

Nexus 5500 Adapter–FEX Configuration Example



Document ID: 117691

Contributed by Charles Youssef, Cisco TAC Engineer.
Sep 04, 2014

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

- Adapter–FEX Overview

Configure

- Ethernet vNICs Configuration
- vHBAs Configuration

Verify

Troubleshoot

- Virtual Ethernet Interface Does Not Come Up
- Collect the Adapter Technical Support Information from the Server Side

Introduction

This document describes how to configure, operate, and troubleshoot the Adapter–Fabric Extender (FEX) feature on Nexus 5500 switches.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these software and hardware versions:

- Nexus 5548UP which runs Version 5.2(1)N1(4)
- Unified Computing System (UCS) C–Series C210 M2 Rack Server with UCS P81E Virtual Interface Card (VIC) that runs Firmware Version 1.4(2)

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command or packet capture setup.

Background Information

Adapter-FEX Overview

This feature allows a Nexus 5500 switch to manage virtual interfaces (both Ethernet virtual Network Interface Controllers (vNICs) and Fibre Channel Virtual Host Bus Adapters (FC vHBAs)) on the server's VIC. This is independent from any hypervisor that runs on the server. Whatever virtual interfaces are created will be visible to the main Operating System (OS) installed on the server (provided that the OS has the appropriate drivers).

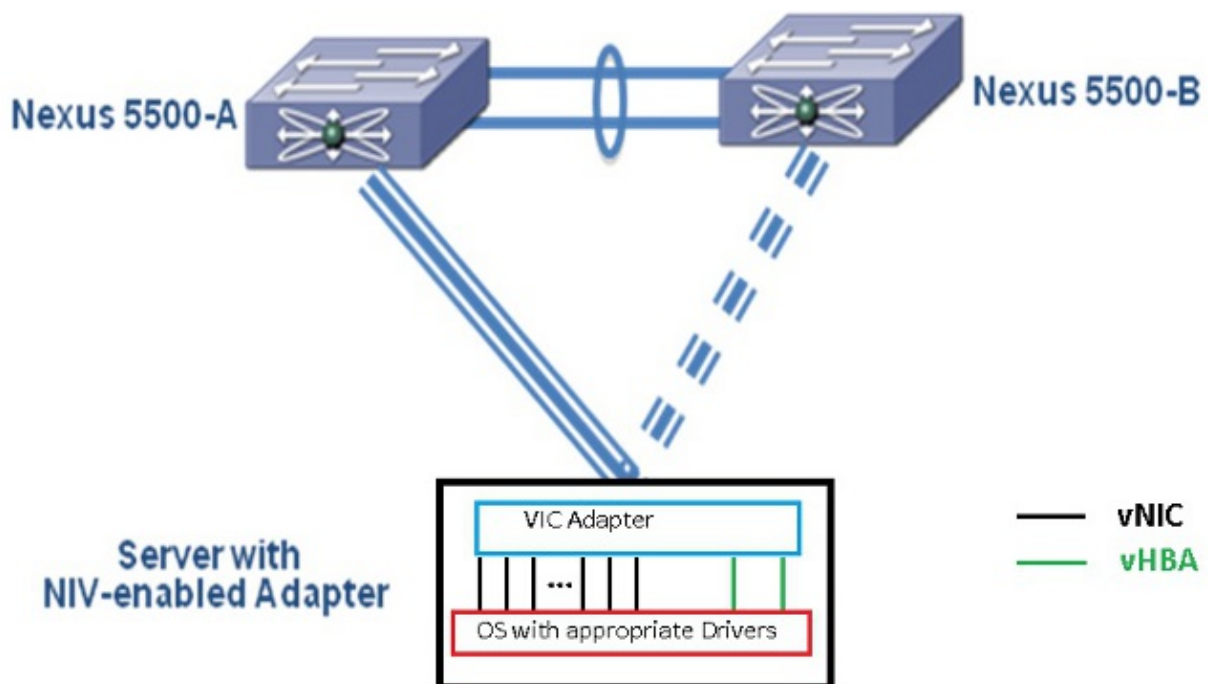
Supported platforms can be found in this section of the Cisco Nexus 5000 Series NX-OS Adapter FEX Operations Guide, Release 5.1(3)N1(1).

Supported topologies for Adapter-FEX can be found in this section of the Cisco Nexus 5000 Series NX-OS Adapter FEX Operations Guide, Release 5.1(3)N1(1).

The supported topologies are:

- Server single-homed to a Nexus 5500 switch
- Server single-homed to a Straight-Through FEX
- Server single-homed to an Active/Active FEX
- Server dual-homed via Active/Standby uplinks to a pair of Nexus 5500 switches
- Server dual-homed via Active/Standby uplinks to a pair of Virtual Port Channel (vPC) Active/Active FEXs

The subsequent configuration section discusses 'Server dual-homed via Active/Standby uplinks to a pair of Nexus 5500 switches' which is depicted here:



Each vNIC will have a corresponding virtual Ethernet interface on the Nexus 5000. Similarly each vHBA will have a corresponding Virtual Fibre Channel (VFC) interface on the Nexus 5000.

Configure

Note: Use the Command Lookup Tool (registered customers only) in order to obtain more information on the commands used in this section.

Ethernet vNICs Configuration

Complete these steps on both Nexus 5000 switches:

1. Normally vPC is defined and operational on the two Nexus 5000 switches. Verify that the vPC domain is defined, peer-keepalive is UP, and peer-link is UP.
2. Enter these commands in order to enable the virtualization feature set.

```
(config)# install feature-set virtualization  
(config)# feature-set virtualization
```

3. (Optional) Allow the Nexus 5000 to auto-create its virtual Ethernet interfaces when the corresponding vNICs are defined on the server. Note that this does not apply to the VFC interfaces which can only be manually defined on the Nexus 5000.

```
(config)# vethernet auto-create
```

4. Configure the Nexus 5000 interface that connects to the servers in Virtual Network Tag (VNTag) mode.

```
(config)# interface Eth 1/10  
(config-if)# switchport mode vntag  
(config-if)# no shutdown
```

5. Configure the port profile(s) to be applied to the vNICs.

The port profiles are configuration templates that can be applied (inherited) by the switch interfaces. In the context of Adapter-FEX, the port profiles can be either applied to the virtual Ethernet interfaces that are manually defined or to the ones that are automatically created when the vNICs are configured on the UCS C-Series Cisco Integrated Management Controller (CIMC) GUI interface.

The port-profile is of type 'vethernet'.

A sample port-profile configuration is shown here:

```
(config)# port-profile type vethernet VNIC1  
(config-port-prof)# switchport mode access  
(config-port-prof)# switchport access vlan 10  
(config-port-prof)# no shutdown  
(config-port-prof)# state enabled
```

Complete these steps on the UCS C-Series server:

1. Connect to the CIMC interface via HTTP and log in with the administrator credentials.
2. Choose **Inventory > Network Adapters > Modify Adapter Properties**.
3. Check the **Enable NIV Mode** check box.
4. Click **Save Changes**.
5. Power off and then power on the server.



6. After the server comes up, choose **Inventory > Network Adapters > vNICs > Add** in order to create vNICs. The most important fields to be defined are:
- VIC Uplink port to be used (P81E has 2 uplink ports referenced as 0 and 1).
 - Channel Number – a unique channel ID of the vNIC on the adapter. This is referenced in the **bind** command under the virtual Ethernet interface on the Nexus 5000. The scope of the channel number is limited to the VNTag physical link. The channel can be thought of as a 'virtual link' on the physical link between the switch and server adapter.
 - Port Profile – the list of port profiles defined on the upstream Nexus 5000 can be selected. A virtual Ethernet interface will be automatically created on the Nexus 5000 if the Nexus 5000 is configured with the **vethernet auto-create** command. Note that only the virtual Ethernet port profile names (port-profile configuration is not) are passed to the server. This occurs after the VNTag link connectivity is established and the initial handshake and negotiation steps are performed between the switch and server adapter.

vNIC Properties

PCI Order: ANY (0 - 17)

Default VLAN: (1 - 4094) N/A

VLAN Mode: N/A

Rate Limit: (1 - 10000 Mbps) N/A

Enable PXE Boot:

Channel Number: 1 (1 - 1000)

Port Profile: [Empty]

Enable Uplink Failover: UPLINK

Failback Timeout: (0 - 600)

Ethernet Interrupt

Interrupt Count: 8 (1 - 514)

Coalescing Time: 125 (0 - 65535 us)

Coalescing Type: MTM

Save Changes Reset Values Cancel

7. Click *Save Changes*.

8. Power off and then power on the server again.

vHBAs Configuration

When you create vHBAs on the server adapter, the corresponding switch interfaces are not automatically created. Instead, they should be manually defined. The steps for the switch and server side are shown here.

Complete these steps on the switch side:

1. Create a virtual Ethernet trunk interface that is bound to the VNTag interface's channel of the server vHBA interface. The Fibre Channel over Ethernet (FCoE) VLAN should not be the native VLAN. The virtual Ethernet numbers should be unique across the two Nexus 5000 switches.

Example:

```
(config)# interface veth 10
(config-if)# switchport mode trunk
(config-if)# switchport trunk allowed vlan 1,100
(config-if)# bind interface eth1/1 channel 3
(config-if)# no shutdown
```

2. Create a VFC interface that is bound to the virtual Ethernet interface defined earlier.

Example:

```
(config)# interface vfc10
(config-if)# bind interface veth 10
(config-if)# no shut
```

The Virtual Storage Area Network (VSAN) membership for this interface is defined under the VSAN database:

```
(config)# vsan database
(config-vsan-db)# vsan 100 interface vfc10
(config-vsan-db)# vlan 100
(config-vlan)# fcoe vsan 100
(config-vlan)# show vlan fcoe
```

Complete these steps on the server side:

1. Choose *Inventory > Network Adapters > vHBAs* in order to create a vHBA interface.

The main fields to be defined are:

- ◆ Port World Wide Name (pWWN)/Node World Wide Name (nWWN)
 - ◆ FCOE VLAN
 - ◆ Uplink ID
 - ◆ Channel number
 - ◆ Boot from Storage Area Network (SAN) if used
2. Power cycle the server.

Verify

Use this section to confirm that your configuration works properly.

The list of virtual Ethernet interfaces can be displayed with these commands:

```
n5k1# show interface virtual summary
Veth      Bound      Channel/  Port      Mac      VM
Interface Interface  DV-Port   Profile   Address  Name
-----
Veth32770 Eth1/2     1         UPLINK
Total 1 Veth Interfaces
n5k1#
n5k1# show interface virtual status
Interface VIF-index  Bound If      Chan  Vlan  Status  Mode  Vntag
-----
Veth32770 VIF-17    Eth1/2      1     10  Up      Active  2
Total 1 Veth Interfaces
```

The automatically created virtual Ethernet interfaces do appear in the running configuration and will be saved to the startup configuration when copy run start is performed:

```
n5k1# show run int ve32770

!Command: show running-config interface Vethernet32770
!Time: Thu Apr 10 12:56:23 2014

version 5.2(1)N1(4)

interface Vethernet32770
  inherit port-profile UPLINK
  bind interface Ethernet1/2 channel 1

n5k1# show int ve32770 brief
```

```
-----
Vethernet      VLAN  Type Mode  Status Reason  Speed
-----
```

```
Veth32770      10      virt access up      none      auto
n5k1#
```

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

Virtual Ethernet Interface Does Not Come Up

Verify the Data Center Bridging Capabilities Exchange Protocol (DCBX) information for the switch VNTag interface with this command:

```
# show system internal dcbx info interface ethernet <>
```

Verify that:

- Data Center Bridging Exchange (DCX) protocol is Converged Ethernet (CEE)
- CEE Network IO Virtualization (NIV) extension is enabled
- NIV Type Length Value (TLV) is present

As highlighted below:

```
n5k1# show sys int dcbx info interface e1/2
```

```
Interface info for if_index: 0x1a001000(Eth1/2)
tx_enabled: TRUE
rx_enabled: TRUE
dcbx_enabled: TRUE
DCX Protocol: CEE <<<<<<<
DCX CEE NIV extension: enabled <<<<<<<<
<output omitted>
```

```
Feature type NIV (7) <<<<<<<
feature type 7(DCX CEE-NIV)sub_type 0
Feature State Variables: oper_version 0 error 0 local error 0 oper_mode 1
    feature_seq_no 0 remote_feature_tlv_present 1 remote_tlv_aged_out 0
    remote_tlv_not_present_notification_sent 0
Feature Register Params: max_version 0, enable 1, willing 0 advertise 1
    disruptive_error 0 mts_addr_node 0x2201 mts_addr_sap 0x193
Other server mts_addr_node 0x2301, mts_addr_sap 0x193
```

```
Desired config cfg length: 8 data bytes:9f ff 68 ef bd f7 4f c6
```

```
Operating config cfg length: 8 data bytes:9f ff 68 ef bd f7 4f c6
```

```
Peer config cfg length: 8 data bytes:10 00 00 22 bd d6 66 f8
```

Common problems include:

- DCX protocol is CIN
 - ◆ Check for L1 problems: cables, SFP, port bring up, adapter.
 - ◆ Check the switch configuration: feature set, switchport VNTag, enable Link Layer Discovery Protocol (LLDP)/DCBX.
- NIV TLV is absent
 - ◆ Check that NIV mode is enabled under the adapter configuration.
 - ◆ Check VNIC Interface Control (VIC) communication has been complete and the port profiles information has been exchanged. Ensure that the current Virtual Interface Manager (VIM) event state is VIM_NIV_PHY_FSM_ST_UP_OPENED_PP.

n5k1# *show sys int vim event-history interface e1/2*

>>>FSM: <Ethernet1/2> has 18 logged transitions<<<<<

- 1) FSM:<Ethernet1/2> Transition at 327178 usecs after Thu Apr 10 12:22:27 2014
Previous state: [VIM_NIV_PHY_FSM_ST_WAIT_DCBX]
Triggered event: [VIM_NIV_PHY_FSM_EV_PHY_DOWN]
Next state: [VIM_NIV_PHY_FSM_ST_WAIT_DCBX]
- 2) FSM:<Ethernet1/2> Transition at 327331 usecs after Thu Apr 10 12:22:27 2014
Previous state: [VIM_NIV_PHY_FSM_ST_WAIT_DCBX]
Triggered event: [VIM_NIV_PHY_FSM_EV_DOWN_DONE]
Next state: [VIM_NIV_PHY_FSM_ST_WAIT_DCBX]
- 3) FSM:<Ethernet1/2> Transition at 255216 usecs after Thu Apr 10 12:26:15 2014
Previous state: [VIM_NIV_PHY_FSM_ST_WAIT_DCBX]
Triggered event: [VIM_NIV_PHY_FSM_EV_RX_DCBX_CC_NUM]
Next state: [VIM_NIV_PHY_FSM_ST_WAIT_3SEC]
- 4) FSM:<Ethernet1/2> Transition at 250133 usecs after Thu Apr 10 12:26:18 2014
Previous state: [VIM_NIV_PHY_FSM_ST_WAIT_3SEC]
Triggered event: [VIM_NIV_PHY_FSM_EV_DCX_3SEC_EXP]
Next state: [VIM_NIV_PHY_FSM_ST_WAIT_ENCAP]
- 5) FSM:<Ethernet1/2> Transition at 262008 usecs after Thu Apr 10 12:26:18 2014
Previous state: [VIM_NIV_PHY_FSM_ST_WAIT_ENCAP]
Triggered event: [VIM_NIV_PHY_FSM_EV_VIC_OPEN_RECEIVED]
Next state: [FSM_ST_NO_CHANGE]
- 6) FSM:<Ethernet1/2> Transition at 60944 usecs after Thu Apr 10 12:26:19 2014
Previous state: [VIM_NIV_PHY_FSM_ST_WAIT_ENCAP]
Triggered event: [VIM_NIV_PHY_FSM_EV_ENCAP_RESP]
Next state: [VIM_NIV_PHY_FSM_ST_UP]
- 7) FSM:<Ethernet1/2> Transition at 62553 usecs after Thu Apr 10 12:26:19 2014
Previous state: [VIM_NIV_PHY_FSM_ST_UP]
Triggered event: [VIM_NIV_PHY_FSM_EV_VIC_OPEN_ACKD]
Next state: [FSM_ST_NO_CHANGE]
- 8) FSM:<Ethernet1/2> Transition at 62605 usecs after Thu Apr 10 12:26:19 2014
Previous state: [VIM_NIV_PHY_FSM_ST_UP]
Triggered event: [VIM_NIV_PHY_FSM_EV_VIC_OPEN_DONE]
Next state: [VIM_NIV_PHY_FSM_ST_UP_OPENED]
- 9) FSM:<Ethernet1/2> Transition at 62726 usecs after Thu Apr 10 12:26:19 2014
Previous state: [VIM_NIV_PHY_FSM_ST_UP_OPENED]
Triggered event: [VIM_NIV_PHY_FSM_EV_PP_SEND]
Next state: [VIM_NIV_PHY_FSM_ST_UP_OPENED_PP]
- 10) FSM:<Ethernet1/2> Transition at 475253 usecs after Thu Apr 10 12:51:45 2014
Previous state: [VIM_NIV_PHY_FSM_ST_UP_OPENED_PP]
Triggered event: [VIM_NIV_PHY_FSM_EV_PHY_DOWN]
Next state: [VIM_NIV_PHY_FSM_ST_WAIT_VETH_DN]
- 11) FSM:<Ethernet1/2> Transition at 475328 usecs after Thu Apr 10 12:51:45 2014
Previous state: [VIM_NIV_PHY_FSM_ST_WAIT_VETH_DN]
Triggered event: [VIM_NIV_PHY_FSM_EV_DOWN_DONE]
Next state: [VIM_NIV_PHY_FSM_ST_WAIT_DCBX]
- 12) FSM:<Ethernet1/2> Transition at 983154 usecs after Thu Apr 10 12:53:06 2014
Previous state: [VIM_NIV_PHY_FSM_ST_WAIT_DCBX]
Triggered event: [VIM_NIV_PHY_FSM_EV_RX_DCBX_CC_NUM]
Next state: [VIM_NIV_PHY_FSM_ST_WAIT_3SEC]
- 13) FSM:<Ethernet1/2> Transition at 992590 usecs after Thu Apr 10 12:53:09 2014
Previous state: [VIM_NIV_PHY_FSM_ST_WAIT_3SEC]


```

    Triggered event: [VIM_NIV_PHY_FSM_EV_DCX_3SEC_EXP]
    Next state: [VIM_NIV_PHY_FSM_ST_WAIT_ENCAP]

14) FSM:<Ethernet1/2> Transition at 802877 usecs after Thu Apr 10 12:53:10 2014
    Previous state: [VIM_NIV_PHY_FSM_ST_WAIT_ENCAP]
    Triggered event: [VIM_NIV_PHY_FSM_EV_ENCAP_RESP]
    Next state: [VIM_NIV_PHY_FSM_ST_UP]

15) FSM:<Ethernet1/2> Transition at 804263 usecs after Thu Apr 10 12:53:10 2014
    Previous state: [VIM_NIV_PHY_FSM_ST_UP]
    Triggered event: [VIM_NIV_PHY_FSM_EV_VIC_OPEN_ACKD]
    Next state: [FSM_ST_NO_CHANGE]

16) FSM:<Ethernet1/2> Transition at 992390 usecs after Thu Apr 10 12:53:11 2014
    Previous state: [VIM_NIV_PHY_FSM_ST_UP]
    Triggered event: [VIM_NIV_PHY_FSM_EV_VIC_OPEN_RECEIVED]
    Next state: [FSM_ST_NO_CHANGE]

17) FSM:<Ethernet1/2> Transition at 992450 usecs after Thu Apr 10 12:53:11 2014
    Previous state: [VIM_NIV_PHY_FSM_ST_UP]
    Triggered event: [VIM_NIV_PHY_FSM_EV_VIC_OPEN_DONE]
    Next state: [VIM_NIV_PHY_FSM_ST_UP_OPENED]

18) FSM:<Ethernet1/2> Transition at 992676 usecs after Thu Apr 10 12:53:11 2014
    Previous state: [VIM_NIV_PHY_FSM_ST_UP_OPENED]
    Triggered event: [VIM_NIV_PHY_FSM_EV_PP_SEND]
    Next state: [VIM_NIV_PHY_FSM_ST_UP_OPENED_PP]

    Curr state: [VIM_NIV_PHY_FSM_ST_UP_OPENED_PP] <<<<<<<<<<
n5k1#

```

If the virtual Ethernet interface is a fixed virtual Ethernet, check to see if VIC_CREATE appears in this command:

```

# show system internal vim info niv msg logs fixed interface e 1/16 ch 1
Eth1/16(Chan: 1) VIF Index: 605
  REQ MsgId: 56630, Type: VIC ENABLE, CC: SUCCESS
  RSP MsgId: 56630, Type: VIC ENABLE, CC: SUCCESS
  REQ MsgId: 4267, Type: VIC SET, CC: SUCCESS
  RSP MsgId: 4267, Type: VIC SET, CC: SUCCESS
  REQ MsgId: 62725, Type: VIC CREATE, CC: SUCCESS <<<<<<<
  RSP MsgId: 62725, Type: VIC CREATE, CC: SUCCESS <<<<<<<
  REQ MsgId: 62789, Type: VIC ENABLE, CC: SUCCESS
  RSP MsgId: 62789, Type: VIC ENABLE, CC: SUCCESS
  REQ MsgId: 21735, Type: VIC SET, CC: SUCCESS
  RSP MsgId: 21735, Type: VIC SET, CC: SUCCESS

```

Note that a fixed virtual Ethernet interface is a virtual interface that does not support migration across physical interfaces. When Adapter-FEX is discussed, the scope is always on fixed virtual Ethernet because Adapter-FEX refers to the use of network virtualization by a single (that is, nonvirtualized) OS.

If VIC_CREATE does not show up:

1. If the adapter is a Cisco NIV adapter, check the VNIC configuration on the adapter side (channel ID, correct uplink UIF port, any pending commit (server reboot needed for any configuration changes)). A vHBA will not bring up virtual Ethernet on both switches in AA FEX topology. A vHBA fixed virtual Ethernet needs an OS driver to bring this up (wait until the OS loads the driver and boots up completely).
2. If the adapter is a Broadcom NIV adapter, check to see if the interfaces are up from the OS side (for example, in Linux, bring up the interface 'ifconfig eth2 up').
3. If VIC_CREATE shows up, but the switch responds with ERR_INTERNAL:

- ◆ Check the port profiles on both the switch and adapter sides. See if any port profiles strings mismatch.
 - ◆ For dynamic fixed virtual Ethernets, check the 'veth auto-create' configuration.
4. If the problem persists, collect the output listed below and contact the Cisco Technical Assistance Center (TAC).

```
# show system internal vim log
# attach fex <number>
# test vic_proxy dump trace
```

Collect the Adapter Technical Support Information from the Server Side

1. Log in to CIMC from a browser.
2. Click the *Admin* tab.
3. Click *Utilities*.
4. Click *Export Technical Support Data to TFTP or Generate Technical Support Data for Local Download*.

Updated: Sep 04, 2014

Document ID: 117691
