-k9-cimc.3.1.20.26.gbin         CIMC                3.1(20.26)
ucs-b200-m4-k9-cimc.3.1.20.26.gbin         CIMC                3.1(20.26)
ucs-b22-m3-k9-cimc.3.1.20.26.gbin          CIMC                3.1(20.26)
ucs-b230-m2-k9-cimc.3.1.20.26.gbin         CIMC                3.1(20.26)
ucs-b250-m1-k9-cimc.3.1.20.26.gbin         CIMC                3.1(20.26)
ucs-b420-m3-k9-cimc.3.1.20.26.gbin         CIMC                3.1(20.26)
ucs-b420-m4-k9-cimc.3.1.20.26.gbin         CIMC                3.1(20.26)
ucs-b440-m2-k9-cimc.3.1.20.26.gbin         CIMC                3.1(20.26)
ucs-c22-k9-cimc.2.0.12.73.gbin             CIMC                2.0(12.73)
ucs-c220-k9-cimc.2.0.12.73.gbin            CIMC                2.0(12.73)
ucs-c220-m4-k9-cimc.2.0.12.73.gbin         CIMC                2.0(12.73)
ucs-c240-k9-cimc.2.0.12.73.gbin            CIMC                2.0(12.73)
ucs-c240-m4-k9-cimc.2.0.12.73.gbin         CIMC                2.0(12.73)
ucs-c3260-m3-k9-cimc.2.0.12.73.gbin        CIMC                2.0(12.73)
ucs-c3260-m4-k9-cimc.2.0.12.73.gbin        CIMC                2.0(12.73)
ucs-c460-m4-k9-cimc.2.0.12.73.gbin         CIMC                2.0(12.73)
ucs-EXM4-1-k9-cimc.3.1.20.26.gbin          CIMC                3.1(20.26)
ucs-EXM4-2-k9-cimc.3.1.20.26.gbin          CIMC                3.1(20.26)
...

UCS-A# /chassis/server/cimc # update firmware 2.0(12.73)
UCS-A# /chassis/server/cimc* # commit-buffer
UCS-A# /chassis/server/cimc # show firmware
Running-Vers    Update-Status   Activate-Status
--------------- --------------- ---------------
2.0(12.73)      Updating        Ready

UCS-A# /chassis/server/cimc # show firmware
Running-Vers    Update-Status   Activate-Status
--------------- --------------- ---------------
2.0(12.73)      Ready           Ready

UCS-A# /chassis/server/cimc # activate firmware 2.0(12.73)
UCS-A# /chassis/server/cimc* # commit-buffer
UCS-A# /chassis/server/cimc # show firmware
Running-Vers    Update-Status   Activate-Status
--------------- --------------- ---------------
2.0(12.73)      Ready           Activating

UCS-A# /chassis/server/cimc # show firmware
Running-Vers    Update-Status   Activate-Status
--------------- --------------- ---------------
2.0(12.73)      Ready           Ready
```

Updating and Activating the BIOS Firmware on a Cisco UCS S3260 Storage Server

Important You can update and activate BIOS firmware on a server using the Cisco UCS Manager CLI on all servers.

Caution Do not remove the hardware that contains the endpoint or perform any maintenance on it until the update process completes. If the hardware is removed or otherwise unavailable due to maintenance, the firmware update fails. This failure might corrupt the backup partition. You cannot update the firmware on an endpoint with a corrupted backup partition.

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 chassis server BIOS mode.
Step 3	UCS-A /chassis/server/bios # **show image**	Displays the available BIOS firmware images.
Step 4	UCS-A /chassis/server/bios # **update firmware** *version-num*	Updates the selected BIOS firmware for the server.
Step 5	(Optional) UCS-A /chassis/server/bios* # **commit-buffer**	Commits the transaction. Use this step only if you intend to use the **show firmware** command in Step 6 to verify that the firmware update completed successfully before activating the firmware in Step 7. You can skip this step and commit the **update-firmware** and **activate-firmware** commands in the same transaction; however, if the firmware update does not complete successfully, the firmware activation does not start. Cisco UCS Manager copies the selected firmware image to the backup memory partition and verifies that image is not corrupt. The image remains as the backup version until you explicitly activate it.
Step 6	(Optional) UCS-A /chassis/server/bios # **show firmware**	Displays the status of the firmware update. Use this step only if you want to verify that the firmware update completed successfully. The firmware update is complete when the update status is Ready. The CLI does not automatically refresh, so you may have to enter the **show firmware** command multiple times until the task state changes from Updating to Ready. Continue to Step 7 when the update status is Ready.
Step 7	UCS-A /chassis/server/bios # **activate firmware** *version-num*	Activates the selected server BIOS firmware version.

	Command or Action	Purpose
Step 8	UCS-A /chassis/server/bios* # **commit-buffer**	Commits the transaction.
Step 9	(Optional) UCS-A /chassis/bios # **show firmware**	Displays the status of the firmware activation. Use this step only if you want to verify that the firmware activation completed successfully. The CLI does not automatically refresh, so you may have to enter the **show firmware** command multiple times until the task state changes from Activating to Ready.

Example

The following example updates and activates the BIOS firmware in the same transaction, without verifying that the firmware update and activation completed successfully:

```
UCS-A# scope server 3/1
UCS-A# /chassis/server # scope bios
UCS-A# /chassis/server/bios # show image
Name                                        Type          Version
------------------------------------------- ------------- -------
ucs-b200-m2-bios.S5500.2.1.3c.0.0151437.bin   Server BIOS   S5500.2.1.3c.
                                                                  0.081120151437
ucs-b200-m3-bios.B200M3.2.2.6d.0.160055.bin   Server BIOS   B200M3.2.2.6d

.0.062220160055
ucs-b200-m4-bios.B200M4.3.1.3c.0.161459.bin   Server BIOS   B200M4.3.1.3c

.0.080120161459
ucs-b200-m4-bios.B200M4.3.1.3e.0.161737.bin   Server BIOS   B200M4.3.1.3e

.0.081120161737
ucs-b22-m3-bios.B22M3.2.2.6d.0.20160114.bin   Server BIOS   B22M3.2.2.6d.
                                                                  0.062220160114
ucs-b230-m2-bios.B230.2.1.3a.0.20151410.bin   Server BIOS   B230.2.1.3a.0
                                                                   .022420151410
ucs-b250-m2-bios.S5500.2.1.3d.0.0161035.bin   Server BIOS   S5500.2.1.3d.
                                                                  0.032520161035
ucs-b420-m3-bios.B420M3.2.2.6e.0.160138.bin   Server BIOS   B420M3.2.2.6e.0.062220160138
ucs-b420-m4-bios.B420M4.3.1.2a.0.161234.bin   Server BIOS   B420M4.3.1.2a.0.072520161234
ucs-b420-m4-bios.B420M4.3.1.2d.0.161622.bin   Server BIOS   B420M4.3.1.2d.0.081120161622
ucs-b440-m2-bios.B440.2.1.3a.0.20151142.bin   Server BIOS   B440.2.1.3a.0.022420151142
ucs-c22-bios.C22M3.2.0.13a.0.0713160955.bin   Server BIOS   C22M3.2.0.13a.0.0713160955
ucs-c220-bios.C220M3.2.0.13a.0.13160937.bin   Server BIOS   C220M3.2.0.13a.0.0713160937
ucs-c220-m4-bios.C220M4.2.0.13a.0.62332.bin   Server BIOS   C220M4.2.0.13a.0.0725162332
ucs-c220-m4-bios.C220M4.2.0.13b.0.61705.bin   Server BIOS   C220M4.2.0.13b.0.0805161705
ucs-c240-bios.C240M3.2.0.13a.0.13160947.bin   Server BIOS   C240M3.2.0.13a.0.0713160947
ucs-c240-m4-bios.C240M4.2.0.13a.0.62345.bin   Server BIOS   C240M4.2.0.13a.0.0725162345
ucs-c240-m4-bios.C240M4.2.0.13b.0.61722.bin   Server BIOS   C240M4.2.0.13b.0.0805161722
ucs-c3260-m3-bios.C3X60M3.2.0.13a.0.044.bin   Server BIOS   C3X60M3.2.0.13a.0.0722160044
ucs-c3260-m4-bios.C3X60M4.2.0.13a.0.350.bin   Server BIOS   C3X60M4.2.0.13a.0.0801162350
ucs-c460-m4-bios.C460M4.2.0.13a.0.60447.bin   Server BIOS   C460M4.2.0.13a.0.072720160447
ucs-c460-m4-bios.C460M4.2.0.13b.0.62321.bin   Server BIOS   C460M4.2.0.13b.0.080320162321
ucs-EXM4-1-bios.EXM4.2.2.7.0.1520161539.bin   Server BIOS   EXM4.2.2.7.0.021520161539
ucs-EXM4-2-bios.EXM4.2.2.7.0.1520161539.bin   Server BIOS   EXM4.2.2.7.0.021520161539
ucs-EXM4-3-bios.EXM4.3.1.2b.0.020161506.bin   Server BIOS   EXM4.3.1.2b.0.062020161506
```

```
UCS-A# /chassis/server/bios # update firmware C3X60M4.2.0.12.11.041320162312
UCS-A# /chassis/server/bios* # activate firmware C3X60M4.2.0.12.11.041320162312
UCS-A# /chassis/server/bios* # commit-buffer
UCS-A# /chassis/server/bios #
```

Activating the Board Controller Firmware on a Cisco UCS S3260 Storage Server

The board controller firmware controls many of the server functions, including eUSBs, LEDs, and I/O connectors.

Note This activation procedure causes the server to reboot. Depending upon whether the service profile associated with the server includes a maintenance policy, the reboot can occur immediately. Cisco recommends that you upgrade the board controller firmware through the host firmware package in the service profile as the last step of upgrading a Cisco UCS domain, along with upgrading the server BIOS. This reduces the number of times a server needs to reboot during the upgrade process.

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 boardcontroller**	Enters board controller mode for the server.
Step 3	(Optional) UCS-A /chassis/server/boardcontroller # **show image**	Displays the available software images for the board controller.
Step 4	(Optional) UCS-A /chassis/server/boardcontroller # **show firmware**	Displays the current running software image for the board controller.
Step 5	UCS-A /chassis/server/boardcontroller # **activate firmware** *version-num*	Activates the selected firmware version on the board controller in the server.
Step 6	UCS-A /chassis/server/boardcontroller* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example activates the board controller firmware:

```
UCS-A# scope server 3/1
UCS-A# /chassis/server # scope boardcontroller
UCS-A# /chassis/server/boardcontroller # show image
Name                                       Type                 Version
------------------------------------------ -------------------- -------
ucs-4308-brdprog.1.0.12.gbin               Chassis Board Controller
                                                                1.0.12
ucs-b200-m3-brdprog.15.0.gbin              Board Controller     15.0
```

```
ucs-b200-m4-brdprog.12.0.gbin                Board Controller     12.0
ucs-b22-m3-brdprog.17.0.gbin                 Board Controller     17.0
ucs-b230-m2-pld.B230100D.gbin                Board Controller     B230100D
ucs-b250-m1-pld.111026-111026.gbin           Board Controller     111026-111026

ucs-b420-m3-brdprog.12.0.gbin                Board Controller     12.0
ucs-b420-m4-brdprog.6.0.gbin                 Board Controller     6.0
ucs-b440-m2-pld.B440100C-B4402008.gbin       Board Controller     B440100C-B440
                                                                                2008
ucs-c22-m3-brdprog.5.0.gbin                  Board Controller     5.0
ucs-c220-m3-brdprog.5.0.gbin                 Board Controller     5.0
ucs-c220-m4-brdprog.27.0.gbin                Board Controller     27.0
ucs-c240-m3-brdprog.5.0.gbin                 Board Controller     5.0
ucs-c240-m4-brdprog.24.0.gbin                Board Controller     24.0
ucs-c3260-brdprog.1.0.11.gbin                Board Controller     1.0.11
ucs-c3260-m3-brdprog.2.0.gbin                Board Controller     2.0
ucs-c460-m4-brdprog.16.0.gbin                Board Controller     16.0
ucs-EXM4-1-brdprog.7.0.gbin                  Board Controller     7.0
ucs-EXM4-2-brdprog.5.0.gbin                  Board Controller     5.0


UCS-A# /chassis/server/boardcontroller # show firmware
BoardController:
    Running-Vers: 1.0.11
    Package-Vers: 3.1(2)B
    Activate-Status: Ready

UCS-A# /chassis/server/boardcontroller # activate firmware 1.0.11
UCS-A# /chassis/server/boardcontroller* # commit-buffer
```

CHAPTER 8

Chassis Management

The Cisco UCS S3260 Chassis

Cisco UCS Manager Release 3.1(2) introduces support for the Cisco UCS S3260 chassis on Cisco UCS 6300 Series, and 6200 Series fabric interconnect setups.

The Cisco UCS S3260 chassis is a 4U chassis that is designed to operate in a standalone environment and also as part of the Cisco Unified Computing System. It has the following main components:

- Four 1050 Watt AC modular power supplies (2 + 2 shared and redundant mode of operation)
- Two System IO Controller (SIOC) slots
- Two storage server slots out of which one can be used for storage expansion

Note The second server slot in the chassis can be utilized by an HDD expansion tray module for an additional four 3.5” drives.

- 56 3.5” drive bays with an optional 4 x 3.5” HDD expansion tray module instead of the second server
- Up to 360 TB storage capacity by using 6 TB HDDs
- Serial Attached SCSI (SAS) expanders that can be configured to assign the 3.5” drives to individual server modules
- The two servers in the chassis can be replaced by a single, dual-height server with an IO expander

Acknowledging a Chassis

Perform the following procedure if you change the port that connects the chassis to the fabric interconnect. Acknowledging the chassis ensures that Cisco UCS Manager is aware of the change in the port.

After you change the port that connects the chassis to the fabric interconnect, wait for at least 1 minute before you reacknowledge the chassis.

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.

Example

The following example acknowledges chassis 3 and commits the transaction:

```
UCS-A# acknowledge chassis 3
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.

Decommissioning may take several minutes to complete.

Example

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.

Note You cannot remove a chassis if it is physically present in the system.

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.

Example

The following example removes chassis 3 and commits the transaction:

```
UCS-A# remove chassis 3
UCS-A* # commit-buffer
UCS-A #
```

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** [**multi-master** | **multi-slave**]	Turns on the chassis locator LED. • **multi-master**—Turns on the LED for the master node only. • **multi-slave**—Turns on the LED for the slave node only.

	Command or Action	Purpose
Step 3	UCS-A /chassis* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example turns on the locator LED for chassis 3 and commits the transaction:

```
UCS-A# scope chassis 3
UCS-A /chassis # enable locator-led
UCS-A /chassis* # commit-buffer
UCS-A /chassis #
```

The following example turns on the locator LED for the master node only and commits the transaction:

```
UCS-A# scope chassis 3
UCS-A /chassis # enable locator-led multi-master
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** [**multi-master** | **multi-slave**]	Turns off the chassis locator LED. • **multi-master**—Turns off the LED for the master node only. • **multi-slave**—Turns off the LED for the slave node only.
Step 3	UCS-A /chassis* # **commit-buffer**	Commits the transaction to the system configuration.

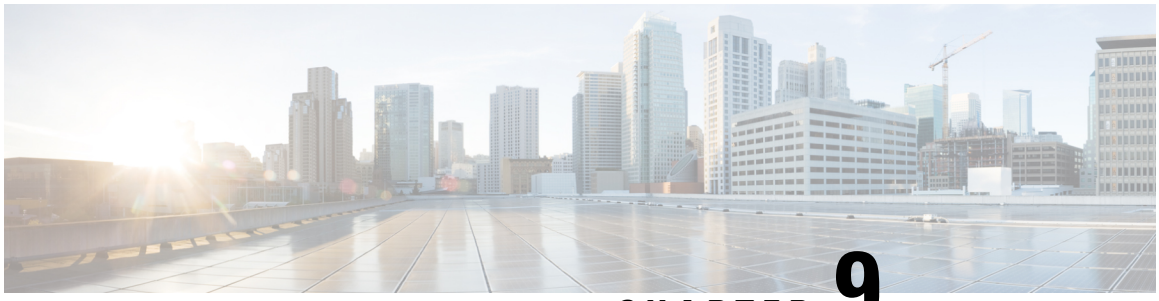
Example

The following example turns off the locator LED for chassis 3 and commits the transaction:

```
UCS-A# scope chassis 3
UCS-A /chassis # disable locator-led
UCS-A /chassis* # commit-buffer
UCS-A /chassis #
```

The following example turns off the locator LED for the master node and commits the transaction:

```
UCS-A# scope chassis 3
UCS-A /chassis # disable locator-led multi-master
UCS-A /chassis* # commit-buffer
UCS-A /chassis #
```



CHAPTER 9

Server Management

Cisco UCS S3260 Server Node Management

You can manage and monitor all Cisco UCS S3260 server nodes in a Cisco UCS domain through Cisco UCS Manager. You can perform some 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.

If a 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 rediscover the server in the slot.

Booting a Server from the Service Profile

Before you begin

Associate a service profile with a 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 server associated with the service profile.
Step 4	UCS-A /org/service-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example boots the 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 #
```

Acknowledging a 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 server.
Step 2	UCS-A*# **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example acknowledges server 1 in chassis 3 and commits the transaction:

```
UCS-A#  acknowledge server 3/1
UCS-A* # commit-buffer
UCS-A #
```

Power Cycling a Server

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 # **cycle** {**cycle-immediate** | **cycle-wait**}	Power cycles the server. Use the **cycle-immediate** keyword to immediately begin power cycling the server; use the **cycle-wait** keyword to schedule the power cycle to begin after all pending management operations have completed.
Step 3	UCS-A /chassis/server* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example immediately power cycles server 1 in chassis 3 and commits the transaction:

```
UCS-A#  scope server 3/1
UCS-A /chassis/server # cycle cycle-immediate
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Shutting Down a 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 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 server associated with the service profile.
Step 4	UCS-A /org/service-profile* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shuts down the 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 #
```

Performing a Hard Reset on a 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 server.

	Command or Action	Purpose
		Use the: • **hard-reset-immediate** keyword to immediately begin hard resetting the server. • **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.

Example

The following example performs an immediate hard reset of server 1 in chassis 3 and commits the transaction:

```
UCS-A#  scope server 3/1
UCS-A /chassis/server # reset hard-reset-immediate
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Resetting a Cisco UCS S3260 Server Node to Factory Default Settings

You can now reset a Cisco UCS S3260 Server Node to its factory settings. By default, the factory reset operation does not affect storage drives. This is to prevent any loss of data. However, you can choose to reset these devices to a known state as well.

The following guidelines apply to Cisco UCS S3260 Server Nodes when using scrub policies:

- For Cisco UCS S3260 Server Nodes, you cannot delete storage by using the scrub policy.
- Cisco UCS S3260 Server Nodes do not support FlexFlash drives.
- For Cisco UCS S3260 Server Nodes, you can only reset the BIOS by using the scrub policy.

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*	Enters chassis 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 **Note** This operation resets the BIOS. • **delete-flexflash-storage**—Resets the server to factory defaults and deletes flexflash storage **Note** This operation is not supported on Cisco UCS S3260 Server Nodes. • **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
Step 3	UCS-A /chassis/server* # **commit-buffer**	Commits any pending transactions.

Example

The following example resets the server settings to factory default without deleting storage, and commits the transaction:

```
UCS-A# scope server 3/1
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 3/1
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 3/1
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 3/1
UCS-A /chassis/server # reset factory-default delete-storage create-initial-storage-volumes
UCS-A /chassis/server* # commit-buffer
```

Removing a Server from a Chassis

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **remove server** *chassis-num* / *server-num*	Removes the specified server.
Step 2	UCS-A*# **commit-buffer**	Commits the transaction to the system configuration.
Step 3	Go to the physical location of the chassis and remove the server hardware from the slot.	For instructions on how to remove the server hardware, see the *Cisco UCS Hardware Installation Guide* for your chassis.

Example

The following example removes server 1 in chassis 3 and commits the transaction:

```
UCS-A#  remove server 3/1
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 Server, on page 94.

Decommissioning a Server

Procedure

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

Example

The following example decommissions server 1 in chassis 3 and commits the transaction:

```
UCS-A# decommission server 3/1
UCS-A* # commit-buffer
UCS-A #
```

Turning On the Locator LED for 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 # **enable locator-led** [**multi-master** | **multi-slave**]	Turns on the server locator LED. The following command options are not applicable to Cisco UCS S3260 Server Nodes: • **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.

Example

The following example turns on the locator LED on server 1 in chassis 3 and commits the transaction:

```
UCS-A# scope server 3/1
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 the master node only on server 1 in chassis 3 and commits the transaction:

```
UCS-A# scope chassis 3/1
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 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 server locator LED. The following command options are not applicable to Cisco UCS S3260 Server Nodes: • **multi-master**—Turns off the LED for the master node only. • **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.

Example

The following example turns off the locator LED on server 1 in chassis 3 and commits the transaction:

```
UCS-A# scope chassis 3/1
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 the master node on server 1 in chassis 3 and commits the transaction:

```
UCS-A# scope chassis 3/1
UCS-A /chassis/server # disable locator-led multi-master
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Resetting All Memory Errors

Use this procedure to reset all correctable and uncorrectable memory errors encountered by .

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-all-memory-errors**	Performs a reset of the memory cards.
Step 3	UCS-A /chassis/server* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example performs an immediate hard reset of server 1 in chassis 3 and commits the transaction:

```
UCS-A#  scope server 3/1
UCS-A /chassis/server # reset-all-memory-errors
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Resetting IPMI to Factory Default Settings

Perform the following procedure if you need to reset IPMI to factory default settings.

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-ipmi**	Resets IPMI settings to factory default.
Step 3	UCS-A /chassis/server* # **commit-buffer**	Commits any pending transactions.

Example

The following example resets the IPMI settings to factory default and commits the transaction:

```
UCS-A# scope server 3/1
UCS-A /chassis/server # reset-ipmi
UCS-A /chassis/server* # commit-buffer
```

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

Resetting the CIMC for a 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 server.
Step 4	UCS-A /chassis/server/cimc* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example resets the CIMC for server 1 in chassis 3 and commits the transaction:

```
UCS-A# scope server 3/1
UCS-A /chassis/server # scope cimc
UCS-A /chassis/server/cimc # reset
UCS-A /chassis/server/cimc* # commit-buffer
UCS-A /chassis/server/cimc #
```

Resetting the CMOS for a 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.

	Command or Action	Purpose
Step 2	UCS-A /chassis/server # **reset-cmos**	Resets the CMOS for the server.
Step 3	UCS-A /chassis/server* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example resets the CMOS for server 1 in chassis 3 and commits the transaction:

```
UCS-A# scope server 3/1
UCS-A /chassis/server # reset-cmos
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Resetting KVM

Perform the following procedure if you need to reset and clear all KVM sessions.

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-kvm**	Resets and clears all KVM sessions.
Step 3	UCS-A /chassis/server* # **commit-buffer**	Commits any pending transactions.

Example

The following example resets and clears all KVM sessions and commits the transaction:

```
UCS-A# scope server 3/1
UCS-A /chassis/server # reset-kvm
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Issuing an NMI from a 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*	Enters chassis 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.

Example

The following example sends an NMI from server 1 in chassis 3 and commits the transaction:

```
UCS-A# scope server 3/1
UCS-A /chassis/server # diagnostic-interrupt
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Recovering a Corrupt BIOS

On rare occasions, an issue with a 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 server boots with the running version of the firmware for that 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 # **recover-bios** *version*	Loads and activates the specified BIOS version.
Step 3	UCS-A /chassis/server* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to recover the BIOS:

```
UCS-A# scope server 3/1
UCS-A /chassis/server # recover-bios S5500.0044.0.3.1.010620101125
UCS-A /chassis/server* # commit-buffer
UCS-A /chassis/server #
```

Health LED Alarms

The server health LED is located on the front of each 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 server health LED blinks amber. This is indicated with a red dot. • Minor - The server health LED is amber. This is indicated with an orange dot.
Description column	A brief description of the alarm.
Sensor ID column	The ID of the sensor that triggered the alarm.
Sensor Name column	The name of the sensor that triggered the alarm.

Viewing Health LED Status

Procedure

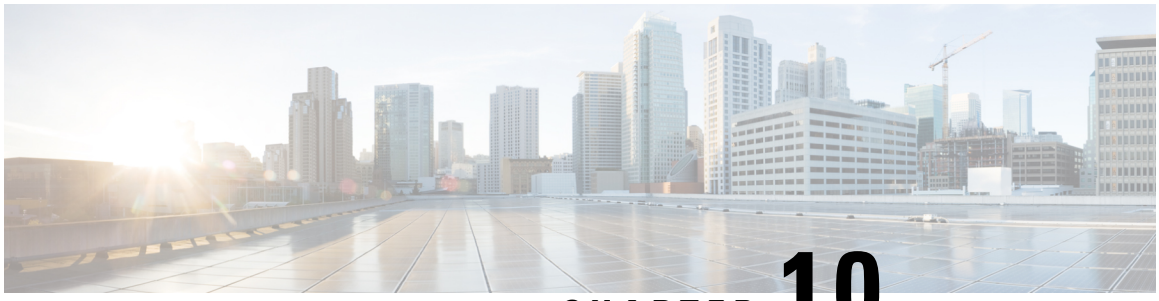
	Command or Action	Purpose
Step 1	UCS-A# **scope server** *chassis-id* / *server-id*	Enters chassis server mode for the specified server.
Step 2	UCS-A /chassis/server # **show health-led expand**	Displays the health LED and sensor alarms for the selected server.

Example

The following example shows how to display the health LED status and sensor alarms for chassis 1 server 3:

```
UCS-A# scope server 1/3
UCS-A /chassis/server # show health-led expand
Health LED:
    Severity: Normal
    Reason:
    Color: Green
    Oper State: On

UCS-A /chassis/server #
```



CHAPTER 10

SIOC Management

SIOC Management in Cisco UCS Manager

You can manage and monitor all System Input/Output Controllers (SIOC) in a Cisco UCS domain through Cisco UCS Manager.

SIOC Removal or Replacement

You can remove or replace an SIOC from a chassis. Removal or replacement of an SIOC is a service-affecting operation, which requires you to power down the entire chassis.

Guidelines for SIOC Removal

- To remove the active SIOC, or both SIOCs, shut down and remove power from the entire chassis. You must disconnect all power cords to completely remove power.
- Removal of SIOCs from a chassis results in the entire chassis being disconnected from Cisco UCS Manager.

SIOC Removal

Do the following to remove an SIOC from the system:

1. Shut down and remove power from the entire chassis. You must disconnect all power cords to completely remove power.
2. Disconnect the cables connecting the SIOC to the system.
3. Remove the SIOC from the system.

SIOC Replacement

Do the following to remove an SIOC from the system and replace it with another SIOC:

1. Shut down and remove power from the entire chassis. You must disconnect all power cords to completely remove power.
2. Disconnect the cables connecting the SIOC to the system.
3. Remove the SIOC from the system.
4. Connect the new SIOC to the system.
5. Connect the cables to the SIOC.
6. Connect power cords and then power on the system.
7. Acknowledge the new SIOC.

   The server connected to the replaced SIOC is rediscovered.

Note If the firmware of the replaced SIOC is not the same version as the peer SIOC, then it is recommended to update the firmware of the replaced SIOC by re-triggering chassis profile association.

Acknowledging an SIOC

Cisco UCS Manager has the ability to acknowledge a specific SIOC in a chassis. Perform the following procedure when you replace an SIOC in a chassis.

Caution This operation rebuilds the network connectivity between the SIOC and the fabric interconnects to which it is connected. The server corresponding to this SIOC becomes unreachable, and traffic is disrupted.

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 sioc** {**1** | **2**}	Acknowledges the specified SIOC in the chassis.
Step 3	UCS-A /chassis* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example acknowledges SIOC 1 and commits the transaction:

```
UCS-A# scope chassis 3
UCS-A /chassis # acknowledge sioc 1
UCS-A /chassis* # commit-buffer
UCS-A /chassis #
```

Resetting the CMC

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 sioc** {**1** | **2**}	Enters the specified SIOC in the chassis.
Step 3	UCS-A /chassis/sioc # **scope cmc**	Enters the CMC of the selected SIOC slot.
Step 4	UCS-A /chassis/sioc/cmc # **reset**	Resets the CMC.
Step 5	UCS-A /chassis/sioc/cmc* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example resets the CMC on SIOC 1 and commits the transaction:

```
UCS-A# scope chassis 1
UCS-A /chassis # scope sioc 1
UCS-A /chassis/sioc # scope cmc
UCS-A /chassis/sioc/cmc # reset
UCS-A /chassis/sioc/cmc* # commit-buffer
```

CMC Secure Boot

With Chassis Management Controller (CMC) secure boot, only Cisco-signed firmware images can be installed and run on the CMC. When the CMC 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 CMC firmware.

Guidelines and Limitations for CMC Secure Boot

- CMC secure boot is supported only on the Cisco UCS S3260 chassis.
- When chassis association is in progress, enabling secure boot on one of the SIOCs will result in a failed operation.
- After CMC secure boot is enabled, it cannot be disabled.
- CMC secure boot is specific to the SIOC on which it is enabled. If you replace the SIOC on which CMC secure boot is enabled, the **Secure boot operational state** field will now display the secure boot status of the new SIOC.
- After CMC secure boot is enabled on a chassis, you cannot move the chassis back to standalone mode and downgrade the firmware to a CMC firmware image earlier than Cisco IMC Release 2.0(13).

- The **Secure boot operational state** field shows the secure boot status. This can be one of the following:
  - **Disabled**—When CMC secure boot is not enabled. This is the default state.
  - **Enabling**—When CMC secure boot is being enabled.
  - **Enabled**—When CMC secure boot is enabled.

Enabling CMC Secure Boot

Cisco UCS Manager Release 3.1(2) introduces the ability to enable Chassis Management Controller (CMC) secure boot so that only Cisco-signed firmware images can be installed and run on the CMC.

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 sioc** {**1** | **2**}	Enters the specified SIOC in the chassis.
Step 3	UCS-A /chassis/sioc # **scope cmc**	Enters the CMC of the selected SIOC slot.
Step 4	UCS-A /chassis/sioc/cmc # **enable secure-boot**	Enables CMC secure boot. If you run this command when the secure boot state is **enabled**, Cisco UCS Manager will display an error message and the operation will fail. **Note** This is an irreversible operation. You cannot disable CMC secure boot.
Step 5	UCS-A /chassis/sioc/cmc* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example enables CMC secure boot on SIOC 1 and commits the transaction:

```
UCS-A# scope chassis 1
UCS-A /chassis # scope sioc 1
UCS-A /chassis/sioc # scope cmc
UCS-A /chassis/sioc/cmc # enable secure-boot
Warning: This is an irreversible operation.
Do you want to proceed? [Y/N] Y
UCS-A /chassis/sioc/cmc* # commit-buffer
```