Cisco Nexus 1000V for VMware vSphere Interface Configuration Guide, Release 5.x

First Published: August 22, 2014

Last Modified: November 05, 2015

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: http://www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2009-2015 Cisco Systems, Inc. All rights reserved.

CONTENTS

CHAPTER 1

New and Changed Information

This chapter contains the following sections:

New and Changed Information

This section lists new and changed content in this document by software release.

To find additional information about new features or command changes, see the *Cisco Nexus 1000V Release Notes* and *Cisco Nexus 1000V Command Reference*.

Table 1: New and Changed Features for the Cisco Nexus 1000V Interface Configuration Guide

Feature	Description	Changed in Release	Where Documented
Port channel vPC-HM	The interface level configuration mode does not support sub-group cdp and manual options for port-channel vPC-HM configuration. Sub-group cdp and manual options for port-channel vPC-HM configuration are supported on the port-profile level configuration.	4.2(1)SV2(1.1)	Configuring Port Channels, on page 45
Backup subgroups	You can assign up to seven backup subgroups when pinning the primary subgroup.	4.2(1)SV1(4a)	Configuring Port Channels, on page 45
Port channel relative numbering	The subgroup numbering begins at zero and is not tied to the vmnic number.	4.2(1)SV1(4a)	Configuring Port Channels, on page 45
Network state tracking for vPC-HM	Pinpoints link failures on a port channel configured for vPC-HM.	4.2(1)SV1(4)	Configuring Port Channels, on page 45

Feature	Description	Changed in Release	Where Documented
VEM management of LACP	You can offload operation of the LACP protocol from the VSM to the VEMs.	4.2(1)SV1(4)	Configuring Port Channels, on page 45
LACP	You can enable the LACP port channel function by turning on the feature using the **feature lacp** command.	4.2(1)SV1(4)	Configuring Port Channels, on page 45
System Jumbo MTU	The system jumbo MTU value is fixed at 9000 and cannot be changed.	4.2(1)SV1(4)	Configuring Interface Parameters, on page 11
Interface MTU	You can configure an interface MTU between 1500 and 9000.	4.2(1)SV1(4)	Configuring Interface Parameters, on page 11
Mapping vEthernet interfaces to connected ports	vEthernet interfaces are now mapped to connected ports by MAC addresses as well as the DVPort number.	4.2(1)SV1(4)	Configuring Virtual Ethernet Interfaces, on page 35

Feature	Description	Changed in Release	Where Documented
Global vEthernet interface controls	You can enable or disable the automatic vEthernet interface controls by deleting unused vEthernet interfaces, purging manual vEthernet configurations, and creating vEthernet interfaces.	4.2(1)SV1(4)	Configuring Virtual Ethernet Interfaces, on page 35
Configuration limits	Configuration limits for vEthernet interfaces, vEthernet trunks, and port profiles were added.	4.0(4)SV1(2)	Interface Configuration Limits, on page 89

Feature	Description	Changed in Release	Where Documented
show interface vethernet command	The **show interface vethernet** command now displays 5-minute input and output packet/bit rate statistics for the interfaces that you specify. The configuration example showing this command output was updated to reflect this change. The **show interface ethernet** command output also provides these new statistics.	4.0(4)SV1(2)	Configuring Virtual Ethernet Interfaces, on page 35
vPC-Host Mode	Support for manual creation of subgroups.	4.0(4)SV1(2)	Configuring Port Channels, on page 45
Static Pinning	Support for attaching (or pinning) a vEthernet interface to a specific port channel subgroup.	4.0(4)SV1(2)	Configuring Port Channels, on page 45

CHAPTER 2

Overview

This chapter contains the following sections:

Information About Interfaces

Ethernet Interfaces

All interfaces on the Cisco Nexus 1000V are Layer 2 Ethernet interfaces, which include access ports, trunk ports, private VLAN ports, and promiscuous ports.

Access Ports

An access port carries traffic for one VLAN. This type of port is a Layer 2 interface only.

Private VLAN Ports

Private VLANs (PVLANs) are used to segregate Layer 2 ISP traffic and convey it to a single router interface. PVLANs achieve device isolation by applying Layer 2 forwarding constraints that allow end devices to share the same IP subnet while being isolated by Layer 2. The use of larger subnets reduces address management overhead. Three separate port designations are used. Each has its own unique set of rules that regulate the ability of each connected endpoint to communicate with other connected endpoints within the same PVLAN domain.

For more information about PVLANs, see the *Cisco Nexus 1000V Layer 2 Switching Configuration Guide*.

Promiscuous Ports

A promiscuous port can talk to all other types of ports. A promiscuous port can talk to isolated ports as well as community ports, and those ports can also talk to promiscuous ports.

For more information about promiscuous ports, see the *Cisco Nexus 1000V Layer 2 Switching Configuration Guide*

Trunk Ports

A trunk port carries traffic for two or more VLANs. This port type is a Layer 2 interface only.

Virtual Ethernet Interfaces

Virtual Ethernet (vEthernet or vEth) interfaces are logical interfaces. Each vEthernet interface corresponds to a switch interface that is connected to a virtual port. The interface types are as follows:

- VM (interfaces connected to VM NICs)
- Service console
- vmkernel

vEthernet interfaces are created on the Cisco Nexus 1000V to represent virtual ports in use on the distributed virtual switch.

Management Interface

You can use the management Ethernet interface to connect the device to a network for remote management using a Telnet client, the Simple Network Management Protocol (SNMP), or other management agents.

Port Channel Interfaces

A port channel is a logical interface that aggregates multiple physical interfaces. You can bundle up to eight individual links to physical ports into a port channel to improve bandwidth and redundancy. You can also use port channeling to load balance traffic across these channeled physical interfaces.

VEM Management of LACP

You can offload operation of the Line Aggregation Control Protocol (LACP) from the VSM to the VEMs to prevent a situation where LACP cannot be negotiated with the upstream switch when the VEM is disconnected from the VSM (referred to as headless mode). VEM management of LACP allows port channels to be reestablished after the reboot of a headless VEM.

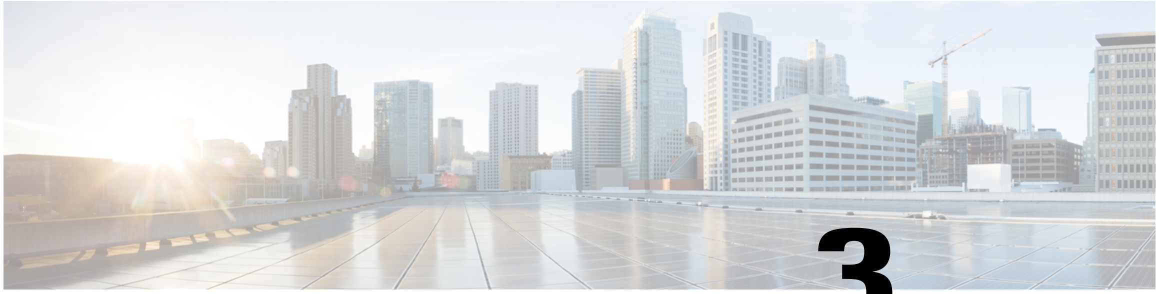
Simplifying the Interface Configuration with Port Profiles

You can use a port profile to simplify the interface configuration. You can configure a port profile and then assign it to multiple interfaces to give them all the same configuration. Changes to the port profile are propagated to the configuration of any interface that is assigned to it.

Note We do not recommend that you override port profile configurations by making changes to the assigned interface configurations. You should make configuration changes to interfaces only to quickly test a change or to disable a port.

High Availability for Interfaces

Interfaces support stateful and stateless restarts. A stateful restart occurs during a supervisor switchover. After the switchover, the Cisco Nexus 1000V applies the run-time configuration.



CHAPTER 3

Configuring Interface Parameters

This chapter contains the following sections:

Information About the Basic Interface Parameters

Description

For the vEthernet, Ethernet, and management interfaces, you can configure the description parameter to provide a name for the interface. Using a unique name for each interface allows you to quickly identify the interface when you are looking at a listing of multiple interfaces.

By default, the description for vEthernet interfaces is automatically formatted to contain information about the connected device. The description for a virtual Network Interface Card (vNIC), for example, contains the VM name and network adapter number. You keep this default description or you can override it with a description of your choosing.

Speed and Duplex Modes

The speed and duplex modes are interrelated for each Ethernet and management interface. By default, each interface autonegotiates its speed and duplex modes with the other interface, but you can change these settings. If you change the settings, be sure to use the same speed and duplex mode settings on both interfaces or use autonegotiation for at least one of the interfaces.

Port MTU Size

The maximum transmission unit (MTU) size specifies the maximum frame size that an Ethernet port can process. For transmissions to occur between two ports, you must configure the same MTU size for both ports. A port drops any frames that exceed its MTU size.

By default, the MTU size for each port is 1500 bytes, which is the IEEE 802.3 standard for Ethernet frames. Larger MTU sizes are possible for more efficient processing of data with less overhead. The larger frames, called jumbo frames, can be up to 9000 bytes in size, which is also the fixed system jumbo MTU size in the Cisco Nexus 1000V.

For a Layer 2 port, you can configure an MTU size as the system default of 1500 bytes or the system default jumbo MTU size of 9000 bytes.

Administrative Status

The administrative-status parameter determines whether an interface is up or down. When an interface is administratively down, it is disabled and unable to transmit data. When an interface is administratively up, it is enabled and able to transmit data.

Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is a Layer 2 protocol that enables two devices that run CDP to learn about each other. You can use CDP to troubleshoot the network by displaying information about the neighboring devices that are linked through each interface. By default, CDP is enabled.

Port Channel

A port channel is an aggregation of multiple physical interfaces that comprise a logical interface. You can bundle up to eight individual interfaces into a port channel to provide increased bandwidth and redundancy. Port channeling also load balances traffic across these physical interfaces. The port channel stays operational if at least one physical interface within the port channel is operational.

Any configuration changes that you apply to the port channel are applied to each interface member of that port channel.

Guidelines and Limitations

Interface parameters have the following configuration guidelines and limitations:

- You usually configure Ethernet port speed and duplex mode parameters to auto to allow negotiation of the speed and duplex modes between ports. If you decide to configure the port speed and duplex modes manually for these ports, consider the following:
    - If you set the Ethernet port speed to auto, the device automatically sets the duplex mode to auto.
    - If you enter the **no speed** command, the device automatically sets both the speed and duplex parameters to auto (the **no speed** command produces the same results as the **speed auto** command).

- ◦ If you configure an Ethernet port speed to a value other than auto (for example, 10, 100, or 1000 Mbps), you must configure the connecting port to match. Do not configure the connecting port to negotiate the speed.

Note The device cannot automatically negotiate the Ethernet port speed and duplex modes if the connecting port is configured to a value other than auto.

Note Changing the Ethernet port speed and duplex mode configuration might shut down and reenable the interface.

- To specify an interface in the CLI, use the following guidelines:
  - ◦ For an Ethernet port, use the **ethernet** *slot/port* command, where *slot* is the module slot number and *port* is the port number.
  - ◦ For a vEthernet port, use the **vethernet** *number* command, where *number* is a number from 1 to 1048575.
  - ◦ A space is not required between the interface type and the slot/port or interface number. For example, for the Ethernet slot 4, port 5 interface, you can specify either the **ethernet 4/5** command or the **ethernet4/5** command.
- Jumbo frames are only supported on the vmxnet3 driver. Attempts to change the MTU appear to succeed but the adapter always drops frames larger than 1500 bytes. For more information, see the VMware KB article http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1015556.

Configuring the Basic Interface Parameters

Specifying an Interface to Configure

You can specify an interface to configure.

Before You Begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface*	Enters interface configuration mode for the specified interface.

	Command or Action	Purpose
Step 3	switch(config-if)# **show interface** *interface*	(Optional) Displays the current configuration of interfaces. The interface argument is defined as follows: • For an Ethernet port, use the **ethernet** *slot/port* command, where *slot* is the module slot number and *port* is the port number. • For the management interface, use the **mgmt 0** or **mgmt0** command. • For a vEthernet port, use the **vethernet** *number* command, where *number* is a number from 1 to 1048575.

This example shows how to specify an interface to configure:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# show interface ethernet 3/1
switch(config-if)#
```

Configuring a Description

You can add a description to an Ethernet, vEthernet, or management interface.

Before You Begin

- Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **description** *string*	Adds a description of up to 80 alphanumeric characters for the interface and saves it in the running configuration.
Step 4	switch(config-if)# **show interface** *interface*	(Optional) Displays the interface status, which includes the description.

	Command or Action	Purpose
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to set the interface description to Ethernet port 24 on module 3:

```
switch# configure terminal
switch(config)# interface ethernet 3/24
switch(config-if)# description Ethernet port 24 on module 3
switch(config-if)#
```

Configuring the Interface Speed and Duplex Modes

You can configure the interface speed and duplex modes.

Before You Begin

- Log in to the CLI in EXEC mode.
- The interface speed and duplex modes are interrelated, so you should configure both at the same time. To see the speeds and duplex modes that you can configure together for Ethernet and management interfaces, see Speed and Duplex Modes, on page 11.

Note The interface speed that you specify can affect the duplex mode used for an interface, so you should set the speed before setting the duplex mode. If you set the speed for autonegotiation, the duplex mode is automatically set to be autonegotiated. If you specify a speed of 10 Mbps or 100 Mbps, the port is automatically configured to use half-duplex mode, but you can specify full-duplex mode instead. If you specify a speed of 1000 Mbps (1 Gbps) or faster, full duplex is automatically used.

- Make sure that the remote port has a speed setting that supports your changes for the local port. If you want to set the local port to use a specific speed, you must set the remote port for the same speed or set the local port to automatically negotiate the speed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **speed** {{**10** | **100** | **1000** | {**auto** [**10 100** [**1000**]]}} | {**10000** | **auto**}}	Designates the port speed as follows:

	Command or Action	Purpose
		• For Ethernet ports on the 48-port, 10/100/1000 modules, sets the speed at 10 Mbps, 100 Mbps, or 1000 Mbps, or sets the port to automatically negotiate its speed with the other 10/100/1000 port on the same link. • For Ethernet ports on the 32-port, 10-Gigabit Ethernet modules, sets the speed at 10,000 Mbps (10 Gbps) or sets the port to automatically negotiate its speed with the other 10-Gigabit Ethernet port on the link. • For management interfaces, sets the speed as 1000 Mbps or sets the port to autonegotiate its speed.
Step 4	switch(config-if)# **duplex** {**full** | **half** | **auto**}	Specifies the duplex mode as full, half, or autonegotiate.
Step 5	switch(config-if)# **show interface** *interface*	(Optional) (Optional) Displays the interface configuration.
Step 6	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to set the port speed and duplex mode to Ethernet port 1 on module 3:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# speed 1000
switch(config-if)# duplex full
switch(config-if)# show interface ethernet 3/1
switch(config-if)#
```

Configuring the MTU Size for an Ethernet Interface

You can configure the size of the maximum transmission unit (MTU) for a Layer 2 Ethernet interface.

Before You Begin

- Log in to the CLI in EXEC mode.
- Specify an MTU size between 1500 and 9000 bytes for an Ethernet interface.
- Make sure that the MTU value you set is supported by the VEM physical NIC. See your VMware documentation for more information about supported MTU for physical NICs.
- Know that Jumbo frames are supported only on the vmxnet3 driver. Attempts to change the MTU appear to succeed but the adapter always drops frames larger than 1500 bytes. For more information, see the VMware KB article http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1015556

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an Ethernet interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **mtu** *size*	Specifies an MTU size between 1500 (the default) and 9000 bytes.
Step 4	switch(config-if)# **show interface ethernet** *slot/port*	(Optional) Displays the interface status, which includes the MTU size.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure the Ethernet interface 3/1 with the default MTU size of 1500 bytes:

```
switch# configure terminal
switch# interface ethernet 3/1
switch(config-if)# mtu 1500
switch(config-if)#
```

Shutting Down and Activating an Interface

You can shut down and restart Ethernet or management interfaces.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that when you shut down an interface, it becomes disabled and the output of monitoring commands show it as being down.
- Know that when you activate an interface that has been shut down, you must restart the device.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface*	Specifies an Ethernet interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **shutdown**	Disables the interface in the running configuration .

	Command or Action	Purpose
Step 4	switch(config-if)# **show interface** *interface*	(Optional) Displays the interface status, which includes the administrative status.
Step 5	switch(config-if)# **no shutdown**	Reenables the interface in the running configuration .
Step 6	switch(config-if)# **show interface** *interface*	(Optional) Displays the interface status, which includes the administrative status. The interface argument are defined as follows: • For an Ethernet port, use the **ethernet** *slot/port* command, where *slot* is the module slot number and *port* is the port number. • For the management interface, use the **mgmt 0** or **mgmt0** command.
Step 7	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to disable the interface Ethernet port 1 on module 3:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# shutdown
switch(config-if)# no shutdown
switch(config-if)#
```

Enabling or Disabling CDP

You can enable or disable the Cisco Discovery Protocol (CDP) for Ethernet and management interfaces.

Before You Begin

- Log in to the CLI in EXEC mode.
- Make sure that CDP is enabled at both ends of the link.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface*	Specifies the interface that you are configuring. The *interface* argument is defined as follows:

	Command or Action	Purpose
		• For an Ethernet port, use the **ethernet** *slot/port* command, where *slot* is the module slot number and *port* is the port number. • For the management interface, use the **mgmt 0** or **mgmt0** command.
Step 3	switch(config-if)# [**no**] **cdp enable**	Enables or disables CDP for the interface in the running configuration. This parameter must be enabled for both interfaces on the same link. As soon as you disable CDP for one of two interfaces, CDP is disabled for the link.
Step 4	switch(config-if)# **show interface** *interface*	(Optional) Displays the CDP status for the interface in the running configuration. The *interface* argument is defined as follows: • For an Ethernet port, use the **ethernet** *slot/port* command, where *slot* is the module slot number and *port* is the port number. • For the management interface, use the **mgmt 0** or **mgmt0** command.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable CDP for Ethernet port 3/1:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# cdp enable
switch(config-if)#
```

This example shows how to disable CDP for Ethernet port 3/1:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# no cdp enable
switch(config-if)#
```

Clearing the Interface Counters

You can clear the Ethernet, vEthernet, and management interface counters.

Before You Begin

Log in to the CLI in EXEC mode, configuration mode, or interface configuration mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **clear counters** *interface*	Clears the counters for the specified interface: • **ethernet** *slot/port* • **vethernet** *number* • **mgmt 0** or **mgmt0**
Step 2	switch# **show interface** *interface*	(Optional) Displays the interface status, which includes the counters, for the specified interface: • **ethernet** *slot/port* • **vethernet** *number* • **mgmt 0** or **mgmt0**

This example shows how to clear and reset the counters on Ethernet port 5/5:

```
switch# clear counters ethernet 5/5
switch#
```

Verifying the Basic Interface Parameters

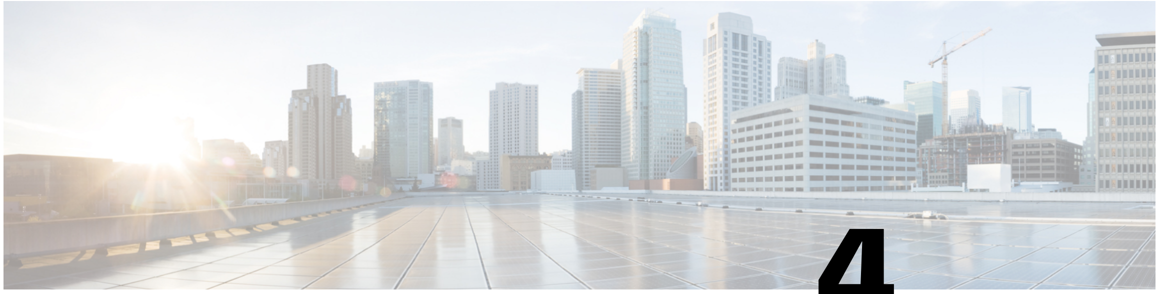
Use one of the following commands to verify the configuration:

Command	Purpose
show cdp	Displays the CDP status.
show interface *interface*	Displays the configured states of one or all interfaces.
show interface brief	Displays a table of interface states.
show interface switchport	Displays the status of Layer 2 ports.

Feature History for Basic Interface Parameters

Feature Name	Releases	Feature Information
System jumbo MTU	4.2(1)SV1(4)	The system jumbo MTU is fixed at 9000 and cannot be changed.

Feature Name	Releases	Feature Information
Interface MTU	4.2(1)SV1(4)	The interface MTU can be configured as a value between 1500 and 9000.
Basic interface parameters	4.0(4)SV1(1)	This feature was introduced.



CHAPTER 4

Configuring Layer 2 Interfaces

This chapter contains the following sections:

Information About Access and Trunk Interfaces

This section describes how to configure Layer 2 switching ports as access or trunk ports.

Note For information about configuring a Switched Port Analyzer (SPAN) destination interface, see the *Cisco Nexus 1000V System Management Configuration Guide*.

Note For information about VLANs, MAC address tables, and private VLANs, see the *Cisco Nexus 1000V Layer 2 Switching Configuration Guide*.

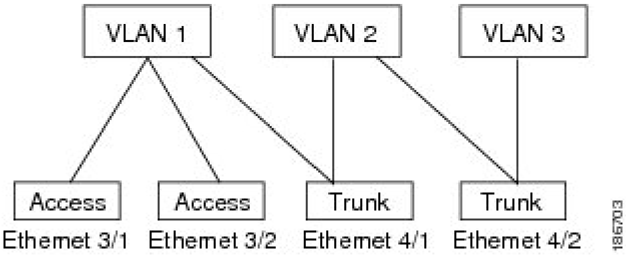
Access and Trunk Interfaces

A Layer 2 port can be configured as an access or a trunk port as follows:

- An access port can have only one VLAN configured on that port; it can carry traffic for only one VLAN.
- A trunk port can have two or more VLANs configured on that port; it can carry traffic for several VLANs simultaneously.

By default, all ports on the Cisco Nexus 1000V are Layer 2 ports. You can change the default port mode (access or trunk). See the *Cisco Nexus 1000V Installation and Upgrade Guide* for information about setting the default port mode. The following figure shows how you can use trunk ports in the network. The trunk port carries traffic for two or more VLANs.

Figure 1: Trunk and Access Ports and VLAN Traffic



In order to correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation, or tagging, method.

To optimize the performance on access ports, you can configure the port as a host port. Once the port is configured as a host port, it is automatically set as an access port and channel grouping is disabled. Use the host designation to decrease the time that it takes the designated port to begin to forward packets.

If an access port receives a packet with an 802.1Q tag in the header other than the access VLAN value, that port drops the packet without learning its MAC source address.

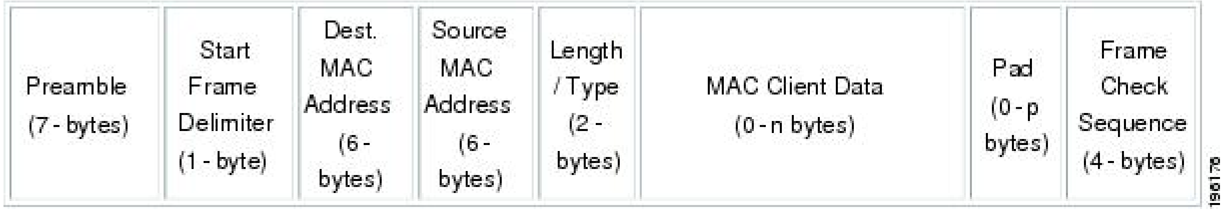
A Layer 2 interface can function as either an access port or a trunk port; it cannot function as both port types simultaneously.

IEEE 802.1Q Encapsulation

A trunk is a point-to-point link between the switch and another networking device. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network.

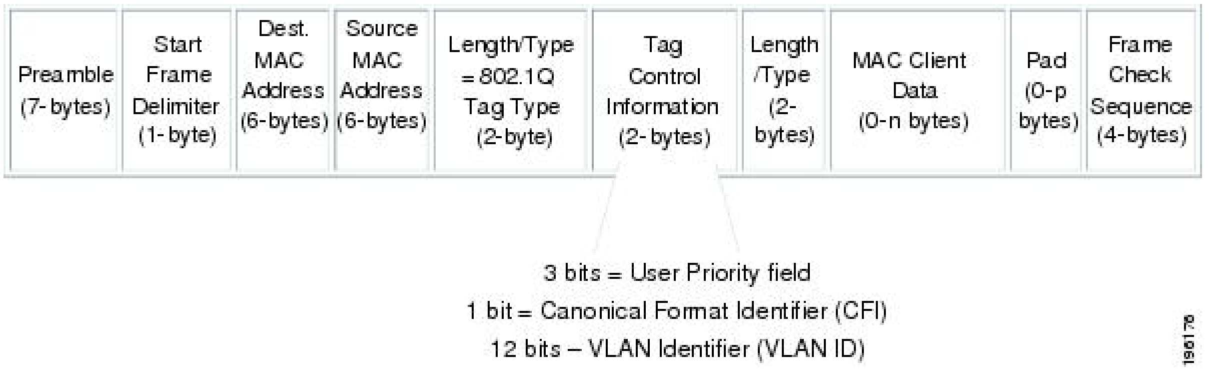
To correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation, or tagging, method that uses a tag that is inserted into the frame header (see the following figures). This tag carries information about the specific VLAN to which the frame and packet belong. This method allows packets that are encapsulated for several different VLANs to traverse the same port and maintain traffic separation between the VLANs. Also, the encapsulated VLAN tag allows the trunk to move traffic end to end through the network on the same VLAN.

Figure 2: Header Without 802.1Q Tag



Preamble (7 - bytes)	Start Frame Delimiter (1 - byte)	Dest. MAC Address (6 - bytes)	Source MAC Address (6 - bytes)	Length / Type (2 - bytes)	MAC Client Data (0 - n bytes)	Pad (0 - p bytes)	Frame Check Sequence (4 - bytes)

Figure 3: Header With 802.1Q Tag



Preamble (7- bytes)	Start Frame Delimiter (1- byte)	Dest. MAC Address (6-bytes)	Source MAC Address (6-bytes)	Length/Type = 802.1Q Tag Type (2-byte)	Tag Control Information (2- bytes)	Length /Type (2- bytes)	MAC Client Data (0-n bytes)	Pad (0-p bytes)	Frame Check Sequence (4-bytes)

3 bits = User Priority field
1 bit = Canonical Format Identifier (CFI)
12 bits – VLAN Identifier (VLAN ID)

High Availability

The software supports high availability for Layer 2 ports.

Prerequisites for VLAN Trunking

VLAN trunking has the following prerequisite:

You are logged into the CLI.

Guidelines and Limitations

VLAN trunking has the following configuration guidelines and limitations:

- Do not connect devices with access links because access links may partition a VLAN.
- When connecting Cisco switches through an 802.1Q trunk, make sure that the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If the native VLAN on one end of the trunk is different from the native VLAN on the other end, spanning tree loops might result.
- You can group trunk ports into port channel groups, but all trunks in the group must have the same configuration. When a group is first created, all ports follow the parameters set for the first port to be

added to the group. If you change the configuration of one of these parameters, the device propagates that setting to all ports in the group, such as the allowed VLANs and the trunk status. For example, if one port in a port group ceases to be a trunk, all ports cease to be trunks.

- If you try to enable 802.1X on a trunk port, an error message appears, and 802.1X is not enabled.
- If you try to change the mode of an 802.1X-enabled port to trunk, the port mode is not changed.

Default Settings

Table 2: Default Settings for Access and Trunk Interfaces

Parameters	Default
Switchport mode	Access
Allowed VLANs	1 to 3967, 4048 to 4094
Access VLAN ID	VLAN1
Native VLAN ID	VLAN1
Native VLAN ID tagging	Disabled
Administrative state	Shut

Configuring Access and Trunk Interfaces

Configuring a LAN Interface as a Layer 2 Access Port

You can configure a Layer 2 port as an access port.

Before You Begin

- Know that the interface can be either Ethernet or vEthernet.
- Know that an access port transmits packets on only one, untagged VLAN. You specify which VLAN traffic that the interface carries, which becomes the access VLAN. If you do not specify a VLAN for an access port, that interface carries traffic only on the default VLAN. The default VLAN is VLAN1.
- Know that the VLAN must exist before you can specify that VLAN as an access VLAN. The system shuts down an access port that is assigned to an access VLAN that does not exist.
- Know that the Cisco Nexus 1000V commands may differ from the Cisco IOS commands.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface*	Specifies the interface that you are configuring and places you in interface configuration mode. The interface argument are defined as follows: • For an Ethernet port, use the **ethernet** *slot/port* command, where *slot* is the module slot number and *port* is the port number. • For a vEthernet port, use the **vethernet** *interface-number* command, where *interface-number* is a number from 1 to 1048575.
Step 3	switch(config-if)# **switchport mode access**	Sets the interface as a nontrunking nontagged, single-VLAN Layer 2 interface in the running configuration.
Step 4	switch(config-if)# **switchport mode access** *vlan-id*	(Optional) Specifies the VLAN for which this access port will carry traffic and saves the change in the running configuration. If you do not enter this command, the access port carries traffic on VLAN1 only for traffic.
Step 5	switch(config-if)# **show interface** *interface*	(Optional) Displays the interface status and information.
Step 6	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to set Ethernet 3/1 as a Layer 2 access port that carries traffic for VLAN 5 only:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 5
switch(config-if)#
```

Configuring Trunk Ports

You can configure a Layer 2 port as a trunk port.

Before You Begin

- Before you configure a trunk port, ensure that you are configuring a Layer 2 interface.
- Know that the interface can be either Ethernet or vEthernet.

- Know that a trunk port transmits untagged packets for one VLAN plus encapsulated, tagged, packets for multiple VLANs. See IEEE 802.1Q Encapsulation, on page 24 for more information.
- The device supports 802.1Q encapsulation only.
- Know that the Cisco Nexus 1000V commands may differ from the Cisco IOS commands.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface*	Specifies the interface that you are configuring and places you in interface configuration mode. The interface argument is defined as follows: • For an Ethernet port, use the **ethernet** *slot/port* command, where *slot* is the module slot number and *port* is the port number. • For a vEthernet port, use the **vethernet** *interface-number* command, where *interface-number* is a number from 1 to 1048575.
Step 3	switch(config-if)# **switchport mode trunk**	Sets the interface as a Layer 2 trunk port in the running configuration. A trunk port can carry traffic in one or more VLANs on the same physical link (VLANs are based on the trunk-allowed VLANs list). By default, a trunk interface can carry traffic for all VLANs. To specify that only certain VLANs are allowed on the specified trunk, use the **switchport trunk allowed vlan** command.
Step 4	switch(config-if)# **show interface** *interface*	(Optional) Displays the interface status and information.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to set Ethernet 3/1 as a Layer 2 trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport mode trunk
switch(config-if)#
```

Configuring the Native VLAN for 802.1Q Trunking Ports

You can configure the native VLAN for 802.1Q trunk ports. If you do not configure this parameter, the trunk port uses the default VLAN as the native VLAN ID.

Before You Begin

Be aware that the Cisco Nexus 1000V commands may differ from the Cisco IOS commands.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface*	Specifies the interface that you are configuring and places you in interface configuration mode. The interface argument is defined as follows: • For an Ethernet port, use the **ethernet** *slot/port* command, where *slot* is the module slot number and *port* is the port number. • For a vEthernet port, use the **vethernet** *interface-number* command, where *interface-number* is a number from 1 to 1048575.
Step 3	switch#(config-if) **switchport trunk native vlan***vlan-id*	Designates the native VLAN for the 802.1Q trunk in the running configuration. Valid values are from 1 to 4094, except those VLANs reserved for internal use. The default value is VLAN1.
Step 4	switch#(config-if) **show vlan**	(Optional) Displays the status and information of VLANs.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to set the native VLAN for the Ethernet 3/1, Layer 2 trunk port to VLAN 5:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport trunk native vlan 5
switch(config-if)#
```

Configuring the Allowed VLANs for Trunking Ports

You can specify the IDs for the VLANs that are allowed on the specific trunk port.

Before You Begin

- Before you configure the allowed VLANs for the specified trunk ports, ensure that you are configuring the correct interfaces and that the interfaces are trunks.
- Know that the Cisco Nexus 1000V commands may differ from the Cisco IOS commands.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface*	Specifies the interface that you are configuring and places you in interface configuration mode. The interface argument is defined as follows: • For an Ethernet port, use the **ethernet** *slot/port* command, where *slot* is the module slot number and *port* is the port number. • For a vEthernet port, use the **vethernet** *interface-number* command, where *interface-number* is a number from 1 to 1048575.
Step 3	switch(config-if)# **switchport trunk allowed vlan** {*vlan-list* **all** | **none** [**add** |**except** | **none** | **remove** {*vlan-list*}]}	Sets the allowed VLANs for the trunk interface in the running configuration. The default is to allow all VLANs on the trunk interface. The range is from 1 to 3967 and 4048 to 4094. VLANs 3968 to 4047 are the default VLANs reserved for internal use by default; this group of VLANs is configurable. By default, all VLANs are allowed on all trunk interfaces. **Note** You cannot add internally allocated VLANs as allowed VLANs on trunk ports. The system returns a message if you attempt to list an internally allocated VLAN as an allowed VLAN.
Step 4	switch(config-if)# **show vlan**	(Optional) Displays the status and information of VLANs.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to add VLANs 15 to 20 to the list of allowed VLANs on the Ethernet 3/1, Layer 2 trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# switchport trunk allowed vlan 15-20
switch(config-if)#
```

Tagging Native VLAN Traffic

When working with 802.1Q trunked interfaces, you can maintain the tagging for all packets that enter with a tag that matches the native VLAN ID. Untagged traffic is dropped (you will still carry control traffic on that interface).

Before You Begin

- Know that the **vlan dot1q tag native** global command changes the behavior of all native VLAN ID interfaces on all trunks on the device.
- Know that this feature applies to the entire device; you cannot apply it to selected VLANs on a device.
- Know that the Cisco Nexus 1000V commands may differ from the Cisco IOS commands.

Note If you enable 802.1Q tagging on one device and disable it on another device, all traffic is dropped on the device with this feature disabled. You must configure this feature identically on each device.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch#(config) **vlan dot1q tag native**	Modifies the behavior of a 802.1Q trunked native VLAN ID interface in the running configuration. The interface maintains the taggings for all packets that enter with a tag that matches the value of the native VLAN ID and drops all untagged traffic. The control traffic is still carried on the native VLAN. The default is disabled.
Step 3	switch(config-if)# **show vlan**	(Optional) Displays the status and information of VLANs.
Step 4	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to change the behavior of the native VLAN on an 802.1Q trunked interface to maintain the tagged packets and drop all untagged traffic (except control traffic):

```
switch# configure terminal
switch(config)# vlan dot1q tag native
switch(config-if)#
```

Verifying the Interface Configuration

Use one of the following commands to verify the access and trunk interface configuration information:

Command	Purpose
show interface ethernet *slot/port* [**brief** | **capabilities** | **counters** | **mac-address** | **status** | **switchport** | **trunk**]	Displays the interface configuration.

Command	Purpose
show interface ethernet *slot/port* **counters** [**brief** | **detailed** | **errors** | **snmp** | **storm-control** | **trunk**]	Displays the counters for a specified Ethernet interface.
show interface ethernet *slot/port* **status** [**err-disable**]	Displays the status for a specified Ethernet interface.
show interface brief	Displays interface configuration information, including the mode.
show interface switchport	Displays information about the access and trunk interface for all Layer 2 interfaces.
show interface trunk [**module** *module-number* | **vlan** *vlan-id*]	Displays trunk configuration information.
show interface capabilities	Displays information about the capabilities of the interfaces.
show running-config interface ethernet *slot/port*	Displays configuration information about the specified interface.

Monitoring the Interface Configuration

Use one of the following commands to display access and trunk interface configuration information:

Command	Purpose
clear counters [*interface*]	Clears the counters.
show interface counters [**module** *module*]	Displays input and output octets unicast packets, multicast packets, and broadcast packets.
show interface counters detailed [**all**]	Displays input packets, bytes, and multicast as well as output packets and bytes.
show interface counters errors [**module** *module*]	Displays information on the number of error packets.

Configuration Examples for Access and Trunk Port Mode

This example shows how to configure a Layer 2 access interface and assign the access VLAN for that interface:

```
switch# configure terminal
switch(config)# interface ethernet 2/30
switch(config-if)# switchport
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 5
switch(config-if)#
```

This example shows how to configure a Layer 2 trunk interface, assign the native VLAN and the allowed VLANs, and configure the device to tag the native VLAN traffic on the trunk interface:

```
switch# configure terminal
switch(config)# interface ethernet 2/35
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk native vlan 10
switch(config-if)# switchport trunk allowed vlan 5, 10
switch(config-if)# exit
switch(config-if)# vlan dot1q tag native
switch(config-if)#
```

Feature History for Layer 2 Interface Parameters

Feature Name	Releases	Feature Information
Layer 2 interface parameters	4.0(4)SV1(1)	This feature was introduced.

CHAPTER 5

Configuring Virtual Ethernet Interfaces

This chapter contains the following sections:

Information About vEthernet Interfaces

Virtual Ethernet (vEthernet or vEth) interfaces are logical interfaces. Each vEthernet interface corresponds to a switch interface that is connected to a virtual port. The interface types are as follows:

- VM (interfaces connected to VM NICs)
- Service console
- vmkernel

vEthernet interfaces are created on the Cisco Nexus 1000V to represent virtual ports in use on the distributed virtual switch.

vEthernet interfaces are mapped to connected ports by MAC address as well as DVPort number. When a server administrator changes the port profile assignment on a vNIC or hypervisor port, the same vEthernet interface is reused.

When bringing up a vEthernet interface where a change in the port profile assignment is detected, the Virtual Supervisor Module (VSM) automatically purges any manual configuration present on the interface. You can use the following command to prevent purging of the manual configuration:

no svs veth auto-config-purge

Guidelines and Limitations

vEthernet interface configuration has the following configuration guideline and limitation:

MTU cannot be configured on a vEthernet interface.

Default Settings

Table 3: Default Settings for vEthernet Interface

Parameters	Default
Switchport mode	Access
Allowed VLANs	1 to 4094
Access VLAN ID	VLAN1
Native VLAN ID	VLAN1
Native VLAN ID tagging	Disabled
Administrative state	Shut
Automatic deletion of vEthernet interfaces	Enabled
Automatic purge of manual configuration on vEthernet interfaces	Enabled
Automatic creation of vEthernet interfaces	Enabled

Configuring vEthernet Interfaces

Configuring Global vEthernet Properties

You can enable or disable the following automatic controls for vEthernet interfaces:

- Deleting unused vEthernet interfaces
- Purging of manual vEthernet configurations
- Creating vEthernet interfaces

Before You Begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **svs veth auto-delete**	(Optional) Enables the VSM to automatically delete DVPorts no longer used by a vNIC or hypervisor port. The default setting is enabled. The **no** form of this command prevents the VSM from deleting unused DVPorts.
Step 3	switch(config)# [**no**] **svs veth auto-config-purge**	(Optional) Enables the VSM to remove all manual configuration on a vEthernet interface when the system administrator changes a port profile on the interface. The default setting is enabled. The **no** form of this command prevents the manual configuration from being deleted in this situation. **Note** Port profiles with ephemeral bindings are purged regardless of this setting.
Step 4	switch(config)# [**no**] **svs veth auto-setup**	(Optional) Enables the VSM to automatically create a vEthernet interface when a new port is activated on a host. The **no** form of this command disables the automatic creation of vEthernet interfaces in this situation. **Note** You can use **no** form of the command to temporary block automatic creation of vEthernet interfaces.
Step 5	switch(config)# **show running-config all | grep "svs-veth"**	(Optional) Displays the default global vEthernet settings that are in effect on the VSM for verification. If a setting is disabled, it does not display in the **show** command output.
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure global vEthernet properties:

```
switch# configure terminal
switch(config)# svs veth auto-delete
switch(config)# svs veth auto-config-purge
switch(config)# svs veth auto-setup
switch(config)# show running-config all | grep "svs veth"
svs veth auto-setup
svs veth auto-delete
svs veth auto-config-purge
switch(config-if)#
```

Configuring a vEthernet Access Interface

You can configure a vEthernet interface for use as an access interface.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that if you do not add a description to the vEthernet interface, one of the following descriptions is added at attach time. If you add a description and then remove it using the **no description** command, then one of the following descriptions is added to the interface:
    - For a VM—*VM-Name, Network Adapter number*
    - For a VMK—*VMware VMkernel, vmk number*
    - For a VSWIF—*VMware Service Console, vswif number*

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	(Optional) Enters the interface configuration mode for the specified vEthernet interface (from 1 to 1048575).
Step 3	switch(config-if)# **description** *string*	(Optional) Adds a description of up to 80 alphanumeric characters to the interface in the running configuration. **Note** If you do not add a description, the default description is added. **Note** You do not need to use quotations around descriptions that include spaces.
Step 4	switch(config-if)# **switchport access vlan** *vlanid*	Configures the vEthernet interface as an access interface and specifies the VLAN ID (1 to 4094) in the running configuration.
Step 5	switch(config-if)# **switchport mode access**	Configures the vEthernet interface for use as an access interface in the running configuration.
Step 6	switch(config-if)# **show interface vethernet** *interface-number*	(Optional) Displays the specified interface for verification.
Step 7	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a vEthernet access interface:

```
switch# configure terminal
switch(config)# interface vethernet 100
switch(config-if)# description accessvlan
switch(config-if)# switchport access vlan 5
switch(config-if)# switchport mode access
switch(config-if)# show interface vethernet1
switch(config-if)#
```

Configuring a Private VLAN on a vEthernet Interface

You can configure a private VLAN (PVLAN) on a vEthernet interface.

Before You Begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Enters the interface configuration mode for the specified vEthernet interface (from 1 to 1048575).
Step 3	switch(config-if)# **description** *string*	(Optional) Adds a description of up to 80 alphanumeric characters to the interface in the running configuration. **Note** If you do not add a description, the default description is added. **Note** You do not need to use quotations around descriptions that include spaces.
Step 4	switch(config-if)# **switchport access vlan** *vlanid*	Configures the vEthernet interface as an access interface and specifies the VLAN ID (1 to 4094) in the running configuration.
Step 5	switch(config-if)# **switchport mode private-vlan host**	Configures the vEthernet interface for a PVLAN host in the running configuration.
Step 6	switch(config-if)# **switchport private-vlan host-association** *primary-vlanid*	Configures the vEthernet interface for a host association with a specific primary VLAN ID (from 1 to 4094) in the running configuration.
Step 7	switch(config-if)# **show interface**	(Optional) Displays the interface status and information.
Step 8	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a vEthernet interface to use in a PVLAN:

```
switch# configure terminal
switch(config)# interface vethernet 1
switch(config-if)# switchport access vlan 5
switch(config-if)# switchport mode private-vlan host
switch(config-if)# switchport private-vlan host-association 5
switch(config-if)# show interface vethernet 1
Vethernet1 is up
Port description is gentoo, Network Adapter 1
Hardware is Virtual, address is 0050.5687.3bac
Owner is VM "gentoo", adapter is Network Adapter 1
Active on module 4
VMware DVS port 1
Port-Profile is vm
Port mode is access
5 minute input rate 1 bytes/second, 0 packets/second
5 minute output rate 94 bytes/second, 1 packets/second
Rx
655 Input Packets 594 Unicast Packets
0 Multicast Packets 61 Broadcast Packets
114988 Bytes
Tx
98875 Output Packets 1759 Unicast Packets
80410 Multicast Packets 16706 Broadcast Packets 0 Flood Packets
6368452 Bytes
0 Input Packet Drops 0 Output Packet Drops
switch(config-if)#
```

Enabling or Disabling a vEthernet Interface

You can enable or disable a vEthernet interface.

Before You Begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Enters the interface configuration mode for the specified vEthernet interface (from 1 to 1048575).
Step 3	switch(config-if)# [**no**] **shutdown**	Enables or disables the vEthernet interface in the running configuration: • **shutdown**: Disables the vEthernet interface. • **no shutdown**: Enables the vEthernet interface.
Step 4	switch(config-if)# **show interface**	(Optional) Displays the interface status and information.

	Command or Action	Purpose
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable a vEthernet interface:

```
switch# configure terminal
switch(config)# interface vethernet 100
switch(config-if)# no shutdown
switch(config-if)# show interface veth100 status
--------------------------------------------------------------------------------
Port           Name               Status   Vlan      Duplex  Speed   Type
--------------------------------------------------------------------------------
Veth100        --                       up          1       auto        auto    --
switch(config-if)#
```

Verifying the vEthernet Interface Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show interface vethernet *interface-number* [**brief** | **counters** [**detailed** [**all**] | **errors**] | **description** | **mac-address** | **status** [**down** | **err-disabled** | **inactive** | **module** *num* | **up**] | **switchport**]	Displays the vEthernet interface configuration.
show interface [**vethernet** *interface-number*]	Displays the complete interface configuration.
show interface [**vethernet** *interface-number*] **brief**	Displays the abbreviated interface configuration.
show interface [**vethernet** *interface-number*] **description**	Displays the interface description.
show interface [**vethernet** *interface-number*] **mac-address**	Displays the interface MAC address. **Note** For vEth interfaces, this command shows the MAC address of the connected device.
show interface [**vethernet** *interface-number*] **status** [**down** | **err-disabled** | **inactive** | **module** *num* | **up**]	Displays the interface line status.
show interface [**vethernet** *interface-number*] **switchport**	Displays interface switchport information.
show interface virtual [**vm** [*vm_name*] | **vmk** | **vswif**] [**module** *mod_no*]	Displays virtual interfaces only.

Command	Purpose
show interface virtual port-mapping [**vm** [*name*] | **vmk** | **vswif** | **description**] [*module_num*]	Displays mappings between the virtual Ethernet and VMware DVPort.

Example: show interface vethernet

This example shows how to display vEthernet 1:

```
switch# show interface veth1
Vethernet1 is up
    Port description is gentoo1, Network Adapter 1
    Hardware is Virtual, address is 0050.56bd.42f6
    Owner is VM "gentoo1", adapter is Network Adapter 1
    Active on module 33
    VMware DVS port 100
    Port-Profile is vlan48
    Port mode is access
    Rx
    491242 Input Packets 491180 Unicast Packets
    7 Multicast Packets 55 Broadcast Packets
    29488527 Bytes
    Tx
    504958 Output Packets 491181 Unicast Packets
    1 Multicast Packets 13776 Broadcast Packets 941 Flood Packets
    714925076 Bytes
    11 Input Packet Drops 0 Output Packet Drops
switch#
```

Example: show interface virtual

This example shows how to display information for all vEthernet interfaces:

```
switch# show interface virtual
--------------------------------------------------------------------------------
Port        Adapter        Owner                    Mod Host
--------------------------------------------------------------------------------
Veth1                      Vm1-kl61                 2
Veth2                      VM1-kl65                 5
Veth3                      VM2-kl61                 2
Veth1       Net Adapter 1  austen-gentoo1          33  austen-strider.austen.
Veth2       Net Adapter 2  austen-gentoo1          33  austen-strider.austen.
switch#
```

Example: show interface virtual description

This example shows how to display the descriptions for all vEthernet interfaces:

```
switch# show interface virtual description
--------------------------------------------------------------------------------
Interface   Description
--------------------------------------------------------------------------------
Veth1       gentoo1, Network Adapter 1
Veth2       gentoo1, Network Adapter 2
Veth3       VMware VMkernel, vmk1
Veth4       VMware Service Console, vswif1

switch#
```

Example: show interface virtual port-mapping

This example shows how to display the virtual port mapping for all vEthernet interfaces:

```
switch# show interface virtual port-mapping
-------------------------------------------------------------------------------
Port    Hypervisor Port    Binding Type    Status    Reason
```

```
-------------------------------------------------------------------------------
Veth1   DVPort5747         static          up        none
Veth2   DVPort3361         static          up        none
switch#
```

Example: show running-config interface veth1

This example shows how to display the running configuration information for all vEthernet interfaces:

```
switch# show running-config interface veth1
version 4.2(1)SV1(4)

interface Vethernet1
  inherit port-profile vlan48
  description gentoo1, Network Adapter 1
  vmware dvport 2968 dvswitch uuid "d4 02 20 50 16 4b 36 97-46 09 dc d8 5b c6 1e c1"
  vmware vm mac 0050.56A0.0000

switch#
```

Monitoring the vEthernet Interface Configuration

Use one of the following commands to monitor the vEthernet interface configuration:

Command	Purpose
show interface [**vethernet** *interface-number*] **counters**	Displays the interface incoming and outgoing counters.
show interface [**vethernet** *interface-number*] **counters detailed** [**all**]	Displays detailed information for all counters. **Note** If **all** is not specified, only nonzero counters are shown.
show interface [**vethernet** *interface-number*] **counters errors**	Displays the interface error counters.

This example shows how to display the counters for all vEthernet interfaces:

```
switch# show interface counters
--------------------------------------------------------------------------------
Port                                 InOctets                      InUcastPkts
--------------------------------------------------------------------------------
mgmt0                               224396010                           562676
Eth8/2                             1106222375                           703688
Veth1                                    2700                                0
control0                            155152164                           501981

--------------------------------------------------------------------------------
Port                              InMcastPkts                      InBcastPkts
--------------------------------------------------------------------------------
mgmt0                                  148552                           656058
Eth8/2                                  15051                          2093981
Veth1                                       0                               45
control0                                  999                           119539

--------------------------------------------------------------------------------
Port                                OutOctets                     OutUcastPkts
--------------------------------------------------------------------------------
mgmt0                                14803330                            56509
Eth8/2                              164990676                           514040
Veth1                               820595312                            17385
control0                               251243                                0
```

```
--------------------------------------------------------------------------------
Port                             OutMcastPkts                      OutBcastPkts
--------------------------------------------------------------------------------
mgmt0                                    1001                              1001
Eth8/2                                  11451                                70
Veth1                                     108                           1383772
control0                                 1001                                 0
```

Configuration Examples for vEthernet Interfaces

This example shows how to configure a vEthernet access interface and assign the access VLAN for that interface:

```
switch# configure terminal
switch(config)# interface vethernet 100
switch(config-if)# switchport
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 5
switch(config-if)#
```

This example shows how to configure a Layer 2 trunk interface, assign the native VLAN and the allowed VLANs, and configure the device to tag the native VLAN traffic on the trunk interface:

```
switch# configure terminal
switch(config)# interface vethernet 1
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk native vlan 10
switch(config-if)# switchport trunk allowed vlan 5, 10
switch(config-if)#
```

Feature History for vEthernet Interfaces

Feature Name	Releases	Feature Information
Global vEthernet interface controls	4.2(1)SV1(4)	You can enable or disable the following automatic vEthernet interface controls: • Deleting unused vEthernet interfaces • Purging of manual vEthernet configurations • Creating vEthernet interfaces
vEthernet interface parameters	4.0(4)SV1(1)	This feature was introduced

CHAPTER 6

Configuring Port Channels

This chapter contains the following sections:

Information About Port Channels

A port channel is an aggregation of multiple physical interfaces that creates a logical interface. You can bundle up to eight individual active links into a port channel to provide increased bandwidth and redundancy. Port channeling also load balances traffic across these physical interfaces. The port channel stays operational as long as at least one physical interface within the port channel is operational.

You can use static port channels, with no associated aggregation protocol, for a simplified configuration.

Port Channels

A port channel bundles physical links into a channel group to create a single logical link that provides the aggregate bandwidth of up to eight physical links. If a member port within a port channel fails, the traffic previously carried over the failed link switches to the remaining member ports within the port channel.

You can bundle up to eight ports into a static port channel without using any aggregation protocol.

Note The device does not support Port Aggregation Protocol (PAgP) for port channels.

Each port can be in only one port channel. All the ports in a port channel must be compatible; they must use the same speed and duplex mode. When you run static port channels with no aggregation protocol, the physical links are all in the on channel mode.

You can create port channels directly by creating the port channel interface, or you can create a channel group that acts to aggregate individual ports into a bundle. When you associate an interface with a channel group, the software creates a matching port channel automatically if the port channel does not already exist. In this instance, the port channel assumes the Layer 2 configuration of the first interface. You can also create the port channel first. In this instance, the Cisco Nexus 1000V creates an empty channel group with the same channel number as the port channel and takes the default Layer 2 configuration, as well as the compatibility configuration.

Note The port channel is operationally up when at least one of the member ports is up and is in the channeling state. The port channel is operationally down when all member ports are operationally down.

Compatibility Checks

When you add an interface to a port channel group, the following compatibility checks are made before allowing the interface to participate in the port channel:

- Network layer
- (Link) speed capability
- Speed configuration
- Duplex capability

- Duplex configuration
- Port mode
- Access VLAN
- Trunk native VLAN
- Tagged or untagged
- Allowed VLAN list
- MTU size
- SPAN—Cannot be a SPAN source or a destination port

To view the full list of compatibility checks performed by the Cisco Nexus 1000V, use the **show port-channel compatibility-parameters**.

You can only add interfaces configured with the channel mode set to on to static port channels. You can configure these attributes on an individual member port. If you configure a member port with an incompatible attribute, the Cisco Nexus 1000V suspends that port in the port channel.

When the interface joins a port channel, some of its individual parameters are removed and replaced with the values on the port channel as follows:

- Bandwidth
- Delay
- Extended Authentication Protocol over UDP
- VRF
- IP address (v4 and v6)
- MAC address
- Spanning Tree Protocol
- Network Access Control
- Service policy
- Quality of Service (QoS)
- Access control lists (ACLs)

The following interface parameters remain unaffected when the interface joins or leaves a port channel:

- Description
- CDP
- MDIX
- Rate mode
- Shutdown
- SNMP trap

Note When you delete the port channel, the software sets all member interfaces as if they were removed from the port channel.

Load Balancing Using Port Channels

The Cisco Nexus 1000V load balances traffic across all operational interfaces in a port channel by hashing the addresses in the frame to a numerical value that selects one of the links in the channel. Port channels provide load balancing by default. Port channel load balancing uses MAC addresses, IP addresses, or Layer 4 port numbers to select the link. Port channel load balancing uses either source or destination addresses or ports, or both source and destination addresses or ports.

You can configure the load balancing mode to apply to all port channels that are configured on the entire device or on specified modules. The per-module configuration takes precedence over the load-balancing configuration for the entire device. You can configure one load balancing mode for the entire device, a different mode for specified modules, and another mode for the other specified modules. You cannot configure the load balancing method per port channel.

You can configure the type of load balancing algorithm used. You can choose the load balancing algorithm that determines which member port to select for egress traffic by looking at the fields in the frame.

Note The default load balancing method uses source MAC addresses.

You can configure one of the following methods to load balance across the port channel:

- Destination MAC address
- Source MAC address
- Source and destination MAC addresses
- Destination IP address and VLAN
- Source IP address and VLAN
- Source and destination IP address and VLAN
- Destination TCP/UDP port number
- Source TCP/UDP port number
- Source and destination TCP/UDP port number
- Destination IP address and TCP/UDP port number
- Source IP address and TCP/UDP port number
- Source and destination IP address and TCP/UDP port number
- Destination IP address, TCP/UDP port number, and VLAN
- Source IP address, TCP/UDP port number, and VLAN
- Source and destination IP address, TCP/UDP port number, and VLAN

- Destination IP address
- Source IP address
- Source and destination IP addresses
- VLAN only
- Source virtual port ID

When you configure source MAC address load balancing, the source MAC address is used to balance the traffic load. When you configure the destination MAC address load-balancing method, the traffic load is balanced using the destination MAC address.

When you configure source IP address load balancing, the source IP address is used to balance the traffic load. When you configure the destination IP address load-balancing method, the traffic load is balanced using the destination IP address.

Note Starting from Release 5.2(1)SV3(1.1), IPv6 support is added for IP and TCP/UDP-based load balancing.

The load balancing methods that use port channels do not apply to multicast traffic. Regardless of the method configured, multicast traffic uses the following methods for load balancing with port channels:

- Multicast traffic with Layer 4 information—Source IP address, source port, destination IP address, and destination port
- Multicast traffic without Layer 4 information—Source IP address and destination IP address
- Non-IP multicast traffic—Source MAC address and destination MAC address

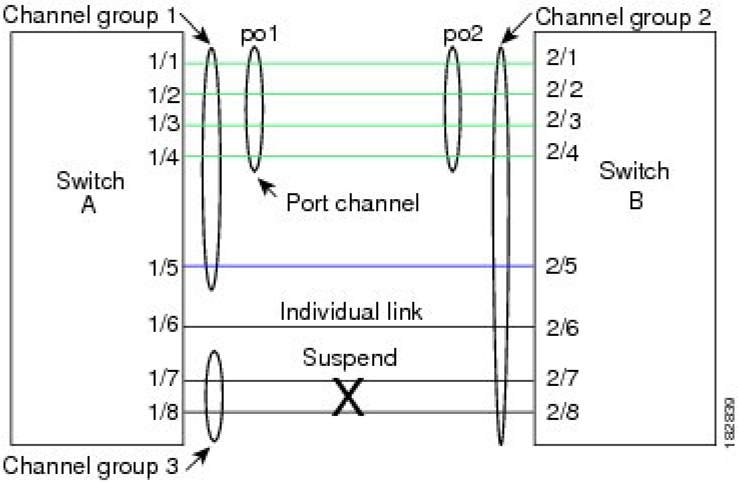
LACP

The Link Aggregation Control Protocol (LACP) allows you to configure up to 16 interfaces into a port channel. A maximum of eight interfaces can be active, and a maximum of eight interfaces can be placed in a standby state. The following figure shows how individual links can be combined into LACP port channels and channel groups as well as function as individual links.

Note

- When you delete the port channel, the associated channel group is automatically deleted. All member interfaces revert to their original configuration.
- LACP port channels on Cisco virtual interface cards do not support more than two vNICs.

Figure 4: Individual Links Combined into a Port Channel



VEM Management of LACP

You can offload operation of the LACP from the Virtual Supervisor Module (VSM) to the Virtual Ethernet Ports (VEMs) to prevent a situation where the VSM cannot negotiate LACP with the upstream switch when the VEM is disconnected from the VSM (referred to as headless mode). VEM management of LACP allows it to reestablish port channels after the reboot of a headless VEM.

Port Channel Modes

Individual interfaces in port channels are configured with channel modes. When you run static port channels with no aggregation protocol, the channel mode is always set to on.

You enable LACP for each channel by setting the channel mode for each interface to active or passive. You can configure either channel mode for individual links in the LACP channel group when you are adding the links to the channel group.

The following table describes the channel modes.

Table 4: Channel Modes for Individual Links in a Port Channel

Channel Mode	Description
passive	LACP mode that places a port into a passive negotiating state in which the port responds to LACP packets that it receives but does not initiate LACP negotiation.
active	LACP mode that places a port into an active negotiating state in which the port initiates negotiations with other ports by sending LACP packets.
on	All static port channels (that are not running LACP) remain in this mode. If you attempt to change the channel mode to active or passive before enabling LACP, the device displays an error message. You enable LACP on each channel by configuring the interface in that channel for the channel mode as either **active** or **passive**. When an LACP attempts to negotiate with an interface in the on state, it does not receive any LACP packets and becomes an individual link with that interface; it does not join the LACP channel group. The default port channel mode is **on**.

Both the passive and active modes allow LACP to negotiate between ports to determine if they can form a port channel based on criteria such as the port speed and the trunking state. The passive mode is useful when you do not know whether the remote system, or partner, supports LACP.

Ports can form an LACP port channel when they are in different LACP modes if the modes are compatible as in these examples:

- A port in **active** mode can form a port channel successfully with another port that is in **active** mode.
- A port in **active** mode can form a port channel with another port in **passive** mode.
- A port in **passive** mode cannot form a port channel with another port that is also in **passive** mode, because neither port will initiate negotiation.
- A port in **on** mode is not running LACP and cannot form a port channel with another port that is in **active** or **passive** mode.

LACP ID Parameters

This section describes the LACP parameters.

LACP System Priority

Each system that runs LACP has an LACP system priority value. It has a default value of 32768 and is not configurable. LACP uses the system priority with the MAC address to form the system ID and also uses the system priority during negotiation with other devices. A higher system priority value means a lower priority.

Note The LACP system ID is the combination of the LACP system priority value and the MAC address.

LACP Port Priority

Each port that is configured to use LACP has an LACP port priority. It has a default value of 32768 and is not configurable. LACP uses the port priority with the port number to form the port identifier.

LACP uses the port priority to decide which ports should be put in standby mode when there is a limitation that prevents all compatible ports from aggregating and which ports should be put into active mode. A higher port priority value means a lower priority for LACP. You can configure the port priority so that specified ports have a lower priority for LACP and are most likely to be chosen as active links, rather than as hot-standby links.

LACP Administrative Key

LACP automatically configures an administrative key value that isequal to the channel entry index (1 through 8) for each port on the VEM configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port’s ability to aggregate with other ports is determined by these factors:

- Port physical characteristics, such as the data rate and the duplex capability
- Configuration restrictions that you establish

LACP Marker Responders

You can dynamically redistribute the data traffic by using port channels. This redistribution may result from a removed or added link or a change in the load-balancing scheme. Traffic redistribution that occurs in the middle of a traffic flow can cause misordered frames.

LACP uses the Marker Protocol to ensure that frames are not duplicated or reordered due to this redistribution. The Marker Protocol detects when all the frames of a given traffic flow are successfully received at the remote end. LACP sends Marker PDUs on each of the port-channel links. The remote system responds to the Marker PDU once it receives all the frames received on this link prior to the Marker PDU. The remote system then sends a Marker Responder. Once the Marker Responders are received by the local system on all member links of the port channel, the local system can redistribute the frames in the traffic flow with no chance of misordering. The software supports only Marker Responders.

LACP-Enabled and Static Port Channels Differences

The following table summarizes the major differences between port channels with LACP enabled and static port channels.

Table 5: Port Channels with LACP Enabled and Static Port Channels

Configurations	Port Channels with LACP Enabled	Static Port Channels
Protocol applied	Enable globally	Not applicable
Channel mode of links	Can be either: • Active • Passive	Can only be On
Maximum number of links in channel	16	8

vPC Host Mode

You use vPC-HM mode to create a port channel when the switch is connected to multiple upstream switches that are not clustered. In the Cisco Nexus 1000V, the port channel is divided into subgroups or logical smaller port channels, each representing one or more uplinks to one upstream physical switch.

Links that connect to the same physical switch are bundled in the same subgroup automatically by using information gathered from the Cisco Discovery Protocol (CDP) packets from the upstream switch. Interfaces can also be manually assigned a specific subgroup.

When you use vPC-HM, each vEthernet interface on the VEM is mapped to one of two subgroups in a round-robin method. All traffic from the vEthernet interface uses the assigned subgroup unless it is unavailable, in which case the vEthernet interface fails over to the remaining subgroup. When the original subgroup becomes available again, traffic shifts back to it. Traffic from each vEthernet interface is then balanced based on the configured hashing algorithm.

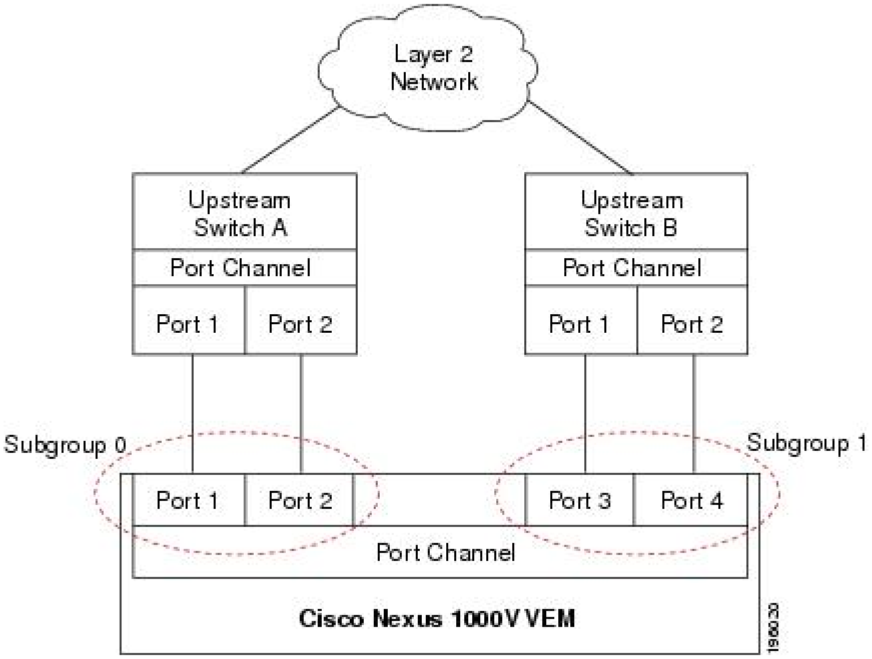
When multiple uplinks are attached to the same subgroup, you must configure the upstream switch in a port channel with the links bundled together. The port channel must also be configured with the **channel-group auto mode on** (active and passive modes use LACP).

If the upstream switches do not support port channels, you can use MAC pinning to assign each Ethernet port member to a particular port channel subgroup.

Note Do not configure vPC-HM on the Cisco Nexus 1000V when the upstream switch ports that connect to the VEMs have vPC configured. If vPC is configured, the connection can be interrupted or disabled.

The following figure shows how to use vPC-HM to assign member ports 1 and 2 to subgroup ID 0 and member ports 3 and 4 to subgroup ID 1.

Figure 5: Using vPC-HM to Connect a Port Channel to Multiple Upstream Switches



Subgroup Creation

If Cisco Discovery Protocol (CDP) is enabled on the upstream switches, subgroups are automatically created using information gathered from the CDP packets. If not, you must manually create subgroups.

Static Pinning

Static pinning allows you to pin the virtual ports behind a VEM to a particular subgroup within the channel. Instead of allowing round robin dynamic assignment between the subgroups, you can assign (or pin) a static vEthernet interface, control VLAN, or packet VLAN to a specific port channel subgroup. With static pinning, traffic is forwarded only through the member ports in the specified subgroup.

You can also pin vEthernet interfaces to subgroups in interface configuration mode.

MAC Pinning

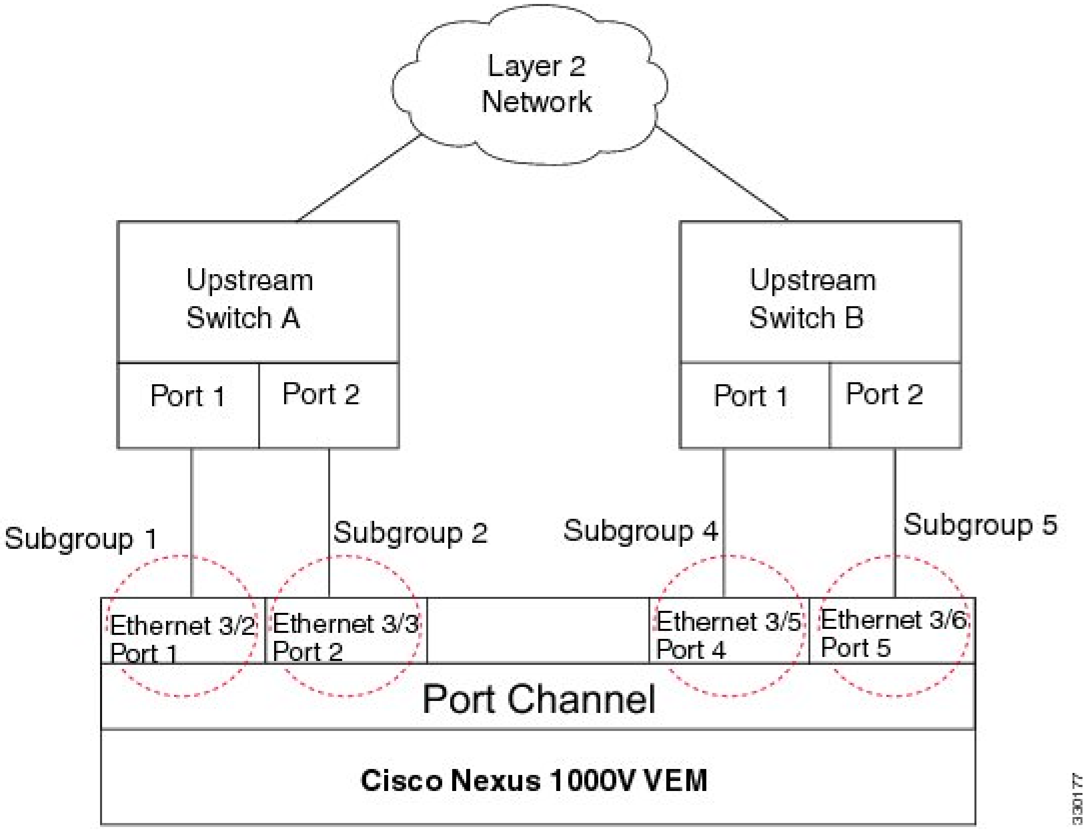
If you are connecting to multiple upstream switches that do not support port channels, MAC pinning is the preferred configuration. MAC pinning divides the uplinks from your server into standalone links and pins the MAC addresses to those links in a round-robin method to ensure that the MAC address of a virtual machine is never seen on multiple upstream switch interfaces. Therefore, no upstream configuration is required to connect the VEM to upstream switches.

MAC pinning does not rely on any protocol to distinguish upstream switches so the configuration is independent of upstream hardware or design.

In the case of a failure, the Cisco Nexus 1000V first sends a gratuitous ARP packet to the upstream switch indicating that the VEM MAC address will now be learned on a different link. It also allows for subsecond failover time.

The following figure shows each member port that is assigned to a specific port channel subgroup using MAC pinning.

Figure 6: Using MAC Pinning to Connect a Port Channel to Multiple Upstream Switches



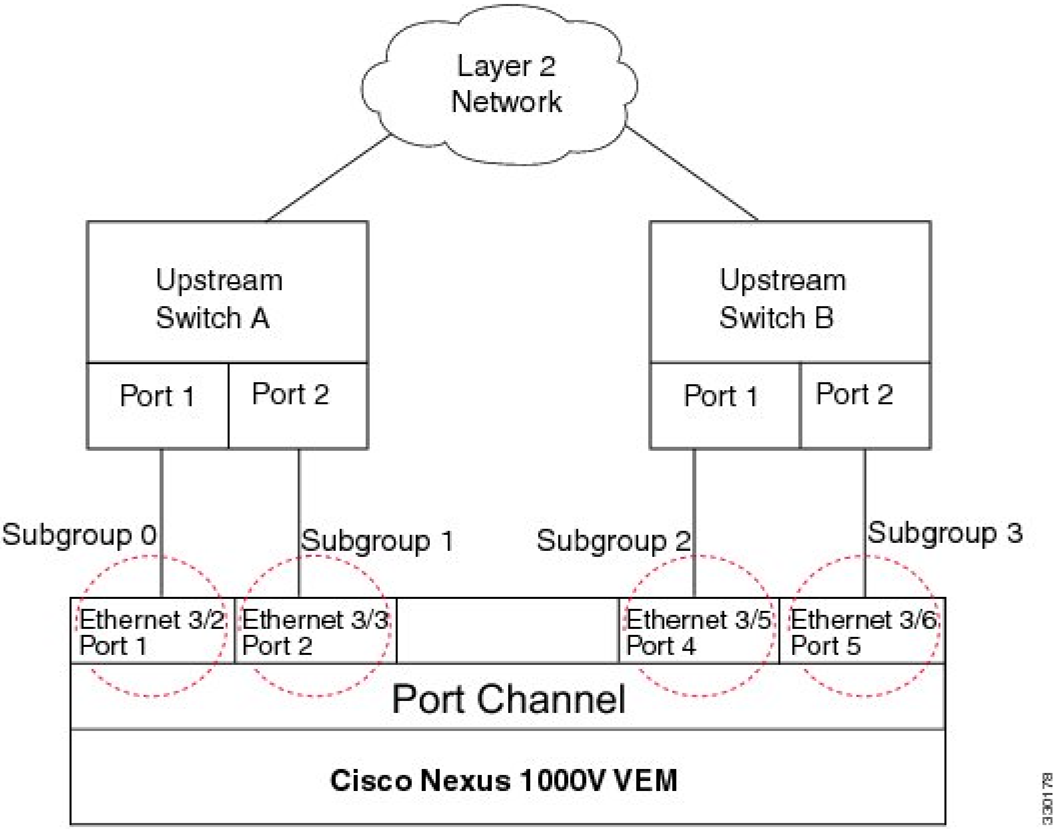
MAC Pinning Relative

This feature modifies the existing algorithm for MAC pinning where the port channel uses the port number (vmnic number) as the subgroup ID for an Ethernet member port.

The new algorithm assigns zero-based logical subgroup IDs to Ethernet member ports. The member port that has the lowest port number (vmnic number) is assigned subgroup ID 0.

The following figure shows each member port that is assigned to a specific port channel subgroup using MAC pinning relative.

Figure 7: Using MAC Pinning Relative to Connect a Port Channel to Multiple Upstream Switches



Network State Tracking for vPC-HM

Network state tracking for vPC-HM identifies link failures where other detection methods fail, and verifies Layer 2 connectivity between vPC-HM channel subgroups. It is not intended to detect network configuration problems.

Network state tracking selects one uplink interface in each sub group for broadcasting packets to a tracking VLAN. The tracking VLAN is usually the lowest forwarding VLAN for trunk ports and the primary VLAN for promiscuous access ports. The packets that are received back from the network on each subgroup are tracked as are the number of consecutively missed broadcasts. If the missed broadcasts for a subgroup exceed the threshold, the port channel is considered to be in split mode. In split mode, the interfaces are marked as inactive, and traffic is pinned to active interfaces.

System messages indicate when a port channel enters or recovers from split mode; and interfaces are marked active or inactive.

High Availability

Port channels provide high availability by load balancing traffic across multiple ports. If a physical port fails, the port channel is still operational if there is an active member in the port channel.

Port channels support stateful and stateless restarts. A stateful restart occurs on a supervisor switchover. After the switchover, the Cisco Nexus 1000V applies the runtime configuration after the switchover.

Prerequisites for Port Channels

Port channeling has the following prerequisites:

- You are logged into the Cisco Nexus 1000V in EXEC mode.
- All ports for a single port channel must meet the compatibility requirements. See Compatibility Checks, on page 46 for more information about the compatibility requirements.
- You can use virtual vPC-HM to configure a port channel even when the physical ports are connected to two different switches.

Guidelines and Limitations

Port channeling has the following guidelines and restrictions:

- All ports in the port channel must be in the same Cisco Nexus 1000V module; you cannot configure port channels across Cisco Nexus 1000V modules.
- Port channels can be formed with multiple upstream links only when they satisfy the compatibility requirements and under the following conditions:
    - The uplinks from the host are going to the same upstream switch.
    - The uplinks from the host going to multiple upstream switches are configured with vPC-HM.
- You can configure multiple port channels on a device.
- After you configure a port channel, the configuration that you apply to the port channel interface affects the port channel member ports. The configuration that you apply to the member ports affects only the member port where you apply the configuration.
- You must remove the port security information from a port before you can add that port to a port channel. You cannot apply the port security configuration to a port that is a member of a channel group.
- You can configure ports that belong to a port channel group as PVLAN ports.
- Any configuration changes that you apply to the port channel is applied to every member interface of that port channel.
- Channel member ports cannot be source or destination SPAN ports.
- To support LACP when inband/AIPC are also carried over the link, you must configure the following commands on the ports connected to the ESX host:

- ◦ **spanning-tree portfast trunk**
- ◦ **spanning-tree bpdufilter enable**

Note If you have a separate dedicated NIC for control traffic, these settings are not required.

- There should be at least two links that connect two switches when inband/AIPC are also carried over the LACP channel.
- If you configure LACP and your upstream switch uses the LACP suspend feature, make sure this feature is disabled. For more information, see the documentation for your upstream switch.
- If you are connecting to an upstream switch or switches that do not support port channels, MAC pinning is the preferred configuration. MAC pinning divides the uplinks from your server into standalone links and pins the MAC addresses to those links in a round-robin method. The drawback is that you cannot leverage the load sharing performance that LACP provides.
- Once a port profile is created, you cannot change its type (Ethernet or vEthernet).
- The server administrator should not assign more than one uplink on the same VLAN without port channels. The server administrator cannot assign more than one uplink on the same host to a profile without port channels or port profiles that share one or more VLANs.

Caution Disruption of connectivity might result if you configure vPC-HM on the Cisco Nexus 1000V when vPC is also configured on the ports of upstream switches that connect to its VEMs.

- You must have already configured the Cisco Nexus 1000Vsoftware using the setup routine. For information, see the *Cisco Nexus 1000V Installation and Upgrade Guide*.
- The Cisco Nexus 1000V must be connected to the vCenter Server.
- You are logged in to the CLI in EXEC mode.
- When you create a port channel, an associated channel group is automatically created.
- If the Link Aggregation Control Protocol (LACP) support is required for the port channel, you must enable the LACP feature before you can configure it.
- Network State Tracking is only supported with HP Virtual Connect where one physical link from the Flex-10 fabric appears as four Flex-10 NICs (physical NICs) to the VMkernel.

Default Settings

Table 6: Default Settings for Port Channels

Parameters	Default
Port profile type	vEthernet

Parameters	Default
Port profile administrative state	All ports are disabled.
Port channel	Admin up
LACP	Disabled
Load balancing method for Layer 2 interfaces	Source and destination MAC address
Load balancing per module	Disabled
Channel mode	on
LACP offload (Offloading LACP management to VEMs)	Enabled **Note** When upgrading toRelease 4.2(1)SV1(5.2) or higher, you disable the LACP offload feature by default. Starting in Release 5.2(1)SV3(1.1), The LACP offload is the only mode that we support. After you upgrade, The VEMs will automatically go into offload mode for LCAP. We recommend that you add the LACP offload configuration to VSM for consistency.
Network State Tracking: Broadcast interval	5 seconds
Network State Tracking: Split-network mode action	repin
Network State Tracking: Maximum threshold miss count	5 seconds
Network State Tracking: State	Disabled

Configuring Port Channels

Creating a Port Profile for a Port Channel

You can define a port channel in a port profile and, if needed, to configure and pin interface or VLAN subgroups.

Procedure

Step 1 Connect to a single upstream switch. See Connecting to a Single Upstream Switch, on page 60.

Step 2 Connect to multiple upstream switches. SeeConnecting to Multiple Upstream Switches, on page 61.

Step 3 Manually configure interface subgroups. See Manually Configuring Interface Subgroups, on page 65.

Step 4 Pin a vEthernet interface to a subgroup. See Pinning a vEthernet Interface to a Subgroup, on page 66.

Step 5 Pin a control or packet VLAN to a subgroup. See Pinning a Control or Packet VLAN to a Subgroup, on page 67.

Connecting to a Single Upstream Switch

You can configure a port channel whose ports are connected to the same upstream switch. If the ports are connected to multiple upstream switches, see Connecting to Multiple Upstream Switches, on page 61

.

Before You Begin

Know that the channel group number assignment is made automatically when the port profile is assigned to the first interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*	Enters port profile configuration mode for the named port profile. • *name*—Specifies the port profile name, which can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V. • **type**—Specifies the port profile as an Ethernet or vEthernet type. Once configured, this setting cannot be changed. The default is the vEthernet type. For configuring port channels, specify the port profile as an Ethernet type. Defining a port profile as an Ethernet type allows the port profile to be used for physical (Ethernet) ports. In the vCenter Server, the corresponding port group can be selected and assigned to physical ports (PNICs). **Note** If a port profile is configured as an Ethernet type, it cannot be used to configure VMware virtual ports.

	Command or Action	Purpose
Step 3	switch(config-port-prof)# **channel-group auto** [**mode** {**on** | **active** | **passive**}] [**mac-pinning** [**relative**]]	Defines a port channel group in which a unique port channel is created and automatically assigned when the port profile is assigned to the first interface. Each additional interface that belongs to the same module is added to the same port channel. In VMware environments, a different port channel is created for each module. • **mode—**Sets the port channel mode to **on**, **active**, or **passive** (active and passive use LACP). • **mac-pinning—**Designates that one subgroup per Ethernet member port must be automatically assigned if the upstream switch does not support port channels. • **relative—**Specifies that the subgroup numbering begins at zero and continues numbering the subgroups consecutively.
Step 4	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	(Optional) Displays the configuration for verification.
Step 5	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a port channel that connects to one upstream switch:

```
switch# configure terminal
switch(config)# port-profile AccessProf
switch(config-port-prof)# channel-group auto mode on
switch(config-port-prof)# show port-profile name AccessProf
port-profile AccessProf
  description: allaccess4
  status: disabled
capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    channel-group auto mode on
  evaluated config attributes:
    channel-group auto mode on
  assigned interfaces:
switch(config-port-prof)#
```

Connecting to Multiple Upstream Switches

You can create a port channel that connects to multiple upstream switches.

Before You Begin

- Log in to the CLI in EXEC mode.
- If the ports are connected to a single upstream switch, see Connecting to a Single Upstream Switch.
- Configure an uplink port profile to be used by the physical NICs in the VEM in virtual port channel-host mode (vPC-HM) when the ports connect to multiple upstream switches.
- If you are connecting to multiple upstream switches that do not support port channels, know that MAC pinning is the preferred configuration. You can configure MAC pinning using this procedure.
- The channel group mode must be set to on (active and passive modes use LACP).
- You must know whether CDP is configured in the upstream switches.
    - If configured, CDP packets from the upstream switch are used to automatically create a subgroup for each upstream switch to manage its traffic separately.
    - If not configured, after completing this procedure, you must manually configure subgroups to manage the traffic flow on the separate switches. See Manually Configuring Interface Subgroups.

Caution Connectivity may be disrupted for up to 60 seconds if the CDP timer is set to 60 seconds (the default).

Caution The VMs behind the Cisco Nexus 1000V receive duplicate packets from the network for unknown unicasts, multicast floods, and broadcasts if vPC-HM is not configured when port channels connect to two different upstream switches.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*	Enters port profile configuration mode for the named port profile. • *name*—Specifies the port profile name, which can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V. • **type**—Specifies the port profile as an Ethernet or vEthernet type. Once configured, this setting cannot be changed. The default is the vEthernet type. For configuring port channels, specify the port profile as an Ethernet type. Defining a port profile as an Ethernet type allows the port profile to be used for physical (Ethernet) ports. In the vCenter Server, the corresponding port group can be selected and assigned to physical ports (PNICs).

	Command or Action	Purpose
		Note If a port profile is configured as an Ethernet type, it cannot be used to configure VMware virtual ports.
Step 3	switch(config-port-prof)# **channel-group auto mode on** [**sub-group** {**cdp** | **manual**}] [**mac-pinning**[**relative**]]	Creates a unique asymmetric port channel (also known as vPC-HM) and automatically assigns it when the port profile is assigned to the first interface. Each additional interface that belongs to the same module is added to the same port channel. In VMware environments, a different port channel is created for each module. The following options are also defined: • **mode**—Sets the port channel mode to on (active and passive use LACP) • **sub-group**—Identifies this channel group as asymmetric, or connected to more than one switch. ◦ **cdp**—Specifies that CDP information is used to automatically create subgroups for managing the traffic flow. ◦ **manual**—Specifies that subgroups are configured manually. This option is used if CDP is not configured on the upstream switches. To configure subgroups, see Manually Configuring Interface Subgroups. • **mac-pinning**—Specifies that Ethernet member ports are assigned to subgroups automatically, one subgroup per member port. This option is used if the upstream switch does not support port channels. • **relative**—The subgroup numbering begins at zero and continues numbering the subgroups consecutively.
Step 4	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	(Optional) Displays the configuration for verification.
Step 5	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to create a port channel that connects to multiple upstream switches that support CDP:

```
switch# configure terminal
switch(config)# port-profile UpLinkProfile2
switch(config-port-prof)# channel-group auto mode on sub-group cdp
switch(config-port-prof)# show port-profile name UpLinkProfile2
```

```
port-profile UpLinkProfile2
  description:
  type: ethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    channel-group auto mode on sub-group cdp
  evaluated config attributes:
    channel-group auto mode on sub-group cdp
  assigned interfaces:
switch(config-port-prof)# copy running-config startup-config
```

This example shows how to create a port channel that connects to multiple upstream switches that do not support CDP:

```
switch# configure terminal
switch(config)# port-profile UpLinkProfile3
switch(config-port-prof)# exit
switch(config)# interface ethernet3/2-3
switch(config-if)# sub-group-id 0
switch(config-port-prof)# show port-profile name
switch(config-port-prof)# show port-profile name UplinkProfile3
port-profile UplinkProfile3
  description:
  type: ethernet
  status: enabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group: UplinkProfile3
  max ports: -
  inherit:
  config attributes:
    channel-group auto mode on sub-group manual
  evaluated config attributes:
    channel-group auto mode on sub-group manual
  assigned interfaces:
switch(config-port-prof)# copy running-config startup-config
```

This example shows how to create a port channel that connects to multiple upstream switches that do not support port channels:

```
switch# configure terminal
switch(config)# port-profile UpLinkProfile1
switch(config-port-prof)# channel-group auto mode on mac-pinning
switch(config-port-prof)# show port-profile name UpLinkProfile1
port-profile UpLinkProfile1
  description:
  type: ethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    channel-group auto mode on mac-pinning
  evaluated config attributes:
    channel-group auto mode on mac-pinning
  assigned interfaces:
switch(config-port-prof)# copy running-config startup-config
```

Manually Configuring Interface Subgroups

You can manually configure port channel subgroups to manage the traffic flow on multiple upstream switches. This action is required for a port channel that connects to multiple upstream switches where CDP is not configured.

Before You Begin

- Log in to the CLI in EXEC mode.
- Configure the port profile for the port channel using the procedure in Connecting to Multiple Upstream Switches, on page 61.
- Know the interface range and the subgroup IDs (0 to 31) for traffic to the upstream switches.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *range*	Enters interface configuration mode for the specified interface range.
Step 3	switch(config-if)# **sub-group id** *number*	Manually configures a subgroup to manage traffic for the upstream switch. Allowable subgroup numbers are from 0 to 31.
Step 4	Repeat Step 2 and 3.	Perform this step for each port connected to an upstream switch where CDP is not configured.
Step 5	switch(config-if)# **show interface ethernet** *range*	(Optional) Displays the configuration for verification.
Step 6	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to manually configure port channel subgroups for a host in module 3 which has four physical ports. The upstream switches do not support CDP. Ethernet ports 3/2 and 3/3 connect to one upstream switch and the Ethernet ports 3/4 and 3/5 connect to another upstream switch.

```
switch# configure terminal
switch(config)# int eth3/2
switch(config-if)# sub-group-id 0
switch(config-if)# int eth3/3
switch(config-if)# sub-group-id 0
switch(config-if)# int eth3/4
switch(config-if)# sub-group-id 1
switch(config-if)# int eth3/5
switch(config-if)# sub-group-id 1
switch(config-if)# copy running-config interface
. . .
interface Ethernet3/2
```

```
  inherit port-profile system-uplink-pvlan
  sub-group-id 0
interface Ethernet3/3
  inherit port-profile system-uplink-pvlan
  sub-group-id 0
interface Ethernet3/4
  inherit port-profile system-uplink-pvlan
  sub-group-id 1
interface Ethernet3/5
  inherit port-profile system-uplink-pvlan
  sub-group-id 1
switch(config-if)#
```

Pinning a vEthernet Interface to a Subgroup

You can pin a vEthernet interface to a specific port channel subgroup in the port profile configuration.

Note You can also pin a subgroup to a vEthernet interface in the interface configuration. See Configuring Static Pinning for an Interface, on page 72.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know the subgroup ID (0 to 31) for the vEthernet interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile type vethernet** *name*	Enters port profile configuration mode for the named port profile.
Step 3	switch(config-port-prof)# **pinning id** *subgroup_id* [**backup** *subgroup_id1...subgroup_id7*]	For the named port profile, assigns (or pins) a vEthernet interface to a port channel subgroup (0–31). **backup**—Optionally specifies an ordered list of backup subgroups for pinning to be used if the primary subgroup is not available.
Step 4	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	(Optional) Displays the configuration for verification.
Step 5	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to create a vEthernet port profile and pin it to port channel subgroup 3:

```
switch# configure terminal
switch(config)# port-profile type vethernet PortProfile1
switch(config-port-prof)# pinning id 3
switch(config-port-prof)# show port-profile name PortProfile1
port-profile PortProfile1
  description:
  type: vethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    pinning id 3
  evaluated config attributes:
    pinning id 3
  assigned interfaces:
switch(config-port-prof)# copy running-config startup-config
```

This example shows how to create a vEthernet port profile and pin it to port channel subgroup 3 and backup subgroups 4 and 6:

```
switch# configure terminal
switch(config)# port-profile type vethernet PortProfile1
switch(config-port-prof)# pinning id 3 backup 4 6
switch(config-port-prof)# show port-profile name PortProfile1
port-profile PortProfile1
  description:
  type: vethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    pinning id 3 backup 4 6
  evaluated config attributes:
    pinning id 3
  assigned interfaces:
switch(config-port-prof)# copy running-config startup-config
```

Pinning a Control or Packet VLAN to a Subgroup

You can pin a control or packet VLAN to a specific subgroup.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that the existing port profile must be a system port profile.
- Know that the port profile must be an Ethernet type.
- If you are pinning a control or packet VLAN, know that it must already be in the port profile.
- If you are pinning a control VLAN, know that the control VLAN must already be one of the system VLANs in the port profile.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** *name*	Enters port profile configuration mode for the named port profile.
Step 3	switch(config-port-prof)# **pinning** {**control-vlan** | **packet-vlan**} *subgroup_id*	Assigns (or pins) a control VLAN or packet VLAN to a port channel subgroup (0 to 31).
Step 4	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	(Optional) Displays the configuration for verification.
Step 5	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure static pinning on a control VLAN:

```
switch# configure terminal
switch(config)# port-profile SystemProfile1
switch(config-port-prof)# pinning control-vlan 3
switch(config-port-prof)# show port-profile SystemProfile1
port-profile SystemProfile1
  description:
  type: ethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: 3
  pinning packet-vlan: -
  system vlans: 1
  port-group: SystemProfile1
  max ports: -
  inherit:
  config attributes:
    switchport mode trunk
    switchport trunk allowed vlan 1-5
    no shutdown
  evaluated config attributes:
    switchport mode trunk
    switchport trunk allowed vlan 1-5
    no shutdown
  assigned interfaces:
switch(config-port-prof)# copy running-config startup-config
```

This example shows how to configure static pinning on a packet VLAN:

```
switch# configure terminal
switch(config)# port-profile SystemProfile1
switch(config-port-prof)# pinning packet-vlan 0
switch(config-port-prof)# show port-profile name SystemProfile1
port-profile SystemProfile1
  description:
  type: ethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: 0
  system vlans: 1
  port-group:
  max ports: -
  inherit:
  config attributes:
    switchport mode access
    switchport access vlan 1
    switchport trunk native vlan 1
    no shutdown
  evaluated config attributes:
    switchport mode access
    switchport access vlan 1
    switchport trunk native vlan 1
    no shutdown
assigned interfaces:
switch(config-port-prof)# copy running-config startup-config
```

Migrating a Channel Group to a Port Profile

You can migrate a channel group to a port profile.

Before You Begin

Log in to the CLI in EXEC mode.

Procedure

Step 1 Place the host in maintenance mode.

Step 2 Do one of the following:

- If distributed resource scheduling (DRS) is enabled, make sure to wait until the virtual machines are migrated to other host(s).
- Otherwise, manually migrate the virtual machines.

Step 3 When all the virtual machines are successfully migrated, from the Cisco Nexus 1000V CLI, create a new Ethernet type port profile for the uplink ports on this host.

- Enter one of the following commands:
    - **channel-group auto mode active | passive**
    - **channel-group auto mode on mac-pinning**
- Perform a CLI override on the existing port channels.

Step 4 Remove the port channel configuration from the uplink switches.

Note The new port channel has a new port channel ID.

Step 5 When all the port(s) are moved from the old port profile, use the following command from the Cisco Nexus 1000V CLI to delete the port channels with zero members:**no interface port-channel** *id*

Step 6 Bring the host out of maintenance mode.

Step 7 To save the running configuration persistently through reboots and restarts by copying it to the startup configuration, enter the following command:
copy running-config startup-config

Step 8 Create the port channel type in the upstream switch. See Creating a Port Profile for a Port Channel.

Migrating Port Profile Types in a Port Profile

To move port profile types in a port profile, you tear down the existing port channel then recreate the port channel.

Before You Begin

Log in to the CLI in EXEC mode.

Procedure

Step 1 Place the host in maintenance mode.

Step 2 Do one of the following:

- If distributed resource scheduling (DRS) is enabled, make sure to wait until the virtual machines are migrated to other host(s).
- Otherwise, manually migrate the virtual machines.

Step 3 When all the virtual machines are successfully migrated, from the Cisco Nexus 1000V CLI, create a new Ethernet type port profile for the uplink ports on this host.

- Enter one of the following commands:
    - **channel-group auto mode active | passive**
    - **channel-group auto mode on mac-pinning**
- Perform a CLI override on the existing port channels.

Step 4 Remove the port channel that you want to migrate in the upstream switch. See Removing a Port Channel Group from a Port Profile.

Step 5 Remove the port channel in the upstream switch.

Step 6 Manually configure subgroup IDs in the Cisco Nexus 1000V Ethernet interface. See Manually Configuring Interface Subgroups

Step 7 Change the port channel type in the Cisco Nexus 1000V port profile. See Migrating a Channel Group to a Port Profile

Step 8 Change the port channel type in the Cisco Nexus 1000V port profile. See Connecting to a Single Upstream Switch

Step 9 Bring the host out of maintenance mode.

Step 10 Migrate the virtual machines back to this host.

Step 11 Save the running configuration persistently through reboots and restarts by copying it to the startup configuration by entering the following command:
copy running-config startup-config

Step 12 Create the port channel type that you want in the upstream switch. See Creating a Port Profile for a Port Channel.

Configuring Network State Tracking for vPC-HM

You can configure Network State Tracking to pinpoint link failures on port channels configured for vPC-HM.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that once you enable Network State Tracking, it is used on every VEM that is configured with a vPC-HM port profile.
- If you specify repinning (the default) and a split network is detected, know that Ethernet interfaces are inactivated, and the vEths are redistributed among all interfaces including the reactivated Ethernet interfaces. Restoration to the earlier pinned state is not guaranteed.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **track network-state enable**	Enables Network State Tracking on all interfaces in vPC-HM port-channels.
Step 3	switch(config)# **track network-state interval** *seconds*	(Optional) Specifies the interval of time, from 1 to 10 seconds, between which tracking broadcasts are sent; and the interval for tracking packets. The default interval is 5 seconds between broadcasts.
Step 4	switch(config)# **track network-state split action** [**repin** | **log-only**]	(Optional) Specifies the action to be taken if a split network is detected. • **repin—**Pins traffic to another uplink (the default). • **no repin—**Leaves vEthernet interfaces where they are.
Step 5	switch(config)# **track network-state threshold miss-count** *count*	(Optional) Specifies the maximum number of broadcasts that can be missed successively (from 3 to 7) before a split network is declared. The default is 5 missed broadcasts.

	Command or Action	Purpose
Step 6	switch(config)# **show network-state tracking config**	(Optional) Displays the Network State Tracking configuration for verification.
Step 7	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure Network State Tracking with an 8-second interval between each sent broadcast, repinning traffic to another uplink if a split network is detected, and a maximum of 7 missed broadcasts before declaring a split network:

```
switch# configure terminal
switch(config)# track network-state enable
switch(config)# track network-state interval 8
switch(config)# track network-state split action repin
switch(config)# track network-state threshold miss-count 7
switch(config)# show network-state tracking config
Tracking mode         : enabled
Tracking Interval     : 8 sec
Miss count threshold : 7 pkts
Split-network action : repin
switch(config)#
```

Configuring Static Pinning for an Interface

You can configure static pinning on a vEthernet interface.

Note You can also pin a subgroup to a vEthernet interface in the port profile configuration. See Pinning a vEthernet Interface to a Subgroup.

Before You Begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Enters interface configuration mode for the specified interface (from 1 to 1048575).
Step 3	switch(config-if)# **pinning id subgroup_id** [**backup** *subgroup_id1...subgroup_id7*]	Assigns (or pins) a vEthernet interface to a specific port channel subgroup (from 0 to 31).

	Command or Action	Purpose
		backup—Optionally specifies an ordered list of backup subgroups for pinning to be used if the primary subgroup is not available.
Step 4	switch(config-if)# **show running-config interface vethernet** *interface-number*	(Optional) Displays the pinning configuration of the specified interface.
Step 5	switch(config-if)# **module vem** *module_number* **execute vemcmd show pinning**	(Optional) Displays the pinning configuration on the specified VEM.
Step 6	switch(config-if)# **module vem** *module_number* **execute vemcmd show static pinning config**	(Optional) Displays the VSM configured pinning subgroups.
Step 7	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to pin subgroup ID 0 to vEthernet interface 1:

```
switch# configure terminal
switch(config)# interface vethernet 1
switch(config-if)# pinning id 0
switch(config-if)# show running-config interface vethernet 1
version 4.0(4)SV1(2)

interface Vethernet3
  service-policy type qos input policy1
  pinning id 0

switch(config-if)# exit
switch(config)# exit
switch# module vem 3 execute vemcmd show pinning
  LTL    IfIndex  PC_LTL  VSM_SGID  VEM_SGID  Eff_SGID
   48   1b040000     304         0         0         0
switch#
```

This example shows the output after configuring backup subgroups for pinning:

```
switch(config-if)# module vem 4 execute vemcmd show static pinning config
  LTL    IfIndex   VSM_SGID  Backup_SGID
   48   1c0000a0      0,          1,2
   50   1c000100      0,          1

switch(config-if)# copy running-config startup-config
```

Removing a Port Channel Group from a Port Profile

You can remove a port channel group from a port profile.

Before You Begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** *name*	Specifies the port profile from which the port channel will be removed.
Step 3	switch(config-port-prof)# **no channel-group auto**	Removes the channel group configuration from all member interfaces in the specified port profile.
Step 4	switch(config-port-prof)# **show port-profile** *name*	(Optional) Displays the configuration for verification.
Step 5	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to remove a port channel group from a port profile:

```
switch# configure terminal
switch(config)# port-profile testProf
switch(config-port-prof)# no channel-group auto
switch(config-port-prof)# show port-profile testProf
switch(config-port-prof)#
```

Shutting Down and Restarting a Port Channel Interface

You can shut down and restart a port channel interface.

Before You Begin

- Log in to the CLI in EXEC mode.
- When you shut down a port channel interface, know that no traffic passes, and the interface is administratively down.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Enters interface configuration mode for the specified port channel interface.
Step 3	switch(config-if)# **shutdown** | **no shutdown**	The **shutdown** keyword shuts down the interface. No traffic passes and the interface displays as administratively down. The default is **no shutdown**.

	Command or Action	Purpose
		Brings the interface back up. The interface displays as administratively up. If there are no operational problems, traffic passes. The default is **no shutdown**.
Step 4	switch(config-if)# **show interface port-channel** *channel-number*	(Optional) Displays interface information for the specified port channel.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to bring up the interface for port channel 2:

```
switch# configure terminal
switch(config)# interface port-channel 2
switch(config-if)# no shutdown
```

Adding a Description to a Port Channel Interface

You can add a description to a port channel interface.

Before You Begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Enters interface configuration mode for the specified port channel interface. For the channel number, the range is from 1 to 4096. The port channel associated with this channel group is automatically created if the port channel does not already exist.
Step 3	switch(config-if)# **description** *string*	Adds a description to the port channel interface. For the string, the description can be up to 80 alphanumeric characters. **Note** You do not need to use quotations around descriptions that include spaces.

	Command or Action	Purpose
Step 4	switch(config-if)# **show interface port-channel** *channel-number*	(Optional) Displays interface information for the specified port channel.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to add a description to port channel 2:

```
switch# configure terminal
switch(config)# interface port-channel 2
switch(config-if)# description engineering
```

Configuring Port Channel Load Balancing

You can configure port channel load balancing.

Before You Begin

- Log in to the CLI in EXEC mode.
- Configure port channel load balancing for the entire device or for a single module.
- Know that module-based load balancing takes precedence over device-based load balancing.
- Know that the default load balancing method is the source MAC address.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-channel load-balance ethernet** {**dest-ip-port** | **dest-ip-port-vlan** | **destination-ip-vlan** | **destination-mac** | **destination-port** | **source-dest-ip-port** | **source-dest-ip-port-vlan** | **source-dest-ip-vlan** | **source-dest-mac** | **source-dest-port** | **source-ip-port** | **source-ip-port-vlan** | **source-ip-vlan** | **source-mac** | **source-port** | **source-virtual-port-id** | **vlan-only** }	Configures the load balance method for the device or module. The range depends on the device. The default load balancing method uses the source MAC address.
Step 3	switch(config)# **show interface port-channel load balance**	(Optional) Displays the port channel load-balancing method.

	Command or Action	Purpose
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure the source IP load-balancing method for port channels on module 5:

```
switch# configure terminal
switch# interface port channel 2
switch# port-channel load-balance ethernet source-ip module 5
```

Configuring the Speed and Duplex Settings for a Port Channel Interface

You can configure the speed and duplex settings for a port channel interface.

Before You Begin

- Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Specifies the port channel interface that you want to configure and enters the interface mode. Allowable channel numbers are from 1 to 4096.
Step 3	switch(config-if)# **speed** {**10** | **100** | **1000** | **auto**}	Sets the speed for the port channel interface. The default is auto for autonegotiation.
Step 4	switch(config-if)# **duplex** {**auto** | **full** | **half**}	Sets the duplex mode for the port channel interface. The default is auto for autonegotiation.
Step 5	switch(config-if)# **show interface port-channel** *channel-number*	(Optional) Displays interface information for the specified port channel.
Step 6	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to set port channel 2 to 100 Mbps:

```
switch# configure terminal
switch(config)# interface port channel 2
switch(config-if)# speed 100
```

Restoring the Default Load-Balancing Method

You can restore the default load-balancing method.

Before You Begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no port-channel load-balance ethernet**	Restores the default load-balancing method, which is the source MAC address.
Step 3	switch(config)# **show interface port-channel load balance**	(Optional) Displays the port channel load-balancing method.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to restore the default load-balancing method:

```
switch# configure terminal
switch(config)# no port-channel load-balance ethernet
switch(config)# show port-channel load-balance
```

Configuring an LACP Port Channel

You can configure the following requirements for LACP:

- Enable LACP support for port channels.
- Configure the individual port channel links so that they are allowed to operate with LACP.
- Configure a system uplink port profile for LACP.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that the default port channel mode is on.

- Enable the LACP feature support before you configure LACP. This procedure has a step for enabling the LACP feature.
- When you configure port channels with no associated aggregation protocol, know that all interfaces on both sides of the link remain in the on channel mode.
- Know that the LACP mode for individual links in an LACP port channel indicates that the link is allowed to operate with LACP.
- Define a native VLAN for the trunk port. Although it may not be used for data, the native VLAN is used for LACP negotiation. If you want traffic forwarded on the native VLAN of the trunk port, the native VLAN must be in the allowed VLAN list and system VLAN list.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature lacp**	Enables LACP support for port channels.
Step 3	switch(config-if)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*	Enters port profile configuration mode for the named port profile. • **name—**Specifies the port profile name, which can be up to 80 characters and must be unique for each port profile on the . • **type—**(Optional) Specifies the port profile as an Ethernet or vEthernet type. Once configured, this setting cannot be changed. The default is the vEthernet type. For configuring port channels, specify the port profile as an Ethernet type. Defining a port profile as an Ethernet type allows the port profile to be used for physical (Ethernet) ports. In the vCenter Server, the corresponding port group can be selected and assigned to physical ports (PNICs). **Note** If a port profile is configured as an Ethernet type, it cannot be used to configure VMware virtual ports.
Step 4	switch(config-port-prof)# **vmware port-group** [**pg_name**]	Designates the port profile as a VMware port group. The port profile is mapped to a VMware port group of the same name unless you specify a name here. When you connect the VSM to vCenter Server, the port group is distributed to the virtual switch on the vCenter Server.
Step 5	switch(config-port-prof)# **switchport mode** {**access** | **private-vlan** {**host** | **promiscuous**} | **trunk**}	Designates how the interfaces are to be used. Allowable port modes: • access • private-vlan ◦ host ◦ promiscuous

	Command or Action	Purpose
		- trunk A trunk port transmits untagged packets for the native VLAN and transmits encapsulated, tagged packets for all other VLANs.
Step 6	switch(config-port-prof)# **switchport trunk allowed vlan** *vlan-id-list*	Designates the port profile as trunking and defines VLAN access to it as follows: - allowed-vlans—Defines VLAN IDs that are allowed on the port. - add—Lists VLAN IDs to add to the list of those allowed on the port. - except—Lists VLAN IDs that are not allowed on the port. - remove—Lists VLAN IDs whose access is to be removed from the port. - all—Indicates that all VLAN IDs are allowed on the port, unless exceptions are also specified. - none—Indicates that no VLAN IDs are allowed on the port. If you do not configure allowed VLANs, the default VLAN 1 is used as the allowed VLAN. If you want traffic forwarded on the native VLAN of the trunk port, the native VLAN must be in the allowed VLAN list.
Step 7	switch(config-port-prof)# **show port-profile** *name*	(Optional) Displays the configuration for verification.
Step 8	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to remove a port channel group from a port profile:

```
switch# configure terminal
switch(config)# port-profile testProf
switch(config-port-prof)# no channel-group auto
switch(config-port-prof)# show port-profile testProf
switch(config-port-prof)#
```

Configuring VEM Management of LACP

You can offload management of LACP from the VSM to the VEMs.

Before You Begin

- Log in to the CLI in EXEC mode.

- After offloading the management of LACP from the VSM to the VEM, you must preserve the running configuration in the startup configuration and reload the VSM before the offload takes effect. This procedure has steps for doing this.
- Know that offloading of LACP management to the VEMs is enabled by default on the VSM.

Note When you upgrade to 4.2(1)SV1(4) or before, the LACP offload management to the VEMs is disabled by default. However, LACP offload is the only supportable mode starting with release 5.2(1)SV3(1.1) and VEMs will automatically go into that mode after you upgrade. We recommend that you add the LACP offload configuration to VSM for consistency.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **lacp offload**	(Optional) Offloads LACP management from the VSM to the VEMs. If enabling LACP offload, a message displays to let you know that a reload is required. Offload of LACP management to the VEMs is enabled by default. **Note** When you upgrade to 4.2(1)SV1(4) or before, the LACP offload management to the VEMs is disabled by default. However, LACP offload is the only supportable mode starting with release 5.2(1)SV3(1.1) and VEMs will automatically go into that mode after you upgrade. We recommend that you add the LACP offload configuration to VSM for consistency.
Step 3	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 4	switch(config)# **show lacp offload status**	(Optional) Displays the LACP offload status for verification. **Note** The current status does not change to enabled until after a reload.
Step 5	switch(config)# **reload**	Reboots both the primary and secondary VSM.
Step 6	switch(config)# **show lacp offload status**	(Optional) Displays the LACP offload status for verification. **Note** The current status does not change to enabled until after a reload.

This example shows how to offload management of LACP from the VSM to the VEMs:

```
switch# configure terminal
switch(config)# lacp offload
Please do a "copy running startup" to ensure the new setting takes effect on next reboot
LACP Offload Status can be verified using "show lacp offload status"
Change in LACP Offload Status takes effect only on the next VSM Reboot
This can potentially cause modules with LACP uplinks to flap

switch(config)# copy running-config startup-config
switch(config)# show lacp offload status
    Current Status        : Disabled
    Running Config Status : Enabled
    Saved Config Status   : Enabled
switch(config)# reload
This command will reboot the system. (y/n)?  [n] y
2010 Sep  3 11:33:35 n1000v %PLATFORM-2-PFM_SYSTEM_RESET: Manual system restart from Command
 Line Interface
n1000v(config)# show lacp offload status
Current Status        : Enabled
    Running Config Status : Enabled
    Saved Config Status   : Enabled
switch(config)#
```

Verifying the Port Channel Configuration

Use the following commands to verify the port channel configuration:

Command	Purpose
show feature	Displays the features available and whether they are enabled.
show interface port-channel *channel-number*	Displays the status of a port channel interface.
show lacp port-channel [**interface port-channel** *channel-number*]	Displays information about LACP port channels.
show lacp interface ethernet *slot/port*	Displays information about specific LACP interfaces.
show lacp offload status	Displays whether LACP management is offloaded to the VEMs. • Enabled—LACP is managed by VEMs. • Disabled—LACP is managed by the VSM.
show network-state tracking config	Displays the Network State Tracking configuration for verification.
show network-state tracking { **module** *modID* | **interface** *channelID*}	Displays the Network State Tracking status for a module or interface.
show port-channel compatibility-parameters	Displays the parameters that must be the same among the member ports in order to join a port channel.

Command	Purpose
show port-channel database [**interface port-channel** *channel-number*]	Displays the aggregation state for one or more port channel interfaces.
show port-channel load-balance	Displays the type of load balancing in use for port channels.
show port-channel summary	Displays a summary for the port channel interfaces.
show port-channel traffic	Displays the traffic statistics for port channels.
show port-channel usage	Displays the range of used and unused channel numbers.
show running-config interface ethernet *port/slot*	Displays information about the running configuration of the specified Ethernet interface.
show running-config interface port-channel *channel-number*	Displays information about the running configuration of the port channel.
show running-config interface vethernet *interface-number*	Displays information about the running configuration of the specified vEthernet interface.

Monitoring Port Channels

Use the following commands to monitor the port channel interface configuration:

Command	Purpose
clear counters interface port-channel *channel-number*	Clears the counters.
show interface counters [**module** *module*]	Displays input and output octets unicast packets, multicast packets, and broadcast packets.
show interface counters detailed [**all**]	Displays input packets, bytes, and multicast and output packets and bytes.
show interface counters errors [**module** *module*]	Displays information about the number of error packets.
show lacp counters [**interface port-channel** *channel-number*]	Displays information about LACP statistics.

Configuration Examples for Port Channels

Configuration Example: Create an LACP Port Channel

This example shows how to set the LACP-enabled interface to the active port channel mode for Ethernet interface 1/4 in channel group 5 and then configure an LACP port profile for the port channel:

```
switch# configure terminal
switch(config)# feature lacp
switch(config)# interface ethernet 1/4
switch(config-if)# channel-group 5 mode active
switch(config-if)# port-profile type ethernet system-uplink
switch(config-port-prof)# vmware port-group lacp
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# switchport trunk allowed vlan 1-100
switch(config-port-prof)# channel-group auto mode active
switch(config-port-prof)# system vlan 1,10,20
switch(config-port-prof)# state enabled
switch(config-port-prof)# show port-channel summary
switch(config-port-prof)# copy running-config startup-config
```

Configuration Example: Configuring Network State Tracking for vPC-HM

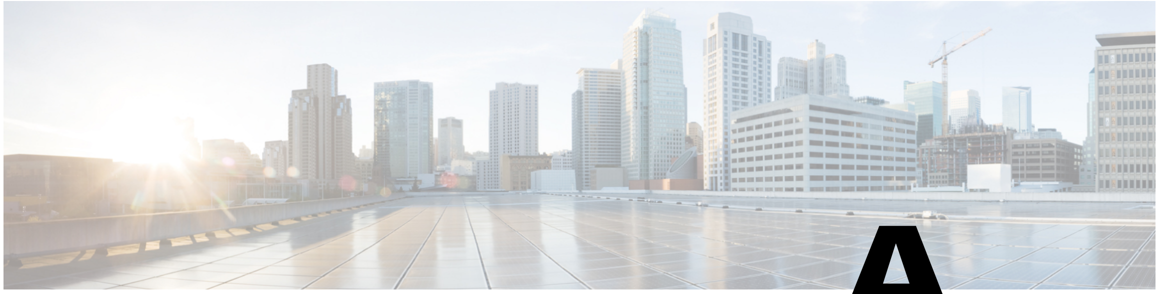
This example shows how to configure Network State Tracking with an 8-second interval between sent broadcasts, with a maximum of 7 missed broadcasts before declaring a split network, and how to repin traffic to another uplink if a split network is detected:

```
switch# configure terminal
switch(config)# track network-state
switch(config)# track network-state interval 8
switch(config)# track network-state split action repin
switch(config)# track network-state threshold miss-count 7
switch(config)# show network-state tracking config
Tracking mode        : enabled
Tracking Interval    : 8 sec
Miss count threshold : 7 pkts
Split-network action : repin
switch(config)#
```

Feature History for Port Channels

Feature Name	Releases	Feature Information
IPv6 support is added for IP and TCP/UDP-based load balancing.	5.2(1)SV3(1.1)	IPv6 packets will be parsed and matching IPv6 fields will be used for load balancing.
Backup subgroups	4.2(1)SV1(4a)	You can assign up to seven backup subgroups when pinning the primary subgroup.
Port channel relative numbering	4.2(1)SV1(4a)	The subgroup numbering begins at zero and is not tied to the vmnic number.

Feature Name	Releases	Feature Information
Port channel vPC-HM	4.2(1)SV1(4)	The interface **sub-group cdp** command is removed from the port channel vPC-HM configuration when connecting to multiple upstream switches.
Network State Tracking for vPC-HM port channels	4.2(1)SV1(4)	Pinpoints link failure on a port channel configured for vPC-HM.
VEM management of LACP	4.2(1)SV1(4)	Offloading management of LACP from the VSM to the VEMs.
Enabling the LACP port channel function	4.2(1)SV1(4)	The **feature lacp** command is added to enable support of LACP port channels. Previously LACP was enabled automatically.
vPC-Host Mode	4.0(4)SV1(2)	Support for manual creation of subgroups.
Static Pinning	4.0(4)SV1(2)	Support for attaching (or pinning) a vEthernet interface to a specific port channel subgroup.
Port Channels	4.0(4)SV1(1)	This feature was introduced.



APPENDIX A

Supported RFCs

This chapter contains the following sections:

Supported RFCs

The following table lists the supported IETF RFCs for interfaces.

Table 7: IP Services RFCs

RFCs	Title
RFC 786	UDP
RFC 791	IP
RFC 792	ICMP
RFC 793	TCP
RFC 826	ARP
RFC 1027	Proxy ARP
RFC 1591	DNS Client
RFC 1812	IPv4 routers

APPENDIX B

Interface Configuration Limits

This chapter contains the following sections:

Interface Configuration Limits

The configuration limits are documented in the *Cisco Nexus 1000V Resource Availability Reference*.