



## Network-Related Policies

---

- [vNIC Template, on page 1](#)
- [Ethernet Adapter Policies, on page 8](#)
- [Ethernet and Fibre Channel Adapter Policies, on page 14](#)
- [GENEVE Offload, on page 22](#)
- [Configuring a Default vNIC Behavior Policy, on page 23](#)
- [Deleting a vNIC from a LAN Connectivity Policy, on page 24](#)
- [Creating a LAN Connectivity Policy, on page 24](#)
- [Deleting a LAN Connectivity Policy, on page 25](#)
- [About the LAN and SAN Connectivity Policies, on page 26](#)
- [Network Control Policy, on page 34](#)
- [Creating a Multicast Policy, on page 39](#)
- [Deleting a Multicast Policy, on page 40](#)
- [Entering Multicast Policy Mode, on page 41](#)
- [Enter a Multicast Policy, on page 41](#)
- [Assigning a Global VLAN Multicast Policy, on page 42](#)
- [Disassociating a Global VLAN Multicast Policy, on page 42](#)
- [Disassociating a VLAN Multicast Policy, on page 43](#)
- [Configuring Ethernet Adapter Policies, on page 44](#)
- [Configuring the Default vNIC Behavior Policy, on page 47](#)
- [Configuring a Network Control Policy, on page 48](#)
- [Deleting a Network Control Policy, on page 50](#)
- [Configuring Multicast Policies, on page 51](#)
- [LACP Policy, on page 56](#)
- [Configuring UDLD Link Policies, on page 59](#)
- [VMQ Connection Policy, on page 67](#)

## vNIC Template

The vNIC LAN connectivity policy defines how a vNIC on a server connects to the LAN.

Cisco UCS Manager does not automatically create a VM-FEX port profile with the correct settings when you create a vNIC template. If you want to create a VM-FEX port profile, you must configure the target of the vNIC template as a VM. You must include this policy in a service profile for it to take effect.

You can select VLAN groups in addition to any individual VLAN while creating a vNIC template.



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

## Creating vNIC Template Pairs

### SUMMARY STEPS

1. UCS-A/ org # **create vnic-templ** *vnic-primary* .
2. UCS-A/ # org vnic-templ **set type updating-template** .
3. UCS-A/ # org vnic-templ [**set fabric** {a | b}] .
4. UCS-A/ # org vnic-templ **set descr primaryinredundancypair** .
5. UCS-A/ # org vnic-templ **set redundancy-type** *primary*.
6. UCS-A/ # org vnic-templ **exit** .
7. UCS-A/ # org vnic-templ **create vNIC-templ** *vNICsecondary* .
8. UCS-A/ # org vnic-templ **set type updating-template** .
9. UCS-A/ org # vnic-templ [**set fabric** {a | b}] .
10. UCS-A/ # org vnic-templ **set descr secondaryredundancypair**.
11. UCS-A/ # org vnic-templ **set redundancy-type** *secondary*.
12. UCS-A/ # org vnic-templ **set peer-template-name** *vNIC-primary*.
13. UCS-A/ # org vnic-templ **commit-buffer** .

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A/ org # <b>create vnic-templ</b> <i>vnic-primary</i> .	Creates a Primary vNIC template.
<b>Step 2</b>	UCS-A/ # org vnic-templ <b>set type updating-template</b> .	Set the template type to updating, which drives the configurations in the Primary vNIC template for shared configurations to the peer vNIC template. See the shared configurations listed below.
<b>Step 3</b>	UCS-A/ # org vnic-templ [ <b>set fabric</b> {a   b}] .	Specifies the fabric for the Primary vNIC template. If you specify Fabric A for the Primary vNIC template, the Secondary vNIC template must be Fabric B or vice versa.
<b>Step 4</b>	UCS-A/ # org vnic-templ <b>set descr primaryinredundancypair</b> .	Sets the template as the Primary vNIC template.
<b>Step 5</b>	UCS-A/ # org vnic-templ <b>set redundancy-type</b> <i>primary</i> .	Sets the redundancy template type as the Primary vNIC template.

	Command or Action	Purpose
		<p>Following are descriptions of the <b>Redundancy Types</b>:</p> <p><b>Primary</b>—Creates configurations that can be shared with the Secondary vNIC template. Any shared changes on the Primary vNIC template are automatically synchronized to the Secondary vNIC template.</p> <p><b>Secondary</b> — All shared configurations are inherited from the Primary template.</p> <p><b>No Redundancy</b>— Legacy vNIC template behavior.</p> <p>Following is a list of shared configurations:</p> <ul style="list-style-type: none"> <li>• <b>Network Control Policy</b></li> <li>• <b>QoS Policy</b></li> <li>• <b>Stats Threshold Policy</b></li> <li>• <b>Template Type</b></li> <li>• <b>Connection Policies</b></li> <li>• <b>VLANS</b></li> <li>• <b>MTU</b></li> </ul> <p>Following is a list of non-shared configurations:</p> <ul style="list-style-type: none"> <li>• <b>Fabric ID</b></li> <li>• <b>CDN Source</b></li> <li>• <b>MAC Pool</b></li> <li>• <b>Description</b></li> <li>• <b>Pin Group Policy</b></li> </ul>
<b>Step 6</b>	UCS-A/ # org vnic-templ <b>exit</b> .	<p>Exits creating the redundancy template pairing.</p> <p><b>Note</b>      Ensure to commit the transaction after linking the Primary vNIC template to a peer Secondary vNIC template to create the redundancy pair.</p>
<b>Step 7</b>	UCS-A/ # org vnic-templ <b>create vNIC-templ vNICsecondary</b> .	Creates the Secondary vNIC template.
<b>Step 8</b>	UCS-A/ # org vnic-templ <b>set type updating-template</b> .	Sets the template type to updating, which automatically inherits the configurations from the Primary vNIC template.
<b>Step 9</b>	UCS-A/ org # vnic-templ [ <b>set fabric {a   b}</b> ] .	Specifies the fabric for the Secondary vNIC template. If you specify Fabric A for the Primary vNIC template, the Secondary vNIC template must be Fabric B or vice versa.

	Command or Action	Purpose
<b>Step 10</b>	UCS-A/ # org vnic-templ <b>set descr secondaryredundancypair</b> .	Sets the secondary vNIC template as a redundancy pair template.
<b>Step 11</b>	UCS-A/ # org vnic-templ <b>set redundancy-type secondary</b> .	Sets the vNIC template type as Secondary.
<b>Step 12</b>	UCS-A/ # org vnic-templ <b>set peer-template-name vNIC-primary</b> .	Sets the Primary vNIC template as the peer to the Secondary vNIC template.
<b>Step 13</b>	UCS-A/ # org vnic-templ <b>commit-buffer</b> .	Commits the transaction to the system configuration.

### Example

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

```
UCS-A /org* # create vnic-template vnic-primary
UCS-A /org/vnic-templ* # set type updating-template
UCS-A /org/vnic-templ* # set fabric a
UCS-A /org/vnic-templ* # set descr primaryinredundancypair
UCS-A /org/vnic-templ* # set redundancy-type primary
UCS-A /org/vnic-templ* # exit
UCS-A /org* # create vnic-templ vnicsecondary
UCS-A /org/vnic-templ* # set fabric b
UCS-A /org/vnic-templ* # set descr secondaryinredundancypair
UCS-A /org/vnic-templ* # set redundancy-type secondary
UCS-A /org/vnic-templ* # set peer-template-name vnic-primary
UCS-A /org/vnic-templ* # commit-buffer
UCS-A /org/vnic-templ #
```

### What to do next

After you create the vNIC redundancy template pair, you can use the redundancy template pair to create redundancy vNIC pairs for any service profile in the same organization or sub- organization.

## Undo vNIC Template Pairs

You can undo the vNIC template pair by changing the Peer Redundancy Template so that there is no peer template for the Primary or the Secondary template. When you undo a vNIC template pair, the corresponding vNIC pairs also becomes undone.

### SUMMARY STEPS

1. UCS-A /org # **scope vnic-templ template1**.
2. UCS-A /org/ vnic-templ # **set redundancy-type no redundancy**.
3. UCS-A /org/vnic-templ\* # **commit-buffer** .

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A /org # <b>scope vnic-templ</b> <i>template1</i> .	Specifies the name of the vNIC template that you want to undo from the template pair.
<b>Step 2</b>	UCS-A /org/ vnic-templ # <b>set redundancy-type</b> <i>no redundancy</i> .	Removes the pairing between the peer Primary or Secondary redundancy template used to perform the template pairing.
<b>Step 3</b>	UCS-A /org/vnic-templ* # <b>commit-buffer</b> .	Commits the transaction to the system configuration.

**Example**

The following example shows how to undo a template pairing:

```
UCS-A /org # scope vnic-templ template1
UCS-A /org/vnic-templ # set redundancy-type no-redundancy
UCS-A /org/vnic-templ* # commit buffer
```

## Configuring a vNIC Template

## SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create vnic-templ** *vnic-templ-name* [**eth-if** *vlan-name*] [**fabric** {**a** | **b**}] [**target** [**adapter** | **vm**]]
3. (Optional) UCS-A /org/vnic-templ # **set descr** *description*
4. (Optional) UCS-A /org/vnic-templ # **set fabric** {**a** | **a-b** | **b** | **b-a**}
5. UCS-A /org/vnic-templ # **set mac-pool** *mac-pool-name*
6. UCS-A /org/vnic-templ # **set mtu** *mtu-value*
7. UCS-A /org/vnic-templ # **set nw-control-policy** *policy-name*
8. UCS-A /org/vnic-templ # **set pin-group** *group-name*
9. UCS-A /org/vnic-templ # **set qos-policy** *policy-name*
10. UCS-A /org/vnic-templ # **set stats-policy** *policy-name*
11. UCS-A /org/vnic-templ # **set type** {**initial-template** | **updating-template**}
12. UCS-A /org/vnic-templ # **commit-buffer**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b> <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
<b>Step 2</b>	UCS-A /org # <b>create vnic-templ</b> <i>vnic-templ-name</i> [ <b>eth-if</b> <i>vlan-name</i> ] [ <b>fabric</b> { <b>a</b>   <b>b</b> }] [ <b>target</b> [ <b>adapter</b>   <b>vm</b> ]]	Creates a vNIC template and enters organization vNIC template mode.

	Command or Action	Purpose
		<p>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:</p> <ul style="list-style-type: none"> <li>• <b>Adapter</b>—The vNICs apply to all adapters. No VM-FEX port profile is created if you choose this option.</li> <li>• <b>VM</b>—The vNICs apply to all virtual machines. A VM-FEX port profile is created if you choose this option.</li> </ul>
<b>Step 3</b>	(Optional) UCS-A /org/vnic-templ # <b>set descr</b> <i>description</i>	Provides a description for the vNIC template.
<b>Step 4</b>	(Optional) UCS-A /org/vnic-templ # <b>set fabric</b> { <b>a</b>   <b>a-b</b>   <b>b</b>   <b>b-a</b> }	<p>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.</p> <p>If you want this vNIC to be able to access the second fabric interconnect if the default one is unavailable, choose <b>a-b</b> (A is the primary) or <b>b-a</b> (B is the primary) .</p> <p><b>Note</b> Do not enable fabric failover for the vNIC under the following circumstances:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• If you plan to associate this vNIC to a server with an adapter that does not support fabric failover, such as the Cisco UCS 82598KR-CI 10-Gigabit Ethernet Adapter. If you do so, Cisco UCS Manager generates a configuration fault when you associate the service profile with the server.</li> </ul>
<b>Step 5</b>	UCS-A /org/vnic-templ # <b>set mac-pool</b> <i>mac-pool-name</i>	The MAC address pool that vNICs created from this vNIC template should use.
<b>Step 6</b>	UCS-A /org/vnic-templ # <b>set mtu</b> <i>mtu-value</i>	<p>The maximum transmission unit, or packet size, that vNICs created from this vNIC template should use.</p> <p>Enter an integer between 1500 and 9000.</p>

	Command or Action	Purpose
		<p><b>Note</b></p> <p>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.</p> <p>For VIC 1400 Series adapters, you can change the MTU size of the vNIC from the host interface settings. When the Overlay network is configured, make sure that the new value is equal to or less than the MTU specified in the associated QoS system class or packets could be dropped during data transmission.</p>
<b>Step 7</b>	UCS-A /org/vnic-templ # <b>set nw-control-policy</b> <i>policy-name</i>	The network control policy that vNICs created from this vNIC template should use.
<b>Step 8</b>	UCS-A /org/vnic-templ # <b>set pin-group</b> <i>group-name</i>	The LAN pin group that vNICs created from this vNIC template should use.
<b>Step 9</b>	UCS-A /org/vnic-templ # <b>set qos-policy</b> <i>policy-name</i>	The quality of service policy that vNICs created from this vNIC template should use.
<b>Step 10</b>	UCS-A /org/vnic-templ # <b>set stats-policy</b> <i>policy-name</i>	The statistics collection policy that vNICs created from this vNIC template should use.
<b>Step 11</b>	UCS-A /org/vnic-templ # <b>set type</b> { <b>initial-template</b>   <b>updating-template</b> }	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 <b>initial-template</b> keyword; otherwise, use the <b>updating-template</b> keyword to ensure that all vNIC instances are updated when the vNIC template is updated.
<b>Step 12</b>	UCS-A /org/vnic-templ # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **delete vnic-templ** *vnic-templ-name*
3. UCS-A /org # **commit-buffer**

### DETAILED STEPS

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

### Example

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 #
```

## Ethernet Adapter Policies

### Configuring an Ethernet Adapter Policy

#### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create eth-policy** *policy-name*
3. (Optional) UCS-A /org/eth-policy # **set arfs accelaratedrfs** {**enabled** | **disabled**}
4. (Optional) UCS-A /org/eth-policy # **set comp-queue count** *count*
5. (Optional) UCS-A /org/eth-policy # **set descr** *description*
6. (Optional) UCS-A /org/eth-policy # **set failover timeout** *timeout-sec*
7. (Optional) UCS-A /org/eth-policy # **set interrupt** {**coalescing-time** *sec* | **coalescing-type** {**idle** | **min**} | **count** *count* | **mode** {**intx** | **msi** | **msi-x**}}
8. (Optional) UCS-A /org/eth-policy # **set nvgre adminstate** {**disabled** | **enabled**}



9. (Optional) UCS-A /org/eth-policy # **set offload** {**large-receive** | **tcp-rx-checksum** | **tcp-segment** | **tcp-tx-checksum**} {**disabled** | **enabled**}
10. (Optional) UCS-A /org/eth-policy # **set policy-owner** {**local** | **pending**}
11. (Optional) UCS-A /org/eth-policy # **set recv-queue** {**count** *count* | **ring-size** *size-num*}
12. (Optional) UCS-A /org/eth-policy # **set rss receivesidescaling** {**disabled** | **enabled**}
13. (Optional) UCS-A /org/eth-policy # **set trans-queue** {**count** *count* | **ring-size** *size-num*}
14. (Optional) UCS-A /org/eth-policy # **set vxlan adminstate** {**disabled** | **enabled**}
15. UCS-A /org/eth-policy # **commit-buffer**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b> <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
<b>Step 2</b>	UCS-A /org # <b>create eth-policy</b> <i>policy-name</i>	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
<b>Step 3</b>	(Optional) UCS-A /org/eth-policy # <b>set arfs acceleratdrfs</b> { <b>enabled</b>   <b>disabled</b> }	Configures Accelerated RFS.
<b>Step 4</b>	(Optional) UCS-A /org/eth-policy # <b>set comp-queue</b> <b>count</b> <i>count</i>	Configures the Ethernet completion queue.
<b>Step 5</b>	(Optional) UCS-A /org/eth-policy # <b>set descr</b> <i>description</i>	Provides a description for the policy.  <b>Note</b> 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 <b>show</b> command output.
<b>Step 6</b>	(Optional) UCS-A /org/eth-policy # <b>set failover timeout</b> <i>timeout-sec</i>	Configures the Ethernet failover.
<b>Step 7</b>	(Optional) UCS-A /org/eth-policy # <b>set interrupt</b> { <b>coalescing-time</b> <i>sec</i>   <b>coalescing-type</b> { <b>idle</b>   <b>min</b> }   <b>count</b> <i>count</i>   <b>mode</b> { <b>intx</b>   <b>msi</b>   <b>msi-x</b> }}	Configures the Ethernet interrupt.
<b>Step 8</b>	(Optional) UCS-A /org/eth-policy # <b>set nvgre adminstate</b> { <b>disabled</b>   <b>enabled</b> }	Configures NVGRE.
<b>Step 9</b>	(Optional) UCS-A /org/eth-policy # <b>set offload</b> { <b>large-receive</b>   <b>tcp-rx-checksum</b>   <b>tcp-segment</b>   <b>tcp-tx-checksum</b> } { <b>disabled</b>   <b>enabled</b> }	Configures the Ethernet offload.
<b>Step 10</b>	(Optional) UCS-A /org/eth-policy # <b>set policy-owner</b> { <b>local</b>   <b>pending</b> }	Specifies the owner for the Ethernet adapter policy.
<b>Step 11</b>	(Optional) UCS-A /org/eth-policy # <b>set recv-queue</b> { <b>count</b> <i>count</i>   <b>ring-size</b> <i>size-num</i> }	Configures the Ethernet receive queue.

	Command or Action	Purpose
<b>Step 12</b>	(Optional) UCS-A /org/eth-policy # <b>set rss receivesidescaling</b> {disabled   enabled}	Configures the RSS.
<b>Step 13</b>	(Optional) UCS-A /org/eth-policy # <b>set trans-queue</b> {count <i>count</i>   ring-size <i>size-num</i> }	Configures the Ethernet transmit queue.
<b>Step 14</b>	(Optional) UCS-A /org/eth-policy # <b>set vxlan adminstate</b> {disabled   enabled}	Configures VXLAN.
<b>Step 15</b>	UCS-A /org/eth-policy # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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 #
```

## Deleting an Ethernet Adapter Policy

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **delete eth-policy** *policy-name*
3. UCS-A /org # **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b> <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
<b>Step 2</b>	UCS-A /org # <b>delete eth-policy</b> <i>policy-name</i>	Deletes the specified Ethernet adapter policy.
<b>Step 3</b>	UCS-A /org # <b>commit-buffer</b>	Commits the transaction to the system configuration.

**Example**

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 an Ethernet Adapter Policy to Enable Stateless Offloads with NVGRE

Cisco UCS Manager supports stateless offloads with NVGRE only with Cisco UCS VIC 1340, 1380, 1385, 1387 and/or Cisco UCS VIC 1380 adapters that are installed on servers running Windows Server 2012 R2 operating systems. You cannot use NVGREs stateless offloads with Netflow, usNIC, or VM-FEX.

**SUMMARY STEPS**

1. UCS-A# **scope org org-name**
2. UCS-A /org # **create eth-policy policy-name**
3. To enable stateless offloads with NVGRE, set the following options:
4. UCS-A /org/eth-policy # **commit-buffer**
5. Install an eNIC driver Version 3.0.0.8 or later.
6. Reboot the server.

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org org-name</b>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
<b>Step 2</b>	UCS-A /org # <b>create eth-policy policy-name</b>	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
<b>Step 3</b>	To enable stateless offloads with NVGRE, set the following options:	<ul style="list-style-type: none"> <li>• Transmit Queues = 1</li> <li>• Receive Queues = n (up to 8)</li> <li>• Completion Queues = # of Transmit Queues + # of Receive Queues</li> <li>• Interrupts = # Completion Queues + 2</li> <li>• Network Virtualization using Generic Routing Encapsulation = Enabled</li> <li>• Interrupt Mode = Msi-X</li> </ul>

	Command or Action	Purpose
		<p><b>Note</b> If you set Interrupt Mode as Msi-X, and if pci=noms parameter is enabled in /boot/grub/grub.conf on RHEL system, then pci=noms would block the eNIC/fNIC driver to run in the Msi-X mode, impacting system performance.</p> <p>For more information on creating an Ethernet adapter policy, see <a href="#">Configuring an Ethernet Adapter Policy</a>, on page 8.</p>
<b>Step 4</b>	UCS-A /org/eth-policy # <b>commit-buffer</b>	Commits the transaction to the system configuration.
<b>Step 5</b>	Install an eNIC driver Version 3.0.0.8 or later.	For more information, see <a href="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">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</a> .
<b>Step 6</b>	Reboot the server.	

### Example

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 VXLAN TSO and checksum offloads only with Cisco UCS VIC 1340, 1380, 1385, 1387, adapters that are running on ESXi 5.5 and later releases. Stateless offloads with VXLAN cannot be used with NetFlow, usNIC, VM-FEX, Netqueue, or VMQ.

VXLAN with Receive Side-Scaling (RSS) support starts with the Cisco UCS Manager 3.1(2) release. RSS is supported with VXLAN stateless offload on VIC adapters 1340, 1380, 1385, 1387, and SIOC on Cisco UCS S3260 system for ESXi 5.5 and later releases.



- Note** VXLAN stateless hardware offloads are not supported with Guest OS TCP traffic over IPv6 on UCS VIC 13xx adapters. To run VXLAN encapsulated TCP traffic over IPV6, disable the VXLAN stateless offloads feature.
- To disable the VXLAN stateless offload feature in UCS Manager, disable the 'Virtual Extensible LAN' field in the Ethernet Adapter Policy.
  - To disable the VXLAN stateless offload feature in the CIMC of a Cisco C-Series UCS server, uncheck 'Enable VXLAN' field in the Ethernet Interfaces pane's vNIC properties area.

## SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create eth-policy** *policy-name*
3. To enable stateless offloads with VXLAN, set the following options:
4. UCS-A /org/eth-policy # **commit-buffer**
5. Install an eNIC driver Version 2.3.0.10 or later.
6. Reboot the server.

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b> <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
<b>Step 2</b>	UCS-A /org # <b>create eth-policy</b> <i>policy-name</i>	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
<b>Step 3</b>	To enable stateless offloads with VXLAN, set the following options:	<ul style="list-style-type: none"> <li>• Transmit Queues = 1</li> <li>• Receive Queues = n (up to 8)</li> <li>• Completion Queues = # of Transmit Queues + # of Receive Queues</li> <li>• Interrupts = # Completion Queues + 2</li> <li>• Virtual Extensible LAN = Enabled</li> <li>• Interrupt Mode = Msi-X</li> </ul> <p><b>Note</b> If you set Interrupt Mode as Msi-X, and if pci=noms parameter is enabled in /boot/grub/grub.conf on RHEL system, then pci=noms would block the eNIC/fNIC driver to run in the Msi-X mode, impacting system performance.</p> <ul style="list-style-type: none"> <li>• Receive Side Scaling = Enabled</li> </ul>

	Command or Action	Purpose
		For more information on creating an Ethernet adapter policy, see <a href="#">Configuring an Ethernet Adapter Policy</a> , on page 8.
<b>Step 4</b>	UCS-A /org/eth-policy # <b>commit-buffer</b>	Commits the transaction to the system configuration.
<b>Step 5</b>	Install an eNIC driver Version 2.3.0.10 or later.	For more information, see <a href="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">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</a> .
<b>Step 6</b>	Reboot the server.	

### Example

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 32
UCS-A /org/eth-policy* # set recv-queue count 8
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 #
```

## Ethernet and Fibre Channel Adapter Policies

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

- Queues
- Interrupt handling
- Performance enhancement
- RSS hash
- Failover in a cluster configuration with two fabric interconnects



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

- **Max LUNs Per Target**—SANsurfer has a maximum of 256 LUNs and does not display more than that number. Cisco UCS Manager supports a higher maximum number of LUNs. This parameter is applicable only for FC-Initiator.
- **Link Down Timeout**—In SANsurfer, you configure the timeout threshold for link down in seconds. In Cisco UCS Manager, you configure this value in milliseconds. Therefore, a value of 5500 ms in Cisco UCS Manager displays as 5s in SANsurfer.
- **Max Data Field Size**—SANsurfer has allowed values of 512, 1024, and 2048. Cisco UCS Manager allows you to set values of any size. Therefore, a value of 900 in Cisco UCS Manager displays as 512 in SANsurfer.
- **LUN Queue Depth**—The LUN queue depth setting is available for Windows system FC adapter policies. Queue depth is the number of commands that the HBA can send and receive in a single transmission per LUN. Windows Storport driver sets this to a default value of 20 for physical miniports and to 250 for virtual miniports. This setting adjusts the initial queue depth for all LUNs on the adapter. Valid range for this value is 1 - 254. The default LUN queue depth is 20. This feature only works with Cisco UCS Manager version 3.1(2) and higher. This parameter is applicable only for FC-Initiator.
- **IO TimeOut Retry**—When the target device does not respond to an IO request within the specified timeout, the FC adapter cancels the pending command then resends the same IO after the timer expires. The FC adapter valid range for this value is 1 - 59 seconds. The default IO retry timeout is 5 seconds. This feature only works with Cisco UCS Manager version 3.1(2) and higher.

### Operating System Specific Adapter Policies

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



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

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

Depending on the UCS firmware, your driver interrupt calculations may be different. Newer UCS firmware uses a calculation that differs from previous versions. Later driver release versions on Linux operating systems now use a different formula to calculate the Interrupt Count. In this formula, the Interrupt Count is the maximum of either the Transmit Queue or the Receive Queue plus 2.

### Interrupt Count in Linux Adapter Policies

Drivers on Linux operating systems use differing formulas to calculate the Interrupt Count, depending on the eNIC driver version. The UCS 3.2 release increased the number of Tx and Rx queues for the eNIC driver from 8 to 256 each.

Use one of the following strategies, according to your driver version.

For Linux drivers before the UCS 3.2 firmware release, use the following formula to calculate the Interrupt Count.

$$\begin{aligned} \text{Completion Queues} &= \text{Transmit Queues} + \text{Receive Queues} \\ \text{Interrupt Count} &= (\text{Completion Queues} + 2) \text{ rounded up to nearest power of } 2 \end{aligned}$$

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

$$\begin{aligned} \text{Completion Queues} &= 1 + 8 = 9 \\ \text{Interrupt Count} &= (9 + 2) \text{ rounded up to the nearest power of } 2 = 16 \end{aligned}$$

On drivers for UCS firmware release 3.2 and higher, the Linux eNIC drivers use the following formula to calculate the Interrupt Count.

$$\text{Interrupt Count} = (\# \text{Tx or Rx Queues}) + 2$$

For example:

$$\begin{aligned} \text{Interrupt Count } wq = 32, rq = 32, cq = 64 &- \text{ then Interrupt Count} = \text{Max}(32, 32) + 2 = 34 \\ \text{Interrupt Count } wq = 64, rq = 8, cq = 72 &- \text{ then Interrupt Count} = \text{Max}(64, 8) + 2 = 66 \\ \text{Interrupt Count } wq = 1, rq = 16, cq = 17 &- \text{ then Interrupt count} = \text{Max}(1, 16) + 2 = 18 \end{aligned}$$

### Interrupt Count in Windows Adapter Policies

For Windows OS, the recommended adapter policy in UCS Manager for VIC 1400 series and above adapters is Win-HPN and if RDMA is used, the recommended policy is Win-HPN-SMB. For VIC 1400 series and above adapters, the recommended interrupt value setting is 512 and the Windows VIC driver takes care of allocating the required number of Interrupts.

For VIC 1300 and VIC 1200 series adapters, the recommended UCS Manager adapter policy is Windows and the Interrupt would be TX + RX + 2, rounded to closest power of 2. The maximum supported Windows queues is 8 for Rx Queues and 1 for Tx Queues.

Example for VIC 1200 and VIC 1300 series adapters:

$$\text{Tx} = 1, \text{Rx} = 4, \text{CQ} = 5, \text{Interrupt} = 8 (1 + 4 \text{ rounded to nearest power of } 2), \text{Enable RSS}$$

Example for VIC 1400 series and above adapters:

$$\text{Tx} = 1, \text{Rx} = 4, \text{CQ} = 5, \text{Interrupt} = 512, \text{Enable RSS}$$

### NVMe over Fabrics using Fibre Channel

The NVM Express (NVMe) interface allows host software to communicate with a non-volatile memory subsystem. This interface is optimized for Enterprise non-volatile storage, which is typically attached as a register level interface to the PCI Express (PCIe) interface.

NVMe over Fabrics using Fibre Channel (FC-NVMe) defines a mapping protocol for applying the NVMe interface to Fibre Channel. This protocol defines how Fibre Channel services and specified Information Units (IUs) are used to perform the services defined by NVMe over a Fibre Channel fabric. NVMe initiators can access and transfer information to NVMe targets over Fibre Channel.



FC-NVMe combines the advantages of Fibre Channel and NVMe. You get the improved performance of NVMe along with the flexibility and the scalability of the shared storage architecture. Cisco UCS Manager Release 4.0(2) supports NVMe over Fabrics using Fibre Channel on UCS VIC 1400 Series adapters.

Starting with UCS Manager release 4.3(2b), NVMeoF using RDMA is supported on Cisco UCS VIC 14000 series adapters.

Cisco UCS Manager provides the recommended FC NVMe Initiator adapter policies in the list of pre-configured adapter policies. To create a new FC-NVMe adapter policy, follow the steps in the *Creating a Fibre Channel Adapter Policy* section.

### NVMe over Fabrics Using RDMA

NVMe over Fabrics (NVMeoF) is a communication protocol that allows one computer to access NVMe namespaces available on another computer. NVMeoF is similar to NVMe, but differs in the network-related steps involved in using the NVMeoF storage devices. The commands for discovering, connecting, and disconnecting a NVMeoF storage device are integrated into the `nvme` utility provided in Linux..

The NVMeoF fabric that Cisco supports is RDMA over Converged Ethernet version 2 (RoCEv2). RoCEv2 is a fabric protocol that runs over UDP. It requires a no-drop policy.

The eNIC RDMA driver works in conjunction with the eNIC driver, which must be loaded first when configuring NVMeoF.

Cisco UCS Manager provides the default Linux-NVMe-RoCE adapter policy for creating NVMe RoCEv2 interfaces. Do not use the default Linux adapter policy. For complete information on configuring RoCEv2 over NVMeoF, refer to the *Cisco UCS Manager Configuration Guide for RDMA over Converged Ethernet (RoCE) v2*.

NVMeoF using RDMA is supported on M5 B-Series or C-Series Servers with Cisco UCS VIC 1400 Series adapters.

Starting with UCS Manager release 4.3(2b), NVMeoF using RDMA is supported on Cisco UCS VIC 14000 series adapters.

## 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 usage CPU efficiency and reduce network 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:

- a. Turn off Interrupt Request Queue (IRQ) balance.
- b. Associate IRQ with different CPUs.
- c. 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 1200 Series
  - Cisco UCS VIC 1300 Series
  - Cisco UCS VIC 1400 Series
- ARFS is supported on the following Operating Systems:
  - Red Hat Enterprise Linux 6.5 and higher versions
  - Red Hat Enterprise Linux 7.0 and higher versions
  - SUSE Linux Enterprise Server 11 SP2 and higher versions
  - SUSE Linux Enterprise Server 12 SP1 - SP3
  - SUSE Linux Enterprise Server 15 and higher versions
  - Ubuntu 14.04.2 and higher versions

## 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 CPU 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
  - SUSE Linux Enterprise Server 11 SP2 and higher versions
  - XenServer 6.5 and higher versions
  - Ubuntu 14.04.2 and higher versions

## RDMA Over Converged Ethernet Overview

Remote Direct Memory Access (RDMA) improves performance by enabling direct data exchange in and out of a server. NVMe over Ethernet (NVMeoF) support for RDMA provides faster access to NVMe namespaces on another computer. RDMA over Converged Ethernet (RoCE) allows direct memory access over an Ethernet network. RoCE is a link layer protocol, and hence, it allows communication between any two hosts in the same Ethernet broadcast domain. RoCE delivers superior performance compared to traditional network socket implementations because of lower latency, lower CPU utilization and higher utilization of network bandwidth. Windows 2012 R2 and later versions use RDMA for accelerating and improving the performance of SMB file sharing and Live Migration.

Cisco UCS Manager supports RoCE for Microsoft SMB Direct. It sends additional configuration information to the adapter while creating or modifying an Ethernet adapter policy. Basic RoCE is also referred to as RoCE version 1 (RoCEv1), and is supported on UCS Manager releases from UCS Manager 2.2(4b) to 4.1(1a).

With Cisco UCS Manager 4.1(1a) and later releases, the RoCEv2 protocol is used.

## Guidelines for Using SMB Direct support on Windows using RDMA over converged Ethernet (RoCE) v2

### General Guidelines and Limitations:

- Cisco UCS Manager release 4.1.x and later releases support Microsoft SMB Direct with RoCEv2 on Microsoft Windows Server 2019 and later. Cisco recommends that you have all KB updates from Microsoft for your Windows Server release.



---

**Note** RoCEv2 is not supported on Microsoft Windows Server 2016.

---

- Cisco recommends you check [UCS Hardware and Software Compatibility](#) specific to your UCS Manager release to determine support for Microsoft SMB Direct with RoCEv2 on Microsoft Windows.
- Microsoft SMB Direct with RoCEv2 is supported only with Cisco UCS VIC 1400 Series and 15000 Series adapters. It is not supported with UCS VIC 1200 Series and 1300 Series adapters. SMB Direct with RoCEv2 is supported on all UCS Fabric Interconnects.



---

**Note** RoCEv1 is not supported with Cisco UCS VIC 1400 Series.

---

- RoCEv2 configuration is supported only between Cisco adapters. Interoperability between Cisco adapters and third party adapters is not supported.
- RoCEv2 supports two RoCEv2 enabled vNIC per adapter and four virtual ports per adapter interface, independent of SET switch configuration.
- RoCEv2 cannot be used on the same vNIC interface as NVGRE, NetFlow, and VMQ features.
- RoCEv2 cannot be used with usNIC.
- RoCEv2-enabled vNIC interfaces must have the no-drop QoS system class enabled in UCS Manager.
- The RoCE Properties queue pairs setting must for be a minimum of 4 queue pairs.
- Maximum number of queue pairs per adapter is 2048.
- The QoS No Drop class configuration must be properly configured on upstream switches such as Cisco Nexus 9000 series switches. QoS configurations will vary between different upstream switches.
- The maximum number of memory regions per rNIC interface is 131072.
- UCS Manager does not support fabric failover for vNICs with RoCEv2 enabled.
- SMB Direct with RoCEv2 is supported on both IPv4 and IPv6.
- RoCEv2 cannot be used with GENEVE offload.

#### **MTU Properties:**

- In older versions of the VIC driver, the MTU was derived from either a UCS Manager service profile or from the Cisco IMC vNIC MTU setting in standalone mode. This behavior changes on Cisco UCS VIC 1400 Series and later adapters, where MTU is controlled from the Windows OS Jumbo Packet advanced property. A value configured from UCS Manager or Cisco IMC has no effect.
- The RoCEv2 MTU value is always power-of-two and its maximum limit is 4096.
- RoCEv2 MTU is derived from the Ethernet MTU.
- RoCEv2 MTU is the highest power-of-two that is less than the Ethernet MTU. For example:
  - if the Ethernet value is 1500, then the RoCEv2 MTU value is 1024
  - if the Ethernet value is 4096, then the RoCEv2 MTU value is 4096
  - if the Ethernet value is 9000, then the RoCEv2 MTU value is 4096

#### **Windows NDPKI Modes of Operation:**

- Cisco's implementation of Network Direct Kernel Provider Interface (NDPKI) supports two modes of operation: Mode 1 and Mode 2. Mode 1 and Mode 2 relate to the implementation of Network Direct Kernel Provider Interface (NDKPI): Mode 1 is native RDMA, and Mode 2 involves configuration for the virtual port with RDMA. Cisco does not support NDPKI Mode 3 operation.
- The recommended default adapter policy for RoCEv2 Mode 1 is Win-HPN-SMBd .

- The recommended default adapter policy for RoCEv2 Mode 2 is MQ-SMBd.
- RoCEv2 enabled vNICs for Mode2 operation require the QoS host control policy set to full.
- Mode 2 is inclusive of Mode 1: Mode 1 must be enabled to operate Mode 2.
- On Windows, the RoCEv2 interface supports MSI & MSIx interrupt modes. By default, it is in MSIx interrupt mode. Cisco recommends you avoid changing interrupt mode when the interface is configured with RoCEv2 properties.

**Downgrade Limitations:** Cisco recommends you remove the RoCEv2 configuration before downgrading to any non-supported RoCEv2 release. If the configuration is not removed or disabled, downgrade will fail.

## Guidelines for using NVMe over Fabrics (NVMeoF) with RoCEv2 on Linux

### General Guidelines and Limitations:

- Cisco recommends you check [UCS Hardware and Software Compatibility](#) specific to your UCS Manager release to determine support for NVMeoF. NVMeoF is supported on UCS M5 and later B-Series and C-Series servers.
- NVMe over RDMA with RoCEv2 is supported with the fourth generation Cisco UCS VIC 1400 Series adapters. NVMe over RDMA is not supported on UCS 6324 Fabric Interconnects or on UCS VIC 1200 Series and 1300 Series adapters.
- When creating RoCEv2 interfaces, use Cisco UCS Manager provided Linux-NVMe-RoCE adapter policy.




---

**Note** Do not use the default Linux Adapter policy with RoCEv2; RoCEv2 interfaces will not be created in the OS.

---

- When configuring RoCEv2 interfaces, use both the enic and enic\_rdma binary drivers downloaded from Cisco.com and install the matched set of enic and enic\_rdma drivers. Attempting to use the binary enic\_rdma driver downloaded from Cisco.com with an inbox enic driver will not work.
- RoCEv2 supports maximum two RoCEv2 enabled interfaces per adapter.
- Booting from an NVMeoF namespace is not supported.
- Layer 3 routing is not supported.
- RoCEv2 does not support bonding.
- Saving a crashdump to an NVMeoF namespace during a system crash is not supported.
- NVMeoF cannot be used with usNIC, VMFEX, VxLAN, VMQ, VMMQ, NVGRE, GENEVE Offload, and DPDK features.
- Netflow monitoring is not supported on RoCEv2 interfaces.
- In the Linux-NVMe-RoCE policy, do not change values of Queue Pairs, Memory Regions, Resource Groups, and Priority settings other than to Cisco provided default values. NVMeoF functionality may not be guaranteed with different settings for Queue Pairs, Memory Regions, Resource Groups, and Priority.

- The QoS no drop class configuration must be properly configured on upstream switches such as Cisco Nexus 9000 series switches. QoS configurations will vary between different upstream switches.
- Set MTU size correctly on the VLANs and QoS policy on upstream switches.
- Spanning Tree Protocol (STP) may cause temporary loss of network connectivity when a failover or failback event occurs. To prevent this issue from occurring, disable STP on uplink switches.
- UCS Manager does not support fabric failover for vNICs with RoCEv2 enabled.

### Interrupts

- Linux RoCEv2 interface supports only MSIx interrupt mode. Cisco recommends avoiding changing interrupt mode when the interface is configured with RoCEv2 properties.
- The minimum interrupt count for using RoCEv2 with Linux is 8.

### Downgrade Limitations:

- Cisco recommends you remove the RoCEv2 configuration before downgrading to any non-supported RoCEv2 release.

## GENEVE Offload

Cisco UCS Manager now supports Generic Network Virtualization Encapsulation (GENEVE) Offload on the ESXi platform, which allows essentially any information to be encoded in a packet and passed between tunnel endpoints. GENEVE provides the overlay capability to create isolated, multi-tenant broadcast domains across data center fabrics on UCS VIC 1400 Series adapters. Using the GENEVE protocol allows you to create logical networks that span physical network boundaries.

GENEVE offload is present in all Ethernet adapter policies and is disabled by default.

Refer to the NSX-T documentation for how to implement GENEVE offload end to end configuration.

Cisco recommends configuring the following values in the Ethernet adapter policy when GENEVE offload is enabled:

- Transmit Queues: 1
- TX Ring Size: 4096
- Receive Queues: 8
- RX Ring Size: 4096
- Completion Queues: 16
- Interrupts: 32

The following features are not supported when GENEVE offload is enabled on any interface:

- Azure Stack QoS
- RoCEv2

GENEVE offload-enabled interfaces do not support usNIC, Netflow, advanced filters, NetQueue, or aRFS.

Other limitations with GENEVE offload include:

- External outer IPV6 is NOT supported with GENEVE offload.
- GENEVE offload is supported only with Cisco UCS VIC1400 series adapters. It is not supported on Cisco UCS VIC 1300 Series and 1200 Series adapters.
- GENEVE offload is supported with ESX 7.0 (NSX-T 3.0) and ESX 6.7U3(NSX-T 2.5).
- Cisco recommends that you remove the GENEVE offload configuration before downgrading to any non-supported release.

## Configuring a Default vNIC Behavior Policy

### SUMMARY STEPS

1. UCS-A# **scope org /**
2. UCS-A/org # **scope vnic-beh-policy**
3. UCS-A/org/vnic-beh-policy # **set action {hw-inherit [template\_name name] | none}**
4. UCS-A/org/vnic-beh-policy # **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org /</b>	Enters the root organization mode.
<b>Step 2</b>	UCS-A/org # <b>scope vnic-beh-policy</b>	Enters default vNIC behavior policy mode.
<b>Step 3</b>	UCS-A/org/vnic-beh-policy # <b>set action {hw-inherit [template_name name]   none}</b>	Specifies the default vNIC behavior policy. This can be one of the following: <ul style="list-style-type: none"> <li>• <b>hw-inherit</b>—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 <b>hw-inherit</b>, you can also specify a vNIC template to create the vNICs.</li> <li>• <b>none</b>—Cisco UCS Manager does not create default vNICs for a service profile. All vNICs must be explicitly created.</li> </ul>
<b>Step 4</b>	UCS-A/org/vnic-beh-policy # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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 #
```

## Deleting a vNIC from a LAN Connectivity Policy

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **scope lan-connectivity-policy** *policy-name*
3. UCS-A /org/lan-connectivity-policy # **delete vnic** *vnic-name*
4. UCS-A /org/lan-connectivity-policy # **commit-buffer**

### DETAILED STEPS

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

### Example

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 a LAN Connectivity Policy

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create lan-connectivity-policy** *policy-name*
3. (Optional) UCS-A /org/lan-connectivity-policy # **set descr** *policy-name*



4. UCS-A /org/lan-connectivity-policy # **commit-buffer**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b> <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
<b>Step 2</b>	UCS-A /org # <b>create lan-connectivity-policy</b> <i>policy-name</i>	Creates the specified LAN connectivity policy, and enters organization LAN connectivity policy mode.  This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
<b>Step 3</b>	(Optional) UCS-A /org/lan-connectivity-policy # <b>set descr</b> <i>policy-name</i>	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).
<b>Step 4</b>	UCS-A /org/lan-connectivity-policy # <b>commit-buffer</b>	Commits the transaction to the system configuration.

**Example**

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.

## Deleting a LAN Connectivity Policy

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

**SUMMARY STEPS**

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **delete lan-connectivity-policy** *policy-name*
3. UCS-A /org # **commit-buffer**

**DETAILED STEPS**

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

**Example**

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 #
```

## About the LAN and SAN Connectivity Policies

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



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

## Privileges Required for LAN and SAN Connectivity Policies

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

**Privileges Required to Create Connectivity Policies**

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

- admin—Can create LAN and SAN connectivity policies

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

### Privileges Required to Add Connectivity Policies to Service Profiles

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

## Interactions between Service Profiles and Connectivity Policies

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

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

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

## Creating a LAN Connectivity Policy

### SUMMARY STEPS

1. `UCS-A# scope org org-name`
2. `UCS-A /org # create lan-connectivity-policy policy-name`
3. (Optional) `UCS-A /org/lan-connectivity-policy # set descr policy-name`
4. `UCS-A /org/lan-connectivity-policy # commit-buffer`

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>UCS-A# scope org org-name</code>	Enters the organization mode for the specified organization. To enter the root organization mode, enter <code>/</code> as the <code>org-name</code> .
Step 2	<code>UCS-A /org # create lan-connectivity-policy policy-name</code>	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 .

	Command or Action	Purpose
		(period), and you cannot change this name after the object is saved.
<b>Step 3</b>	(Optional) UCS-A /org/lan-connectivity-policy # <b>set descr</b> <i>policy-name</i>	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).
<b>Step 4</b>	UCS-A /org/lan-connectivity-policy # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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 24](#), begin this procedure at Step 3.

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **scope lan-connectivity-policy** *policy-name*
3. UCS-A /org/lan-connectivity-policy # **create vnic** *vnic-name* [**eth-if** *eth-if-name*] [**fabric** {**a** | **b**}]
4. UCS-A /org/lan-connectivity-policy/vnic # **set fabric** {**a** | **a-b** | **b** | **b-a**}
5. UCS-A /org/lan-connectivity-policy/vnic # **set adapter-policy** *policy-name*
6. UCS-A /org/lan-connectivity-policy/vnic # **set identity** {**dynamic-mac** {*mac-addr* | **derived**} | **mac-pool** *mac-pool-name*}
7. UCS-A /org/lan-connectivity-policy/vnic # **set mtu** *size-num*
8. UCS-A /org/lan-connectivity-policy/vnic # **set nw-control-policy** *policy-name*
9. UCS-A /org/lan-connectivity-policy/vnic # **set order** {*order-num* | **unspecified**}
10. UCS-A /org/lan-connectivity-policy/vnic # **set pin-group** *group-name*
11. UCS-A /org/lan-connectivity-policy/vnic # **set qos-policy** *policy-name*
12. UCS-A /org/lan-connectivity-policy/vnic # **set stats-policy** *policy-name*

13. UCS-A /org/lan-connectivity-policy/vnic # **set template-name** *policy-name*
14. UCS-A /org/lan-connectivity-policy/vnic # **set vcon** {1 | 2 | 3 | 4 | any}
15. UCS-A /org/lan-connectivity-policy/vnic # **commit-buffer**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b> <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
<b>Step 2</b>	UCS-A /org # <b>scope lan-connectivity-policy</b> <i>policy-name</i>	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
<b>Step 3</b>	UCS-A /org/lan-connectivity-policy # <b>create vnic</b> <i>vnic-name</i> [ <b>eth-if</b> <i>eth-if-name</i> ] [ <b>fabric</b> {a   b}]	Creates a vNIC for the specified LAN connectivity policy.  This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
<b>Step 4</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>set fabric</b> {a   a-b   b   b-a}	Specifies the fabric to use for the vNIC. If you did not specify the fabric when you created the vNIC in Step 3, you have the option to specify it with this command.  If you want this vNIC to be able to access the second fabric interconnect if the default one is unavailable, choose <b>a-b</b> (A is the primary) or <b>b-a</b> (B is the primary) .  <b>Note</b> Do not enable fabric failover for the vNIC under the following circumstances: <ul style="list-style-type: none"> <li>• 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.</li> <li>• If you plan to associate this vNIC to a server with an adapter that does not support fabric failover, such as the Cisco UCS 82598KR-CI 10-Gigabit Ethernet Adapter. If you do so, Cisco UCS Manager generates a configuration fault when you associate the service profile with the server.</li> </ul>
<b>Step 5</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>set adapter-policy</b> <i>policy-name</i>	Specifies the adapter policy to use for the vNIC.

	Command or Action	Purpose
<b>Step 6</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>set identity</b> { <b>dynamic-mac</b> { <i>mac-addr</i>   <b>derived</b> }   <b>mac-pool</b> <i>mac-pool-name</i> }	Specifies the identity (MAC address) for the vNIC. You can set the identity using one of the following options: <ul style="list-style-type: none"> <li>• Create a unique MAC address in the form <i>nn:nn:nn:nn:nn</i>.</li> <li>• Derive the MAC address from one burned into the hardware at manufacture.</li> <li>• Assign a MAC address from a MAC pool.</li> </ul>
<b>Step 7</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>set mtu</b> <i>size-num</i>	Specifies the maximum transmission unit, or packet size, that this vNIC accepts. Enter an integer between 1500 and 9216. <b>Note</b> 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.
<b>Step 8</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>set nw-control-policy</b> <i>policy-name</i>	Specifies the network control policy that the vNIC should use.
<b>Step 9</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>set order</b> { <i>order-num</i>   <b>unspecified</b> }	Specifies the relative order for the vNIC.
<b>Step 10</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>set pin-group</b> <i>group-name</i>	Specifies the LAN pin group that the vNIC should use.
<b>Step 11</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>set qos-policy</b> <i>policy-name</i>	Specifies the quality of service policy that the vNIC should use.
<b>Step 12</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>set stats-policy</b> <i>policy-name</i>	Specifies the statistics collection policy that the vNIC should use.
<b>Step 13</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>set template-name</b> <i>policy-name</i>	Specifies the dynamic vNIC connectivity policy to use for the vNIC.
<b>Step 14</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>set vcon</b> { <b>1</b>   <b>2</b>   <b>3</b>   <b>4</b>   <b>any</b> }	Assigns the vNIC to the specified vCon. Use the <b>any</b> keyword to have Cisco UCS Manager automatically assign the vNIC.
<b>Step 15</b>	UCS-A /org/lan-connectivity-policy/vnic # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **scope lan-connectivity-policy** *policy-name*
3. UCS-A /org/lan-connectivity-policy # **delete vnic** *vnic-name*
4. UCS-A /org/lan-connectivity-policy # **commit-buffer**

### DETAILED STEPS

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

### Example

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 24](#), 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.

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **scope lan-connectivity-policy** *policy-name*
3. UCS-A /org/lan-connectivity-policy # **create vnic-iscsi** *iscsi-vnic-name* .
4. (Optional) UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set iscsi-adaptor-policy** *iscsi-adaptor-name*
5. (Optional) UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set auth-name** *authentication-profile-name*
6. UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set identity** { **dynamic-mac** {*dynamic-mac-address* | **derived** } | **mac-pool** *mac-pool-name* }
7. UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set iscsi-identity** {*initiator-name initiator-name* | **initiator-pool-name** *iqn-pool-name*}
8. UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set overlay-vnic-name** *overlay-vnic-name*
9. UCS-A /org/lan-connectivity-policy/vnic-iscsi # **create eth-if**
10. UCS-A /org/ex/vnic-iscsi/eth-if # **set vlanname** *vlan-name*
11. UCS-A /org/lan-connectivity-policy/vnic-iscsi # **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b> <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
<b>Step 2</b>	UCS-A /org # <b>scope lan-connectivity-policy</b> <i>policy-name</i>	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
<b>Step 3</b>	UCS-A /org/lan-connectivity-policy # <b>create vnic-iscsi</b> <i>iscsi-vnic-name</i> .	Creates an iSCSI vNIC for the specified LAN connectivity policy.  This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
<b>Step 4</b>	(Optional) UCS-A /org/lan-connectivity-policy/vnic-iscsi # <b>set iscsi-adaptor-policy</b> <i>iscsi-adaptor-name</i>	Specifies the iSCSI adaptor policy that you have created for this iSCSI vNIC.



	Command or Action	Purpose
Step 5	(Optional) UCS-A /org/lan-connectivity-policy/vnic-iscsi # <b>set auth-name</b> <i>authentication-profile-name</i>	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 <i>Creating an Authentication Profile</i> .
Step 6	UCS-A /org/lan-connectivity-policy/vnic-iscsi # <b>set identity</b> { <b>dynamic-mac</b> { <i>dynamic-mac-address</i>   <b>derived</b> }   <b>mac-pool</b> <i>mac-pool-name</i> }	Specifies the MAC address for the iSCSI vNIC. <b>Note</b> The MAC address is set only for the Cisco UCS NIC M51KR-B Adapters.
Step 7	UCS-A /org/lan-connectivity-policy/vnic-iscsi # <b>set iscsi-identity</b> { <b>initiator-name</b> <i>initiator-name</i>   <b>initiator-pool-name</b> <i>iqn-pool-name</i> }	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 # <b>set overlay-vnic-name</b> <i>overlay-vnic-name</i>	Specifies the Ethernet vNIC that is used by the iSCSI device as the overlay vNIC. For more information, see <i>Configuring a vNIC for a Service Profile</i> .
Step 9	UCS-A /org/lan-connectivity-policy/vnic-iscsi # <b>create eth-if</b>	Creates an Ethernet interface for a VLAN assigned to the iSCSI vNIC.
Step 10	UCS-A /org/ex/vnic-iscsi/eth-if # <b>set vllanname</b> <i>vlan-name</i>	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 # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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 vllanname 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 iSCSI 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

**SUMMARY STEPS**

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **scope lan-connectivity-policy** *policy-name*
3. UCS-A /org/lan-connectivity-policy # **delete vnic-iscsi** *iscsi-vnic-name*
4. UCS-A /org/lan-connectivity-policy # **commit-buffer**

**DETAILED STEPS**

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

**Example**

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 #
```

## 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 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. This configuration might result in an Ethernet teaming driver being unable 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 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.



---

**Note** Cisco UCS Manager Release 4.0(2) introduces support for **MAC Security** on Cisco UCS 6454 Fabric Interconnects.

---

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create nw-ctrl-policy** *policy-name*
3. UCS-A /org/nw-ctrl-policy # **{disable | enable} cdp**

4. UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **lldp transmit**
5. UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **lldp receive**
6. UCS-A /org/nw-ctrl-policy # **set uplink-fail-action** {**link-down** | **warning**}
7. UCS-A /org/nw-ctrl-policy # **set mac-registration-mode**{**all-host-vlans** | **only-native-vlan**}
8. UCS-A /org/nw-ctrl-policy # **create mac-security**
9. UCS-A /org/nw-ctrl-policy/mac-security # **set forged-transmit** {**allow** | **deny**}
10. UCS-A /org/nw-ctrl-policy/mac-security # **commit-buffer**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b> <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
<b>Step 2</b>	UCS-A /org # <b>create nw-ctrl-policy</b> <i>policy-name</i>	Creates the specified network control policy, and enters organization network control policy mode.
<b>Step 3</b>	UCS-A /org/nw-ctrl-policy # { <b>disable</b>   <b>enable</b> } <b>cdp</b>	Disables or enables Cisco Discovery Protocol (CDP).
<b>Step 4</b>	UCS-A /org/nw-ctrl-policy # { <b>disable</b>   <b>enable</b> } <b>lldp transmit</b>	Disables or enables the transmission of LLDP packets on an interface.
<b>Step 5</b>	UCS-A /org/nw-ctrl-policy # { <b>disable</b>   <b>enable</b> } <b>lldp receive</b>	Disables or enables the reception of LLDP packets on an interface.
<b>Step 6</b>	UCS-A /org/nw-ctrl-policy # <b>set uplink-fail-action</b> { <b>link-down</b>   <b>warning</b> }	<p>Specifies the action to be taken when no uplink port is available in end-host mode.</p> <p>Use the <b>link-down</b> 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 <b>warning</b> 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.</p>
<b>Step 7</b>	UCS-A /org/nw-ctrl-policy # <b>set mac-registration-mode</b> { <b>all-host-vlans</b>   <b>only-native-vlan</b> }	<p>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:</p> <ul style="list-style-type: none"> <li>• <b>Only Native Vlan</b>—MAC addresses are only added to the native VLAN. This option is the default, and it maximizes the port+VLAN count.</li> <li>• <b>All Host Vlans</b>—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 <i>not</i> running in Promiscuous mode.</li> </ul>

	Command or Action	Purpose
Step 8	UCS-A /org/nw-ctrl-policy # <b>create mac-security</b>	Enters organization network control policy MAC security mode.
Step 9	UCS-A /org/nw-ctrl-policy/mac-security # <b>set forged-transmit {allow   deny}</b>	Determine whether the server can use different MAC addresses when sending packets to the fabric interconnect. Entering <b>allow</b> means all server packets are accepted by the fabric interconnect, regardless of the MAC address associated with the packets. Entering <b>deny</b> means after the first packet has been sent to the fabric interconnect, all other packets must use the same MAC address or they will be silently rejected by the fabric interconnect.  If you plan to install VMware ESX on the associated server, you must configure the MAC Security to <b>allow</b> for the network control policy applied to the default vNIC. If you do not configure MAC Security for <b>allow</b> , the ESX installation may fail because the MAC security permits only one MAC address while the installation process requires more than one MAC address.
Step 10	UCS-A /org/nw-ctrl-policy/mac-security # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

The following example shows how to create a network control policy named ncp5, enable CDP, enable LLDP transmit and LLDP receive, 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 #
```

The following example shows how to create a network control policy named ncp5, enable CDP, set the uplink fail action to link-down, 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* # set uplink-fail-action link-down
UCS-A /org/nw-ctrl-policy* # commit-buffer
UCS-A /org/nw-ctrl-policy #
```

## Configuring Link Layer Discovery Protocol for Fabric Interconnect vEthernet Interfaces

Cisco UCS Manager 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.

## Displaying Network Control Policy Details

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **scope nw-ctrl-policy** {**default** | *policy-name*}
3. UCS-A /org/nw-ctrl-policy # **show detail**

### DETAILED STEPS

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

### Example

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

### SUMMARY STEPS

1. UCS-A# **scope org /**
2. UCS-A /org # **delete nwctrl-policy** *policy-name*
3. UCS-A /org # **commit-buffer**

### DETAILED STEPS

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

### Example

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 #

```

## Creating a Multicast Policy

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

### SUMMARY STEPS

1. UCS-A# **scope org**
2. UCS-A /org # **create mcast-policy** *policy-name*
3. UCS-A /org/mcast-policy\* # **commit-buffer**

## DETAILED STEPS

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

**Example**

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 #
```

## 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.

## SUMMARY STEPS

1. UCS-A# **scope org**
2. UCS-A /org # **delete mcast-policy** *policy-name*
3. UCS-A /org # **commit-buffer**

## DETAILED STEPS

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

**Example**

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 #
```

## Entering Multicast Policy Mode

### SUMMARY STEPS

1. UCS-A# **scope org**
2. UCS-A /org # **scope mcast-policy** *policy-name*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b>	Enters organization mode for the specified organization.
<b>Step 2</b>	UCS-A /org # <b>scope mcast-policy</b> <i>policy-name</i>	Enters organization multicast policy mode.

### Example

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 #
```

## Enter a Multicast Policy

You can enter an existing multicast policy using the **enter mcast-policy** *policy-name* command.

### Before you begin

Create a multicast policy.

### SUMMARY STEPS

1. UCS-A# **scope org**
2. UCS-A /org # **enter mcast-policy** *policy-name*

### DETAILED STEPS

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

**Example**

The following example shows how to create a multicast policy named `policy1` and enter `mcast-policy` mode:

```
UCS-A# scope org /
UCS-A /org # enter mcast-policy policy1
UCS-A /org/mcast-policy #
```

## Assigning a Global VLAN Multicast Policy

You can assign a multicast policy to a global VLAN in the Ethernet uplink fabric mode.

**Before you begin**

Create a VLAN.

**SUMMARY STEPS**

1. UCS-A# **scope eth-uplink**
2. UCS-A /eth-uplink # **scope vlan default**
3. UCS-A /eth-uplink/vlan # **set mcastpolicy policy-name**
4. UCS-A /eth-uplink/vlan # **commit-buffer**

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope eth-uplink</b>	Enters Ethernet uplink mode.
<b>Step 2</b>	UCS-A /eth-uplink # <b>scope vlan default</b>	Enters Ethernet uplink VLAN mode.
<b>Step 3</b>	UCS-A /eth-uplink/vlan # <b>set mcastpolicy policy-name</b>	Assigns a multicast policy to a global VLAN.
<b>Step 4</b>	UCS-A /eth-uplink/vlan # <b>commit-buffer</b>	Commits the transaction to the system configuration.

## Disassociating a Global VLAN Multicast Policy

You can disassociate a multicast policy from global VLANs in the Ethernet uplink fabric mode.



**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.

**Before you begin**

Create a Global VLAN and associate a multicast policy.

**SUMMARY STEPS**

1. UCS-A# **scope eth-uplink**
2. UCS-A /eth-uplink # **scope vlan default**
3. UCS-A /eth-uplink/vlan # **set mcastpolicy ""**
4. UCS-A /eth-uplink/vlan # **commit-buffer**

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	UCS-A# <b>scope eth-uplink</b>	Enters Ethernet uplink mode.
<b>Step 2</b>	UCS-A /eth-uplink # <b>scope vlan default</b>	Enters Ethernet uplink VLAN mode.
<b>Step 3</b>	UCS-A /eth-uplink/vlan # <b>set mcastpolicy ""</b>	Disassociates any multicast policy from the global VLAN. If you configure <b>set mcastpolicy ""</b> in a VLAN, the VLAN will inherit multicast settings from the default multicast policy.
<b>Step 4</b>	UCS-A /eth-uplink/vlan # <b>commit-buffer</b>	Commits the transaction to the system configuration.

## Disassociating a VLAN Multicast Policy

You can disassociate a VLAN from any multicast policy in the Ethernet uplink fabric mode by entering an empty string ("") as the policy name.

**Before you begin**

Create a VLAN and associate a multicast policy to the VLAN.

**SUMMARY STEPS**

1. UCS-A# **scope eth-uplink**
2. UCS-A /eth-uplink # **scope fabric {a | b}**
3. UCS-A /eth-uplink/fabric # **scope vlan vlan-name**
4. UCS-A /eth-uplink/fabric/vlan # **set mcastpolicy ""**
5. UCS-A /eth-uplink/fabric/vlan # **commit-buffer**

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	UCS-A# <b>scope eth-uplink</b>	Enters Ethernet uplink mode.
<b>Step 2</b>	Required: UCS-A /eth-uplink # <b>scope fabric {a   b}</b>	Enters Ethernet uplink fabric mode for the specified fabric interconnect.

	Command or Action	Purpose
<b>Step 3</b>	UCS-A /eth-uplink/fabric # <b>scope vlan</b> <i>vlan-name</i>	Enters Ethernet uplink fabric VLAN mode.
<b>Step 4</b>	UCS-A /eth-uplink/fabric/vlan # <b>set mcastpolicy</b> ""	Disassociates any multicast policy for the VLAN. If you configure <b>set mcastpolicy ""</b> in a VLAN, the VLAN will inherit multicast settings from the default multicast policy.
<b>Step 5</b>	UCS-A /eth-uplink/fabric/vlan # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

The following example disassociates any multicast policy from a VLAN named `vlan1` 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 #
```

## Configuring Ethernet Adapter Policies

### Configuring an Ethernet Adapter Policy

#### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create eth-policy** *policy-name*
3. (Optional) UCS-A /org/eth-policy # **set arfs accelaratedrfs** {**enabled** | **disabled**}
4. (Optional) UCS-A /org/eth-policy # **set comp-queue count** *count*
5. (Optional) UCS-A /org/eth-policy # **set descr** *description*
6. (Optional) UCS-A /org/eth-policy # **set failover timeout** *timeout-sec*
7. (Optional) UCS-A /org/eth-policy # **set interrupt** {**coalescing-time** *sec* | **coalescing-type** {**idle** | **min**} | **count** *count* | **mode** {**intx** | **msi** | **msi-x**}}
8. (Optional) UCS-A /org/eth-policy # **set nvgre adminstate** {**disabled** | **enabled**}
9. (Optional) UCS-A /org/eth-policy # **set offload** {**large-receive** | **tcp-rx-checksum** | **tcp-segment** | **tcp-tx-checksum**} {**disabled** | **enabled**}
10. (Optional) UCS-A /org/eth-policy # **set policy-owner** {**local** | **pending**}
11. (Optional) UCS-A /org/eth-policy # **set recv-queue** {**count** *count* | **ring-size** *size-num*}
12. (Optional) UCS-A /org/eth-policy # **set rss receivesidescaling** {**disabled** | **enabled**}
13. (Optional) UCS-A /org/eth-policy # **set trans-queue** {**count** *count* | **ring-size** *size-num*}
14. (Optional) UCS-A /org/eth-policy # **set vxlan adminstate** {**disabled** | **enabled**}
15. UCS-A /org/eth-policy # **commit-buffer**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# <b>scope org</b> <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
Step 2	UCS-A /org # <b>create eth-policy</b> <i>policy-name</i>	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 3	(Optional) UCS-A /org/eth-policy # <b>set arfs acceleratedrfs</b> { <b>enabled</b>   <b>disabled</b> }	Configures Accelerated RFS.
Step 4	(Optional) UCS-A /org/eth-policy # <b>set comp-queue count</b> <i>count</i>	Configures the Ethernet completion queue.
Step 5	(Optional) UCS-A /org/eth-policy # <b>set descr</b> <i>description</i>	Provides a description for the policy.  <b>Note</b> 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 <b>show</b> command output.
Step 6	(Optional) UCS-A /org/eth-policy # <b>set failover timeout</b> <i>timeout-sec</i>	Configures the Ethernet failover.
Step 7	(Optional) UCS-A /org/eth-policy # <b>set interrupt</b> { <b>coalescing-time</b> <i>sec</i>   <b>coalescing-type</b> { <b>idle</b>   <b>min</b> }   <b>count</b> <i>count</i>   <b>mode</b> { <b>intx</b>   <b>msi</b>   <b>msi-x</b> }}	Configures the Ethernet interrupt.
Step 8	(Optional) UCS-A /org/eth-policy # <b>set nvgre adminstate</b> { <b>disabled</b>   <b>enabled</b> }	Configures NVGRE.
Step 9	(Optional) UCS-A /org/eth-policy # <b>set offload</b> { <b>large-receive</b>   <b>tcp-rx-checksum</b>   <b>tcp-segment</b>   <b>tcp-tx-checksum</b> } { <b>disabled</b>   <b>enabled</b> }	Configures the Ethernet offload.
Step 10	(Optional) UCS-A /org/eth-policy # <b>set policy-owner</b> { <b>local</b>   <b>pending</b> }	Specifies the owner for the Ethernet adapter policy.
Step 11	(Optional) UCS-A /org/eth-policy # <b>set recv-queue</b> { <b>count</b> <i>count</i>   <b>ring-size</b> <i>size-num</i> }	Configures the Ethernet receive queue.
Step 12	(Optional) UCS-A /org/eth-policy # <b>set rss receivesidescaling</b> { <b>disabled</b>   <b>enabled</b> }	Configures the RSS.
Step 13	(Optional) UCS-A /org/eth-policy # <b>set trans-queue</b> { <b>count</b> <i>count</i>   <b>ring-size</b> <i>size-num</i> }	Configures the Ethernet transmit queue.
Step 14	(Optional) UCS-A /org/eth-policy # <b>set vxlan adminstate</b> { <b>disabled</b>   <b>enabled</b> }	Configures VXLAN.
Step 15	UCS-A /org/eth-policy # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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 #
```

## Deleting an Ethernet Adapter Policy

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **delete eth-policy** *policy-name*
3. UCS-A /org # **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b> <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
<b>Step 2</b>	UCS-A /org # <b>delete eth-policy</b> <i>policy-name</i>	Deletes the specified Ethernet adapter policy.
<b>Step 3</b>	UCS-A /org # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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

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

# Configuring the Default vNIC Behavior Policy

## Default vNIC Behavior Policy

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

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

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



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

## Configuring a Default vNIC Behavior Policy

### SUMMARY STEPS

1. UCS-A# **scope org /**
2. UCS-A/org # **scope vnic-beh-policy**
3. UCS-A/org/vnic-beh-policy # **set action {hw-inherit [template\_name name] | none}**
4. UCS-A/org/vnic-beh-policy # **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org /</b>	Enters the root organization mode.
<b>Step 2</b>	UCS-A/org # <b>scope vnic-beh-policy</b>	Enters default vNIC behavior policy mode.
<b>Step 3</b>	UCS-A/org/vnic-beh-policy # <b>set action {hw-inherit [template_name name]   none}</b>	Specifies the default vNIC behavior policy. This can be one of the following: <ul style="list-style-type: none"> <li>• <b>hw-inherit</b>—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.</li> </ul> <p>If you specify <b>hw-inherit</b>, you can also specify a vNIC template to create the vNICs.</p>

	Command or Action	Purpose
		<ul style="list-style-type: none"> <li><b>none</b>—Cisco UCS Manager does not create default vNICs for a service profile. All vNICs must be explicitly created.</li> </ul>
<b>Step 4</b>	UCS-A/org/vnic-beh-policy # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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 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.



**Note** Cisco UCS Manager Release 4.0(2) introduces support for **MAC Security** on Cisco UCS 6454 Fabric Interconnects.

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create nw-ctrl-policy** *policy-name*
3. UCS-A /org/nw-ctrl-policy # **{disable | enable} cdp**
4. UCS-A /org/nw-ctrl-policy # **{disable | enable} lldp transmit**
5. UCS-A /org/nw-ctrl-policy # **{disable | enable} lldp receive**
6. UCS-A /org/nw-ctrl-policy # **set uplink-fail-action {link-down | warning}**
7. UCS-A /org/nw-ctrl-policy # **set mac-registration-mode {all-host-vlans | only-native-vlan}**
8. UCS-A /org/nw-ctrl-policy # **create mac-security**
9. UCS-A /org/nw-ctrl-policy/mac-security # **set forged-transmit {allow | deny}**
10. UCS-A /org/nw-ctrl-policy/mac-security # **commit-buffer**



## DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# <b>scope org</b> <i>org-name</i>	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the <i>org-name</i> .
Step 2	UCS-A /org # <b>create nw-ctrl-policy</b> <i>policy-name</i>	Creates the specified network control policy, and enters organization network control policy mode.
Step 3	UCS-A /org/nw-ctrl-policy # { <b>disable</b>   <b>enable</b> } <b>cdp</b>	Disables or enables Cisco Discovery Protocol (CDP).
Step 4	UCS-A /org/nw-ctrl-policy # { <b>disable</b>   <b>enable</b> } <b>lldp transmit</b>	Disables or enables the transmission of LLDP packets on an interface.
Step 5	UCS-A /org/nw-ctrl-policy # { <b>disable</b>   <b>enable</b> } <b>lldp receive</b>	Disables or enables the reception of LLDP packets on an interface.
Step 6	UCS-A /org/nw-ctrl-policy # <b>set uplink-fail-action</b> { <b>link-down</b>   <b>warning</b> }	Specifies the action to be taken when no uplink port is available in end-host mode.  Use the <b>link-down</b> 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 <b>warning</b> 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 # <b>set mac-registration-mode</b> { <b>all-host-vlans</b>   <b>only-native-vlan</b> }	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: <ul style="list-style-type: none"> <li>• <b>Only Native Vlan</b>—MAC addresses are only added to the native VLAN. This option is the default, and it maximizes the port+VLAN count.</li> <li>• <b>All Host Vlans</b>—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 <i>not</i> running in Promiscuous mode.</li> </ul>
Step 8	UCS-A /org/nw-ctrl-policy # <b>create mac-security</b>	Enters organization network control policy MAC security mode.
Step 9	UCS-A /org/nw-ctrl-policy/mac-security # <b>set forged-transmit</b> { <b>allow</b>   <b>deny</b> }	Determine whether the server can use different MAC addresses when sending packets to the fabric interconnect. Entering <b>allow</b> means all server packets are accepted by the fabric interconnect, regardless of the MAC address associated with the packets. Entering <b>deny</b> means after the first packet has been sent to the fabric interconnect, all

	Command or Action	Purpose
		<p>other packets must use the same MAC address or they will be silently rejected by the fabric interconnect.</p> <p>If you plan to install VMware ESX on the associated server, you must configure the MAC Security to <b>allow</b> for the network control policy applied to the default vNIC. If you do not configure MAC Security for <b>allow</b>, the ESX installation may fail because the MAC security permits only one MAC address while the installation process requires more than one MAC address.</p>
<b>Step 10</b>	UCS-A /org/nw-ctrl-policy/mac-security # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

The following example shows how to create a network control policy named `ncp5`, enable CDP, enable LLDP transmit and LLDP receive, 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 #
```

The following example shows how to create a network control policy named `ncp5`, enable CDP, set the uplink fail action to link-down, 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* # set uplink-fail-action link-down
UCS-A /org/nw-ctrl-policy* # commit-buffer
UCS-A /org/nw-ctrl-policy #
```

## Deleting a Network Control Policy

### SUMMARY STEPS

1. UCS-A# **scope org /**
2. UCS-A /org # **delete nwctrl-policy policy-name**
3. UCS-A /org # **commit-buffer**

## DETAILED STEPS

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

**Example**

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, IGMP querier, and IGMP source IP proxy. IGMP Snooping dynamically determines hosts in a VLAN that should be included in particular multicast transmissions. You can create, modify, and delete a multicast policy that can be associated to one or more VLANs. When a multicast policy is modified, all VLANs associated with that multicast policy are re-processed to apply the changes. For private VLANs, you can set a multicast policy for primary VLANs but not for their associated isolated VLANs due to a Cisco NX-OS forwarding implementation.

By default, IGMP snooping is enabled and IGMP querier is disabled. When IGMP snooping is enabled, the fabric interconnects send the IGMP queries only to the hosts. They do not send IGMP queries to the upstream network. To send IGMP queries to the upstream, do one of the following:

- Configure IGMP querier on the upstream fabric interconnect with IGMP snooping enabled
- Disable IGMP snooping on the upstream fabric interconnect
- Change the fabric interconnects to switch mode

By default, IGMP Source IP Proxy state is enabled. When IGMP Source IP Proxy is enabled, the fabric interconnect acts as a proxy for its hosts and manages the membership of hosts and routing devices in multicast groups. IP hosts use IGMP to report their multicast group memberships to any immediately neighboring multicast routing devices. When IGMP source IP proxy is disabled, the fabric interconnect will forward the IGMP messages from the hosts towards the upstream router or switch without any change.

The following limitations and guidelines apply to multicast policies:

- Only the default multicast policy is allowed for a global VLAN.
- If a Cisco UCS domain includes 6300 and 6200 series fabric interconnects, any multicast policy can be assigned.

- We highly recommend you use the same IGMP snooping state on the fabric interconnects and the associated LAN switches. For example, if IGMP snooping is disabled on the fabric interconnects, it should be disabled on any associated LAN switches as well.
- The option to enable or disable IGMP source IP proxy is supported on Cisco UCS UCS 6400, UCS 6300, and UCS 6200 Series fabric interconnects.

## Creating a Multicast Policy

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

### SUMMARY STEPS

1. UCS-A# **scope org**
2. UCS-A /org # **create mcast-policy** *policy-name*
3. UCS-A /org/mcast-policy\* # **commit-buffer**

### DETAILED STEPS

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

### Example

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 Parameters

You can configure the following parameters for a multicast policy:

1. Enable or disable IGMP snooping. The default state is enabled.
2. Set the IGMP snooping querier state and IPv4 address. The default state is disabled.
3. Set the IGMP source IP proxy state. The default state is enabled.

### SUMMARY STEPS

1. UCS-A# **scope org**
2. UCS-A /org # **create mcast-policy** *policy-name*

3. UCS-A /org/mcast-policy\* # **set querier** {enabled | disabled}
4. UCS-A /org/mcast-policy\* # **set querierip** *IGMP snooping querier IPv4 address*
5. UCS-A /org/mcast-policy\* # **set snooping** {enabled | disabled}
6. UCS-A /org/mcast-policy\* # **set source-ip-proxy** {enabled | disabled}
7. UCS-A /org/mcast-policy\* # **commit-buffer**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b>	Enters organization mode for the specified organization.
<b>Step 2</b>	UCS-A /org # <b>create mcast-policy</b> <i>policy-name</i>	Creates a new multicast policy with the specified policy name, and enters organization multicast policy mode.
<b>Step 3</b>	UCS-A /org/mcast-policy* # <b>set querier</b> {enabled   disabled}	Enables or disables IGMP snooping querier. By default, IGMP snooping querier is disabled for a multicast policy.
<b>Step 4</b>	UCS-A /org/mcast-policy* # <b>set querierip</b> <i>IGMP snooping querier IPv4 address</i>	Specifies the IPv4 address for the IGMP snooping querier.
<b>Step 5</b>	UCS-A /org/mcast-policy* # <b>set snooping</b> {enabled   disabled}	Enables or disables IGMP snooping. By default, IGMP snooping is enabled for a multicast policy.
<b>Step 6</b>	UCS-A /org/mcast-policy* # <b>set source-ip-proxy</b> {enabled   disabled}	Enables or disables IGMP source IP proxy. By default, IGMP source IP proxy state is enabled for a multicast policy.  <b>Note</b> IGMP source IP proxy is supported on Cisco UCS 6400 Series, Cisco UCS 6300 Series, and Cisco UCS 6200 Series Fabric Interconnects.
<b>Step 7</b>	UCS-A /org/mcast-policy* # <b>commit-buffer</b>	Commits the transaction to the system configuration.  <b>Note</b> Follow these guidelines if you choose to set IGMP Snooping querier IP addresses for a multicast policy: <ol style="list-style-type: none"> <li>a. In the Ethernet Switch-Mode configuration, you must set the querier IP addresses for each FI in the domain.</li> <li>b. In the Ethernet End-Host mode, you can set the querier IP address just for FI A, and optionally for FI B as well. If an IP address is not set explicitly for FI-B, it uses the same address set for FI A.</li> </ol>

### Example

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* # set source-ip-proxy 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, IGMP snooping querier, or IGMP source IP proxy state. When a multicast policy is modified, all VLANs associated with that multicast policy are re-processed to apply the changes.

### SUMMARY STEPS

1. UCS-A# **scope org**
2. UCS-A /org # **scope mcast-policy** *policy-name*
3. UCS-A /org/mcast-policy\* # **set querier**{enabled | disabled}
4. UCS-A /org/mcast-policy\* # **set querierip** *IGMP snooping querier IPv4 address*
5. UCS-A /org/mcast-policy\* # **set snooping**{enabled | disabled}
6. UCS-A /org/mcast-policy\* # **set-source-ip-proxy**{enabled | disabled}
7. UCS-A /org/mcast-policy\* # **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b>	Enters organization mode for the specified organization.
<b>Step 2</b>	UCS-A /org # <b>scope mcast-policy</b> <i>policy-name</i>	Enters organization multicast policy mode.
<b>Step 3</b>	UCS-A /org/mcast-policy* # <b>set querier</b> {enabled   disabled}	Enables or disables IGMP snooping querier. By default, IGMP snooping querier is disabled for a multicast policy.
<b>Step 4</b>	UCS-A /org/mcast-policy* # <b>set querierip</b> <i>IGMP snooping querier IPv4 address</i>	Specifies the IPv4 address for the IGMP snooping querier.
<b>Step 5</b>	UCS-A /org/mcast-policy* # <b>set snooping</b> {enabled   disabled}	Enables or disables IGMP snooping. By default, IGMP snooping is enabled for a multicast policy.
<b>Step 6</b>	UCS-A /org/mcast-policy* # <b>set-source-ip-proxy</b> {enabled   disabled}	Enables or disables IGMP source IP proxy. By default, IGMP source IP proxy state is enabled for a multicast policy.
<b>Step 7</b>	UCS-A /org/mcast-policy* # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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* # set source-ip-proxy 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.

### SUMMARY STEPS

1. UCS-A# **scope eth-uplink**
2. UCS-A /eth-uplink # **scope fabric {a | b}**
3. UCS-A /eth-uplink/fabric # **scope vlan vlan-name**
4. UCS-A /eth-uplink/fabric/vlan # **set mcastpolicy policy-name**
5. UCS-A /eth-uplink/fabric/vlan # **commit-buffer**

### DETAILED STEPS

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

### Example

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.

### SUMMARY STEPS

1. UCS-A# **scope org**
2. UCS-A /org # **delete mcast-policy** *policy-name*
3. UCS-A /org # **commit-buffer**

### DETAILED STEPS

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

### Example

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 #
```

## 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

### SUMMARY STEPS

1. UCS-A# **scope org**
2. UCS-A /org # **create lacppolicy***policy nam.*
3. UCS-A /org # **commit-buffer**

### DETAILED STEPS

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

### Example

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

### SUMMARY STEPS

1. UCS-A# **scope org**
2. UCS-A /org # **scope lacppolicy** *policy-name .*
3. UCS-A /org/lacp policy/ *policy-name* # **set suspend-individual** *true .*
4. UCS-A /org/lacp policy/ *policy-name* # **set lacp-rate** *fast .*
5. UCS-A /org/lacp policy/ *policy-name* # **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# <b>scope org</b>	Enters the root organization mode.

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

### Example

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 procedure describes assigning the lacp policy to port-channels.

### SUMMARY STEPS

1. UCS-A# **scope eth-uplink**
2. UCS-A /eth-uplink # **scope fabric**
3. UCS-A /eth-uplink/fabric # **scope port-channel**
4. UCS-A /eth-uplink/fabric/port-channel # **set lacp-policy-name***policy-name*
5. UCS-A /eth-uplink/ fabric/port-channel **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope eth-uplink</b>	Enters Ethernet uplink mode.
<b>Step 2</b>	UCS-A /eth-uplink # <b>scope fabric</b>	Enters the fabric mode.
<b>Step 3</b>	UCS-A /eth-uplink/fabric # <b>scope port-channel</b>	Enters the port-channel mode.

	Command or Action	Purpose
Step 4	UCS-A /eth-uplink/fabric/port-channel # <b>set lacp-policy-name</b> <i>policy-name</i>	Specifies the lacp policy for this port-channel.
Step 5	UCS-A /eth-uplink/ fabric/port-channel <b>commit-buffer</b>	Commits the transaction to the system.

### Example

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/fabric # 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
UCS-A UCS-A/eth-uplink/port-channel #
```

## 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 UDLD Link Policy

### SUMMARY STEPS

1. UCS-A# **scope org /**
2. UCS-A /org # **create udld-link-policy link-policy-name**
3. UCS-A /org/udld-link-policy # **commit-buffer**
4. UCS-A /org/udld-link-policy # **exit**
5. UCS-A /org # **scope udld-link-policy link-policy-name**
6. UCS-A /org/udld-link-policy # **set mode {aggressive | normal}**
7. UCS-A /org/udld-link-policy # **set admin-state {disabled | enabled}**
8. UCS-A /org/udld-link-policy # **commit-buffer**

### DETAILED STEPS

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

### Example

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

### SUMMARY STEPS

1. UCS-A# **scope org /**
2. UCS-A /org # **show udld-policy**
3. UCS-A /org # **scope udld-policy default**
4. UCS-A /org/udld-policy # **set message-interval seconds**
5. UCS-A /org/udld-policy # **set recovery-action [reset | none]**
6. UCS-A /org/udld-policy # **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org /</b>	Enters the root organization mode.
<b>Step 2</b>	UCS-A /org # <b>show udld-policy</b>	Displays the current UDLD system settings.
<b>Step 3</b>	UCS-A /org # <b>scope udld-policy default</b>	Enters UDLD policy mode for the global UDLD policy.
<b>Step 4</b>	UCS-A /org/udld-policy # <b>set message-interval seconds</b>	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.
<b>Step 5</b>	UCS-A /org/udld-policy # <b>set recovery-action [reset   none]</b>	Specifies the action to be taken on any ports that are disabled when UDLD aggressive mode is enabled. The default is none.
<b>Step 6</b>	UCS-A /org/udld-policy # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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 uddl-policy

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

UCS-A /chassis/org # scope uddl-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 #

```

## Configuring a Link Profile

### SUMMARY STEPS

1. UCS-A# **scope org /**
2. UCS-A /org # **create eth-link-profile** *link-profile-name*
3. UCS-A /org/eth-link-profile # **commit-buffer**
4. UCS-A /org/eth-link-profile # **exit**
5. UCS-A /org # **scope eth-link-profile** *link-profile-name*
6. UCS-A /org/eth-link-profile # **set uddl-link-policy** *link-policy-name*
7. UCS-A /org/eth-link-profile # **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org /</b>	Enters the root organization mode.
<b>Step 2</b>	UCS-A /org # <b>create eth-link-profile</b> <i>link-profile-name</i>	Creates a link profile with the specified name, and enters link profile mode.
<b>Step 3</b>	UCS-A /org/eth-link-profile # <b>commit-buffer</b>	Commits the transaction to the system configuration.
<b>Step 4</b>	UCS-A /org/eth-link-profile # <b>exit</b>	Returns to the previous mode.
<b>Step 5</b>	UCS-A /org # <b>scope eth-link-profile</b> <i>link-profile-name</i>	Enters link profile mode for the specified link profile.
<b>Step 6</b>	UCS-A /org/eth-link-profile # <b>set uddl-link-policy</b> <i>link-policy-name</i>	Assigns the specified UDLD link policy to the link profile.
<b>Step 7</b>	UCS-A /org/eth-link-profile # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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
```

## Assigning a Link Profile to a Port Channel Ethernet Interface

### SUMMARY STEPS

1. UCS-A# **scope eth-uplink**
2. UCS-A /eth-uplink # **scope fabric {a | b}**
3. UCS-A /eth-uplink/fabric # **scope port-channel port-chan-id**
4. UCS-A /eth-uplink/fabric/port-channel # **scope member-port slot-id port-id**
5. UCS-A /eth-uplink/fabric/port-channel/member-port # **set eth-link-profile link-profile-name**
6. UCS-A /eth-uplink/fabric/port-channel/member-port # **commit-buffer**

### DETAILED STEPS

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

### Example

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

### SUMMARY STEPS

1. UCS-A# **scope fc-uplink**
2. UCS-A /fc-uplink # **scope fabric {a | b}**
3. UCS-A /fc-uplink/fabric # **scope fcoe-port-channel port-chan-id**
4. UCS-A /fc-uplink/fabric/fcoe-port-channel # **scope fcoe-member-port slot-id port-id**
5. UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # **set eth-link-profile link-profile-name**
6. UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	UCS-A# <b>scope fc-uplink</b>	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # <b>scope fabric {a   b}</b>	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # <b>scope fcoe-port-channel port-chan-id</b>	Enters Fibre Channel uplink fabric port channel mode for the specified port channel.
Step 4	UCS-A /fc-uplink/fabric/fcoe-port-channel # <b>scope fcoe-member-port slot-id port-id</b>	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 # <b>set eth-link-profile link-profile-name</b>	Assigns the specified link profile.
Step 6	UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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

### SUMMARY STEPS

1. UCS-A# **scope eth-uplink**
2. UCS-A /eth-uplink # **scope fabric {a | b}**

3. UCS-A /eth-uplink/fabric # **scope interface** *slot-num port num*
4. UCS-A /eth-uplink/fabric/interface # **set eth-link-profile** *link-profile-name*
5. UCS-A /eth-uplink/fabric/interface # **commit-buffer**

### DETAILED STEPS

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

### Example

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

### SUMMARY STEPS

1. UCS-A# **scope fc-uplink**
2. UCS-A /fc-uplink # **scope fabric** {a | b}
3. UCS-A /fc-uplink/fabric # **scope fcoeinterface** *slot-num port num*
4. UCS-A /fc-uplink/fabric/fcoeinterface # **set eth-link-profile** *link-profile-name*
5. UCS-A /fc-uplink/fabric/fcoeinterface # **commit-buffer**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope fc-uplink</b>	Enters Fibre Channel uplink mode.
<b>Step 2</b>	UCS-A /fc-uplink # <b>scope fabric</b> {a   b}	Enters Fibre Channel uplink fabric mode for the specified fabric.
<b>Step 3</b>	UCS-A /fc-uplink/fabric # <b>scope fcoeinterface</b> <i>slot-num port num</i>	Enters the Fibre Channel interface command mode for the specified uplink port.

	Command or Action	Purpose
Step 4	UCS-A /fc-uplink/fabric/fcoeinterface # <b>set eth-link-profile</b> <i>link-profile-name</i>	Assigns the specified link profile.
Step 5	UCS-A /fc-uplink/fabric/fcoeinterface # <b>commit-buffer</b>	Commits the transaction to the system configuration.

### Example

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 #
```

## 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

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create vmq-conn-policy** *policy-name*
3. UCS-A /org/vmq-conn-policy\* # **set queue-count***queue count*
4. UCS-A /org/vmq-conn-policy\* # **set interrupt-count***interrupt count*
5. UCS-A /org/vmq-conn-policy\* # **commit-buffer**

### DETAILED STEPS

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

### Example

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
UCS-A /org/vmq-conn-policy #
```

## Creating a VMMQ Connection Policy

### SUMMARY STEPS

1. UCS-A# **scope org** *org-name*
2. UCS-A /org # **create vmq-conn-policy** *policy-name*
3. UCS-A /org/vmq-conn-policy\* # **set multi-queue enabled**
4. UCS-A /org/vmq-conn-policy\* # **set vmmq-sub-vnic-count** *vnic count*
5. UCS-A /org/vmq-conn-policy\* # **set vmmq-adaptor-profile-name** *profile-name*
6. UCS-A /org/vmq-conn-policy\* # **commit-buffer**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope org</b> <i>org-name</i>	Enters organization mode for the specified organization. To enter the root organization mode, type / as the <i>org-name</i> .
<b>Step 2</b>	UCS-A /org # <b>create vmq-conn-policy</b> <i>policy-name</i>	Specifies the name for this VMQ connection policy.
<b>Step 3</b>	UCS-A /org/vmq-conn-policy* # <b>set multi-queue enabled</b>	Enables multi-queue for the VMQ connection policy.
<b>Step 4</b>	UCS-A /org/vmq-conn-policy* # <b>set vmmq-sub-vnic-count</b> <i>vnic count</i>	Specifies the sub-vnic count for the VMMQ connection policy.
<b>Step 5</b>	UCS-A /org/vmq-conn-policy* # <b>set vmmq-adaptor-profile-name</b> <i>profile-name</i>	Specifies the profile name for the VMMQ connection policy.
<b>Step 6</b>	UCS-A /org/vmq-conn-policy* # <b>commit-buffer</b>	Commits the transaction to the system.

## Example

The following example creates a VMMQ connection policy:

```
UCS-A# scope org
UCS-A /org # create vmq-conn-policy vmmq
UCS-A /org/vmq-conn-policy* # set multi-queue enabled
UCS-A /org/vmq-conn-policy* # set vmmq-sub-vnic-count 4
UCS-A /org/vmq-conn-policy* # set vmmq-adaptor-profile-name MQ-SMBd
UCS-A /org/vmq-conn-policy* # commit-buffer
```

## Assigning a VMQ Connection Policy to a vNIC

## SUMMARY STEPS

1. UCS-A# **scope vnic** *vnic-name*
2. UCS-A /org # **create vmq-conn-policy-ref** *policy-name*
3. UCS-A /org/vmq-conn-policy\* # **commit-buffer**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	UCS-A# <b>scope vnic</b> <i>vnic-name</i>	Enters command mode for the specified vnic.
<b>Step 2</b>	UCS-A /org # <b>create vmq-conn-policy-ref</b> <i>policy-name</i>	Specifies the policy name to assign for this VMQ connection policy.
<b>Step 3</b>	UCS-A /org/vmq-conn-policy* # <b>commit-buffer</b>	Commits the transaction to the system.

### Example

The following example assigns a VMQ connection policy to a vNIC:

```
UCS-A# /org/service-profile* scope vnic vnic
UCS-A /org/service-profile/vnic* # create vmq-conn-policy-ref vmq
UCS-A /org/service-profile/vnic* # commit-buffer
```