

# 连结5500适配器FEX配置示例

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[适配器FEX概述](#)

[配置](#)

[以太网vNICs配置](#)

[vHBAs配置](#)

[验证](#)

[故障排除](#)

[虚拟以太网接口不出来](#)

[收集了从服务器端的适配器技术支持信息](#)

## 简介

本文描述如何配置，操作和排除故障在连结的适配器结构扩展器(FEX)功能5500交换机。

## 先决条件

### 要求

本文档没有任何特定的要求。

### 使用的组件

本文档中的信息基于以下软件和硬件版本：

- 运行版本5.2(1)N1(4)的连结5548UP
- 统一计算系统(UCS) C系列C210 M2有UCS P81E虚拟接口界面卡的(VIC)货架服务器该运行固件版本1.4(2)

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络实际，请确保您了解任何命令或数据包捕获设置潜在影响。

# 背景信息

## 适配器FEX概述

此功能允许连结5500交换机管理虚拟接口(以太网虚拟网络网络界面控制器(vNICs)和光纤信道虚拟主机总线适配器(FC vHBAs))在服务器的VIC。这在服务器运行的所有hypervisor是独立。任何虚拟接口创建将是可视对在服务器安装的主要操作系统(OS) (在OS有适当的驱动程序)条件下。

支持的平台可以在[Cisco连结5000系列NX-OS适配器FEX操作指南的此部分找到](#)，[版本5.1\(3\)N1\(1\)](#)。

适配器FEX的支持的拓扑可以在[Cisco连结5000系列NX-OS适配器FEX操作指南的此部分找到](#)，[版本5.1\(3\)N1\(1\)](#)。

支持的拓扑是：

- 服务器单址的对连结5500交换机
- 服务器单址的对直通电缆FEX
- 服务器选拔址的对主动/主动FEX
- 服务器双重址的通过对一个对的活动/等待uplink端口连结5500交换机
- 服务器双重址的通过对一个对的活动/等待uplink端口虚拟端口信道(vPC)主动/主动FEXs

随后的配置部分讨论‘服务器双重址的通过对表示得此处的一个对的活动/等待uplink端口连结5500交换机的：

每vNIC将有在连结5000的一个对应的虚拟以太网接口。同样每vHBA将有在连结5000的一个对应的虚拟光纤信道(VFC)接口。

## 配置

**Note:**使用[命令查找工具](#) ( [仅限注册用户](#) ) 可获取有关本部分所使用命令的详细信息。

## 以太网vNICs配置

完成在两连结的这些步骤5000交换机：

1. 通常vPC是定义和可操作的在两连结5000交换机。验证vpc domain定义，对等体保活是UP，并且对等体林克是UP。
2. 输入这些命令为了启用虚拟化特性组。

```
(config)# install feature-set virtualization
(config)# feature-set virtualization
```
3. (可选)请允许连结5000自动创建其虚拟以太网接口，当对应的vNICs在服务器时定义。注意这不应用对VFC在连结5000可能手工只定义的接口。

```
(config)# vethernet auto-create
```
4. 配置连接到在虚拟网络标记的连结5000接口(VNTag)模式的服务器。

```
(config)# interface Eth 1/10
(config-if)# switchport mode vntag
```

```
(config-if)# no shutdown
```

5. 配置将应用的端口配置文件对vNICs。端口配置文件是可以应用的配置模板(继承)由交换机接口。在适配器FEX中，端口配置文件可以是二者之一应用对手工定义的虚拟以太网接口或对自动地创建的那个，当vNICs在UCS C系列思科集成管理控制器(CIMC)时GUI界面配置。波尔特配置文件是类型‘vethernet’。示例波尔特配置文件配置显示此处：

```
(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
```

完成在UCS C系列服务器的这些步骤：

1. 连接对CIMC接口通过HTTP和登录与管理员凭证。
2. 选择Inventory>网络适配器> Modify适配器属性。
3. 检查Enable (event) NIV Mode复选框。
4. 点击Save Changes。
5. 停电然后启动服务器。
6. 在服务器出现后，请选择Inventory>网络适配器> vNICs >Add为了创建vNICs。将定义的最重要的字段是：将使用的VIC上行链路端口(P81E有作为0和1)被参考的2上行链路端口。信道数-vNIC的唯一信道ID在适配器的。这被参考bind命令在连结5000的虚拟以太网接口下。信道数的范围对VNTag物理链路被限制。信道可以设想作为一‘虚链路’在交换机和服务器适配器之间的物理链路。波尔特配置文件-端口配置文件列表在上行连结定义的5000可以选择。如果连结5000配置与vethernet自动创建命令，虚拟以太网接口在连结5000将自动地创建。注意仅虚拟以太网端口配置文件名称(波尔特配置文件配置不是)通过到服务器。这在VNTag链路连接以后发生设立，并且最初的握手和协商步骤被执行在交换机和服务器适配器之间。
7. 点击Save Changes。
8. 停电再然后启动服务器。

## vHBAs配置

当您创建在服务器适配器时的vHBAs，对应的交换机接口没有自动地创建。反而，应该手工定义他们。交换机和服务器端的步骤显示此处。

完成在转换面的这些步骤：

1. 创建一定对服务器vHBA接口的VNTag界面通道的一个虚拟以太网中继接口。在以太网(FCoE) VLAN的光纤信道不应该是本地VLAN。虚拟以太网编号应该是唯一在两连结间5000交换机。

示例：

```
(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. 创建一定对定义的虚拟以太网接口前的VFC接口。示例：

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

此接口的虚拟存储区域网络(VSAN)会员定义在VSAN数据库下：

```
(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
```

完成在服务器端的这些步骤：

1. 选择Inventory>网络适配器> vHBAs为了创建vHBA接口。将定义的主要字段是：全世界波尔特命名(pwwn)全世界/Node名称(nWWN)FCOE VLAN上行链路ID信道编号从存储区域网络(SAN)启动若被采用
2. 重新通电服务器。

## 验证

使用本部分可确认配置能否正常运行。

虚拟以太网接口列表可以用这些命令显示：

```
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
```

自动地创建的虚拟以太网接口在运行的配置出现，并且请保存对启动配置，当copy run start执行时：

```
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#
```

## 故障排除

本部分提供的信息可用于对配置进行故障排除。

### 虚拟以太网接口不出来

验证交换机VNTag接口的数据中心桥接功能开关协议(DCBX)信息用此命令：

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

检验：

- 桥接Exchange (DCX)协议的数据中心是聚合的以太网(CEE)
- CEE网络IO虚拟化(NIV)分机启用
- NIV类型长度值(TLV)存在

如如下突出显示：

```
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
```

常见问题包括：

- DCX协议是CIN  
检查L1问题：电缆，SFP，端口启动，适配器。检查交换机配置：特性组，switchport VNTag，enable (event)链路层发现协议(LLDP) /DCBX。
- NIV TLV是缺少的 检查NIV模式启用在适配器配置下。检查VNIC接口控制(VIC)通信完成，并且端口配置文件信息交换。保证当前虚拟接口管理器(精力)事件状态是 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#
```

如果虚拟以太网接口是一已修复虚拟以太网，请确认VIC\_CREATE是否在此命令出现：

```
# 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
```

注意一个已修复虚拟以太网接口是不支持在物理接口间的迁移的虚拟接口。当适配器FEX讨论时，范围总是在已修复虚拟以太网，因为适配器FEX由单个(即nonvirtualized) OS是指使用网络虚拟化。

如果VIC\_CREATE没出现：

1. 如果适配器是思科NIV适配器，请检查在适配器侧(信道ID、正确上行链路UIF端口，任何待定进行(为任何配置更改需要的服务器重新启动的vNIC配置))。vHBA不会在AA FEX拓扑方面启动在两交换机的虚拟以太网。vHBA修复的虚拟以太网需要OS驱动程序提出此(等待，直到OS装载驱动程序并且启动完全)。
2. 如果适配器是Broadcom NIV适配器，请确认接口是否是UP从OS侧(例如，在Linux，请启动接口'ifconfig eth2 up')。
3. 如果VIC\_CREATE出现，但是交换机回应ERR\_INTERNAL：检查在交换机和适配器侧的端口配置文件。是否检查任何端口配置文件字符串不匹配。对于动态已修复虚拟以太网，请检查'veth自动创建'配置。
4. 如果问题持续，请收集如下所示的输出并且请与Cisco技术支持中心(TAC)联系。

```
# 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
```

## 收集了从服务器端的适配器技术支持信息

1. 洛金对CIMC从浏览器。
2. 点击Admin选项卡。
3. 点击工具。
4. 单击出口技术支持数据对TFTP或生成技术支持数据为本地下载。