Configuring Network-Related Policies

This chapter includes the following sections:

Configuring vNIC Templates

vNIC Template

This policy defines how a vNIC on a server connects to the LAN. This policy is also referred to as a vNIC LAN connectivity policy.

Cisco UCS Manager does not automatically create a VM-FEX port profile with the correct settings when you create a vNIC template. If you want to create a VM-FEX port profile, you must configure the target of the vNIC template as a VM.

You need to include this policy in a service profile for it to take effect.

Note If your server has two Emulex or QLogic NICs (Cisco UCS CNA M71KR-E or Cisco UCS CNA M71KR-Q), you must configure vNIC policies for both adapters in your service profile to get a user-defined MAC address for both NICs. If you do not configure policies for both NICs, Windows still detects both of them in the PCI bus. Then because the second eth is not part of your service profile, Windows assigns it a hardware MAC address. If you then move the service profile to a different server, Windows sees additional NICs because one NIC did not have a user-defined MAC address.

Configuring a vNIC Template

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **create vnic-templ** *vnic-templ-name* [**eth-if** *vlan-name*] [**fabric** {**a** | **b**}] [**target** [**adapter** | **vm**]]	Creates a vNIC template and enters organization vNIC template mode. The target you choose determines whether or not Cisco UCS Manager automatically creates a VM-FEX port profile with the appropriate settings for the vNIC template. This can be one of the following: • **Adapter—**The vNICs apply to all adapters. No VM-FEX port profile is created if you choose this option. • **VM—**The vNICs apply to all virtual machines. A VM-FEX port profile is created if you choose this option.
Step 3	UCS-A /org/vnic-templ # **set descr** *description*	(Optional) Provides a description for the vNIC template.
Step 4	UCS-A /org/vnic-templ # **set fabric** {**a** | **a-b** | **b** | **b-a**}	(Optional) Specifies the fabric to use for the vNIC. If you did not specify the fabric when creating the vNIC template in Step 2, you have the option to specify it with this command. If you want this vNIC to be able to access the second fabric interconnect if the default one is unavailable, choose **a-b** (A is the primary) or **b-a** (B is the primary) .

	Command or Action	Purpose
		Note Do not enable fabric failover for the vNIC under the following circumstances: • If the Cisco UCS domain is running in Ethernet Switch Mode. vNIC fabric failover is not supported in that mode. If all Ethernet uplinks on one fabric interconnect fail, the vNICs do not fail over to the other. • If you plan to associate this vNIC with a server that has an adapter which does not support fabric failover, such as the Cisco UCS 82598KR-CI 10-Gigabit Ethernet Adapter. If you do so, Cisco UCS Manager generates a configuration fault when you associate the service profile with the server.
Step 5	UCS-A /org/vnic-templ # **set mac-pool** *mac-pool-name*	The MAC address pool that vNICs created from this vNIC template should use.
Step 6	UCS-A /org/vnic-templ # **set mtu** *mtu-value*	The maximum transmission unit, or packet size, that vNICs created from this vNIC template should use. Enter an integer between 1500 and 9216. **Note** If the vNIC template has an associated QoS policy, the MTU specified here must be equal to or less than the MTU specified in the associated QoS system class. If this MTU value exceeds the MTU value in the QoS system class, packets may be dropped during data transmission.
Step 7	UCS-A /org/vnic-templ # **set nw-control-policy** *policy-name*	The network control policy that vNICs created from this vNIC template should use.
Step 8	UCS-A /org/vnic-templ # **set pin-group** *group-name*	The LAN pin group that vNICs created from this vNIC template should use.
Step 9	UCS-A /org/vnic-templ # **set qos-policy** *policy-name*	The quality of service policy that vNICs created from this vNIC template should use.
Step 10	UCS-A /org/vnic-templ # **set stats-policy** *policy-name*	The statistics collection policy that vNICs created from this vNIC template should use.
Step 11	UCS-A /org/vnic-templ # **set type** {**initial-template** | **updating-template**}	Specifies the vNIC template update type. If you do not want vNIC instances created from this template to be automatically updated when the template is updated, use the **initial-template** keyword; otherwise, use the **updating-template** keyword to ensure that all vNIC instances are updated when the vNIC template is updated.
Step 12	UCS-A /org/vnic-templ # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures a vNIC template and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create vnic template VnicTempFoo
UCS-A /org/vnic-templ* # set descr "This is a vNIC template example."
UCS-A /org/vnic-templ* # set fabric a
UCS-A /org/vnic-templ* # set mac-pool pool137
UCS-A /org/vnic-templ* # set mtu 8900
UCS-A /org/vnic-templ* # set nw-control-policy ncp5
UCS-A /org/vnic-templ* # set pin-group PinGroup54
UCS-A /org/vnic-templ* # set qos-policy QosPol5
UCS-A /org/vnic-templ* # set stats-policy ServStatsPolicy
UCS-A /org/vnic-templ* # set type updating-template
UCS-A /org/vnic-templ* # commit-buffer
UCS-A /org/vnic-templ #
```

Deleting a vNIC Template

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **delete vnic-templ** *vnic-templ-name*	Deletes the specified vNIC template.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the vNIC template named VnicTemp42 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete vnic template VnicTemp42
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Ethernet Adapter Policies

Ethernet and Fibre Channel Adapter Policies

These policies govern the host-side behavior of the adapter, including how the adapter handles traffic. For example, you can use these policies to change default settings for the following:

- Queues
- Interrupt handling
- Performance enhancement

- RSS hash
- Failover in an cluster configuration with two fabric interconnects

Note For Fibre Channel adapter policies, the values displayed by Cisco UCS Manager may not match those displayed by applications such as QLogic SANsurfer. For example, the following values may result in an apparent mismatch between SANsurfer and Cisco UCS Manager:

- Max LUNs Per Target—SANsurfer has a maximum of 256 LUNs and does not display more than that number. Cisco UCS Manager supports a higher maximum number of LUNs.
- Link Down Timeout—In SANsurfer, you configure the timeout threshold for link down in seconds. In Cisco UCS Manager, you configure this value in milliseconds. Therefore, a value of 5500 ms in Cisco UCS Manager displays as 5s in SANsurfer.
- Max Data Field Size—SANsurfer has allowed values of 512, 1024, and 2048. Cisco UCS Manager allows you to set values of any size. Therefore, a value of 900 in Cisco UCS Manager displays as 512 in SANsurfer.

Operating System Specific Adapter Policies

By default, Cisco UCS provides a set of Ethernet adapter policies and Fibre Channel adapter policies. These policies include the recommended settings for each supported server operating system. Operating systems are sensitive to the settings in these policies. Storage vendors typically require non-default adapter settings. You can find the details of these required settings on the support list provided by those vendors.

Important We recommend that you use the values in these policies for the applicable operating system. Do not modify any of the values in the default policies unless directed to do so by Cisco Technical Support.

However, if you are creating an Ethernet adapter policy for a Windows OS (instead of using the default Windows adapter policy), you must use the following formulas to calculate values that work with Windows:

Completion Queues = Transmit Queues + Receive Queues
Interrupt Count = (Completion Queues + 2) rounded up to nearest power of 2

For example, if Transmit Queues = 1 and Receive Queues = 8 then:

Completion Queues = 1 + 8 = 9
Interrupt Count = (9 + 2) rounded up to the nearest power of 2 = 16

Accelerated Receive Flow Steering

Accelerated Receive Flow Steering (ARFS) is hardware-assisted receive flow steering that can increase CPU data cache hit rate by steering kernel level processing of packets to the CPU where the application thread consuming the packet is running.

Using ARFS can improve CPU efficiency and reduce traffic latency. Each receive queue of a CPU has an interrupt associated with it. You can configure the Interrupt Service Routine (ISR) to run on a CPU. The ISR

moves the packet from the receive queue to the backlog of one of the current CPUs, which processes the packet later. If the application is not running on this CPU, the CPU must copy the packet to non-local memory, which adds to latency. ARFS can reduce this latency by moving that particular stream to the receive queue of the CPU on which the application is running.

ARFS is disabled by default and can be enabled through Cisco UCS Manager. To configure ARFS, do the following:

1. Create an adapter policy with ARFS enabled.
2. Associate the adapter policy with a service profile.
3. Enable ARFS on a host.
    1. Turn off Interrupt Request Queue (IRQ) balance.
    2. Associate IRQ with different CPUs.
    3. Enable ntuple by using ethtool.

Guidelines and Limitations for Accelerated Receive Flow Steering

- ARFS supports 64 filters per vNIC
- ARFS is supported on the following adapters:
    - Cisco UCS VIC 1280, 1240, 1340, and 1380
    - Cisco UCS VIC 1225, 1225T, 1285, 1223, 1227T, and 1385
- ARFS is supported on the following Operating Systems:
    - Red Hat Enterprise Linux 6.5, and 6.6
    - Red Hat Enterprise Linux 7.0 and higher versions
    - SUSE Linux Enterprise Server 11 SP2 and SP3
    - SUSE Linux Enterprise Server 12 and higher versions
    - Ubuntu 14.04.2

Interrupt Coalescing

Adapters typically generate a large number of interrupts that a host CPU must service. Interrupt coalescing reduces the number of interrupts serviced by the host CPU. This is done by interrupting the host only once for multiple occurrences of the same event over a configurable coalescing interval.

When interrupt coalescing is enabled for receive operations, the adapter continues to receive packets, but the host CPU does not immediately receive an interrupt for each packet. A coalescing timer starts when the first packet is received by the adapter. When the configured coalescing interval times out, the adapter generates one interrupt with the packets received during that interval. The NIC driver on the host then services the multiple packets that are received. Reduction in the number of interrupts generated reduces the time spent by the host CPU on context switches. This means that the CPU has more time to process packets, which results in better throughput and latency.

Adaptive Interrupt Coalescing

Due to the coalescing interval, the handling of received packets adds to latency. For small packets with a low packet rate, this latency increases. To avoid this increase in latency, the driver can adapt to the pattern of traffic flowing through it and adjust the interrupt coalescing interval for a better response from the server.

Adaptive interrupt coalescing (AIC) is most effective in connection-oriented low link utilization scenarios including email server, databases server, and LDAP server. It is not suited for line-rate traffic.

Guidelines and Limitations for Adaptive Interrupt Coalescing

- Adaptive Interrupt Coalescing (AIC) does not provide any reduction in latency when the link utilization is more than 80 percent.
- Enabling AIC disables static coalescing.
- AIC is supported on the following Operating Systems:
  - Red Hat Enterprise Linux 6.4 and higher versions
  - Red Hat Enterprise Linux 7.0 and higher versions
  - SUSE Linux Enterprise Server 11 SP2 and SP3
  - SUSE Linux Enterprise Server 12
  - XenServer 6.5
  - Ubuntu 14.04.2

RDMA Over Converged Ethernet for SMB Direct

RDMA over Converged Ethernet (RoCE) allows direct memory access over an Ethernet network. RoCE is a link layer protocol, and hence, it allows communication between any two hosts in the same Ethernet broadcast domain. RoCE delivers superior performance compared to traditional network socket implementations because of lower latency, lower CPU utilization and higher utilization of network bandwidth. Windows 2012 and later versions use RDMA for accelerating and improving the performance of SMB file sharing and Live Migration.

Cisco UCS Manager Release 2.2(4) supports RoCE for Microsoft SMB Direct. It sends additional configuration information to the adapter while creating or modifying an Ethernet adapter policy.

Guidelines and Limitations for SMB Direct with RoCE

- Microsoft SMB Direct with RoCE is supported only on Windows 2012 R2.
- Microsoft SMB Direct withRoCE is supported only with Cisco UCS VIC 1340 and 1380 adapters.
- Cisco UCS Manager does not support more than 4 RoCE-enabled vNICs per adapter.
- Cisco UCS Manager does not support RoCE with NVGRE, VXLAN, NetFlow, VMQ, or usNIC.
- Maximum number of queue pairs per adapter is 8192.
- Maximum number of memory regions per adapter is 524288.

- If you do not disable RoCE before downgrading Cisco UCS Manager from Release 2.2(4), downgrade will fail.

Configuring an Ethernet Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 2	UCS-A /org # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 3	UCS-A /org/eth-policy # **set arfs accelaratedrfs** {**enabled** | **disabled**}	(Optional) Configures Accelerated RFS.
Step 4	UCS-A /org/eth-policy # **set comp-queue count** *count*	(Optional) Configures the Ethernet completion queue.
Step 5	UCS-A /org/eth-policy # **set descr** *description*	(Optional) Provides a description for the policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 6	UCS-A /org/eth-policy # **set failover timeout** *timeout-sec*	(Optional) Configures the Ethernet failover.
Step 7	UCS-A /org/eth-policy # **set interrupt** {**coalescing-time** *sec* | **coalescing-type** {**idle** | **min**} | **count** *count* | **mode** {**intx** | **msi** | **msi-x**}}	(Optional) Configures the Ethernet interrupt.
Step 8	UCS-A /org/eth-policy # **set nvgre adminstate** {**disabled** | **enabled**}	(Optional) Configures NVGRE.
Step 9	UCS-A /org/eth-policy # **set offload** {**large-receive** | **tcp-rx-checksum** | **tcp-segment** | **tcp-tx-checksum**} {**disabled** | **enabled**}	(Optional) Configures the Ethernet offload.
Step 10	UCS-A /org/eth-policy # **set policy-owner** {**local** | **pending**}	(Optional) Specifies the owner for the Ethernet adapter policy.

	Command or Action	Purpose
Step 11	UCS-A /org/eth-policy # **set recv-queue** {**count** *count* | **ring-size** *size-num*}	(Optional) Configures the Ethernet receive queue.
Step 12	UCS-A /org/eth-policy # **set roce adminstate** {**disabled** | **enabled**} | **memoryregions** *number-of-memory-regions* | **queuepairs** *number-of-queue-pairs* | **resourcegroups** *number-of-resource-groups*	(Optional) Configures RDMA over converged Ethernet (RoCE) by using the following options: • **adminstate—**Enables or disables RoCE. • **memoryregion**s**—**Configures the number of memory regions to be used per adapter. The values range from 1-524288 memory regions, and should be an integer rounded up to the nearest power of 2. • **queuepairs—**Configures the number of queue pairs to be used per adapter. The values range from 1-8192 queue pairs, and should be an integer rounded up to the nearest power of 2. • **resourcegroups—**Configures the number of resource groups to be used. The values range from 1-128 resource groups. The value should be an integer rounded up to the nearest power of 2 and greater than or equal to the number of CPU cores on the system for optimum performance.
Step 13	UCS-A /org/eth-policy # **set rss receivesidescaling** {**disabled** | **enabled**}	(Optional) Configures the RSS.
Step 14	UCS-A /org/eth-policy # **set trans-queue** {**count** *count* | **ring-size** *size-num*}	(Optional) Configures the Ethernet transmit queue.
Step 15	UCS-A /org/eth-policy # **set vxlan adminstate** {**disabled** | **enabled**}	(Optional) Configures VXLAN.
Step 16	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example configures an Ethernet adapter policy, and commits the transaction:

```
UCS-A# scope org
UCS-A /org* # create eth-policy EthPolicy19
UCS-A /org/eth-policy* # set comp-queue count 16
UCS-A /org/eth-policy* # set descr "This is an Ethernet adapter policy example."
UCS-A /org/eth-policy* # set failover timeout 300
UCS-A /org/eth-policy* # set interrupt count 64
UCS-A /org/eth-policy* # set offload large-receive disabled
UCS-A /org/eth-policy* # set recv-queue count 32
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set trans-queue
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

The following example configures an Ethernet adapter policy with RoCE, and commits the transaction:

```
UCS-A# scope org
UCS-A /org* # create eth-policy EthPolicy20
UCS-A /org/eth-policy* # set roce adminstate enable
UCS-A /org/eth-policy* # set roce memoryregions 131072
UCS-A /org/eth-policy* # set roce queuepairs 256
UCS-A /org/eth-policy* # set roce resourcegroups 32
UCS-A /org/eth-policy # commit buffer
UCS-A /org # show eth-policy EthPolicy20 detail expand

Eth Adapter Policy:
    Name: EthPolicy20
    Description:
    Policy Owner: Local

    ARFS:
        Accelarated Receive Flow Steering: Disabled

    Ethernet Completion Queue:
        Count: 2

    Ethernet Failback:
        Timeout (sec): 5

    Ethernet Interrupt:
        Coalescing Time (us): 125
        Coalescing Type: Min
        Count: 4
        Driver Interrupt Mode: MSI-X

    NVGRE:
        NVGRE: Disabled

    Ethernet Offload:
        Large Receive: Enabled
        TCP Segment: Enabled
        TCP Rx Checksum: Enabled
        TCP Tx Checksum: Enabled

    Ethernet Receive Queue:
        Count: 1
        Ring Size: 512

    ROCE:
        RoCE: Enabled
        Resource Groups: 32
        Memory Regions: 131072
        Queue Pairs: 256

    VXLAN:
        VXLAN: Disabled

    Ethernet Transmit Queue:
        Count: 1
        Ring Size: 256

    RSS:
        Receive Side Scaling: Disabled
```

Configuring an Ethernet Adapter Policy to Enable eNIC Support for MRQS on Linux Operating Systems

Cisco UCS Manager includes eNIC support for the Multiple Receive Queue Support (MRQS) feature on Red Hat Enterprise Linux Version 6.x and SUSE Linux Enterprise Server Version 11.x.

Procedure

Step 1 Create an Ethernet adapter policy.
Use the following parameters when creating the Ethernet adapter policy:

- Transmit Queues = 1
- Receive Queues = n (up to 8)
- Completion Queues = # of Transmit Queues + # of Receive Queues
- Interrupts = # Completion Queues + 2
- Receive Side Scaling (RSS) = Enabled
- Interrupt Mode = Msi-X

See Configuring an Ethernet Adapter Policy, on page 8.

Step 2 Install an eNIC driver Version 2.1.1.35 or later.
See Cisco UCS Virtual Interface Card Drivers for Linux Installation Guide.

Step 3 Reboot the server

Configuring an Ethernet Adapter Policy to Enable Stateless Offloads with NVGRE

Cisco UCS Manager supports stateless offloads with NVGRE only with Cisco UCS VIC 1340 and/or Cisco UCS VIC 1380 adapters that are installed on servers running Windows Server 2012 R2 operating systems. Stateless offloads with NVGRE cannot be used with NetFlow, usNIC, VM-FEX, or VMQ.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 2	UCS-A /org # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 3	To enable stateless offloads with NVGRE, set the following options:	• Transmit Queues = 1 • Receive Queues = n (up to 8) • Completion Queues = # of Transmit Queues + # of Receive Queues • Interrupts = # Completion Queues + 2 • Network Virtualization using Generic Routing Encapsulation = Enabled

	Command or Action	Purpose
		• Interrupt Mode = Msi-X For more information on creating an Ethernet adapter policy, see Configuring an Ethernet Adapter Policy, on page 8.
Step 4	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.
Step 5	Install an eNIC driver Version 3.0.0.8 or later.	For more information, see http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/sw/vic_drivers/install/Windows/b_Cisco_VIC_Drivers_for_Windows_Installation_Guide.html.
Step 6	Reboot the server.	

The following example shows how to configure an Ethernet adapter policy to enable stateless offloads with NVGRE and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create eth-policy NVGRE
UCS-A /org/eth-policy* # set descr "Ethernet adapter policy with stateless offloads"
UCS-A /org/eth-policy* # set nvgre adminstate enabled
UCS-A /org/eth-policy* # set comp-queue count 16
UCS-A /org/eth-policy* # set interrupt count 64
UCS-A /org/eth-policy* # set recv-queue count 32
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set trans-queue 1
UCS-A /org/eth-policy* # set interrupt mode mxi-x
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

Configuring an Ethernet Adapter Policy to Enable Stateless Offloads with VXLAN

Cisco UCS Manager supports stateless offloads with VXLAN only with Cisco UCS VIC 1340 and/or Cisco UCS VIC 1380 adapters that are installed on servers running VMWare ESXi Release 5.5 and later releases of the operating system. Stateless offloads with VXLAN cannot be used with NetFlow, usNIC, VM-FEX, or VMQ.

Procedure

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

	Command or Action	Purpose
Step 3	To enable stateless offloads with VXLAN, set the following options:	• Transmit Queues = 1 • Receive Queues = n (up to 8) • Completion Queues = # of Transmit Queues + # of Receive Queues • Interrupts = # Completion Queues + 2 • Virtual Extensible LAN = Enabled • Interrupt Mode = Msi-X For more information on creating an Ethernet adapter policy, see Configuring an Ethernet Adapter Policy, on page 8.
Step 4	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.
Step 5	Install an eNIC driver Version 2.1.2.59 or later.	For more information, see http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/sw/vic_drivers/install/ESX/2-0/b_Cisco_VIC_Drivers_for_ESX_Installation_Guide.html.
Step 6	Reboot the server.	

The following example shows how to configure an Ethernet adapter policy to enable stateless offloads with VXLAN and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create eth-policy VXLAN
UCS-A /org/eth-policy* # set descr "Ethernet adapter policy with stateless offloads"
UCS-A /org/eth-policy* # set vxlan adminstate enabled
UCS-A /org/eth-policy* # set comp-queue count 16
UCS-A /org/eth-policy* # set interrupt count 64
UCS-A /org/eth-policy* # set recv-queue count 32
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set trans-queue 1
UCS-A /org/eth-policy* # set interrupt mode mxi-x
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

Deleting an Ethernet Adapter Policy

Procedure

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

	Command or Action	Purpose
Step 2	UCS-A /org # **delete eth-policy** *policy-name*	Deletes the specified Ethernet adapter policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the Ethernet adapter policy named EthPolicy19 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete eth-policy EthPolicy19
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring the Default vNIC Behavior Policy

Default vNIC Behavior Policy

Default vNIC behavior policy allows you to configure how vNICs are created for a service profile. You can choose to create vNICS manually, or you can allow them to be created automatically

You can configure the default vNIC behavior policy to define how vNICs are created. This can be one of the following:

- **None—**Cisco UCS Manager does not create default vNICs for a service profile. All vNICs must be explicitly created.
- **HW Inherit—**If a service profile requires vNICs and none have been explicitly defined, Cisco UCS Manager creates the required vNICs based on the adapter installed in the server associated with the service profile.

Note If you do not specify a default behavior policy for vNICs, **HW Inherit** is used by default.

Configuring a Default vNIC Behavior Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A/org # **scope vnic-beh-policy**	Enters default vNIC behavior policy mode.

	Command or Action	Purpose
Step 3	UCS-A/org/vnic-beh-policy # **set action** {**hw-inherit** [**template_name** *name*] | **none**}	Specifies the default vNIC behavior policy. This can be one of the following: • **hw-inherit—**If a service profile requires vNICs and none have been explicitly defined, Cisco UCS Manager creates the required vNICs based on the adapter installed in the server associated with the service profile. If you specify **hw-inherit**, you can also specify a vNIC template to create the vNICs. • **none—**Cisco UCS Manager does not create default vNICs for a service profile. All vNICs must be explicitly created.
Step 4	UCS-A/org/vnic-beh-policy # **commit-buffer**	Commits the transaction to the system configuration.

This example shows how to set the default vNIC behavior policy to **hw-inherit**:

```
UCS-A # scope org /
UCS-A/org # scope vnic-beh-policy
UCS-A/org/vnic-beh-policy # set action hw-inherit
UCS-A/org/vnic-beh-policy* # commit-buffer
UCS-A/org/vnic-beh-policy #
```

Configuring LAN Connectivity Policies

LAN and SAN Connectivity Policies

Connectivity policies determine the connections and the network communication resources between the server and the LAN or SAN on the network. These policies use pools to assign MAC addresses, WWNs, and WWPNs to servers and to identify the vNICs and vHBAs that the servers use to communicate with the network.

Note We do not recommend that you use static IDs in connectivity policies, because these policies are included in service profiles and service profile templates and can be used to configure multiple servers.

Privileges Required for LAN and SAN Connectivity Policies

Connectivity policies enable users without network or storage privileges to create and modify service profiles and service profile templates with network and storage connections. However, users must have the appropriate network and storage privileges to create connectivity policies.

Privileges Required to Create Connectivity Policies

Connectivity policies require the same privileges as other network and storage configurations. For example, you must have at least one of the following privileges to create connectivity policies:

- admin—Can create LAN and SAN connectivity policies
- ls-server—Can create LAN and SAN connectivity policies
- ls-network—Can create LAN connectivity policies
- ls-storage—Can create SAN connectivity policies

Privileges Required to Add Connectivity Policies to Service Profiles

After the connectivity policies have been created, a user with ls-compute privileges can include them in a service profile or service profile template. However, a user with only ls-compute privileges cannot create connectivity policies.

Interactions between Service Profiles and Connectivity Policies

You can configure the LAN and SAN connectivity for a service profile through either of the following methods:

- LAN and SAN connectivity policies that are referenced in the service profile
- Local vNICs and vHBAs that are created in the service profile
- Local vNICs and a SAN connectivity policy
- Local vHBAs and a LAN connectivity policy

Cisco UCS maintains mutual exclusivity between connectivity policies and local vNIC and vHBA configuration in the service profile. You cannot have a combination of connectivity policies and locally created vNICs or vHBAs. When you include a LAN connectivity policy in a service profile, all existing vNIC configuration is erased, and when you include a SAN connectivity policy, all existing vHBA configuration in that service profile is erased.

Creating a LAN Connectivity Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **create lan-connectivity-policy** *policy-name*	Creates the specified LAN connectivity policy, and enters organization LAN connectivity policy mode. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object has been saved.

	Command or Action	Purpose
Step 3	UCS-A /org/lan-connectivity-policy # **set descr** *policy-name*	(Optional) Adds a description to the policy. We recommend that you include information about where and how the policy should be used. Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote).
Step 4	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy* # set descr "LAN connectivity policy"
UCS-A /org/lan-connectivity-policy* # commit-buffer
UCS-A /org/lan-connectivity-policy #
```

What to Do Next

Add one or more vNICs and/or iSCSI vNICs to this LAN connectivity policy.

Creating a vNIC for a LAN Connectivity Policy

If you are continuing from Creating a LAN Connectivity Policy, on page 16, begin this procedure at Step 3.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **scope lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **create vnic** *vnic-name* [**eth-if** *eth-if-name*] [**fabric** {**a** | **b**}]	Creates a vNIC for the specified LAN connectivity policy. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object has been saved.
Step 4	UCS-A /org/lan-connectivity-policy/vnic # **set fabric** {**a** | **a-b** | **b** | **b-a**}	Specifies the fabric to use for the vNIC. If you did not specify the fabric when you created the vNIC in Step 3, you have the option to specify it with this command.

	Command or Action	Purpose
		If you want this vNIC to be able to access the second fabric interconnect if the default one is unavailable, choose **a-b** (A is the primary) or **b-a** (B is the primary) . **Note** Do not enable fabric failover for the vNIC under the following circumstances: • If the Cisco UCS domain is running in Ethernet Switch Mode. vNIC fabric failover is not supported in that mode. If all Ethernet uplinks on one fabric interconnect fail, the vNICs do not fail over to the other. • If you plan to associate this vNIC with a server that has an adapter which does not support fabric failover, such as the Cisco UCS 82598KR-CI 10-Gigabit Ethernet Adapter. If you do so, Cisco UCS Manager generates a configuration fault when you associate the service profile with the server.
Step 5	UCS-A /org/lan-connectivity-policy/vnic # **set adapter-policy** *policy-name*	Specifies the adapter policy to use for the vNIC.
Step 6	UCS-A /org/lan-connectivity-policy/vnic # **set identity** {**dynamic-mac** {*mac-addr* | **derived**} | **mac-pool** *mac-pool-name*}	Specifies the identity (MAC address) for the vNIC. You can set the identity using one of the following options: • Create a unique MAC address in the form *nn***:***nn***:***nn* **:***nn***:***nn*. • Derive the MAC address from one burned into the hardware at manufacture. • Assign a MAC address from a MAC pool.
Step 7	UCS-A /org/lan-connectivity-policy/vnic # **set mtu** *size-num*	Specifies the maximum transmission unit, or packet size, that this vNIC accepts. Enter an integer between 1500 and 9216. **Note** If the vNIC has an associated QoS policy, the MTU specified here must be equal to or less than the MTU specified in the associated QoS system class. If this MTU value exceeds the MTU value in the QoS system class, packets might get dropped during data transmission.
Step 8	UCS-A /org/lan-connectivity-policy/vnic # **set nw-control-policy** *policy-name*	Specifies the network control policy that the vNIC should use.

	Command or Action	Purpose
Step 9	UCS-A /org/lan-connectivity-policy/vnic # **set order** {*order-num* | **unspecified**}	Specifies the relative order for the vNIC.
Step 10	UCS-A /org/lan-connectivity-policy/vnic # **set pin-group** *group-name*	Specifies the LAN pin group that the vNIC should use.
Step 11	UCS-A /org/lan-connectivity-policy/vnic # **set qos-policy** *policy-name*	Specifies the quality of service policy that the vNIC should use.
Step 12	UCS-A /org/lan-connectivity-policy/vnic # **set stats-policy** *policy-name*	Specifies the statistics collection policy that the vNIC should use.
Step 13	UCS-A /org/lan-connectivity-policy/vnic # **set template-name** *policy-name*	Specifies the dynamic vNIC connectivity policy to use for the vNIC.
Step 14	UCS-A /org/lan-connectivity-policy/vnic # **set vcon** {**1** | **2** | **3** | **4** | **any**}	Assigns the vNIC to the specified vCon. Use the **any** keyword to have Cisco UCS Manager automatically assign the vNIC.
Step 15	UCS-A /org/lan-connectivity-policy/vnic # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to configure a vNIC for a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy* # create vnic vnic3 fabric a
UCS-A /org/lan-connectivity-policy/vnic* # set fabric a-b
UCS-A /org/lan-connectivity-policy/vnic* # set adapter-policy AdaptPol2
UCS-A /org/lan-connectivity-policy/vnic* # set identity mac-pool MacPool3
UCS-A /org/lan-connectivity-policy/vnic* # set mtu 8900
UCS-A /org/lan-connectivity-policy/vnic* # set nw-control-policy ncp5
UCS-A /org/lan-connectivity-policy/vnic* # set order 0
UCS-A /org/lan-connectivity-policy/vnic* # set pin-group EthPinGroup12
UCS-A /org/lan-connectivity-policy/vnic* # set qos-policy QosPol5
UCS-A /org/lan-connectivity-policy/vnic* # set stats-policy StatsPol2
UCS-A /org/lan-connectivity-policy/vnic* # set template-name VnicConnPol3
UCS-A /org/lan-connectivity-policy/vnic* # set vcon any
UCS-A /org/lan-connectivity-policy/vnic* # commit-buffer
UCS-A /org/lan-connectivity-policy/vnic #
```

What to Do Next

If desired, add another vNIC or an iSCSI vNIC to the LAN connectivity policy. If not, include the policy in a service profile or service profile template.

Deleting a vNIC from a LAN Connectivity Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **scope lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **delete vnic** *vnic-name*	Deletes the specified vNIC from the LAN connectivity policy.
Step 4	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete a vNIC named vnic3 from a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy # delete vnic vnic3
UCS-A /org/lan-connectivity-policy* # commit-buffer
UCS-A /org/lan-connectivity-policy #
```

Creating an iSCSI vNIC for a LAN Connectivity Policy

If you are continuing from Creating a LAN Connectivity Policy, on page 16, begin this procedure at Step 3.

Before You Begin

The LAN connectivity policy must include an Ethernet vNIC that can be used as the overlay vNIC for the iSCSI device.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **scope lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **create vnic-iscsi** *iscsi-vnic-name* .	Creates an iSCSI vNIC for the specified LAN connectivity policy.

	Command or Action	Purpose
		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 has been saved.
Step 4	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set iscsi-adaptor-policy** *iscsi-adaptor-name*	(Optional) Specifies the iSCSI adapter policy that you have created for this iSCSI vNIC.
Step 5	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set auth-name** *authentication-profile-name*	(Optional) Sets the authentication profile to be used by the iSCSI vNIC. The authentication profile must already exist for it to be set. For more information, see Creating an Authentication Profile.
Step 6	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set identity** { **dynamic-mac** {*dynamic-mac-address* | **derived** } | **mac-pool** *mac-pool-name* }	Specifies the MAC address for the iSCSI vNIC. **Note** The MAC address is set only for the Cisco UCS NIC M51KR-B Adapters.
Step 7	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set iscsi-identity** {**initiator-name** *initiator-name* | **initiator-pool-name** *iqn-pool-name*}	Specifies the name of the iSCSI initiator or the name of an IQN pool from which the iSCSI initiator name will be provided. The iSCSI initiator name can be up to 223 characters.
Step 8	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set overlay-vnic-name** *overlay-vnic-name*	Specifies the Ethernet vNIC that is used by the iSCSI device as the overlay vNIC. For more information, see Configuring a vNIC for a Service Profile.
Step 9	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **create eth-if**	Creates an Ethernet interface for a VLAN assigned to the iSCSI vNIC.
Step 10	UCS-A /org/ex/vnic-iscsi/eth-if # **set vlanname** *vlan-name*	Specifies the VLAN name. The default VLAN is default. For the Cisco UCS M81KR Virtual Interface Card and the Cisco UCS VIC-1240 Virtual Interface Card, the VLAN that you specify must be the same as the native VLAN on the overlay vNIC. For the Cisco UCS M51KR-B Broadcom BCM57711 Adapter, the VLAN that you specify can be any VLAN assigned to the overlay vNIC.
Step 11	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to configure an iSCSI vNIC for a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy # create vnic-iscsi iSCSI1
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set iscsi-adaptor-policy iscsiboot
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set auth-name initauth
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set identity dynamic-mac derived
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set iscsi-identity initiator-name iSCSI1
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set overlay-vnic-name eth1
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # create eth-if
UCS-A /org/lan-connectivity-policy/vnic-iscsi/eth-if* # set vlanname default
UCS-A /org/lan-connectivity-policy/vnic-iscsi/eth-if* # commit buffer
UCS-A /org/lan-connectivity-policy/vnic-iscsi/eth-if
```

What to Do Next

If desired, add another iSCI vNIC or a vNIC to the LAN connectivity policy. If not, include the policy in a service profile or service profile template.

Deleting an iSCSI vNIC from a LAN Connectivity Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **scope lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **delete vnic-iscsi** *iscsi-vnic-name*	Deletes the specified iSCSI vNIC from the LAN connectivity policy.
Step 4	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete an iSCSI vNIC named iscsivnic3 from a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy # delete vnic-iscsi iscsivnic3
UCS-A /org/lan-connectivity-policy* # commit-buffer
UCS-A /org/lan-connectivity-policy #
```

Deleting a LAN Connectivity Policy

If you delete a LAN connectivity policy that is included in a service profile, you will delete all vNICs and iSCSI vNICs from that service profile and disrupt LAN data traffic for the server associated with the service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **delete lan-connectivity-policy** *policy-name*	Deletes the specified LAN connectivity policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete the LAN connectivity policy named LanConnectiSCSI42 from the root organization and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete lan-connectivity-policy LanConnectiSCSI42
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Network Control Policies

Network Control Policy

This policy configures the network control settings for the Cisco UCS domain, including the following:

- Whether the Cisco Discovery Protocol (CDP) is enabled or disabled
- How the virtual interface (VIF) behaves if no uplink port is available in end-host mode
- The action that Cisco UCS Manager takes on the remote Ethernet interface, vEthernet interface , or vFibre Channel interface when the associated border port fails
- Whether the server can use different MAC addresses when sending packets to the fabric interconnect
- Whether MAC registration occurs on a per-VNIC basis or for all VLANs

Action on Uplink Fail

By default, the **Action on Uplink Fail** property in the network control policy is configured with a value of link-down. For adapters such as the Cisco UCS M81KR Virtual Interface Card, this default behavior directs Cisco UCS Manager to bring the vEthernet or vFibre Channel interface down if the associated border port fails. For Cisco UCS systems using a non-VM-FEX capable converged network adapter that supports both

Ethernet and FCoE traffic, such as Cisco UCS CNA M72KR-Q and the Cisco UCS CNA M72KR-E, this default behavior directs Cisco UCS Manager to bring the remote Ethernet interface down if the associated border port fails. In this scenario, any vFibre Channel interfaces that are bound to the remote Ethernet interface are brought down as well.

Note if your implementation includes those types of non-VM-FEX capable converged network adapters mentioned in this section and the adapter is expected to handle both Ethernet and FCoE traffic, we recommend that you configure the **Action on Uplink Fail** property with a value of warning. Note that this configuration might result in an Ethernet teaming driver not being able to detect a link failure when the border port goes down.

MAC Registration Mode

MAC addresses are installed only on the native VLAN by default, which maximizes the VLAN port count in most implementations.

Note If a trunking driver is being run on the host and the interface is in promiscuous mode, we recommend that you set the Mac Registration Mode to All VLANs.

Configuring Link Layer Discovery Protocol for Fabric Interconnect vEthernet Interfaces

Cisco UCS Manager Release 2.2.4 allows you to enable and disable LLDP on a vEthernet interface. You can also retrieve information about these LAN uplink neighbors. This information is useful while learning the topology of the LAN connected to the UCS system and while diagnosing any network connectivity issues from the Fabric Interconnect (FI). The FI of a UCS system is connected to LAN uplink switches for LAN connectivity and to SAN uplink switches for storage connectivity. When using Cisco UCS with Cisco Application Centric Infrastructure (ACI), LAN uplinks of the FI are connected to ACI leaf nodes. Enabling LLDP on a vEthernet interface will help the Application Policy Infrastructure Controller (APIC) to identify the servers connected to the FI by using vCenter.

To permit the discovery of devices in a network, support for Link Layer Discovery Protocol (LLDP), a vendor-neutral device discovery protocol that is defined in the IEEE 802.1ab standard, is introduced. LLDP is a one-way protocol that allows network devices to advertise information about themselves to other devices on the network. LLDP transmits information about the capabilities and current status of a device and its interfaces. LLDP devices use the protocol to solicit information only from other LLDP devices.

You can enable or disable LLDP on a vEthernet interface based on the Network Control Policy (NCP) that is applied on the vNIC in the service profile.

Configuring a Network Control Policy

MAC address-based port security for Emulex converged Network Adapters (N20-AE0102) is not supported. When MAC address-based port security is enabled, the fabric interconnect restricts traffic to packets that contain the MAC address that it first learns. This is either the source MAC address used in the FCoE Initialization Protocol packet, or the MAC address in an ethernet packet, whichever is sent first by the adaptor. This configuration can result in either FCoE or Ethernet packets being dropped.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **create nw-ctrl-policy** *policy-name*	Creates the specified network control policy, and enters organization network control policy mode.
Step 3	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **cdp**	Disables or enables Cisco Discovery Protocol (CDP).
Step 4	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **lldp transmit**	Disables or enables the transmission of LLDP packets on an interface.
Step 5	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **lldp receive**	Disables or enables the reception of LLDP packets on an interface.
Step 6	UCS-A /org/nw-ctrl-policy # **set uplink-fail-action** {**link-down** | **warning**}	Specifies the action to be taken when no uplink port is available in end-host mode. Use the **link-down** keyword to change the operational state of a vNIC to down when uplink connectivity is lost on the fabric interconnect, and facilitate fabric failover for vNICs. Use the **warning** keyword to maintain server-to-server connectivity even when no uplink port is available, and disable fabric failover when uplink connectivity is lost on the fabric interconnect. The default uplink failure action is link-down.
Step 7	UCS-A /org/nw-ctrl-policy # **set mac-registration-mode**{**all-host-vlans** | **only-native-vlan**	Whether adapter-registered MAC addresses are added only to the native VLAN associated with the interface or added to all VLANs associated with the interface. This can be one of the following: • **Only Native Vlan—**MAC addresses are only added to the native VLAN. This option is the default, and it maximizes the port+VLAN count. • **All Host Vlans—**MAC addresses are added to all VLANs with which they are associated. Select this option if your VLANs are configured to use trunking but are *not* running in Promiscuous mode.
Step 8	UCS-A /org/nw-ctrl-policy # **create mac-security**	Enters organization network control policy MAC security mode
Step 9	UCS-A /org/nw-ctrl-policy/mac-security # **set forged-transmit** {**allow** | **deny**}	Allows or denies the forging of MAC addresses when sending traffic. MAC security is disabled when forged MAC addresses are allowed, and MAC security is enabled when forged MAC addresses are denied. By default,

	Command or Action	Purpose
		forged MAC addresses are allowed (MAC security is disabled).
Step 10	UCS-A /org/nw-ctrl-policy/mac-security # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a network control policy named ncp5, enable CDP, enable LLDP transmit and LLDP recive, set the uplink fail action to link-down, deny forged MAC addresses (enable MAC security), and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create nw-ctrl-policy ncp5
UCS-A /org/nw-ctrl-policy* # enable cdp
UCS-A /org/nw-ctrl-policy* # enable lldp transmit
UCS-A /org/nw-ctrl-policy* # enable lldp receive
UCS-A /org/nw-ctrl-policy* # set uplink-fail-action link-down
UCS-A /org/nw-ctrl-policy* # create mac-security
UCS-A /org/nw-ctrl-policy/mac-security* # set forged-transmit deny
UCS-A /org/nw-ctrl-policy/mac-security* # commit-buffer
UCS-A /org/nw-ctrl-policy/mac-security #
```

Displaying Network Control Policy Details

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **scope nw-ctrl-policy** {**default** | *policy-name*}	Enters organization network control policy mode for the specified network control policy.
Step 3	UCS-A /org/nw-ctrl-policy # **show detail**	Displays details about the specified network control policy.

The following example shows how to display the details of a network control policy named ncp5:

```
UCS-A# scope org /
UCS-A /org # scope nw-ctrl-policy ncp5
UCS-A /org/nw-ctrl-policy* # show detail

Network Control Policy:
     Name: ncp5
     CDP: Enabled
     LLDP Transmit: Enabled
     LLDP Receive: Enabled
     Uplink fail action: Link Down
     Adapter MAC Address Registration: Only Native Vlan
     Policy Owner: Local
     Description:
```

```
UCS-A /org/nw-ctrl-policy #
```

Deleting a Network Control Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A /org # **delete nwctrl-policy** *policy-name*	Deletes the specified network control policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example deletes the network control policy named ncp5 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete nwctrl-policy ncp5
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Multicast Policies

Multicast Policy

This policy is used to configure Internet Group Management Protocol (IGMP) snooping and IGMP querier. IGMP Snooping dynamically determines hosts in a VLAN that should be included in particular multicast transmissions. You can create, modify, and delete a multicast policy that can be associated to one or more VLANs. When a multicast policy is modified, all VLANs associated with that multicast policy are re-processed to apply the changes. By default, IGMP snooping is enabled and IGMP querier is disabled. For private VLANs, you can set a multicast policy for primary VLANs but not for their associated isolated VLANs due to a Cisco NX-OS forwarding implementation.

The following limitations apply to multicast policies on the Cisco UCS 6100 series fabric interconnect and the 6200 series fabric interconnect:

- If a Cisco UCS domain includes only 6100 series fabric interconnects, only the default multicast policy is allowed for local VLANs or global VLANs.
- If a Cisco UCS domain includes one 6100 series fabric interconnect and one 6200 series fabric interconnect:
  - Only the default multicast policy is allowed for a local VLAN on a 6100 series fabric interconnect.
  - On a 6200 series fabric interconnect, user-defined multicast policies can also be assigned along with the default multicast policy.

- Only the default multicast policy is allowed for a global VLAN (as limited by one 6100 series fabric interconnect in the cluster.

- If a Cisco UCS domain includes only 6200 series fabric interconnects, any multicast policy can be assigned.

Creating a Multicast Policy

A multicast policy can be created only in the root organization and not in a sub-organization.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **create mcast-policy** *policy-name*	Creates a multicast policy with the specified policy name, and enters organization multicast policy mode.
Step 3	UCS-A /org/mcast-policy* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # create mcast-policy policy1
UCS-A /org/mcast-policy* # commit-buffer
UCS-A /org/mcast-policy #
```

Configuring IGMP Snooping Parameters

You can enable or disable IGMP snooping for a multicast policy. By default, the IGMP snooping state is enabled for a multicast policy. You can also set the IGMP snooping querier state and IPv4 address for the multicast policy.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **create mcast-policy** *policy-name*	Creates a new multicast policy with the specified policy name, and enters organization multicast policy mode.

	Command or Action	Purpose
Step 3	UCS-A /org/mcast-policy* # **set querier**{**enabled** | **disabled**}	Enables or disables IGMP snooping querier. By default, IGMP snooping querier is disabled for a multicast policy.
Step 4	UCS-A /org/mcast-policy* # **set querierip** *IGMP snooping querier IPv4 address*	Specifies the IPv4 address for the IGMP snooping querier.
Step 5	UCS-A /org/mcast-policy* # **set snooping**{**enabled** | **disabled**}	Enables or disables IGMP snooping. By default, IGMP snooping is enabled for a multicast policy.
Step 6	UCS-A /org/mcast-policy* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create and enter a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # create mcast-policy policy1
UCS-A /org/mcast-policy* # set querier enabled
UCS-A /org/mcast-policy* # set querierip 1.2.3.4
UCS-A /org/mcast-policy* # set snooping enabled
UCS-A /org/mcast-policy* # commit-buffer
UCS-A /org/mcast-policy #
```

Modifying Multicast Policy Parameters

You can modify an existing multicast policy to change the state of IGMP snooping or IGMP snooping querier. When a multicast policy is modified, all VLANs associated with that multicast policy are re-processed to apply the changes.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **scope mcast-policy** *policy-name*	Enters organization multicast policy mode.
Step 3	UCS-A /org/mcast-policy* # **set querier**{**enabled** | **disabled**}	Enables or disables IGMP snooping querier. By default, IGMP snooping querier is disabled for a multicast policy.
Step 4	UCS-A /org/mcast-policy* # **set querierip** *IGMP snooping querier IPv4 address*	Specifies the IPv4 address for the IGMP snooping querier.
Step 5	UCS-A /org/mcast-policy* # **set snooping**{**enabled** | **disabled**}	Enables or disables IGMP snooping. By default, IGMP snooping is enabled for a multicast policy.

	Command or Action	Purpose
Step 6	UCS-A /org/mcast-policy* # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # scope mcast-policy policy1
UCS-A /org/mcast-policy* # set querier enabled
UCS-A /org/mcast-policy* # set querierip 1.2.3.4
UCS-A /org/mcast-policy* # set snooping enabled
UCS-A /org/mcast-policy* # commit-buffer
UCS-A /org/mcast-policy #
```

Assigning a VLAN Multicast Policy

You can set a multicast policy for a VLAN in the Ethernet uplink fabric mode. You cannot set a multicast policy for an isolated VLAN.

Before You Begin

Create a VLAN.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric**{**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric interconnect.
Step 3	UCS-A /eth-uplink/fabric # **scope vlan** *vlan-name*	Enters Ethernet uplink fabric VLAN mode.
Step 4	UCS-A /eth-uplink/fabric/vlan # **set mcastpolicy** *policy-name*	Assigns a multicast policy for the VLAN.
Step 5	UCS-A /eth-uplink/fabric/vlan # **commit-buffer**	Commits the transaction to the system configuration.

The following example sets a named VLAN accessible to one fabric interconnect and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope vlan vlan1
UCS-A /eth-uplink/fabric/vlan # set mcastpolicy policy1
UCS-A /eth-uplink/fabric/vlan* # commit-buffer
UCS-A /eth-uplink/fabric/vlan #
```

Deleting a Multicast Policy

Note If you assigned a non-default (user-defined) multicast policy to a VLAN and then delete that multicast policy, the associated VLAN inherits the multicast policy settings from the default multicast policy until the deleted policy is re-created.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **delete mcast-policy** *policy-name*	Deletes a multicast policy with the specified policy name.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to delete a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # delete mcast-policy policy1
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring LACP Policies

LACP Policy

Link Aggregation combines multiple network connections in parallel to increase throughput and to provide redundancy. Link aggregation control protocol (LACP) provides additional benefits for these link aggregation groups. Cisco UCS Manager enables you to configure LACP properties using LACP policy.

You can configure the following for a lacp policy:

- **Suspended-individual**: If you do not configure the ports on an upstream switch for lacp, the fabric interconnects treat all ports as uplink Ethernet ports to forward packets. You can place the lacp port in suspended state to avoid loops. When you set suspend-individual on a port-channel with lacp, if a port that is part of the port-channel does not receive PDUs from the peer port, it will go into suspended state.
- **Timer values**: You can configure rate-fast or rate-normal. In rate-fast configuration, the port is expected to receive 1 PDU every 1 second from the peer port. The time out for this is 3 seconds. In rate-normal configuration, the port is expected to receive 1 PDU every 30 seconds. The timeout for this is 90 seconds.

System creates a default lacp policy at system start up. You can modify this policy or create new. You can also apply one lacp policy to multiple port-channels.

Creating a LACP Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters the root organization mode.
Step 2	UCS-A /org # **create lacppolicy***policy nam*.	Creates the specified lacp policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

The following example creates the lacp policy and commits the transaction:

```
UCS-A# scope org
UCS-A /org # create lacppolicy lacp1
UCS-A /org* # commit-buffer
UCS-A /org #
```

Editing a LACP Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters the root organization mode.
Step 2	UCS-A /org # **scope lacppolicy** *policy-name* .	Enters the specified lacp policy.
Step 3	UCS-A /org/lacp policy/ policy-name # **set suspend-individual** *true* .	Sets suspend individual for the policy.
Step 4	UCS-A /org/lacp policy/ policy-name # **set lacp-rate** *fast* .	Sets LACP rate for the policy.
Step 5	UCS-A /org/lacp policy/ policy-name # **commit-buffer**	Commits the transaction to the system configuration.

The following example modifies the lacp policy and commits transaction:

```
UCS-A# scope org
UCS-A/org # scope lacppolicy policy-name
UCS-A /org/lacp policy policy-name# set suspend-individual true
UCS-A/prg/policy policy-name# set lacp-rate fast
UCS-A /org* # commit-buffer
UCS-A /org #
```

Assigning LACP Policy to Port-Channels

Default lacp policy is assigned to port channels by default. You can assign a different lacp policy to the port channel. If the assigned policy does not exist, system generates a fault. You can create the same policy to clear the fault.

Note You can assign lacp policy to port-channels, FCoE port-channels, and ethernet storage port-channels. This procedures describes assigning the lacp policy to port-channels.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric**	Enters the fabric mode.
Step 3	UCS-A /eth-uplink/fabric # **scope port-channel**	Enters the port-channel mode.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **set lacp-policy-name***policy-name*	Specifies the lacp policy for this port-channel.
Step 5	UCS-A /eth-uplink/ fabric/port-channel **commit-buffer**	Commits the transaction to the system.

The following example shows assigning a lacp policy to a port-channel:

```
UCS-A# scope eth-uplink
UCS-A UCS-A/eth-uplink # scope fabric
UCS-A UCS-A/eth-uplink/facric # scope port-channel
UCS-A UCS-A/eth-uplink/port-channel# set lacp-policy-name
UCS-A UCS-A/eth-uplink/port-channel# commit-buffer
```

Configuring UDLD Link Policies

Understanding UDLD

UniDirectional Link Detection (UDLD) is a Layer 2 protocol that enables devices connected through fiber-optic or twisted-pair Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists. All connected devices must support UDLD for the protocol to successfully identify and disable unidirectional links. When UDLD detects a unidirectional link, it marks the link as unidirectional. Unidirectional links can cause a variety of problems, including spanning-tree topology loops.

UDLD works with the Layer 1 mechanisms to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected interfaces. When you enable both autonegotiation and UDLD, the Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

A unidirectional link occurs whenever traffic sent by a local device is received by its neighbor but traffic from the neighbor is not received by the local device.

Modes of Operation

UDLD supports two modes of operation: normal (the default) and aggressive. In normal mode, UDLD can detect unidirectional links due to misconnected interfaces on fiber-optic connections. In aggressive mode, UDLD can also detect unidirectional links due to one-way traffic on fiber-optic and twisted-pair links and to misconnected interfaces on fiber-optic links.

In normal mode, UDLD detects a unidirectional link when fiber strands in a fiber-optic interface are misconnected and the Layer 1 mechanisms do not detect this misconnection. If the interfaces are connected correctly but the traffic is one way, UDLD does not detect the unidirectional link because the Layer 1 mechanism, which is supposed to detect this condition, does not do so. In case, the logical link is considered undetermined, and UDLD does not disable the interface. When UDLD is in normal mode, if one of the fiber strands in a pair is disconnected and autonegotiation is active, the link does not stay up because the Layer 1 mechanisms did not detect a physical problem with the link. In this case, UDLD does not take any action, and the logical link is considered undetermined.

UDLD aggressive mode is disabled by default. Configure UDLD aggressive mode only on point-to-point links between network devices that support UDLD aggressive mode. With UDLD aggressive mode enabled, when a port on a bidirectional link that has a UDLD neighbor relationship established stops receiving UDLD packets, UDLD tries to reestablish the connection with the neighbor and administratively shuts down the affected port. UDLD in aggressive mode can also detect a unidirectional link on a point-to-point link on which no failure between the two devices is allowed. It can also detect a unidirectional link when one of the following problems exists:

- On fiber-optic or twisted-pair links, one of the interfaces cannot send or receive traffic.
- On fiber-optic or twisted-pair links, one of the interfaces is down while the other is up.
- One of the fiber strands in the cable is disconnected.

Methods to Detect Unidirectional Links

UDLD operates by using two mechanisms:

- Neighbor database maintenance

  UDLD learns about other UDLD-capable neighbors by periodically sending a hello packet (also called an advertisement or probe) on every active interface to keep each device informed about its neighbors. When the switch receives a hello message, it caches the information until the age time (hold time or time-to-live) expires. If the switch receives a new hello message before an older cache entry ages, the switch replaces the older entry with the new one.

  UDLD clears all existing cache entries for the interfaces affected by the configuration change whenever an interface is disabled and UDLD is running, whenever UDLD is disabled on an interface, or whenever the switch is reset. UDLD sends at least one message to inform the neighbors to flush the part of their caches affected by the status change. The message is intended to keep the caches synchronized.

- Event-driven detection and echoing

  UDLD relies on echoing as its detection mechanism. Whenever a UDLD device learns about a new neighbor or receives a resynchronization request from an out-of-sync neighbor, it restarts the detection window on its side of the connection and sends echo messages in reply. Because this behavior is the same on all UDLD neighbors, the sender of the echoes expects to receive an echo in reply.

If the detection window ends and no valid reply message is received, the link might shut down, depending on the UDLD mode. When UDLD is in normal mode, the link might be considered undetermined and might not be shut down. When UDLD is in aggressive mode, the link is considered unidirectional, and the interface is shut down.

If UDLD in normal mode is in the advertisement or in the detection phase and all the neighbor cache entries are aged out, UDLD restarts the link-up sequence to resynchronize with any potentially out-of-sync neighbors.

If you enable aggressive mode when all the neighbors of a port have aged out either in the advertisement or in the detection phase, UDLD restarts the link-up sequence to resynchronize with any potentially out-of-sync neighbor. UDLD shuts down the port if, after the fast train of messages, the link state is still undetermined.

UDLD Configuration Guidelines

The following guidelines and recommendations apply when you configure UDLD:

- A UDLD-capable interface also cannot detect a unidirectional link if it is connected to a UDLD-incapable port of another switch.
- When configuring the mode (normal or aggressive), make sure that the same mode is configured on both sides of the link.
- UDLD should be enabled only on interfaces that are connected to UDLD capable devices. The following interface types are supported:
  - Ethernet uplink
  - FCoE uplink
  - Ethernet uplink port channel member
  - FCoE uplink port channel member

Configuring a Link Profile

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** /	Enters the root organization mode.
Step 2	UCS-A /org # **create eth-link-profile** *link-profile-name*	Creates a link profile with the specified name, and enters link profile mode.
Step 3	UCS-A /org/eth-link-profile # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /org/eth-link-profile # **exit**	Returns to the previous mode.
Step 5	UCS-A /org # **scope eth-link-profile** *link-profile-name*	Enters link profile mode for the specified link profile.

	Command or Action	Purpose
Step 6	UCS-A /org/eth-link-profile # **set udld-link-policy** *link-policy-name*	Assigns the specified UDLD link policy to the link profile.
Step 7	UCS-A /org/eth-link-profile # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a link profile called LinkProfile1 and assign the default UDLD link policy.

```
UCS-A# scope org /
UCS-A /chassis/org # create eth-link-profile LinkProfile1
UCS-A /chassis/org/eth-link-profile* # commit-buffer
UCS-A /chassis/org/eth-link-profile # exit
UCS-A /chassis/org # scope eth-link-profile LinkProfile1
UCS-A /chassis/org/eth-link-profile # set udld-link-policy default
UCS-A /chassis/org/eth-link-profile* # commit-buffer
```

Configuring a UDLD Link Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** /	Enters the root organization mode.
Step 2	UCS-A /org # **create udld-link-policy** *link-policy-name*	Creates a UDLD link policy with the specified name, and enters UDLD link policy mode.
Step 3	UCS-A /org/udld-link-policy # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /org/udld-link-policy # **exit**	Returns to the previous mode.
Step 5	UCS-A /org # **scope udld-link-policy** *link-policy-name*	Enters UDLD link policy mode for the specified UDLD link policy.
Step 6	UCS-A /org/udld-link-policy # **set mode** {**aggressive** | **normal**}	Specifies the mode for the UDLD link policy.
Step 7	UCS-A /org/udld-link-policy # **set admin-state** {**disabled** | **enabled**}	Disables or enables UDLD on the interface.
Step 8	UCS-A /org/udld-link-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to create a link profile called UDLDPol1, sets the mode to aggressive, and enables UDLD on the interface.

```
UCS-A# scope org /
UCS-A /chassis/org # create udld-link-policy UDLDPol1
```

```
UCS-A /chassis/org/udld-link-policy* # commit-buffer
UCS-A /chassis/org/udld-link-policy # exit
UCS-A /chassis/org # scope udld-link-policy UDLDPol1
UCS-A /chassis/org/udld-link-policy # set mode aggressive
UCS-A /chassis/org/udld-link-policy* # set admin-state enabled
UCS-A /chassis/org/udld-link-policy* # commit-buffer
UCS-A /chassis/org/udld-link-policy #
```

Modifying the UDLD System Settings

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** /	Enters the root organization mode.
Step 2	UCS-A /org # **show udld-policy**	Displays the current UDLD system settings.
Step 3	UCS-A /org # **scope udld-policy default**	Enters UDLD policy mode for the global UDLD policy.
Step 4	UCS-A /org/udld-policy # **set message-interval** *seconds*	Specifies the time interval (in seconds) between UDLD probe messages on ports that are in advertisement mode. Enter an integer between 7 and 60. The default is 15 seconds.
Step 5	UCS-A /org/udld-policy # **set recovery-action** [**reset** | **none**]	Specifies the action to be taken on any ports that are disabled when UDLD aggressive mode is enabled. The default is none.
Step 6	UCS-A /org/udld-policy # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to update the default UDLD system settings for a 30 second time interval.

```
UCS-A# scope org /
UCS-A /chassis/org # show udld-policy

UDLD system settings:
    Name       Message interval (sec) Recovery action
    ---------- ---------------------- ---------------
    default    15                     None

UCS-A /chassis/org # scope udld-policy default
UCS-A /chassis/org/udld-policy # set message-interval 30
UCS-A /chassis/org/udld-policy* # commit-buffer
UCS-A /chassis/org/udld-policy #
```

Assigning a Link Profile to a Port Channel Ethernet Interface

Procedure

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

The following example shows how to assign link profile LinkProfile1 to a port channel Ethernet interface:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope port-channel 88
UCS-A /eth-uplink/fabric/port-channel # scope member-port 1 31
UCS-A /eth-uplink/fabric/port-channel/member-port # set eth-link-profile LinkProfile1
UCS-A /eth-uplink/fabric/port-channel/member-port* # commit-buffer
UCS-A /eth-uplink/fabric/port-channel/member-port #
```

Assigning a Link Profile to a Port Channel FCoE Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope fcoe-port-channel** *port-chan-id*	Enters Fibre Channel uplink fabric port channel mode for the specified port channel.

	Command or Action	Purpose
Step 4	UCS-A /fc-uplink/fabric/fcoe-port-channel # **scope fcoe-member-port** *slot-id port-id*	Enters Fibre Channel server fabric, fabric port channel mode for the specified member port.
Step 5	UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # **set eth-link-profile** *link-profile-name*	Assigns the specified link profile.
Step 6	UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to assign link profile LinkProfile1 to a port channel FCoE interface:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope fcoe-port-channel 192
UCS-A /fc-uplink/fabric/fcoe-port-channel # scope fcoe-member-port 1 20
UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # set eth-link-profile LinkProfile1
UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port* # commit-buffer
UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port #
```

Assigning a Link Profile to an Uplink Ethernet Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **scope interface** *slot-num port num*	Enters the interface command mode for the specified uplink port.
Step 4	UCS-A /eth-uplink/fabric/interface # **set eth-link-profile** *link-profile-name*	Assigns the specified link profile.
Step 5	UCS-A /eth-uplink/fabric/interface # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to assign link profile LinkProfile1 to an uplink Ethernet interface:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope interface 2 2
UCS-A /eth-uplink/fabric/interface # set eth-link-profile LinkProfile1
UCS-A /eth-uplink/fabric/interface* # commit-buffer
UCS-A /eth-uplink/fabric/interface #
```

Assigning a Link Profile to an Uplink FCoE Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope fcoeinterface** *slot-num port num*	Enters the Fibre Channel interface command mode for the specified uplink port.
Step 4	UCS-A /fc-uplink/fabric/fcoeinterface # **set eth-link-profile** *link-profile-name*	Assigns the specified link profile.
Step 5	UCS-A /fc-uplink/fabric/fcoeinterface # **commit-buffer**	Commits the transaction to the system configuration.

The following example shows how to assign link profile LinkProfile1 to an uplink FCoE interface:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope fcoeinterface 2 2
UCS-A /fc-uplink/fabric/fcoeinterface # set eth-link-profile LinkProfile1
UCS-A /fc-uplink/fabric/fcoeinterface* # commit-buffer
UCS-A /fc-uplink/fabric/fcoeinterface #
```

Configuring VMQ Connection Policies

VMQ Connection Policy

Cisco UCS Manager enables you to configure VMQ connection policy for a vNIC. VMQ provides improved network performance to the entire management operating system. Configuring a VMQ vNIC connection policy involves the following:

- Create a VMQ connection policy
- Create a static vNIC in a service profile
- Apply the VMQ connection policy to the vNIC

If you want to configure the VMQ vNIC on a service profile for a server, at least one adapter in the server must support VMQ. Make sure the servers have at least one the following adapters installed:

- UCS-VIC-M82-8P
- UCSB-MLOM-40G-01
- UCSC-PCIE-CSC-02

The following are the supported Operating Systems for VMQ:

- Windows 2012
- Windows 2012R2

You can apply only any one of the vNIC connection policies on a service profile at any one time. Make sure to select one of the three options such as Dynamic, usNIC or VMQ connection policy for the vNIC. When a VMQ vNIC is configured on service profile, make sure you have the following settings:

- Select SRIOV in the BIOS policy.
- Select Windows in the Adapter policy.

Creating a VMQ Connection Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 2	UCS-A /org # **create vmq-conn-policy** *policy-name*	Specifies the name for this VMQ connection policy.
Step 3	UCS-A /org/vmq-conn-policy* # **set queue-count***queue count*	Specifies the queue count for the VMQ connection policy.
Step 4	UCS-A /org/vmq-conn-policy* # **set interrupt-count***interrupt count*	Specifies the interrupt count for the VMQ connection policy.
Step 5	UCS-A /org/vmq-conn-policy* # **commit-buffer**	Commits the transaction to the system.

The following example creates a VMQ connection policy:

```
UCS-A# scope org
UCS-A /org # create vmq-conn-policy policy name
UCS-A /org/vmq-conn-policy* # set queue-count queue count (number)
UCS-A /org/vmq-conn-policy* # set interrupt-count queue count (number)
UCS-A /org/vmq-conn-policy* # commit-buffer
```

NetQueue

Information About NetQueue

NetQueue improves traffic performance by providing a network adapter with multiple receive queues. These queues allow the data interrupt processing that is associated with individual virtual machines to be grouped.

Note NetQueue is supported on servers running VMware ESXi operating systems.

Configuring NetQueue

Procedure

Step 1 Create a Virtual Machine Queue (VMQ) connection policy.

Step 2 Configure NetQueues in a service profile by selecting the VMQ connection policy.
Use the following when you are configuring NetQueue:

- The default ring size is rx512, tx256
- The interrupt count on each VNIC is VMQ count x 2 +2

  Note The number of interrupts depends on the number of NetQueues enabled.

- The driver supports up to 16 NetQueues per port for standard frame configurations.

  Note VMware recommends that you use up to eight NetQueues per port for standard frame configurations.

- NetQueue should be enabled only on MSIX systems.
- You should disable NetQueue on 1 GB NICs.

Step 3 Enable the MSIX mode in the adapter policy for NetQueue.

Step 4 Associate the service profile with the server.