



# Configuring FlexAttach Virtual pWWN

---

This chapter provides information about FlexAttach virtual pWWN and how to configure FlexAttach virtual pWWN.

- [Finding Feature Information, on page 2](#)
- [Information About FlexAttach Virtual pWWN, on page 3](#)
- [Guidelines and Limitations, on page 5](#)
- [Configuring FlexAttach Virtual pWWN, on page 6](#)
- [Verifying FlexAttach Virtual pWWN Configuration, on page 9](#)
- [Monitoring FlexAttach Virtual pWWN, on page 10](#)

## Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at <https://tools.cisco.com/bugsearch/> and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the New and Changed chapter or the Feature History table below.

# Information About FlexAttach Virtual pWWN

## FlexAttach Virtual pWWN

FlexAttach virtual pWWN feature facilitates server and configuration management. In a SAN environment, the server installation or replacement, requires interaction and coordination among the SAN and server administrators. For coordination, it is important that the SAN configuration does not change when a new server is installed, or when an existing server is replaced. FlexAttach virtual pWWN minimizes the interaction between the server administrator and the SAN administrator by abstracting the real pWWN using virtual pWWNs.

When FlexAttach virtual pWWN is enabled on an interface, a virtual pWWN is assigned to the server interface. The real pWWN is replaced by a virtual pWWN, which is used for a SAN configuration such as zoning.

Server administrators can benefit from FlexAttach in the following scenarios:

- **Pre-configure**—Pre-configure SAN for new servers that are not available physically yet. For example, they may be on order. FlexAttach can be enabled on the ports designated for the new servers and use the virtual WWNs assigned for configuring SAN. The new servers are then plugged into the fabric without any change needed in the SAN.
- **Replacement to the same port**—A failed server can be replaced onto the same port without changing the SAN. The new server gets a same pWWN as the failed server because the virtual pWWN is assigned to the port.
- **Replacement to (spare)**—A spare server, which is on the same NPV device or a different NPV device) can be brought online without changes to the SAN. This action is achieved by moving the virtual port WWN from the current server port to the spare port.
- **Server Mobility**—A server can be moved to another port on the same NPV device or another NPV device without changing the SAN. This is accomplished by moving the virtual pWWN to the new port. No change is needed if FlexAttach was configured using the physical port WWN of the server to the virtual port WWN mapping.

## Difference Between San Device Virtualization and FlexAttach Port Virtualization

Table describes the difference between SAN device virtualization (SDV) and FlexAttach port virtualization.

**Table 1: Difference Between SDV and FlexAttach Virtualization**

<b>SAN Device Virtualization (SDV)</b>	<b>FlexAttach Virtualization</b>
Facilitates target and disk management, and only facilitates disk and data migration.	Facilitates server management and has no restriction on the end devices used.
WWN NAT and Fibre Channel ID (FC-ID) are allocated on the virtual device, both primary and secondary.	WWN and Network Address Transport (NAT) is allocated to host bus adapter (HBA).

SAN Device Virtualization (SDV)	FlexAttach Virtualization
FC-ID rewrite on the switch indicates a rewrite-capable switch on the path.	No rewrite requirements.
Configuration is distributed. This allows programming rewrites and connectivity anywhere.	Configuration distribution is not required for any of the interface-based configurations.
Configuration is secured to device alias.	Does not require device alias for virtual pWWN.
Does not allow automapping to the secondary device.	Allows automapping to the new HBA. Mapping process is manual for NPIV.

## FlexAttach Virtual pWWN CFS Distribution

The FlexAttach virtual pWWN configuration is distributed for CFS through IPv4, and is enabled by default. The FlexAttach virtual pWWN distribution, by default, is on CFS region 201. The CFS region 201 links only to the NPV-enabled switches. Other CFS features such as syslog is on region 0. Region 0 will be linked through IPv4 for all NPV switches on the same physical fabric. If CFS has an option to link through IPv4 or ISL, then CFS will select the ISL path.




---

**Note** NPV switches do not have ISL (E or TE ports) and are linked through IPv4.

---

## Security Settings for FlexAttach Virtual pWWN

Security settings for the FlexAttach virtual pWWN feature are done by port security at the NPV core. Node WWN of the end device is used to provide physical security.

For more details on enabling port security, refer to the [Cisco MDS 9000 Series NX-OS Security Configuration Guide](#).

## Guidelines and Limitations

Following are recommended guidelines and requirements when deploying FlexAttach virtual pWWN:

- FlexAttach configuration is supported only on NPV switches.
- Cisco Fabric Services (CFS) IP version 4 (IPv4) distribution should be enabled.
- Virtual WWNs should be unique across the fabric.

# Configuring FlexAttach Virtual pWWN

## Automatically Assigning FlexAttach Virtual pWWN

Automatic assignment of virtual pWWN can be configured on an NPV switch globally, per VSAN, or per port. When assigned automatically, a virtual WWN is generated from the device local switch WWN.

To assign a virtual pWWN automatically, perform this task:

### Before you begin

The port must be in a shut state when the virtual pWWN is enabled.

---

**Step 1** Enter configuration mode:

**Example:**

```
switch# configure terminal
```

**Step 2** Assign FlexAttach virtual pWWN automatically for the interfaces:

**Example:**

```
switch(config)# flex-attach virtual-pwwn auto [interface interface-list]
```

To assign FlexAttach virtual pWWN automatically for the VSANs:

```
switch# (config)# flex-attach virtual-pwwn auto [vsan vsan-range]
```

**Step 3** Commit the configuration:

**Example:**

```
switch(config)# flex-attach commit
```

---

## Manually Assigning FlexAttach Virtual pWWN

### Restrictions

The interface mentioned in the interface value must be in a shut state.

To assign virtual pWWN manually, perform this task:

### Before you begin

- Some ports may be in automode, some in manual mode, and the virtual pWWNs need not be assigned.

- The port must be in a shut state when a virtual pWWN is assigned.

---

**Step 1** Enter configuration mode:

**Example:**

```
switch# configure terminal
```

**Step 2** Configure the FlexAttach virtual pWWN for the interface:

**Example:**

```
switch(config)# flex-attach virtual-pwwn vppwn interface interface
```

(Optional) Configure the FlexAttach virtual pWWN for the interface in the VSAN:

```
switch(config)# flex-attach virtual-pwwn vppwn interface interface [ vsan vsan]
```

**Step 3** Commit the configuration:

```
switch(config)# flex-attach commit
```

---

## Mapping pWWN to Virtual pWWN

You can configure virtual pWWNs through real pWWNs. This process is required for NPIV hosts containing multiple pWWNs, of which only FLOGI is mapped to the virtual pWWN. Subsequent FDSIDs will have different mappings.

Several checks are done by the NPV core to ensure the uniqueness of virtual pWWNs in the switch across the NPV switches. When duplicate virtual pWWNs are configured, the subsequent logins are rejected by the NPV core switch.

### Restrictions

- The specified virtual pWWN and the real pWWN must not be logged in.
- To map pWWN to virtual pWWN, perform this task:

### Before you begin

The interface must be in a shut state and the specified virtual pWWN should not be logged in.

---

**Step 1** Enter configuration mode:

**Example:**

```
switch# configure terminal
```

**Step 2** Map the pWWN to the virtual pWWN:

**Example:**

```
switch(config)# flex-attach virtual-pwwn vpwwn pwwn pwwn
```

**Step 3** Commit the configuration:

```
switch(config)# flex-attach commit
```

---



# Verifying FlexAttach Virtual pWWN Configuration

To display FlexAttach configuration information, perform one of the following tasks:

Command	Purpose
<code>show flex-attach virtual-pwwn</code>	Displays the type and value of virtual pWWNs.
<code>show fcns database</code>	Displays if the end device is logged with the correct virtual WWNs.

For detailed information about the fields in the output from these commands, refer to the [Cisco MDS 9000 Series NX-OS Command Reference](#).

To view and confirm that the type and value of virtual pWWNs are correct, enter the `show flex-attach virtual-pwwn` command.

## Example: Displaying the Type and Value of Virtual pWWNs

```
switch# show flex-attach virtual-pwwn
VIRTUAL PORT WWNS ASSIGNED TO INTERFACES
-----
VSAN INTERFACE VIRTUAL-PWWN AUTO LAST-CHANGE
-----
1 fc1/1 00:00:00:00:00:00:00:00
1 fc1/2 22:73:00:05:30:01:6e:1e TRUE Thu Jan 31 01:58:52 2008
1 fc1/3 22:5e:00:05:30:01:6e:1e TRUE Thu Jan 31 01:58:52 2008
1 fc1/4 22:5f:00:05:30:01:6e:1e TRUE Thu Jan 31 01:58:52 2008
1 fc1/5 22:74:00:05:30:01:6e:1e TRUE Thu Jan 31 01:26:24 2008
1 fc1/6 22:60:00:05:30:01:6e:1e TRUE Thu Jan 31 01:58:52 2008
1 fc1/7 22:61:00:05:30:01:6e:1e TRUE Thu Jan 31 01:58:52 2008
1 fc1/8 22:62:00:05:30:01:6e:1e TRUE Thu Jan 31 01:58:52 2008
1 fc1/9 22:63:00:05:30:01:6e:1e TRUE Thu Jan 31 01:58:52 2008
1 fc1/10 22:64:00:05:30:01:6e:1e TRUE Thu Jan 31 01:58:52 2008
1 fc1/11 22:65:00:05:30:01:6e:1e TRUE Thu Jan 31 01:58:52 2008
1 fc1/12 22:66:00:05:30:01:6e:1e TRUE Thu Jan 31 01:58:52 2008
```

## Verifying the End Device

To verify that the end device is logged with the correct virtual WWNs, use the `show fcns database` command on the NPV core.

### Example: Verifying the End Device

```
switch# show fcns database
VSAN 1:
-----
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
-----
0x010000 N 20:01:00:0d:ec:2f:c1:40 (Cisco) npv
0x010001 N 20:02:00:0d:ec:2f:c1:40 (Cisco) npv
0x010200 N 21:00:00:e0:8b:83:01:a1 (Qlogic) scsi-fcp:init
0x010300 N 21:01:00:e0:8b:32:1a:8b (Qlogic) scsi-fcp:init
Total number of entries = 4
```

## Monitoring FlexAttach Virtual pWWN

Table lists the errors that might be displayed and provides the workarounds.

**Table 2: FlexAttach Errors and Workarounds**

Error	Description	Workaround
fc1/1 : interface is not down	FlexAttach configuration fails because the configuration is enabled for an active interface with the operation state as up.	To move the port to the shut state, enable the FlexAttach configuration, and then move the port to no shut state.
FlexAttach configuration is not distributed to the peers	The FlexAttach configuration on one peer NPV is not available to any other peer NPV.	FlexAttach configuration will not be distributed if <b>cfs ipv4 distribute</b> , or <b>cfs ipv6 distribute</b> is disabled. Enable <b>cfs ipv4 distribute</b> , or <b>cfs ipv6 distribute</b> .
Even with CFS distribution enabled Inagua does not become a peer with other NPVs	CFS over IP is enabled, and the Inagua in one blade center is not the peer NPV for other NPVs.	CFS over IP uses IP multicast to discover the NPV peers in the network. IBM MM does not support multicast and cannot act as a peer with NPV. This prevents the FlexAttach configuration from getting distributed to other peer NPVs in the network.
NP port uses physical pWWN instead of virtual pWWN configured through FlexAttach	This occurs when NP port uses physical pWWN instead of virtual pWWN, that is configured through FlexAttach.	FlexAttach is supported on server interfaces such as F ports, and not on external interfaces such as NP ports.
real port WWN and virtual WWN cannot be same	This occurs when you try to configure FlexAttach with a similar value for pWWN and virtual pWWN.	Use different values for pWWN and virtual pWWN, as similar values for pWWN and virtual pWWN are not allowed.
Virtual port WWN already exists	This occurs when you try to configure an already defined pWWN to a different interface.	Use an undefined virtual pWWN for a new interface.