Configuring Adapter-FEX

The chapter contains the following sections:

Adding a Server with a Server Network Adapter

The following sections describe how to configure a server with a server network adapter for Adapter-FEX.

Enabling Switch for Adapter-FEX

Enable the Adapter-FEX feature on each of the switches that the server is connecting to.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**install feature-set virtualization**
	Example:
	`switch(config)# install feature-set virtualization`
	`switch(config)#`
		Installs the Cisco Virtual Machine feature set on the switch.

	Command or Action	Purpose

Step 3	**feature-set virtualization**
	Example:
	`switch(config)# feature-set virtualization`
	`switch(config)#`
		Enables the Cisco Virtual Machine features on the switch.

Step 4	**vethernet auto-create**
	Example:
	`switch(config)# vethernet auto-create`
	`switch(config)#`
		Enables the automatic creation of virtual Ethernet interfaces on the switch.

Example:

```
switch# configure terminal
switch(config)# install feature-set virtualization
switch(config)# feature-set virtualization
switch(config)# vethernet auto-create
switch(config)#
```

Configuring the Adapter FEX Feature

Configure the switch to support Adapter-FEX.

Before You Begin

Enable the Adapter FEX feature by enabling the virtualization feature set.

Enable network interface virtualization (NIV) on the physical server adapters using the Cisco Integrated Management Controller (CIMC).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)#**port-profile type vethernet** *port-profile_name*	Create port-profiles (type vethernet) that is associated with the vNICs of the adapter. Configure the relevant properties and policies in the port-profile, such as VLAN, bandwidth, quality of service (QoS), and access control lists (ACLs).
Step 3	switch(config-port-profile)#**state enabled**	Enable the vNIC.
Step 4	switch(config-port-profile)#**exit**	Exits the current configuration mode.
Step 5	switch(config)#**interface ethernet** *type /port*	Enters configuration mode for the specified interface.

	Command or Action	Purpose
Step 6	switch(config-if)#**switchport mode vntag**	On the Ethernet interfaces of the switch that the network adapter's uplink interfaces are connected to, configure the switchport for VNTag mode.
Step 7	switch(config-if)#**exit**	Exits the current configuration mode.
Step 8	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 9	Connect the server network adapter uplinks.	Connect the server network adapter uplinks from the server to the switch or FEX.

The following example show how to configure a Cisco Nexus switch with four vNICs by creating one port-profile for each type of vNIC:

```
switch# configure terminal
switch(config)# port-profile type vethernet user_data1
    switchport trunk allowed vlan 2-100
    switchport trunk native vlan 2
    switchport mode trunk
switch(config)# state enabled

switch(config)# port-profile type vethernet user_data2
    switchport trunk allowed vlan 2-100
    switchport trunk native vlan 2
    switchport mode trunk
switch(config)# state enabled

switch(config)# port-profile type vethernet user_management
    switchport access vlan 1
switch(config)# state enabled

switch(config)# port-profile type vethernet user_backup
    switchport mode trunk
    switchport trunk allowed vlan 2-100
    switchport trunk native vlan 2
    mac port access-group mac_acl1
    ip port access-group ip_acl1 in
    ipv6 port traffic-filter ipv6_acl1 in
switch(config)# state enabled

switch(config)# interface ethernet1/5
   description ucs_vic2/0
switch(config-if)# switchport mode vntag
switch(config-if)#
```

Configuring a Server Network Adapter

Using the network adapter configuration utility on the server, enable Network Interface Virtualization (NIV) mode on the network adapter. See the documentation of your specific server network adapter for details. To complete the configuration, you might have to reboot the server to reset the network adapter.

- For the Cisco UCS P81E or Cisco UCS VIC1225 Virtual Interface Card, refer to the Cisco UCS Virtual Interface Card documentation in the *Cisco UCS C-Series Servers Integrated Management Controller CLI Configuration Guide*.
- For the Broadcom BCM57810 Convergence Network Interface Card, refer to your source for Broadcom Corp. documentation.

Server Network Adapter Initial Handshake and Negotiation

When connecting the server network adapter to the enabled VNTag mode Ethernet port on a switch or to a FEX, a handshake is initiated. An exchange of information about NIV capabilities takes place and communications begin in VNTag mode. The switch passes the list of configured port-profiles (type vethernet) down to the adapter. These port-profile names are displayed in the server network adapter's configuration utility as options for selection.

Only port-profile names are passed to the server network adapter. The configurations in the port-profile are not passed to the server network adapter.

If the server network adapter is not connected to the switch, it is still possible to configure the virtual network interface cards (vNICs) on the server. However the port-profile names are not available on the adapter.

Configuring vNICs on Server Network Adapter

Using the network adapter configuration utility on the server, create the appropriate number of vNICs. Create each vNIC with the appropriate properties, such as unique channel number, MAC address, uplink failover properties, or port-profile names.

Note After the initial handshake and negotiation, the list of port-profiles configured on the switch are automatically available on the server adapter. You can associate these port-profile names to the vNICs.

Note For each vNICs or virtual hub bus adapter (vHBAs), configure a unique virtual Ethernet (vEthernet or vEth) interface. For example, vNIC1's virtual ethernet will be vEth1 on both Cisco Nexus switches. Since, each vNIC maps to a single unique vEth it is then synced between both Cisco Nexus switches MAC tables.

Each vNIC has a unique channel number associated with it. A vNIC is identified on the switch by the bind command that associates a physical port and the vNIC's channel number to a vethernet interface. Refer to the documentation of your specific server network adapter for details.

- For the Cisco UCS P81E or Cisco UCS VIC1225 Virtual Interface Card, see the Cisco UCS Virtual Interface Card documentation in the *Cisco UCS C-Series Servers Integrated Management Controller CLI Configuration Guide*.
- For the Broadcom BCM57810 Convergence Network Interface Card, refer to your source for Broadcom Corp. documentation.

When VNtag connectivity is established, only port-profile names (type vethernet) are passed to the server network adapter. The configurations in the port-profile are not passed to the server network adapter.

The port-profiles names are displayed in the server network adapter's configuration utility as options for selection.

Initializing the Server Network Adapter

After you configure the vNICs on the server network adapter, you might have to consider the following to complete the configuration.

- Reboot the server and reset the server network adapter.
- Reload the drivers.

See the documentation for your specific server network adapter for details.

When the configuration is complete, the server network adapter and the switch re-establishes a link and performs the initial handshake and negotiation process. The server network adapter and the switch also establish higher level control plane connectivity using the Virtual Interface Configuration (VIC) protocol.

Note The VIC protocol allows you to provision and manage virtual interfaces on a remote device.

When the VIC protocol connectivity is established, the server network adapter requests that the switch create a Vethernet interface for each vNIC that is configured on the server network adapter. The server network adapter also passes the following attributes over the uplink in addition to the create Vethernet interface request:

- Port-profile name
- Channel number
- Active/standby status

The switch responds by creating a Vethernet interface for each vNIC on the server network adapter and associates the port-profile and channel number to the Vethernet interface.

The server bootup process might be held at the BIOS configuration phase until the Vethernet creation has completed. After the Vethernet is created, the bootup process resumes and the operating system is loaded. See to the documentation for your specific adapter for details.

The global configuration command, **no vethernet auto-create**, stops the Vethernet interfaces from being automatically created. If the switch is not configured with the **vethernet auto-create** command, you must configure the Vethernet interfaces manually with the appropriate binding and port-profile configurations.

Vethernet interfaces created by the switch are numbered automatically as they are created. These vethernet numbers start from 32769. The switch picks the lowest unused number when creating a Vethernet interface.

When you manually create Vetethernet interfaces, you may select any number for the Vethernet. However, as a best practice, you should choose a number that is less that 32678.

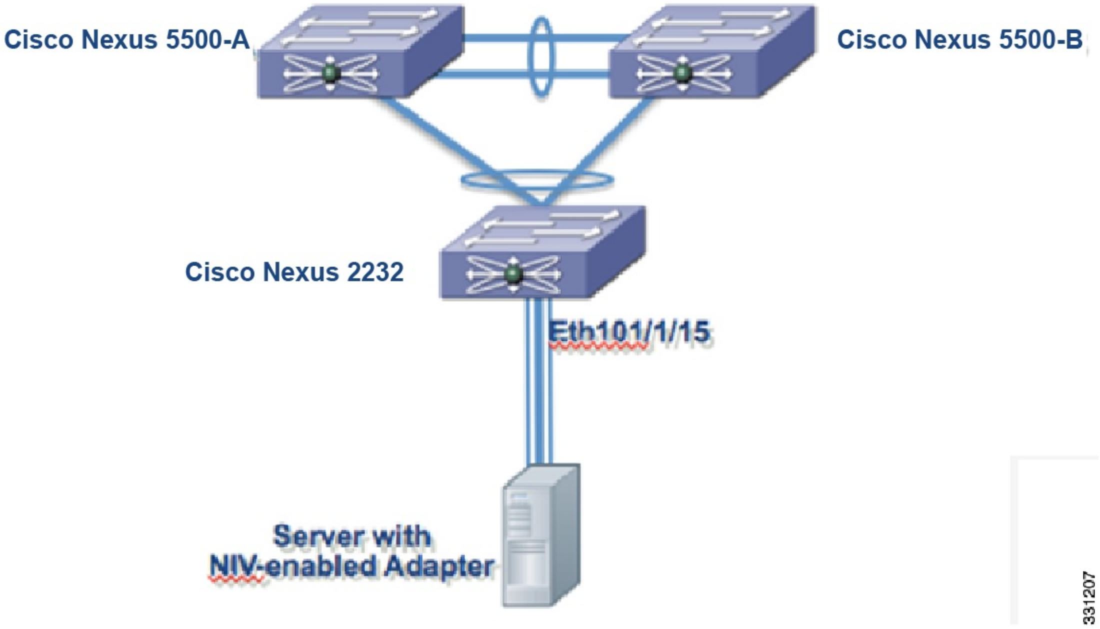
Example of configuring Vethernet interfaces

```
switch# interface vethernet 21
    bind interface ethernet 1/5 channel 1
    inherit port-profile user_data
switch# interface vethernet 22
    bind interface ethernet 1/5 channel 2
    inherit port-profile user_data
switch# interface vethernet 23
    bind interface ethernet 1/5 channel 3
    inherit port-profile user_management
switch# interface vethernet 24
    bind interface ethernet 1/5 channel 4
    inherit port-profile user_backup
```

Cisco Nexus Adapter FEX Initialization Example for a Dual-Homed FEX Topology

The following figure shows a topology with a dual-homed FEX.

Figure 1: Cisco Nexus Adapter FEX Dual-Homed Topology



Vethernet interface initialization results of the Cisco Nexus Swtich-A

```
switch# interface vethernet 21
    bind interface ethernet 101/1/15 channel 1
    inherit port-profile user_data
switch# interface vethernet 22
    bind interface ethernet 101/1/15 channel 2
    inherit port-profile user_data
switch# interface vethernet 23
    bind interface ethernet 101/1/15 channel 3
    inherit port-profile user_management
switch# interface vethernet 24
    bind interface ethernet 101/1/15 channel 4
    inherit port-profile user_backup
```

Vethernet interface initialization results of the Cisco Nexus Switch-B

```
switch# interface vethernet 21
    bind interface ethernet 101/1/15 channel 1
    inherit port-profile user_data
switch# interface vethernet 22
    bind interface ethernet 101/1/15 channel 2
    inherit port-profile user_data
switch# interface vethernet 23
    bind interface ethernet 101/1/15 channel 3
    inherit port-profile user_management
```

```
switch# interface vethernet 24
    bind interface ethernet 101/1/15 channel 4
    inherit port-profile user_backup
```

Note The initialization results are identical for both switches.

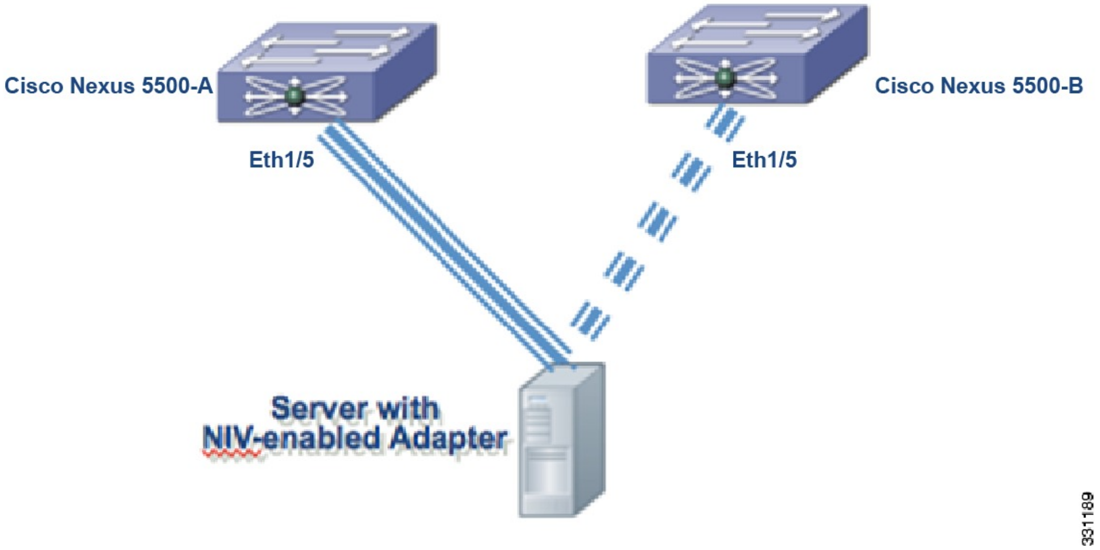
Cisco Nexus Adapter-FEX Initialization Example with Active-Standby Topology

The active-standby topologies described here have one uplink as active and the other uplink as standby. With some server network adapters, you may select the active and standby uplinks per vNIC. In this case, each uplink is an active uplink for a specific vNIC and becomes a standby for the remainder uplinks. For each vNIC, you should choose the active and standby uplinks as a best practice.

Channel numbers for Vethernet interfaces created on the standby uplink device are the same as the channel numbers on the active uplink.

Consider the following figure showing a topology with active-standby uplinks.

Figure 2: Cisco Nexus Adapter FEX Active-Standby Topology



In a virtual port channel (vPC) topology, the Vethernet interfaces that are associated with a single vNIC has the same interface number on the vPC primary and secondary switch.

Assume that the server network adapter in this example has four vNICs (eth0, eth1, eth2, eth3) and two uplinks (uplink_0 and uplink_1) with the configuration described in the following table:

vNIC	Uplink	Port-profile Name	Channel Number
eth0	Uplink_0	user_data	1
eth1	Uplink_1	user_management	2

eth2	Uplink_0	user_data	3
eth3	Uplink_1	user_backup	4

Also assume that uplink_0 is connected to Ethernet 1/5 on Swtich-A and that uplink_1 is connected to Ethernet 1/15 on Switch-B.

In a vPC active-standby topology, Vethernet numbers are allocated by the vPC primary switch. When a Vethernet create request is received, the vPC secondary switch requests the primary switch for an allocation. The **show vpc brief** command displays the vPC role of a switch.

If a server is moved from one port to another port on either the primary vPC switch or the secondary vPC switch, the server network adapter connectivity is not established until the old Vethernet interfaces are deleted.

Vethernet interface initialization results of Cisco Nexus Switch-A

```
switch# interface vethernet 37
    bind interface ethernet 1/5 channel 1
    inherit port-profile user_data
switch# interface vethernet 38
    bind interface ethernet 1/5 channel 2
    inherit port-profile user_data
switch# interface vethernet 39
    bind interface ethernet 1/5 channel 3
    inherit port-profile user_management
switch# interface vethernet 40
    bind interface ethernet 1/5 channel 4
    inherit port-profile user_backup
```

Vethernet interface initialization results of Cisco Nexus Switch-B

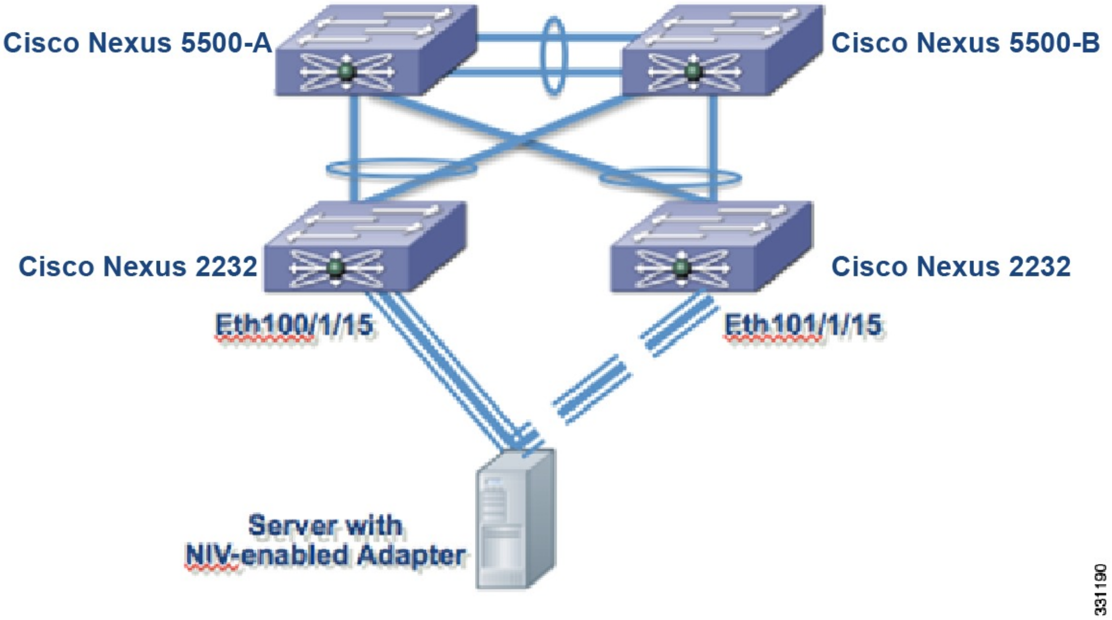
```
switch# interface vethernet 37
    bind interface ethernet 1/5 channel 1
    inherit port-profile user_data
switch# interface vethernet 38
    bind interface ethernet 1/5 channel 2
    inherit port-profile user_data
switch# interface vethernet 39
    bind interface ethernet 1/5 channel 3
    inherit port-profile user_management
switch# interface vethernet 40
    bind interface ethernet 1/5 channel 4
    inherit port-profile user_backup
```

Note The initialization results are identical for both switches.

Cisco Nexus Adapter FEX Initialization Example for a Dual-Homed FEX with Active-Standby Topology

The following figure shows a topology with active-standby uplinks.

Figure 3: Cisco Nexus Adapter FEX Dual-Homed Active-Standby Topology



Assume that the adapter in this example has four vNICs (eth0, eth1, eth2, eth3) and two uplinks (uplink_0 and uplink_1) with the configuration described in the following table:

vNIC	Uplink	Port-profile Name	Channel Number
eth0	Uplink_0	user_data	1
eth1	Uplink_1	user_management	2
eth2	Uplink_0	user_data	3
eth3	Uplink_1	user_backup	4

Also assume that uplink_0 is connected to Ethernet 100/1/15 on and that uplink_1 is connected to Ethernet 101/1/15.

Vethernet interface initialization results of Cisco Nexus Switch-A

```
switch# interface vethernet 37
    bind interface ethernet 100/1/15 channel 1
    bind interface ethernet 101/1/15 channel 1
    inherit port-profile user_data
```

```
switch# interface vethernet 38
    bind interface ethernet 100/1/15 channel 2
    bind interface ethernet 101/1/15 channel 2
    inherit port-profile user_data
switch# interface vethernet 39
    bind interface ethernet 100/1/15 channel 3
    bind interface ethernet 101/1/15 channel 3
    inherit port-profile user_management
switch# iinterface vethernet 40
    bind interface ethernet 100/1/15 channel 4
    bind interface ethernet 101/1/15 channel 4
    inherit port-profile user_backup
```

Vethernet interface initialization results of Cisco Nexus Switch-B

```
switch# interface vethernet 37
    bind interface ethernet 100/1/15 channel 1
    bind interface ethernet 101/1/15 channel 1
    inherit port-profile user_data
switch# interface vethernet 38
    bind interface ethernet 100/1/15 channel 2
    bind interface ethernet 101/1/15 channel 2
    inherit port-profile user_data
switch# interface vethernet 39
    bind interface ethernet 100/1/15 channel 3
    bind interface ethernet 101/1/15 channel 3
    inherit port-profile user_management
switch# interface vethernet 40
    bind interface ethernet 100/1/15 channel 4
    bind interface ethernet 101/1/15 channel 4
    inherit port-profile user_backup
```

Note The initialization results are identical for both switches.

Creating Vethernet Interfaces

The server network adapter makes requests to the switch to create vethernet interfaces one at a time.

Example of results of creating vethernet interface

```
switch# show running-config interface Vethernet32773

interface Vethernet32773
  inherit port-profile sample
  bind interface Ethernet1/20 channel 11

switch# show interface Vethernet32773 detail
vif_index: 29
--------------------------
  veth is bound to interface Ethernet1/20 (0x1a013000)
  priority: 0
  vntag: 52
  status: standby
  channel id: 11
  registered mac info:
      vlan 0 - mac 00:00:00:00:00:00
      vlan 0 - mac 00:22:bd:d6:5f:4b
      vlan 0 - mac ff:ff:ff:ff:ff:ff
```

Saving Vethernet Interfaces in the Startup-config

Vethernet interfaces that are created with a request to create the VIC protocol interface are kept in the running-configuration. When the running-configuration is saved, the Vethernet interfaces and their configurations are saved in the startup-config.

When the switch is reloaded, the switch loads the startup-config and the Vethernet interface is created statically from the startup-config. When the VNTag interface is initialized and the server network adapter makes a request for Vethernet creation, the already existing Vethernet interface is used.

Note You can delete a Vethernet interface from the system with the **no interface vethernet** *veth-id* command. Since vethernet interfaces are saved in the startup-config, a best practice is to save the running-config to the startup-config.

Configuring Vethernet Interfaces Manually

When a server network adapter is connected to a VNTag-mode Ethernet interface on a switch, the Vethernet interfaces are created automatically. However, Vethernet interfaces can also be created manually from the configuration mode with the **interface vethernet** *veth-id* command.

Note When manually configuring Vethernet interfaces, use the **no vethernet auto-create** command to disable the automatic creation of Vethernet interfaces.

When a server network adapter is connected and a request to create a Vethernet is received, the switch checks for a manually configured Vethernet interface that matches the channel number of the Vethernet creation request. If the manually configured Vethernet interface already exists, the manually configured Vethernet interface is used and a new Vethernet interface is not created.

Configuration Conflicts Between Switch and Server Network Adapter

Conflicts with port-profile names might occur between the configuration of a Vethernet interface that was manually configured or one that was saved in the start-up config and the configuration on the server network adapter.

For example, the server network adapter might associate the vNIC with channel number 5 to the port-profile user_backup, but the switch might associate the Vethernet interface bound to channel number 5 to the port-profile user_data.

When a conflict of this kind occurs, the configuration from the server network adapter takes precedence and overwrites the configuration on the switch.

Note When a port-profile is associated to a vNIC, but the port-profile does not exist on the switch, the Vethernet creation and initialization fails.

Configuration Conflicts Between Switches

In a vPC topology, the configuration of a specific Vethernet interface must be identical on both the switches. If there is a mismatch in the configurations on the switches, the resulting configuration conflict causes the Vethernet interfaces to fail.

When a conflict of this kind occurs, you must resolve the conflict manually before the Vethernet interface can be brought up successfully.

Removing a Server with a Server Network Adapter

When a server network adapter is connected to a switch, Vethernet interfaces are created to support the server network adapter. When you want to remove a server network adapter from the topology, disconnecting or powering down the server does not remove these Vethernet interfaces. Stale Vethernet interfaces exist until they are manually deleted.

When the stale Vethernet interfaces are manually deleted, the server that is connected to the server network adapter can be removed from the topology.

Note As a best practice, you should delete unused Vethernet interfaces to conserve switch resources.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no interface vethernet** *veth-id*
	Example:
	`switch(config)# no interface vethernet 1`
	`switch(config)#`
		Removes the Vethernet interface.

Step 3	**copy running-config startup-config** **Example:** `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.

Example of removing Vethernet interfaces

```
switch# configure terminal
switch(config)# no interface vethernet 1
switch(config)# copy running-config startup-config
switch(config)#
```

Relocating a Server to a Different Physical Interface

When a server network adapter is relocated from one physical interface on a switch to another identical switch, new Vethernet interfaces are created and are bound to the new physical interface. The original Vethernet interfaces that were bound to the original physical interfaces exist until they are deleted.

When a new server is connected to the physical interface where the original server was connected, the Vethernet interfaces are reused (assuming that the Vethernet interfaces were not deleted). However, this action might cause a conflict between the configuration on the switch and the configuration on the server network adapter.

Note When a server network adapter is moved from one server to another, the server network adapter configuration, such as the information about vNICs, moves with the server network adapter. The configuration is saved on the server network adapter.

In a vPC topology, Vethernet numbers for a vNIC must be identical on both vPC peers. Moving a server that has a server network adapter from one physical interface to another for one of the vPC peers without first deleting the Vethernet interfaces on that switch, results in new Vethernet interfaces with different numbers. This results in having the two vPC peers to be not synchronized. Vethernet creation failure occurs when the vPC peers are not synchronized. As a best practice, the original Vethernet interfaces should be deleted before the server is connected to the new physical interface.

Procedure

	Command or Action	Purpose
Step 1	**shutdown** **Example:** `switch# shutdown`	Issues shutdown on original physical interface.

Step 2	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 3	**no interface vethernet** *veth-id*
	Example:
	`switch(config)# no interface vethernet 21`
	`switch(config)#`
		Delete each vethernet interface associated with the server.

Step 4	**no switchport mode vntag**
	Example:
	`switch(config-if)# no switchport mode vntag`
	`switch(config)#`
		Unconfigures switchport mode vntag on the original physical interface on both vPC switches.

Step 5	Unplug the server from both vPC switches.	Disconnects server from both vPC switches.

	Command or Action	Purpose
Step 6	Connect the server to the new physical interface.	Connects the server to the new physical interface.
Step 7	Configure the server network adapter.	Configures the server network adapter for the new switch.

Example:

- Perform the following commands:

```
switch# shutdown
switch# configure terminal
switch(config)# no interface vethernet 21
switch(config-if)# no switchport mode vntag
switch(config-if)#
```

- Unplug the server from both vPC switches.
- Connect the server to the new physical interface.
- Configure the server network adapter.

Adding or Deleting Port-Profiles

When a new port-profile of type vethernet is added, it is immediately passed to all connected server network adapters and becomes available. Similarly, when a port-profile is removed, its name is withdrawn from every connected server network adapter.

Note Because only port-profile names are passed to the server network adapter, changing the configuration within a port-profile does not trigger any information to be passed to the server network adapter.

Note When a newly added port-profile is used, such as being associated with an existing or new vNIC, you might need to reboot the server to reset the server network adapter. See the documentation for the your specific adapter for details.

When a port-profile is deleted, all Vethernet interfaces that are associated with it are brought down.

Disabling Adapter-FEX on a Switch

Disable the Adapter-FEX feature on each of the switches that the server is connected to.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no interface vethernet** *veth-id* **Example:** `switch(config)# no interface vethernet 21`	Delete each vethernet interface in the running-config.

Step 3	**no port-profile type vethernet** *port-profile_name*
	Example:
	`switch(config-if)# no port-profile type vethernet user_data`
	`switch(config)#`
		Delete port-profiles (type vethernet) that associate with the vNICs of the adapter.

Step 4	**no switchport mode vntag**
	Example:
	`switch(config-if)# no switchport mode vntag`
	`switch(config)#`
		Unconfigures the switchport for VNTag mode.

Step 5	**no feature-set virtualization**
	Example:
	`switch(config)# no feature-set virtualization`
	`switch(config)#`
		Disables virtualization.

Example:

```
switch# configure terminal
switch(config)# no interface vethernet 21
switch(config-if)# no port-profile type vethernet user_data
switch(config-if)# no switchport mode vntag
switch(config-if)# no feature-set virtualization
switch(config-if)#
```