CONTENTS

Preface v
  Audience v
  Document Conventions v
  Related Documentation for Cisco Nexus 5500 Series NX-OS Software vi
  Documentation Feedback viii
  Obtaining Documentation and Submitting a Service Request viii

CHAPTER 1 Overview of Adapter-FEX 1
  Overview 1
  Multiple Virtual Links 1
  Supported Platforms 2
  Limitations for Adapter-FEX 2
  Cisco Nexus Adapter-FEX Topology Examples 3

CHAPTER 2 Configuring Adapter-FEX 7
  Adding a Server with a Server Network Adapter 7
    Enabling Switch for Adapter-FEX 7
    Configuring the Adapter FEX Feature 8
    Configuring a Server Network Adapter 9
    Server Network Adapter Initial Handshake and Negotiation 10
    Configuring vNICs on Server Network Adapter 10
    Initializing the Server Network Adapter 11
      Cisco Nexus Adapter FEX Initialization Example for a Dual-Homed FEX Topology 12
      Cisco Nexus Adapter-FEX Initialization Example with Active-Standby Topology 13
      Cisco Nexus Adapter FEX Initialization Example for a Dual-Homed FEX with Active-Standby Topology 15
    Creating Vethernet Interfaces 16
Saving Vethernet Interfaces in the Startup-config 17
Configuring Vethernet Interfaces Manually 17
Configuration Conflicts Between Switch and Server Network Adapter 17
Configuration Conflicts Between Switches 18
Removing a Server with a Server Network Adapter 18
  Relocating a Server to a Different Physical Interface 19
  Adding or Deleting Port-Profiles 20
Disabling Adapter-FEX on a Switch 20
Preface

The Preface contains the following sections:

- Audience, page v
- Document Conventions, page v
- Related Documentation for Cisco Nexus 5500 Series NX-OS Software, page vi
- Documentation Feedback, page viii
- Obtaining Documentation and Submitting a Service Request, page viii

Audience

This publication is for network administrators who configure and maintain Cisco Nexus devices and Cisco Nexus 2000 Series Fabric Extenders.

Document Conventions

Command descriptions use the following conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bold</td>
<td>Bold text indicates the commands and keywords that you enter literally as shown.</td>
</tr>
<tr>
<td>Italic</td>
<td>Italic text indicates arguments for which the user supplies the values.</td>
</tr>
<tr>
<td>[x]</td>
<td>Square brackets enclose an optional element (keyword or argument).</td>
</tr>
<tr>
<td>[x</td>
<td>y]</td>
</tr>
<tr>
<td>{x</td>
<td>y}</td>
</tr>
</tbody>
</table>
## Related Documentation for Cisco Nexus 5500 Series NX-OS Software

The entire Cisco NX-OS 5500 Series documentation set is available at the following URL:

**Release Notes**
The release notes are available at the following URL:

**Configuration Guides**
These guides are available at the following URL:
The documents in this category include:

- Cisco Nexus 5500 Series NX-OS Adapter-FEX Configuration Guide
- Cisco Nexus 5500 Series NX-OS FabricPath Configuration Guide
- Cisco Nexus 5500 Series NX-OS FCoE Configuration Guide
- Cisco Nexus 5500 Series NX-OS Fundamentals Configuration Guide
- Cisco Nexus 5500 Series NX-OS Interfaces Configuration Guide
- Cisco Nexus 5500 Series NX-OS Layer 2 Switching Configuration Guide
- Cisco Nexus 5500 Series NX-OS Multicast Routing Configuration Guide
- Cisco Nexus 5500 Series NX-OS Quality of Service Configuration Guide
- Cisco Nexus 5500 Series NX-OS SAN Switching Configuration Guide
- Cisco Nexus 5500 Series NX-OS Security Configuration Guide
- Cisco Nexus 5500 Series NX-OS System Management Configuration Guide
- Cisco Nexus 5500 Series NX-OS Unicast Routing Configuration Guide

**Installation and Upgrade Guides**
These guides are available at the following URL:
The document in this category include:

- Cisco Nexus 5500 Series NX-OS Software Upgrade and Downgrade Guides

**Licensing Guide**

**Command References**
These guides are available at the following URL:
The documents in this category include:

- Cisco Nexus 5500 Series NX-OS Fabric Extender Command Reference
- Cisco Nexus 5500 Series NX-OS FabricPath Command Reference
- Cisco Nexus 5500 Series NX-OS Fundamentals Command Reference
- Cisco Nexus 5500 Series NX-OS Interfaces Command Reference
- Cisco Nexus 5500 Series NX-OS Layer 2 Interfaces Command Reference
- Cisco Nexus 5500 Series NX-OS Multicast Routing Command Reference
- Cisco Nexus 5500 Series NX-OS Quality of Service Command Reference
- Cisco Nexus 5500 Series NX-OS Security Command Reference
- Cisco Nexus 5500 Series NX-OS System Management Command Reference
- Cisco Nexus 5500 Series NX-OS TrustSec Command Reference
- Cisco Nexus 5500 Series NX-OS Unicast Routing Command Reference
- Cisco Nexus 5500 Series NX-OS Virtual Port Channel Command Reference

Technical References

Error and System Messages

Troubleshooting Guide

Documentation Feedback
To provide technical feedback on this document, or to report an error or omission, please send your comments to: ciscodfa-docfeedback@cisco.com.
We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request
Subscribe to What's New in Cisco Product Documentation, which lists all new and revised Cisco technical documentation as an RSS feed and delivers content directly to your desktop using a reader application. The RSS feeds are a free service.
Overview of Adapter-FEX

This chapter provides an overview of the Cisco NX-OS Adapter-FEX feature and includes the following sections:

- Overview, page 1
- Multiple Virtual Links, page 1
- Supported Platforms, page 2
- Limitations for Adapter-FEX, page 2
- Cisco Nexus Adapter-FEX Topology Examples, page 3

Overview

The Cisco NX-OS Adapter-FEX feature combines the advantages of the FEX link architecture with server I/O virtualization to create multiple virtual interfaces over a single Ethernet interface. This allows you to deploy a dual port NIC on the server and to configure more than two virtual interfaces that the server sees as a regular Ethernet interface. The advantage of this approach is that it allows you to reduce power and cooling needs and to reduce the number of network ports.

Multiple Virtual Links

Adapter-FEX can be thought of as a way to divide a single physical link into multiple virtual links or channels. Each channel is identified by a unique channel number and its scope is limited to the physical link.

The physical link connects a port on a server network adapter with an Ethernet port on the switch. This allows the channel to connect a vNIC on the server with a Vethernet interface on the switch.

Packets on each channel are tagged with a VNTag that has a specific source virtual interface identifier (VIF). The VIF allows the receiver to identify the channel that the source used to transmit the packet.
Supported Platforms

Adapter-FEX requires a server network adapter that is connected to a parent switch that supports Adapter-FEX functionality.

This implementation is designed to work with server network adapters, such as the Cisco UCS P81E or Cisco UCS VIC1225 Virtual Interface Cards (VICs) for the Cisco UCS C-Series Rack-Mount Server or third-party adapters that support the VNTag technology, such as the Broadcom BCM57810 Convergence Network Interface Card.

Limitations for Adapter-FEX

Adapter-FEX has the following limitations:

- In Cisco NX-OS Release 5.0(3)N2(1)), the Cisco Nexus 5000 Platform switch does not support the Adapter-FEX feature.

- You can perform a non-disruptive ISSU only when you reload the switch after disabling Adapter-FEX feature.

- To verify that the current STP topology is consistent with ISSU requirements, use the `show spanning-tree issu-impact` command to display the STP configuration and whether or not there are potential STP issues.

This example shows how to display information about the STP impact when performing an ISSU:

```
switch# show spanning-tree issu-impact
For ISSU to Proceed, Check the Following Criteria:
1. No Topology change must be active in any STP instance
2. Bridge assurance(BA) should not be active on any port (except MCT)
3. There should not be any Non Edge Designated Forwarding port (except MCT)
4. ISSU criteria must be met on the VPC Peer Switch as well
Following are the statistics on this switch
No Active Topology change Found!
Criteria 1 PASSED !
No Ports with BA Enabled Found!
Criteria 2 PASSED !
No Non-Edge Designated Forwarding Ports Found!
Criteria 3 PASSED !
ISSU Can Proceed! Check Peer Switch.
```
Cisco Nexus Adapter-FEX Topology Examples

The following figures show examples of Cisco Nexus Adapter-FEX topologies with server network adapters.

Figure 1: Cisco Nexus Adapter-FEX Topologies

Figure 2: Cisco Nexus Adapter-FEX Dual-Homed Topology
The following figures show topologies that are supported for server network adapters that support active-standby teaming of uplinks.

The active-standby topologies shown here have one uplink as active and the other uplink as standby. With some server network adapters, you can 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 remaining uplinks.

Selecting the active and standby uplinks per vNIC is a best practice.
The Cisco UCS P81E and Cisco UCS VIC1225 Virtual Interface Cards support active-standby uplink teaming. These Virtual Interface Cards allow each virtual NIC (vNIC) to choose the active uplink. The other uplink is configured as a standby uplink.

See the Cisco UCS Virtual Interface Card documentation in the *Cisco UCS C-Series Servers Integrated Management Controller CLI Configuration Guide* for details.

---

**Figure 3: Cisco Nexus Adapter-FEX Active-Standby Topology**

**Figure 4: Cisco Nexus Adapter-FEX Dual-Homed Active-Standby Topology**
Overview of Adapter-FEX

Cisco Nexus Switch-A

Cisco Nexus FEX

Cisco Nexus Switch-B

Cisco Nexus FEX

Server with NIV-enabled Adapter

Eth100/1/15

Eth101/1/15
Configuring Adapter-FEX

The chapter contains the following sections:

- Adding a Server with a Server Network Adapter, page 7
- Removing a Server with a Server Network Adapter, page 18
- Disabling Adapter-FEX on a Switch, page 20

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> install feature-set virtualization</td>
<td>Installs the Cisco Virtual Machine feature set on the switch.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# install feature-set virtualization</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
</tbody>
</table>
### 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>switch# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>switch(config)# port-profile type vethernet port-profile_name</td>
<td>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).</td>
</tr>
<tr>
<td>3</td>
<td>switch(config-port-profile)#state enabled</td>
<td>Enable the vNIC.</td>
</tr>
<tr>
<td>4</td>
<td>switch(config-port-profile)#exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>5</td>
<td>switch(config)#interface ethernet type /port</td>
<td>Enters configuration mode for the specified interface.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>6</td>
<td>switch(config-if)#switchport mode vntag</td>
<td>On the Ethernet interfaces of the switch that the network adapter’s uplink interfaces are connected to, configure the switchport for VNTag mode.</td>
</tr>
<tr>
<td>7</td>
<td>switch(config-if)#exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td>8</td>
<td>switch(config-if)# copy running-config startup-config</td>
<td>(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>9</td>
<td>Connect the server network adapter uplinks</td>
<td>Connect the server network adapter uplinks from the server to the switch or FEX.</td>
</tr>
</tbody>
</table>

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

```plaintext
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
```

**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.
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 Vethernet 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
```
Cisco Nexus Adapter FEX Initialization Example for a Dual-Homed FEX Topology

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

Figure 5: Cisco Nexus Adapter FEX Dual-Homed Topology

Vethernet interface initialization results of the Cisco Nexus Switch-A

```bash
switch# interface vethernet 24
bind interface ethernet 1/5 channel 4
inherit port-profile user_backup
```

Vethernet interface initialization results of the Cisco Nexus Switch-B

```bash
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
```
 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 6: 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:

<table>
<thead>
<tr>
<th>vNIC</th>
<th>Uplink</th>
<th>Port-profile Name</th>
<th>Channel Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>Uplink_0</td>
<td>user_data</td>
<td>1</td>
</tr>
<tr>
<td>eth1</td>
<td>Uplink_1</td>
<td>user_management</td>
<td>2</td>
</tr>
</tbody>
</table>
also assume that uplink_0 is connected to ethernet 1/5 on switch-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 7: Cisco Nexus Adapter FEX Dual-Homed Active-Standby Topology*

<table>
<thead>
<tr>
<th>vNIC</th>
<th>Uplink</th>
<th>Port-profile Name</th>
<th>Channel Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>Uplink_0</td>
<td>user_data</td>
<td>1</td>
</tr>
<tr>
<td>eth1</td>
<td>Uplink_1</td>
<td>user_management</td>
<td>2</td>
</tr>
<tr>
<td>eth2</td>
<td>Uplink_0</td>
<td>user_data</td>
<td>3</td>
</tr>
<tr>
<td>eth3</td>
<td>Uplink_1</td>
<td>user_backup</td>
<td>4</td>
</tr>
</tbody>
</table>

Also 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:

```
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
```
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
```

---

**Note**

The initialization results are identical for both switches.
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.

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.

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.

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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td></td>
</tr>
<tr>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Removes the Vethernet interface.</td>
</tr>
<tr>
<td>no interface vethernet  <em>veth-id</em></td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td></td>
</tr>
<tr>
<td>switch(config)# no interface vethernet 1</td>
<td></td>
</tr>
<tr>
<td>switch(config)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Saves the change persistently through reboots</td>
</tr>
<tr>
<td>copy running-config startup-config</td>
<td>and restarts by copying the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><em>Example:</em></td>
<td></td>
</tr>
<tr>
<td>switch(config)# copy running-config startup-config</td>
<td></td>
</tr>
</tbody>
</table>

**Example of removing Vethernet interfaces**

```bash
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>shutdown</td>
<td>Issues shutdown on original physical interface.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch# shutdown</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch# configure terminal switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>no interface vethernet veth-id</td>
<td>Delete each vethernet interface associated with the server.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)# no interface vethernet 21 switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>no switchport mode vntag</td>
<td>Unconfigures switchport mode vntag on the original physical interface on both vPC switches.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-if)# no switchport mode vntag switch(config)#</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>Unplug the server from both vPC switches.</td>
<td>Disconnects server from both vPC switches.</td>
</tr>
</tbody>
</table>
### Adding or Deleting Port-Profiles

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 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
</tbody>
</table>
| **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:**

```bash
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)#
```
INDEX

A
Adapter-FEX 1, 2, 3, 12, 13, 15, 20
  disabling 20
  multiple virtual links 1
  overview 1
  supported platforms 2
  topology example 3, 12, 13, 15

B
Broadcom Convergence Network Interface Card 9, 10

C
Cisco UCS Virtual Interface Card 9, 10
  configuration conflict 17, 18
    between switch and server network adapter 17
    between switches 18

N
Network Interface Virtualization (NIV) 9

P
port-profiles 20
  adding or deleting 20

S
server 19
  relocating 19
server network adapter 9, 10, 11
  configuring 9
  handshake and negotiation 10
  initializing 11
  virtual NIC (vNIC) 10
switch 7, 8
  configuring 8
  enabling 7

V
vethernet interface 16, 17, 18
  configuring manually 17
  creating 16
  deleting 18
    in startup-config 17
  virtual interface identifier (VIF) 1