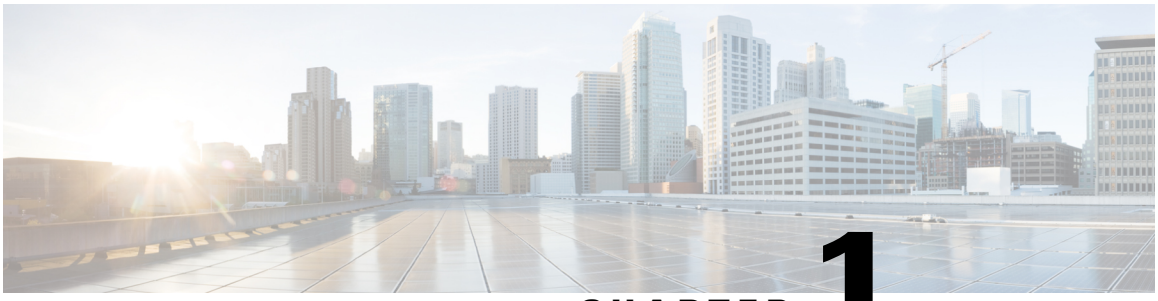
Cisco Nexus 1000V for KVM Port Profile Configuration Guide, Release 5.x

First Published: August 01, 2014

Last Modified: November 09, 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

CONTENTS



CHAPTER 1

New and Changed Information

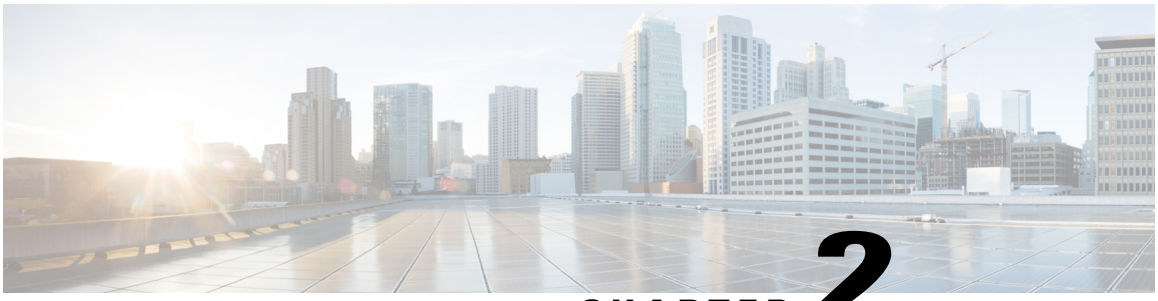
This chapter contains the following sections:

New and Changed Information

Table 1: New and Changed Features

Content	Description	Changed in Release	Where Documented
Private VLAN Port Profiles	This feature is introduced.	5.2(1)SK3(2.1)	Configuring a Private VLAN in a Port Profile, on page 59

New and Changed Information



CHAPTER 2

Overview

This chapter contains the following sections:

Cisco Nexus 1000V for KVM and OpenStack

The Cisco Nexus 1000V for KVM consists of two main components:

- Virtual Ethernet Module (VEM)—A software component that is deployed on each KVM host. Each VM on the host is connected to the VEM through virtual Ethernet (vEth) ports.
- Virtual Supervisor Module (VSM)—The Management component that controls multiple VEMs and helps in the definition of VM-focused network policies. It is deployed either as a virtual appliance on any KVM host or on the Cisco Cloud Services Platform appliance.

Each of these components is tightly integrated with the OpenStack environment:

- The VEM is a hypervisor-resident component and is tightly integrated with the KVM architecture.
- The VSM is integrated with OpenStack using the OpenStack Neutron Plug-in.
- The OpenStack Neutron API has been extended to include two additional user-defined resources:
  - Network profiles as logical groupings of network segments.

Note In Cisco Nexus 1000V for KVM Release 5.2(1)SK3(2.2), network profiles are created automatically for each network type. Network profile creation by administrators is not supported.

- Policy profiles group port policy information, including security.

Using OpenStack, you create VMs, networks, and subnets on the Cisco Nexus 1000V for KVM, by defining components such as the following:

- Tenants
- Network segments, such as VLANs, VLAN trunks, and VXLANs
- IP address pools (subnets)

Using the Cisco Nexus 1000V for KVM VSM, you create port profiles (called policy profiles in OpenStack), which define the port policy information, including security settings.

When a VM is deployed, a port profile is dynamically created on the Cisco Nexus 1000V for KVM for each unique combination of policy port profile and network segment. All other VMs deployed with the same policy to this network reuse this dynamic port profile.

Note You must consistently use OpenStack for all VM network and subnet configuration. If you use *both* OpenStack and the VSM to configure VM networks and subnets, the OpenStack and the VSM configurations can become out-of-sync and result in faulty or inoperable network deployments.

Information About Port Profiles

A port profile is a collection of interface-level configuration attributes that are combined to create a port classification on the OpenStack controller. Using port profiles allows you, as the system administrator, to configure a consistent network policy on the virtual Ethernet (vEthernet) and physical Ethernet interfaces across all of the hosts managed by the Cisco Nexus 1000V Virtual Supervisor Module (VSM) . Both Ethernet and vEthernet port profiles are created on the VSM and published to the VEMs. However, vEthernet port profiles are also pushed to the OpenStack controller.

Live Policy Changes

Port profiles are not static entities but dynamic policies that can change as network needs change. Changes to active port profiles are applied to each switch port that is using the profile, which simplifies the process of applying new network policies or changing an existing policy.

Atomic Inheritance

To maintain a consistent configuration among the interfaces in a port profile, the entire port profile configuration is applied to its member interfaces (this process is sometimes referred to as inheritance).

Rollbacks to a Consistent Configuration

When you update the configuration in a port profile, its member interfaces are also updated. If the configuration fails, the port profile and its member interfaces are rolled back to the last known good configuration for the port profile.

Interface Quarantines

Port profile interfaces are sectioned off and shut down when a port profile configuration is in error. This process is called an Interface Quarantine.

If you create a port profile with a command error, such as a private VLAN mapping error or service policy map error, and then attempt to apply this port profile to an interface, the interface shuts down. The error is not copied to the interface and a system message is generated with details of the error. In this case, you must correct the error in the port profile, return the interface to service, and apply the corrected port profile to the interface.

VLAN Ranges for Trunking and Access Port Profiles

You can configure VLANs only on Ethernet port profiles, not vEthernet policy profiles. Use the information in the following table while configuring trunking and access port profiles. In accordance with the IEEE 802.1Q standard, up to 4094 VLANs are supported.

Table 2: VLAN Ranges

VLAN Numbers	Range	Usage
1	Normal	Cisco default. You can use this VLAN, but you cannot modify or delete it.
2—1005	Normal	You can create, use, modify, and delete these VLANs.

VLAN Numbers	Range	Usage
1006—4094	Extended	You can create, name, and use these VLANs. You cannot change the following parameters: • The state is always active. • The VLAN is always enabled. You cannot shut down these VLANs.
3968—4047 and 4094	Internally allocated	These 80 VLANs, plus VLAN 4094, are allocated for internal device use. You cannot create, delete, or modify any VLANs within the block reserved for internal use.

CHAPTER 3

Creating Basic Port Profiles

This chapter contains the following sections:

Information About Creating Port Profiles

You can create Ethernet or virtual Ethernet (vEthernet) type port profiles.

When you configure Ethernet port profiles, you must configure all attributes; however, when you configure vEthernet port profiles, you configure some attributes in the port profile on the Virtual Supervisor Module (VSM) and other attributes in the network profile on the OpenStack controller. For example, you configure the port profiles with VLANs and VXLANs on the OpenStack controller.

After creating and configuring the port profile, you must enable it so that its configuration is applied to the assigned ports and the port profile is placed in an operational state.

Finally, you must publish the port profile. When you publish an Ethernet type port profile, the port profile configuration is pushed to the VEMs. When you publish a vEthernet type port profile, the port profile configuration is pushed to the VEMs as well as to the OpenStack controller, where the port profile and network profile configurations are combined to automatically generate the configuration that is applied to the vEthernet interface. You cannot modify or delete these automatically generated and dynamic port profiles. For more information about these port profiles, see the *Cisco Nexus 1000V for KVM Virtual Network Configuration Guide*.

Guidelines and Limitations for Creating Port Profiles

- Once you create a port profile as either an Ethernet or vEthernet type, you cannot change the type.
- The **channel-group** command is not supported by vEthernet type port profiles.
- In an installation where multiple Ethernet port profiles are active on the same Virtual Ethernet Module (VEM), we recommend that they do not carry the same VLAN(s). The allowed VLAN list should be mutually exclusive. You can configure overlapping VLANs but they might cause duplicate packets to be received by virtual machines in the network.
- To maintain consistency between the port profile definition and what is applied to an interface, if a port profile modification is rejected by any port, the modification is rejected by the port profile too.
- A maximum transmission unit (MTU) can only be configured for uplink Ethernet type port profiles.
- User-configured port profile inheritance is not supported on the Cisco Nexus 1000V for KVM.

Default Settings

The following table lists the default settings in the port profile configuration.

Parameter	Default
cdp	Enabled
description	None
max-ports	512
min-ports	1
shutdown	All ports administratively disabled
state	Disabled
type	vEthernet

Creating Port Profiles

Creating a Basic Ethernet Port Profile

Before You Begin

- You are logged in to the CLI in EXEC mode.
- You know whether the ports need to be initialized with system settings.

- You have identified the characteristics needed for this port profile.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile type ethernet** *name*	Enters port profile configuration mode for the named port profile. If the port profile does not already exist, it is created. Once configured, the type cannot be changed. Defining an Ethernet type port profile allows the port profile to be used for physical (Ethernet) ports.
Step 3	switch(config-port-prof)# **channel-group auto mode on**	(Optional) Configures the port-channels in the port profile.
Step 4	switch(config-port-prof)# **no shutdown**	Enables all ports in the port profile.
Step 5	switch(config-port-prof)# **state enabled**	Enables the operational state of the port profile.
Step 6	switch(config-port-prof)# **publish port-profile** [*name*]	Pushes the port profile to the VEMs.
Step 7	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 new Ethernet type port profile:

```
switch(config)# port-profile type ethernet UplinkPortChannel
switch(config-port-prof)# channel-group auto mode on
switch(config-port-prof)# no shutdown
switch(config-port-prof)# state enabled
switch(config-port-prof)# publish port-profile
switch(config)# copy running-config startup-config
```

Configuring Access Ports Using Port Profiles

An access port transmits packets on only one untagged VLAN. You can specify the VLAN, and it becomes the access VLAN. If you do not specify a VLAN for an access port, that interface carries traffic only on the default VLAN 1.

Procedure

Step 1 switch# **configure terminal**
Enters global configuration mode.

Step 2 switch(config)# [no] **vlan** *vlan-id*
Creates or deletes, and saves in the running configuration, a VLAN or a range or VLANs.

Step 3 switch(config)# **port-profile type ethernet** *name*
Enters port profile configuration mode for the named port profile. If the port profile does not already exist, it is created using the following characteristics:

- *name*—The port profile name can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V.
- **type**—(Optional) The default port profile type is Ethernet.

Step 4 switch(config-port-prof)# **switchport mode access**
Sets port mode access.

Step 5 switch(config-port-prof)# **switchport access vlan** [*vlan-id-access*]
Assigns an access VLAN ID to this port profile.

Note An access port transmits packets on only one untagged VLAN. You can specify the VLAN, and it becomes the access VLAN. If you do not specify a VLAN for an access port, that interface carries traffic only on the default VLAN 1. If you do not specify a VLAN ID, then VLAN 1 is used automatically.

Step 6 switch(config-port-prof)# **no shutdown**
Administratively enables all ports in the profile.

Step 7 switch(config-port-prof)# **state enabled**
Enables the port profile and applies its configuration to the assigned ports.

Step 8 switch(config-port-prof)# **publish port-profile** [*name*]
Pushes the port profile to the VEMs.

Step 9 (Optional) switch(config-port-prof)# **copy running-config startup-config**
Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

```
switch# configure terminal
switch(config)# port-profile mgmt-access
switch(config-port-prof)# switchport mode access
switch(config-port-prof)# switchport access vlan 72
switch(config-port-prof)# no shutdown
switch(config-port-prof)# state enabled
switch(config-port-prof)# publish port-profile
switch(config-port-prof)#
```

Configuring Trunk Ports Using Port Profiles

You can use this procedure to 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.
- A trunk port transmits untagged packets for one VLAN plus encapsulated, tagged, packets for multiple VLANs.
- The device supports 802.1Q encapsulation only.
- Be aware that the Cisco Nexus 1000V commands may differ from the Cisco IOS commands.

Procedure

Step 1 switch# **configure terminal**
Enters global configuration mode.

Step 2 switch(config)# **vlan** *vlan-id*
Creates or deletes, and saves in the running configuration, a VLAN or a range or VLANs.

Step 3 switch(config)# **port-profile** [**type ethernet**] *name*
Enters port profile configuration mode for the named port profile. If the port profile does not already exist, it is created using the following characteristics:

- *name*—The port profile name can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V.
- **type**—(Optional) The default port profile type is vEthernet.

Step 4 switch(config-port-prof)# **switchport mode trunk**
Designates that the interfaces are to be used as a trunking ports.

A trunk port transmits untagged packets for the native VLAN and transmits encapsulated, tagged packets for all other VLANs.

Step 5 switch(config-port-prof)# **switchport trunk allowed vlan** {*allowed-vlans* | **add** *add-vlans* | **except** *except-vlans* | **remove** *remove-vlans* | **all** | **none**}
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.

Note If you do not configure allowed VLANs, then the default VLAN 1 is used as the allowed VLAN.

Step 6 switch(config-port-prof)# **no shutdown**
Administratively enables all ports in the profile.

Step 7 switch(config-port-prof)# **state enabled**
Enables the port profile and applies its configuration to the assigned ports.

Step 8 switch(config-port-prof)# **publish port-profile** [*name*]
Pushes the port profile to the VEMs.

Step 9 (Optional) switch(config-port-prof)# **copy running-config startup-config**
Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

This example shows how to configure a trunk port profile.

```
switch# configure terminal
switch(config)# port-profile Trunk_To_Cloud
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# switchport trunk allowed vlan 72,2315-2350
switch(config-port-prof)# no shutdown
switch(config-port-prof)# state enabled
switch(config-port-prof)# publish port-profile
switch(config-port-prof)# copy running-config startup-config
```

Creating a vEthernet Port Profile

Before You Begin

- You are logged in to the CLI in EXEC mode.
- The template profiles should not have the **switchport mode access vlan** command configured. If the command is configured, the configuration is not applied.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile type vethernet** *profile-name*	Enters port profile configuration mode for the named vEthernet port profile. If the port profile does not already exist, it is created. Once configured, the type cannot be changed. Defining a vEthernet type port profile allows the port profile to be used for virtual (vEthernet) ports.
Step 3	switch(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.
Step 4	switch(config-port-prof)# **state enabled**	Enables the port profile and applies its configuration to the assigned ports.
Step 5	switch(config-port-prof)# **publish port-profile** [*name*]	Pushes the port profile to the VEMs as well as to the OpenStack controller.
Step 6	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 enable a port profile.

```
switch# configure terminal
switch(config)# port-profile type vethernet AccessProf
switch(config-port-prof)# state enabled
switch(config-port-prof)# no shut
switch(config-port-prof)# publish port-profile
switch(config-port-prof)# copy running-config startup-config
```

Configuring a Trunk Policy Profile for a vEthernet Port

You can use this procedure to configure a trunk policy profile for a vEthernet port.

Note Native VLAN configuration in the trunk policy profile for vEthernet ports is not supported.

Procedure

Step 1 switch# **configure terminal**
Enters global configuration mode.

Step 2 switch(config)# **port-profile** [**type vethernet**] *TrunkProfile*

Step 3 switch(config-port-prof)# **switchport mode trunk**
Designates that the interfaces are to be used as a trunking ports.

A trunk port transmits untagged packets for the native VLAN and transmits encapsulated, tagged packets for all other VLANs.

Step 4 switch(config-port-prof)# **no shutdown**
Enables all ports in the port profile.

Step 5 switch(config-port-prof)# **state enabled**
Enables the port profile and applies its configuration to the assigned ports.

Step 6 switch(config-port-prof)# **publish port-profile**
Pushes the port profile to the VEMs

Step 7 (Optional) switch(config-port-prof)# **copy running-config startup-config**
Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

Verifying the Port Profile Configuration

Use one of the following commands to verify the configuration:

- **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]
- **show port-profile virtual usage** [**name** *profile-name*]
- **show running-config port-profile** [*prof-name*]
- **show interface virtual nsm**

For detailed information about the command output, see the *Cisco Nexus 1000V Command Reference*.

show port profile

```
switch# show port-profile

port-profile DEFAULT_DATA_VNIC1
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  switchport mode access
  switchport access vlan 2170
  no shutdown
 evaluated config attributes:
  switchport mode access
  switchport access vlan 2170
  no shutdown
 assigned interfaces:
 port-group:
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vservice: no
 port-profile role: none
 port-binding: static

port-profile DEFAULT_DATA_VNIC2
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  switchport mode access
  switchport access vlan 2167
  no shutdown
 evaluated config attributes:
  switchport mode access
  switchport access vlan 2167
  no shutdown
 assigned interfaces:
 port-group:
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vservice: no
 port-profile role: none
 port-binding: static

port-profile DEFAULT_DATA_VNIC3
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  switchport mode access
  switchport access vlan 2169
  no shutdown
 evaluated config attributes:
  switchport mode access
  switchport access vlan 2169
  no shutdown
 assigned interfaces:
 port-group:
 system vlans: none
```

```
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vservice: no
 port-profile role: none
 port-binding: static

port-profile hsrp-1
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  switchport mode trunk
  disable-loop-detection hsrp
  no shutdown
 evaluated config attributes:
  switchport mode trunk
  disable-loop-detection hsrp
  no shutdown
 assigned interfaces:
 port-group:
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vservice: no
 port-profile role: none
 port-binding: static

port-profile uplink_sys
 type: Ethernet
 description:
 status: enabled
 max-ports: 512
 min-ports: 1
 inherit:
 config attributes:
  switchport mode trunk
  switchport trunk allowed vlan 2167-2170
  no shutdown
 evaluated config attributes:
  switchport mode trunk
  switchport trunk allowed vlan 2167-2170
  no shutdown
 assigned interfaces:
 port-group:
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vservice: no
 port-profile role: none
 port-binding: static

port-profile uplink_sys_pc
 type: Ethernet
 description:
 status: enabled
 max-ports: 512
 min-ports: 1
 inherit:
 config attributes:
  switchport mode trunk
  switchport trunk allowed vlan 2167
  channel-group auto mode active
  no shutdown
 evaluated config attributes:
  switchport mode trunk
  switchport trunk allowed vlan 2167
  channel-group auto mode active
```

```
  no shutdown
 assigned interfaces:
 port-group:
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vservice: no
 port-profile role: none
 port-binding: static

port-profile vm_access_sys
 type: Vethernet
 description:
 status: disabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  switchport mode access
 evaluated config attributes:
  switchport mode access
 assigned interfaces:
 port-group:
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vservice: no
 port-profile role: none
 port-binding: static

switch#
```

show port-profile name UpLinkProfile3

```
switch# show port-profile name UpLinkProfile3
port-profile UpLinkProfile3
  description:
  type: vethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  publish:
  max ports: 32
  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#
```

show port-profile brief

```
switch# show port-profile brief
--------------------------------------------------------------------------------
-
Port                          Profile    Profile  Conf   Eval   Assigned  Child

Profile                       Type       State    Items  Items  Intfs     Profs

--------------------------------------------------------------------------------
-
DEFAULT_DATA_VNIC1            Vethernet  1        3      3      0         0

DEFAULT_DATA_VNIC2            Vethernet  1        3      3      0         0
```

```
DEFAULT_DATA_VNIC3            Vethernet  1        3      3      0          0

hsrp-1                        Vethernet  1        3      3      0          0

mx-nlb                        Vethernet  0        0      0      0          0

N1K_Cloud_Default_Trunk       Vethernet  1        2      2      0          0

uplink_sys                    Ethernet   1        3      3      0          0

uplink_sys_pc                 Ethernet   1        4      4      0          0

vm_access_sys                 Vethernet  0        1      1      0          0

vrrp-1                        Vethernet  1        3      3      0          0

--------------------------------------------------------------------------------
-
Profile    Assigned  Total  Sys    Parent Child  UsedBy
Type       Intfs     Prfls  Prfls  Prfls  Prfls  Prfls
--------------------------------------------------------------------------------
-
Vethernet  0         10     0      12     0      0
Ethernet   0         2      0      3      0      0
switch#
```

show port-profile virtual usage

```
switch# show port-profile virtual usage
--------------------------------------------------------------------------------
Port Profile               Port        Adapter        Owner
--------------------------------------------------------------------------------
PVLAN_Macpin               Po2
                           Po4
                           Po6
                           Po8
                           Po9
                           Eth3/1      vmnic0         NODE-135
                           Eth3/2      vmnic1         NODE-135
                           Eth4/1      vmnic0         NODE-137
                           Eth4/2      vmnic1         NODE-137
                           Eth5/2      vmnic1         NODE-139
                           Eth5/3      vmnic2         NODE-139
                           Eth6/1      vmnic0         NODE-141
                           Eth6/2      vmnic1         NODE-141
                           Eth7/1      vmnic0         NODE-UCS-158
                           Eth7/2      vmnic1         NODE-UCS-158
dynpp_03ac7d00-933d-4fc6-8 Veth1                      LINUX-RHEL-01
                           Veth2                      LINUX-RHEL-02
                           Veth3                      LINUX-RHEL-03
                           Veth4                      LINUX-RHEL-04
                           Veth5                      LINUX-RHEL-05
                           Veth6                      LINUX-RHEL-06
                           Veth7                      LINUX-RHEL-07
switch#
```

show port-profile expand-interface name PVLAN_Macpin

```
switch# show port-profile expand-interface name  PVLAN_Macpin

port-profile lacp-uplink
port-channel2
  switchport mode trunk
  switchport trunk allowed vlan 2000-2200
  channel-group auto mode active
  no shutdown
port-channel4
  switchport mode trunk
  switchport trunk allowed vlan 2000-2200
  channel-group auto mode active
```

```
  no shutdown
Ethernet3/1
  switchport mode trunk
  switchport trunk allowed vlan 2000-2200
  channel-group auto mode active
  no shutdown
Ethernet3/3
  switchport mode trunk
  switchport trunk allowed vlan 2000-2200
  channel-group auto mode active
  no shutdown
Ethernet4/2
  switchport mode trunk
  switchport trunk allowed vlan 2000-2200
  channel-group auto mode active
  no shutdown
Ethernet4/5
  switchport mode trunk
  switchport trunk allowed vlan 2000-2200
  channel-group auto mode active
  no shutdown

switch#
```

show port-profile expand-interface

```
switch# show port-profile expand-interface
port-profile DATA-Lacp
 port-channel3
  switchport mode trunk
  switchport trunk allowed vlan 150,205,207,209,211,213,215,217,219,221
  switchport trunk allowed vlan add 223,225,227,229,231,233,235,237,239
  switchport trunk allowed vlan add 241,243,245,247,249,251,253,255,257
  switchport trunk allowed vlan add 261-263,265,267,269,271,273,275,277
  switchport trunk allowed vlan add 281,283,285,287,289,291,293,295,297
  switchport trunk allowed vlan add 299
  channel-group auto mode active
  no shutdown
 port-channel5
  switchport mode trunk
  switchport trunk allowed vlan 150,205,207,209,211,213,215,217,219,221
  switchport trunk allowed vlan add 223,225,227,229,231,233,235,237,239
  switchport trunk allowed vlan add 241,243,245,247,249,251,253,255,257
  switchport trunk allowed vlan add 261-263,265,267,269,271,273,275,277
  switchport trunk allowed vlan add 281,283,285,287,289,291,293,295,297
  switchport trunk allowed vlan add 299
  channel-group auto mode active
  no shutdown
Ethernet4/3
  switchport mode trunk
  switchport trunk allowed vlan 150,205,207,209,211,213,215,217,219,221
  switchport trunk allowed vlan add 223,225,227,229,231,233,235,237,239
switch#
```

show port-profile expand-interface name

```
switch# show port-profile expand-interface name
vmn_f58d3545-a0a1-4441-8b7e-1a7c8339524b_0200362d-0d69-44bc-8f2d-40685f474ddf
port-profile vmn_f58d3545-a0a1-4441-8b7e-1a7c8339524b_0200362d-0d69-44bc-8f2d-40685f474ddf
Vethernet33
  switchport mode access
  switchport access vlan 63
  no shutdown
Vethernet157
  switchport mode access
  switchport access vlan 63
  no shutdown
```

show running-config port-profile

```
switch# show running-config port-profile

!Command: show running-config port-profile
!Time: Mon Aug 26 09:04:05 2013

version 5.2(1)SK1(1.1)
port-profile default max-ports 32
port-profile default port-binding static
port-profile type vethernet N1K_Cloud_Default_Trunk
  switchport mode trunk
  no shutdown
  guid 51e1095a-61ea-50b5-9f3c-19842dcff6e7
  max-ports 64
  description Port Profile created for Nexus 1000V internal usage. Do not use.
  state enabled
port-profile type ethernet uplink_sys
  switchport mode trunk
  switchport trunk allowed vlan 2167-2170
  no shutdown
  guid 53502d18-9ffb-411a-b665-d830081136e5
  max-ports 512
  state enabled
port-profile type ethernet uplink_sys_pc
  switchport mode trunk
  switchport trunk allowed vlan 2167
  channel-group auto mode active
  no shutdown
  guid 7aa26801-1e00-2684-97ec-a7cc1a4615af
  max-ports 512
  state enabled
port-profile type vethernet vm_access_sys
  switchport mode access
  guid 78dc356e-1fe5-7c72-8c2c-6286065720a8
port-profile type vethernet DEFAULT_DATA_VNIC1
  switchport mode access
  switchport access vlan 2170
  no shutdown
  guid 5cb014fe-3d4f-014a-b673-869700f70425
  state enabled
port-profile type vethernet DEFAULT_DATA_VNIC2
  switchport mode access
  switchport access vlan 2167
  no shutdown
  guid 42dbc174-30ec-2ab7-8796-c92e15ea4167
  state enabled
port-profile type vethernet DEFAULT_DATA_VNIC3
  switchport mode access
  switchport access vlan 2169
  no shutdown
  guid 090dc703-caca-102c-869a-86e433531d77
  state enabled
port-profile type vethernet mx-nlb
  guid 2505614c-2107-5f97-9f21-45d70b57aa3e
port-profile type vethernet hsrp-1
  switchport mode trunk
  disable-loop-detection hsrp
  no shutdown
  guid 6d2b8903-94c5-2e9a-923d-182408301feb
  state enabled
port-profile type vethernet vrrp-1
  disable-loop-detection vrrp
  switchport mode trunk
  no shutdown
  guid 3262b6ec-1333-2665-bc78-37a31ea6a71e
  state enabled

switch#
```

Enabling a Port Profile

You enable a port profile to apply its configuration to the assigned port.

Before You Begin

- You are logged in to the CLI in EXEC mode.
- You have already created the port profile.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**vethernet**}] *profile-name*	Enters port profile configuration mode for the named vEthernet port profile.
Step 3	switch(config-port-prof)# **state enabled**	Enables the port profile and applies its configuration to the assigned ports.
Step 4	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	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 enable a port profile.

```
switch# configure terminal
switch(config)# port-profile AccessProf
switch(config-port-prof)# state enabled
switch(config-port-prof)# show port-profile name AccessProf
port-profile AccessProf
  description: allaccess4
  status: enabled
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)#
```

Publishing a Port Profile

You publish an Ethernet port profile to push its configuration to the VEMs. You publish a vEthernet port profile to publish its configuration to the OpenStack controllerwhere it can be created and implemented in a virtual network.

Before You Begin

- You are logged in to the CLI in EXEC mode.
- You have already created and enabled the port profile.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**vethernet**}] *profile-name*	Enters port profile configuration mode for the named port profile.
Step 3	switch(config-port-prof)# **publish port-profile** [*name*]	If you are publishing an Ethernet port profile, the port profile configuration is pushed to the VEMs. If you are publishing a vEthernet port profile, the port profile configuration is published to the OpenStack controller.
Step 4		
Step 5	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	Displays the configuration for verification.
Step 6	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 publish a port profile.

```
switch# configure terminal
switch(config)# port-profile AccessProf
switch(config-port-prof)# publish port-profile
switch(config-port-prof)#
```

Removing a Port Profile

Before You Begin

- You are logged in to the CLI in EXEC mode.

- You must remove all associations to the vEthernet port profile in the OpenStack controller before removing the port profile from the VSM.

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **show interface virtual nsm**	(Optional) Shows policy port profile usage for all interfaces. **Note** You cannot remove a port profile if there are active interfaces associated with it.
Step 2	switch# **configure terminal**	Enters global configuration mode.
Step 3	switch(config)# **no port-profile** *profile_name*	Removes the port profile configuration and operational settings. When a port profile is removed from the VSM, you have to refresh the extension manager from the OpenStack controller to remove the port profile from the OpenStack controller. If the extension manager from the OpenStack controller is not refreshed, the profile is displayed as Marked for Deletion in the OpenStack controller.
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.

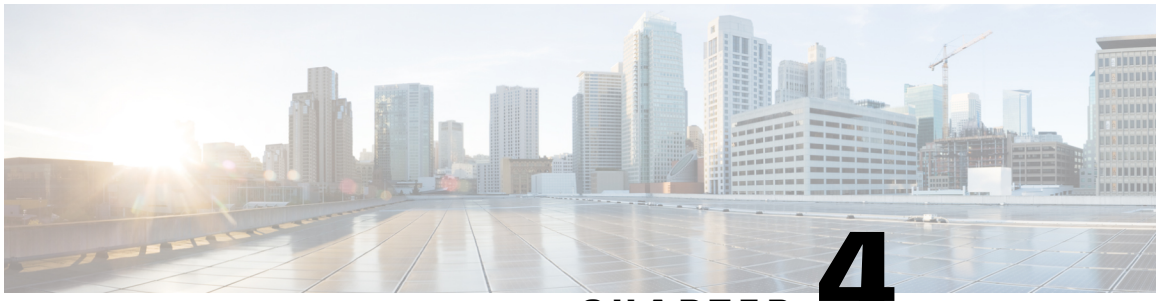
This example shows how to remove a port profile:

```
switch# show port-profile virtual usage
--------------------------------------------------------------------------------
Port Profile               Port        Adapter        Owner
--------------------------------------------------------------------------------
trunk-uplink               Po1
                           Po3
                           Po5
                           Eth3/6      eth3           site1-uvem
                           Eth4/6      eth3           site2-uvem
                           Eth5/1      eth3           site3-uvem
                           Eth5/4      eth2           site3-uvem
                           Eth5/6      eth4           site3-uvem
vm-access-vlan-1601        Veth14      vnet6          85badf15-244d-4719-a2da-8
                           Veth73      vnet2          85badf15-244d-4719-a2da-8
                           Veth99      vnet4          85badf15-244d-4719-a2da-8
vm-access-vlan-1701        Veth3       vnet1          85badf15-244d-4719-a2da-8
                           Veth8       vnet6          85badf15-244d-4719-a2da-8
                           Veth11      vnet8          85badf15-244d-4719-a2da-8
                           Veth15      vnet1          85badf15-244d-4719-a2da-8
                           Veth125     vnet1          85badf15-244d-4719-a2da-8
                           Veth171     vnet104        85badf15-244d-4719-a2da-8
vm-access-vlan-1801        Veth9       vnet7          85badf15-244d-4719-a2da-8
                           Veth20      vnet9          85badf15-244d-4719-a2da-8
1G-uplink                  Eth3/4      eth5           site1-uvem
                           Eth4/4      eth5           site2-uvem
                           Eth5/5      eth5           site3-uvem
                           Eth6/1      eth1           test-vem1
                           Eth7/1      eth1           test-vem2
                           Eth8/1      eth1           test-vem3
                           Eth9/1      eth1           test-vem4
                           Eth10/1     eth1           test-vem5
```

```
                         Eth11/1     eth1           test-vem6
                         Eth12/1     eth1           test-vem7
                         Eth13/1     eth1           test-vem8
                         Eth14/1     eth1           test-vem101
                         Eth15/1     eth1           test-vem102
                         Eth16/1     eth1           test-vem103
                         Eth17/1     eth1           test-vem104
                         Eth18/1     eth1           test-vem105
vm-access-vlan-2000      Veth4       vnet2          85badf15-244d-4719-a2da-8
                         Veth18      vnet2          85badf15-244d-4719-a2da-8
vm-access-vlan-2001      Veth5       vnet3          85badf15-244d-4719-a2da-8
                         Veth19      vnet3          85badf15-244d-4719-a2da-8
vm-access-vlan-1611      Veth17      vnet7          85badf15-244d-4719-a2da-8
                         Veth22      vnet14         85badf15-244d-4719-a2da-8
                         Veth173     vnet3          85badf15-244d-4719-a2da-8
lacp-uplink              Po2
                         Po4
                         Eth3/1      eth4           site1-uvem
                         Eth3/3      eth2           site1-uvem
                         Eth4/2      eth4           site2-uvem
                         Eth4/5      eth2           site2-uvem
vm-access-vlan-2100      Veth6       vnet4          85badf15-244d-4719-a2da-8
                         Veth13      vnet4          85badf15-244d-4719-a2da-8
10G-uplink               Eth3/5      eth0           site1-uvem
                         Eth4/3      eth0           site2-uvem
                         Eth5/3      eth0           site3-uvem
vm-access-vlan-2         Veth1       vnet0          85badf15-244d-4719-a2da-8
                         Veth12      vnet0          85badf15-244d-4719-a2da-8
                         Veth31      vnet0          85badf15-244d-4719-a2da-8
vm-access-vlan-1602      Veth126     vnet5          85badf15-244d-4719-a2da-8
vm-access-vlan-1603      Veth127     vnet6          85badf15-244d-4719-a2da-8
vm-access-vlan-1604      Veth169     vnet7          85badf15-244d-4719-a2da-8
.
.
.
vm-access-vlan-1747      Veth151     vnet150        85badf15-244d-4719-a2da-8
vm-access-vlan-1748      Veth134     vnet151        85badf15-244d-4719-a2da-8
vm-access-vlan-1749      Veth107     vnet152        85badf15-244d-4719-a2da-8
1G-VXLAN-uplink          Eth3/2      eth6           site1-uvem
                         Eth4/1      eth6           site2-uvem
                         Eth5/2      eth6           site3-uvem
vm-access-vlan-1799      Veth2       vxlan-nic0
                         Veth10      vxlan-nic0
bd1                      Veth7       vnet5          85badf15-244d-4719-a2da-8
                         Veth16      vnet5          85badf15-244d-4719-a2da-8
mcast-access-port        Veth177     vnet8          85badf15-244d-4719-a2da-8
                         Veth178     vnet10         85badf15-244d-4719-a2da-8
                         Veth179     vnet156        85badf15-244d-4719-a2da-8
                         Veth180     vnet9          020b89af-9fba-0118-0013-0
switch# configure terminal
switch(config)# no port-profile trunk-uplink
switch(config)# copy running-config startup-config
switch(config)#
```

Feature History for Port Profiles

Feature Name	Releases	Feature Information
Port profiles	Release 5.2(1)SK1(2.1)	This feature was introduced.



CHAPTER 4

Configuring Port Channels Using Port Profiles

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
- IP address (v4 and v6)
- MAC address
- Spanning Tree Protocol
- Network Access Control
- Service policy
- Access control lists (ACLs)

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

- Description
- CDP
- 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.

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

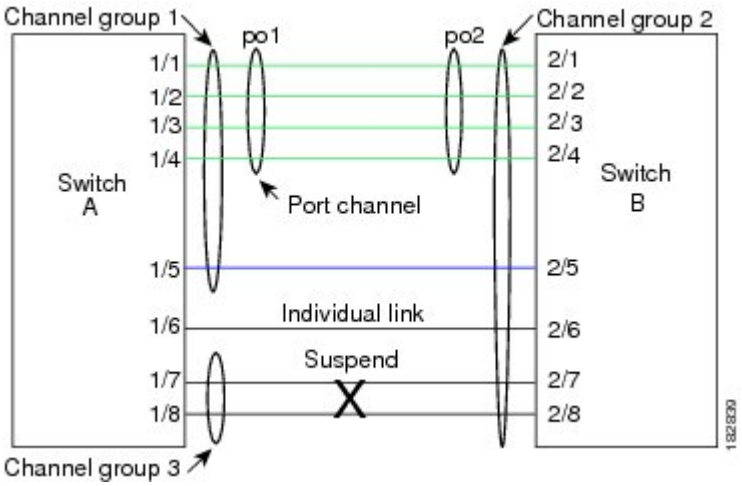
LACP

The Link Aggregation Control Protocol (LACP) allows you to configure interfaces into a port channel. 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 1: Individual Links Combined into a Port Channel



VEM Management of LACP

LACP is offloaded to VEM from the VSM 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 3: 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 4: 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.

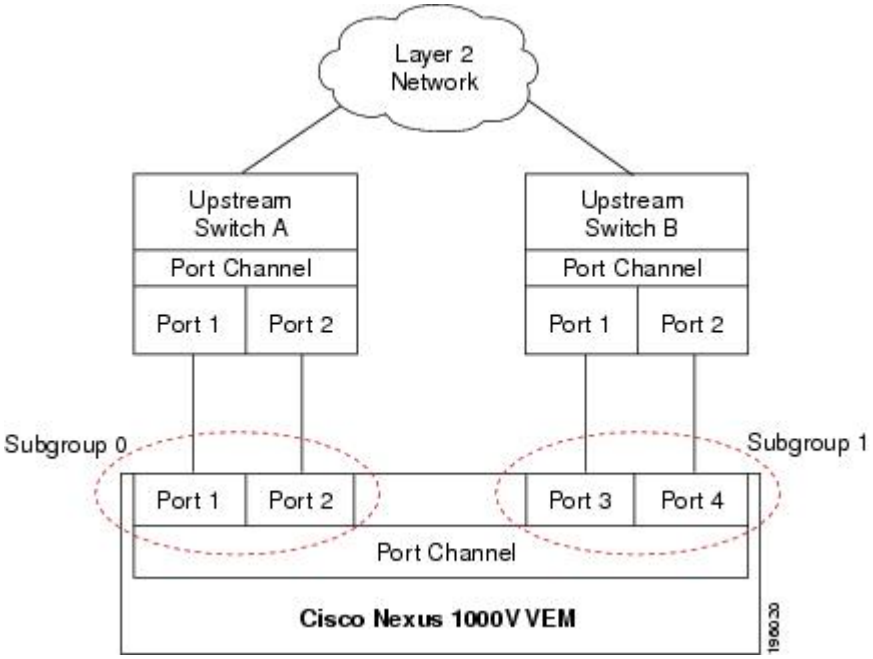
When multiple uplinks are attached to the same subgroup, you must configure the upstream switch in a port channel with the links bundled together. The port channel must also be configured with the **channel-group auto mode on** (active and passive modes use LACP).

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



Subgroup Creation

If the virtual port channel host mode (vPC-HM) type is configured for sub-group manual, you must manually create subgroups. Otherwise, the switch creates the subgroups automatically.

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

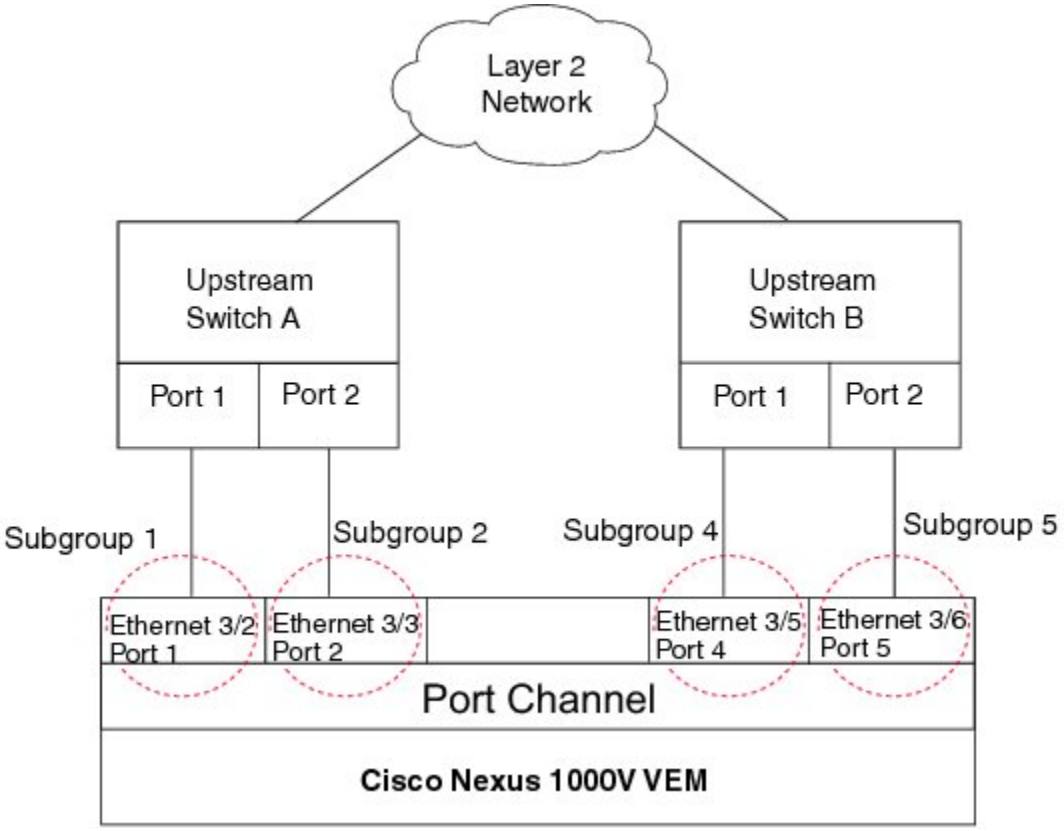
If you are connecting to multiple upstream switches that do not support port channels, MAC pinning is the preferred configuration. MAC pinning divides the uplinks from your server into standalone links and pins the MAC addresses to those links in a round-robin method to ensure that the MAC address of 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 3: Using MAC Pinning to Connect a Port Channel to Multiple Upstream Switches



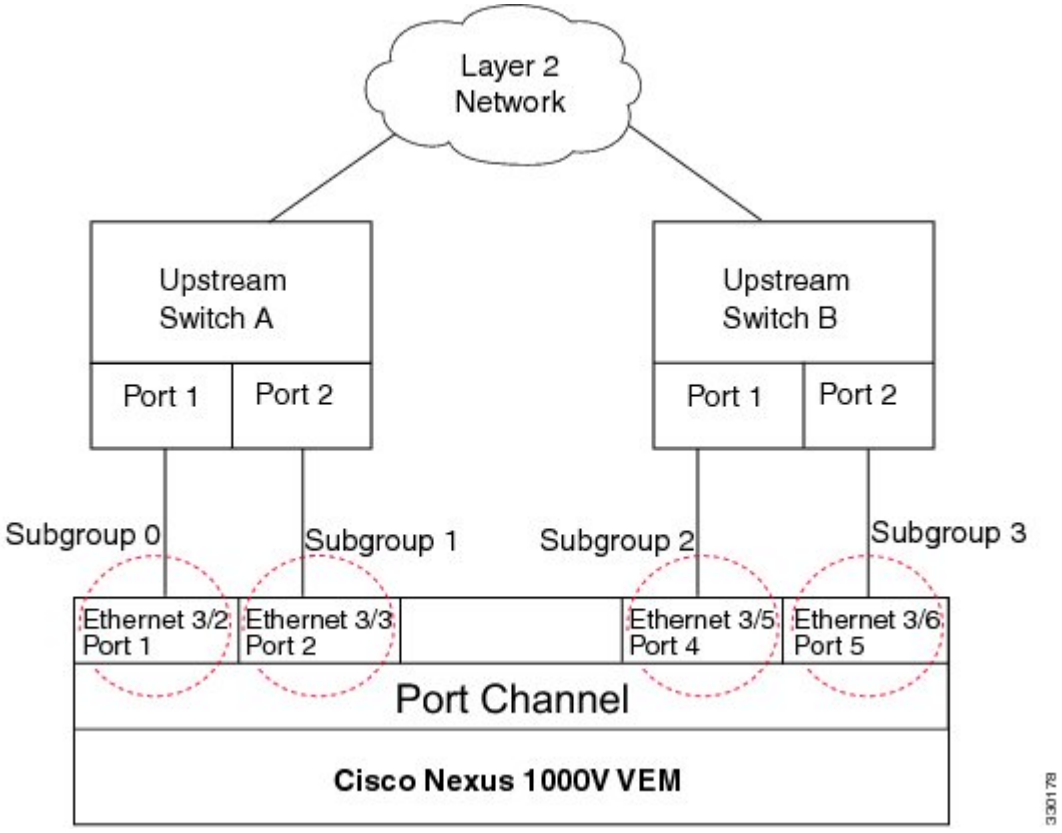
MAC Pinning Relative

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



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.
- 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. Similarly, 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 a source or destination SPAN port.
- In order to support LACP when inband/AIPC are also carried over the link, you must configure the following commands on the ports connected to the hypervisor 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, then 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. It is not supported to 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 may 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*.
- When you create a port channel, an associated channel group is automatically created.
- If LACP support is required for the port channel, then the LACP feature must be enabled before you can configure it.
- When the LACP feature is enabled, it is placed in the offload mode by default, and you cannot disable this mode.

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.
Step 2 Connect to multiple upstream switches. SeeConnecting to Multiple Upstream Switches.
Step 3 Manually configure interface subgroups. See Manually Configuring Interface Subgroups.
Step 4 Pin a vEthernet interface to a subgroup. See Pinning a vEthernet Interface to a Subgroup.
Step 5 Pin a control or packet VLAN to a subgroup. See Pinning a Control or Packet VLAN to a Subgroup.

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.

Before You Begin

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 OpenStack Horizon Server, the corresponding port group can be selected and assigned to physical ports (PNICs).
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 and port profile is added to the same port channel. A separate port channel is created for each module using that port profile. • **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 or if the members of the port channel are connected to two or more upstream switches. • **relative**—Specifies that the subgroup numbering begins at zero and continues numbering the subgroups consecutively.
Step 4	switch(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.
Step 5	switch(config-port-prof)# **state enabled**	Enables the port profile and applies its configuration to the assigned ports.
Step 6	switch(config-port-prof)# **publish port-profile** [*name*]	Pushes the port profile to the VEMs as well as to the OpenStack controller.

	Command or Action	Purpose
Step 7	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *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 configure a port channel that connects to one upstream switch:

```
switch# configure terminal
switch(config)# port-profile type ethernet UplinkProf
switch(config-port-prof)# channel-group auto mode on
switch(config-port-prof)# no shutdown
switch(config-port-prof)# state enabled
switch(config-port-prof)# publish port-profile
switch(config-port-prof)# show port-profile name UplinkProf
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, then 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 OpenStack Horizon Server, the corresponding port group can be selected and assigned to physical ports (PNICs).
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 and port profile is added to the same port channel. A separate port channel is created for each module using that port profile. The following options are also defined: • **mode**—Sets the port channel mode to on. • **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.

	Command or Action	Purpose
		◦ **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
```

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.

Before You Begin

- You are logged in to the CLI in EXEC mode.
- You 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.

	Command or Action	Purpose
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.
- The existing port profile must be a system port profile.
- 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 Port Channel Types in a Port Profile

To move member ports to another port profile, you must tear down the existing port channel, and then recreate it.

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** [**sub-group** { **cdp** | **manual**}] [**mac-pinning** [**relative**]]
- 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 Bring the host out of maintenance mode.

Step 9 Migrate the virtual machines back to this host.

Step 10 To 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 11 Create the port channel type that you want in the upstream switch. See Creating a Port Profile for a Port Channel.

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). **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
!Command: show running-config interface Vethernet1
!Time: Wed Jul 17 06:48:47 2013

version 5.2(1)SK1(1.1)

interface Vethernet1
  inherit port-profile DEFAULT_DATA_VNIC1
  description 51c91ae5, vnet24
  pinning id 0
  dvport uuid "51c91ae9-4dff-dff2-ff2d-572657e64757"

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.
- We recommend that you shut down the port channel from the upstream switch, not from the local CCisco Nexus 1000V switch.

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**. 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.

	Command or Action	Purpose
Step 2	switch(config)# **interface port-channel** *channel-number*	Places you into 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 string, the description can be up to 80 alphanumeric characters. **Note** You do not need to use quotations around descriptions that include spaces.
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.
- Module-based load balancing takes precedence over device-based load balancing.
- The default load balancing method is the source MAC address.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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 an uplink port profile for LACP.

Before You Begin

- Log in to the CLI in EXEC mode.
- 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.
- 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** *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. 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.
Step 4	switch(config-if)# **channel-group auto mode** {**active** | **passive**}	Creates a port channel group in one of the following modes: • **active—**When the LACP feature is enabled, LACP is enabled on the specified interface.

	Command or Action	Purpose
		• **passive**—When the LACP feature is enabled, LACP is enabled on the specified interface only if an LACP device is detected.
Step 5	switch(config-port-prof)# **switchport mode** {**access** | **trunk**}	Designates how the interfaces are to be used. Allowable port modes: • access • 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)# **port-group** [**pg_name**]	Designates the port profile as a port group.
Step 9	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 an LACP port profile for a port channel:

```
switch# configure terminal
switch(config)# feature lacp
switch(config-if)# port-profile type ethernet system-uplink
```

```
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
```

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 port-channel compatibility-parameters	Displays the parameters that must be the same among the member ports in order to join a port channel.
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.

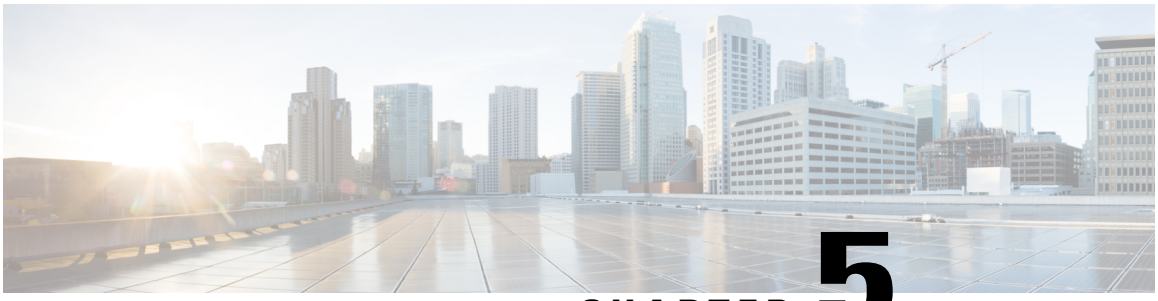
Monitoring Port Channels

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 on the number of error packets.
show lacp counters [**interface port-channel** *channel-number*]	Displays information about LACP statistics.

Feature History for Port Channels

Feature Name	Releases	Feature Information
Port Channels	Release 5.2(1)SK1(2.1)	This feature was introduced.



CHAPTER 5

Configuring a Private VLAN in a Port Profile

This chapter contains the following sections:

Information About Private VLANs

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 Layer 2 isolated. In turn, the use of larger subnets reduces address management overhead.

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

Configuring a Port Profile as a Private VLAN

Before You Begin

- You are logged in to the CLI in EXEC mode.
- You know the VLAN IDs for both the primary and secondary VLAN in the private VLAN pair.
- You know whether this private VLAN inherits its configuration.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*	Enters port profile configuration mode for the named port profile. If the port profile does not already exist, it is created using the following characteristics: • *name*—The port profile name can be up to 80 alphanumeric characters and must be unique for each port profile on the Cisco Nexus 1000V. • **type**—(Optional) The port profile type can be Ethernet or vEthernet. Once configured, the type cannot be changed. The default is the vEthernet type. Defining a port profile type as Ethernet allows the port profile to be used for physical (Ethernet) ports. In the OpenStack Horizon 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 3	switch(config-port-prof)# **switchport mode private-vlan** {**host**| **promiscuous**|**trunk promiscuous**}	Designates the port profile for use as a private VLAN and defines the ports as follows: • **promiscuous**—vEthernet ports that belong to the primary VLAN and communicate with the Layer 3 gateway. Promiscuous ports can communicate with any interface in the PVLAN domain, including those associated with secondary VLANs. • **host**—vEthernet ports that belong to the secondary VLAN as one of the following: ◦ Community PVLAN host port ◦ Isolated PVLAN host port • **trunk promiscuous**—A physical Ethernet trunk port that carries both regular non-PVLAN traffic and PVLAN traffic. When traffic comes from a PVLAN host port, the packet is translated to the primary VLAN packet.
Step 4	switch(config-port-prof)# **switchport private-vlan host-association** *primary-vlan secondary-vlan*	Assigns the primary and secondary VLAN IDs to the port profile and saves this association in the running configuration. • *primary-vlan*—Specifies a primary VLAN ID. You can specify only one primary VLAN ID. • *secondary-vlan*—Specifies the secondary VLAN ID. You can specify only one secondary VLAN ID.

	Command or Action	Purpose
Step 5	switch(config-port-prof)# **switchport private-vlan trunk allowed vlan** *vlan-range*	Sets the allowed VLANs and VLAN IDs when interface is in private-vlan trunking mode.
Step 6	switch(config-port-prof)# **switchport private-vlan mapping** *primary_vlan* [**add** | **remove**] *secondary_vlan*	Maps the primary VLAN ID to the secondary VLAN ID for the port profile. • *primary-vlan*—Specifies a primary VLAN ID. You can specify only one primary VLAN ID. • **add**—Associates the secondary VLAN to the primary VLAN. • **remove**—Clears the association between the secondary VLAN and the primary VLAN. • *secondary-vlan*—Specifies the secondary VLAN ID. You can specify only one secondary VLAN ID.
Step 7	switch(config-port-prof)# **switchport private-vlan mapping trunk** *primary_vlan* [**add** | **remove**] *secondary_vlan*	Designates the primary private VLAN. The range of valid values is 1 to 3967.
Step 8	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [*name profile-name*]	(Optional) Displays the configuration for verification.
Step 9	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.

These examples show different ways that port profiles can be configured as private VLANs:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# port-profile type vethernet pv154
switch(config-port-prof)# publish port-profile
switch(config-port-prof)# switchport mode private-vlan host
switch(config-port-prof)# switchport private-vlan host-association 153 154
switch(config-port-prof)# no shutdown
switch(config-port-prof)# state enabled
switch(config-port-prof)# show run port-profile pv154

!Command: show running-config port-profile pv154
!Time: Wed Nov  5 11:48:03 2014

version 5.2(1)SV3(2.1)
port-profile type vethernet pv154
  publish port-profile
  switchport mode private-vlan host
  switchport private-vlan host-association 153 154
  no shutdown
  max-ports 1024
  state enabled
```

```
switch(config-port-prof)# port-profile type vethernet pvprom
switch(config-port-prof)# publish port-profile
switch(config-port-prof)# switchport mode private-vlan promiscuous
switch(config-port-prof)# switchport private-vlan mapping 153 154-155
switch(config-port-prof)# no shutdown
switch(config-port-prof)# state enabled
switch(config-port-prof)# show run port-profile p-c-154

!Command: show running-config port-profile p-c-154
!Time: Wed Nov 5 11:48:03 2014

version 5.2(1)SV3(2.1)
port-profile type vethernet p-c-154
  switchport mode private-vlan host
  switchport private-vlan host-association 153 154
  no shutdown
  guid b92d5f70-50ad-49e9-99a4-2b13fba802ff
  state enabled
  publish port-profile
```

Feature History for Private VLAN Port Profiles

Feature Name	Release	Feature Information
Private VLAN Port Profiles	5.2(1)SK3(2.1)	This feature was introduced.