C H A P T E R **20**

SAN Device Virtualization

This chapter describes how to configure virtual devices to represent physical end devices for switches running Cisco MDS SAN-OS Release 3.1(2) and later.

Cisco SAN device virtualization (SDV) is a licensed feature included in the Cisco MDS 9000 Family Enterprise package (ENTERPRISE_PKG). See Chapter 3, "Obtaining and Installing Licenses," for details about acquiring licenses.

This chapter includes the following sections:

- About SDV, page 20-1
- Configuring SDV, page 20-4
- SDV Requirements and Guidelines, page 20-9
- SDV Configuration Example, page 20-12
- Displaying SDV Information, page 20-14
- Default Settings, page 20-14

About SDV

As of Cisco SAN-OS Release 3.1(2) and later, you can use Cisco SDV to create virtual devices that represent physical end-devices. Virtualization of SAN devices accelerates swapout or failover to a replacement disk array, and it also minimizes downtime when replacing host bus adapters (HBAs) or when re-hosting an application on a different server.

SAN devices that are virtualized can be either initiators or targets. You can virtualize targets to create a *virtual target*, and also virtualize initiators to create a *virtual initiator*. Such configurations do not distinguish between virtual initiators and virtual targets (see Figure 20-1 and Figure 20-2).

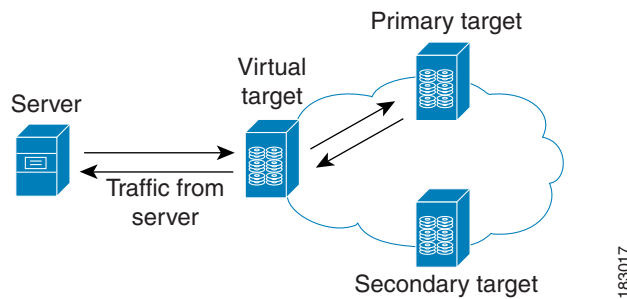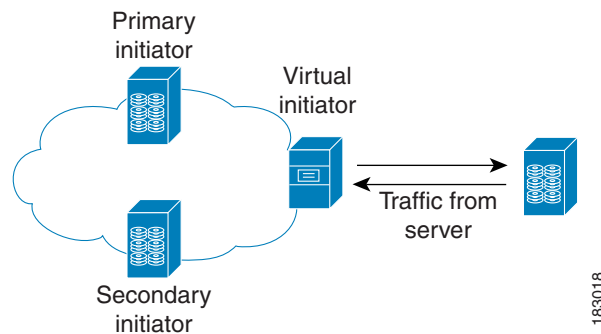*Figure 20-1   Target Virtualization*

Figure 20-2   Initiator Virtualization

Note   While most of the examples in this chapter describe target virtualization, the same behaviors apply to initiator virtualization as well.

Typically, today's deployments for handling device failures are designed for high availability (HA), with redundancy being a key part of this design. Let's consider the case where a target is designed to be redundant. Here, two arrays are deployed–a primary and secondary. Enterprises often use some type of consistency technology (such as EMF SRDF) between the primary and secondary arrays to ensure that the secondary is a mirrored copy of the production LUN. However, if the primary array fails, it must be replaced by the secondary, as all I/O must occur on the secondary. Problems can occur because the time required to bring the secondary array up and have it working often takes longer than most can afford (Figure 20-3 illustrates this dilemma).

Figure 20-3   Typical Deployment for Handling Device Failures Before SDV

If a storage array is replaced *without* using Cisco SDV, then it may require the following:

- Taking down a server to modify zoning and account for the new array.
- Changing the Cisco SAN-OS configuration to accommodate Fibre Channel IDs (FC IDs) and pWWNs of the new array.
- Changing a server configuration to accommodate the new FC IDs and pWWNs.

More specifically, without SDV you might experience the following:

- It can take a considerable amount of time to configure a secondary device for a typical production environment.
- In the zoning configuration, all the initiators must be re-zoned with the secondary device, and certain initiators must also be reconfigured. For example, the WWN and FC ID of the secondary device are different, so driver files must be changed and the server must be rebooted.
- Clustering (multiple initiators) compounds the problem, and the failover procedure must be repeated for each server of the cluster. Think of a server cluster as a set of HBAs–any storage array FC ID changes must be performed for each HBA.

SDV enables you to:

- Reduce the amount of time it takes for data migration, and ultimately the overall amount of downtime.
- Easily scale to larger numbers of devices.

Figure 20-4 illustrates the benefits of SDV. In this configuration, disk array Y replaces disk array X. When disk array X was deployed, the user created virtual devices for all the Fibre Channel interfaces using SDV. After data replication from disk array X was completed, the user briefly pauses activity on the application server and re-linked disk array Y to the virtual devices used by the server, completing the swapout of disk array X. No zoning changes or host operating system configuration changes were required during the time-critical period when the swap was performed; this significantly minimized application downtime.

> **Note** The array administrator will likely have to perform actions on array Y for it to become a primary device and accept server log ins before linking the virtual device to the array Y pWWN.

Figure 20-4   SDV Example
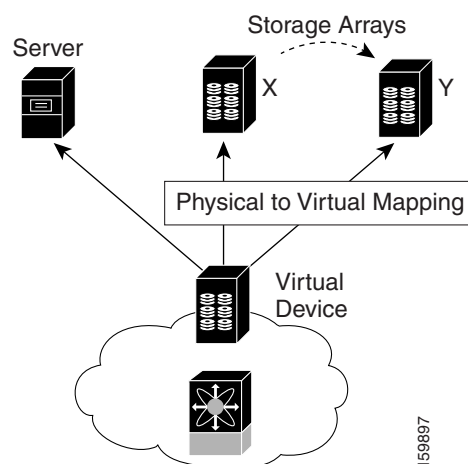
Key Concepts

The following terms are used throughout this chapter:

- Virtual device
  The virtualized or proxy representation of the real device, which is registered with the name server and has a pWWN and FC ID. A virtual device exists as long as its real (physical) counterpart is online. The virtual device pWWN and FC ID must be unique and cannot clash with any real device pWWNs and FC IDs.

- Virtual domain
  Reserved by SDV to assign FC IDs to virtual devices. If the switch that reserved the domain goes down, another switch takes over its role using the same domain.

Configuring SDV

SDV is a distributed service and uses CFS (Cisco Fabric Services) distribution to synchronize the databases. When you configure SDV it starts a CFS session and locks the fabric. When a fabric is locked, Cisco SAN-OS software does not allow any configuration changes from a switch–other than the switch holding the lock–and issues a message to inform users about the locked status. Configuration changes are held in a pending database for the application. You must perform a commit operation to make the configuration active and to release the lock for all switches. You can discard or stop changes from being distributed by issuing the **abort/clear** command.

See Chapter 6, "Using the CFS Infrastructure" for more details about CFS,

> **Note** When you enable SDV, CFS distribution is also enabled; CFS distribution cannot be disabled for SDV.

The following sections describe how to configure SDV:

- Configuring a Virtual Device, page 20-4
- Configuring a Zone for a Virtual Device, page 20-6
- Linking a Virtual Device with a Physical Device, page 20-8
- Configuring LUN Zone Members for SDV Devices, page 20-8
- Resolving Fabric Merge Conflicts, page 20-9

Configuring a Virtual Device

A virtual device is identified by an alphanumeric name of up to 32 characters and defines all the real devices (one primary and one or more secondary) that it represents. Upon the successful creation of a virtual device, the virtual device name is internally registered as the device alias name with the device alias database; the pWWN is automatically assigned by the system using Cisco OUI (Organizational Unique Identifier). A virtual device appears as a real, physical device. You can enumerate up to 128 devices for a virtual device. There is a limit of 4095 on the number of virtual devices that you can create in a single VSAN.
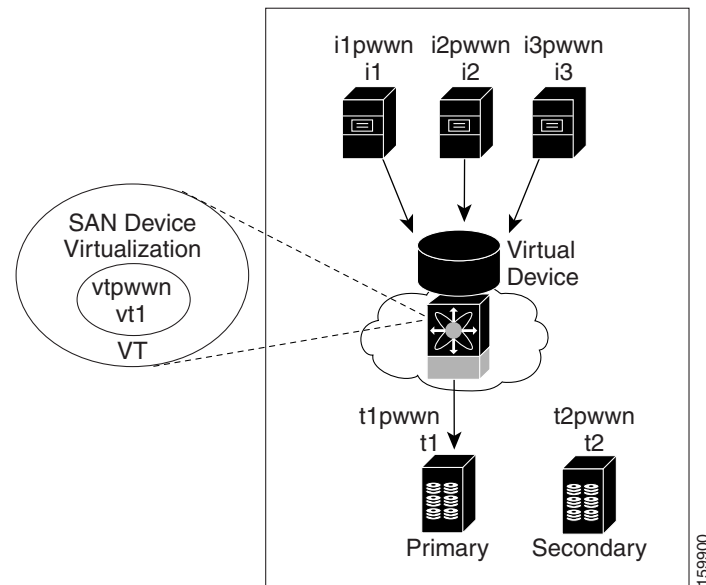
> **Note** For Cisco MDS SAN-OS Release 3.1(2) and later, SDV has been tested to work with up to 1024 virtual devices per VSAN.

Figure 20-5 shows a configuration that includes a new virtual device, vt1.

Figure 20-5   Creating a Virtual Device

To configure a virtual device and commit it to the fabric configuration, follow these steps:

	Command	Purpose
Step 1	switch# **config t** switch(config)#	Enters configuration mode.
Step 2	switch(config)# **sdv enable**	Enables the SDV feature. **Note** If you do not have the Cisco MDS 9000 Family Enterprise package license (ENTERPRISE_PKG) installed, this command will fail.
Step 3	switch(config)# **sdv virtual-device name vdev1 vsan 2**	Configures a virtual device alias name (vdev1). Enters SDV manager configuration submode.
Step 4	switch(config-sdv-virt-dev)# **pwwn 21:00:00:04:cf:cf:45:40 primary** switch(config-sdv-virt-dev)# **pwwn 21:00:00:04:cf:cf:38:d6**	Maps primary virtual device to the pWWN of real devices. Maps secondary virtual device to pWWN of a real device.
Step 5	switch(config-sdv-virt-dev)# **exit**	Exits SDV manager configuration submode.
Step 6	switch(config)# **sdv commit vsan 2**	Commits the VSAN configuration to the fabric.
Step 7	switch(config)# **exit** switch# **show device-alias database**	Verifies that the virtualized device is registered in the device alias database.
Step 8	switch# **show fcns database vsan vsan 2**	If the primary device is online, verifies that the virtual device appears in the name server database and has the correct FC4 type (Fibre Channel layer 4).

Configuring a Zone for a Virtual Device

After configuring a virtual device, you must create a zone that includes all the other real devices and the virtual device as members, and add this zone to a zone set, which you can activate. You can add the virtual device to the zone using the configured name and member type as the device alias.
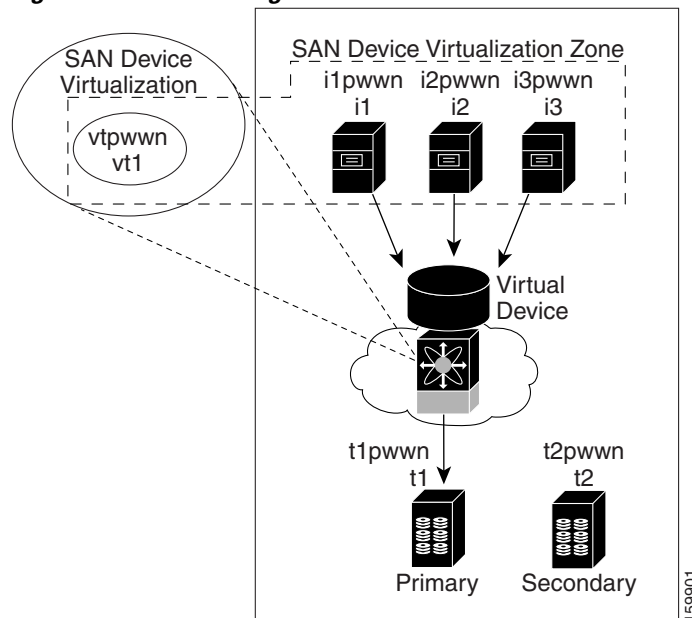
Note   This configuration process does not support interoperability. If you are working in interop-VSANs, we recommend that you configure the zone directly using the system-assigned pWWN of the virtual device.

Figure 20-6 shows a virtual device-name device alias (vt1) zoned with the real devices activated; the primary device is online.

Figure 20-6   Zoning the Virtual Device with Real Devices

To add the virtual device to a zone as a zone member, follow these steps:

Step 1	switch# **config t**	Enters configuration mode.
Step 2	switch(config)# **zoneset name zs1 vsan 1**	Groups zones under one zone set named zs1 in the VSAN, vsan 1. Enters zone set submode.
Step 3	switch(config-zoneset)# **zone name zone1**	Creates the zone zone1.
Step 4	switch(config-zoneset-zone)# **member device-alias vdev1** or switch (config-zoneset-zone)# **member pwwn 50:00:53:00:01:fa:70:09**	Adds a virtual device (vdev1) as device-alias type zone member to the zone. Alternatively, adds a virtual device as pWWN type zone member to the zone.
Step 5	switch(config-zoneset-zone)# **member pwwn 21:22:5:d8:15:11:8:8**	Adds the real device to the zone.
Step 6	switch(config-zoneset-zone)# **end**	Exits all modes.

Step 7	`switch# `**`show zoneset`**	Displays the zone sets configured for the VSAN, vsan 1, in step 4, as well as the primary and secondary pWWNs.
Step 8	`switch# `**`config t`**	Enters configuration mode.
Step 9	`switch(config)# `**`zoneset activate name zs1 vsan 1`**	Activates the new zone, zs1 for the VSAN, vsan1.
Step 10	`switch(config)# `**`end`**	Exits all modes.
Step 11	`switch# `**`show zoneset active vsan 1`**	Displays the active zone for the VSAN, vsan1. Confirms the zone set and zone names.

⚠

Caution   Set the device alias mode to **enhanced** when using SDV (because the pWWN of a virtual device could change).

For example, SDV is enabled on a switch and a virtual device is defined. SDV assigns a pWWN for the virtual device, and it is zoned based on the pWWN in a zone. If you later disable SDV, this configuration is lost. If you reenable SDV and create the virtual device using the same name, there is no guarantee that it will get the same pWWN again. Hence, you would have to rezone the pWWN-based zone. However, if you perform zoning based on the device-alias name, there are no configuration changes required if or when the pWWN changes.

Be sure you understand how device alias modes work before enabling them. Refer to Chapter 24, "Distributing Device Alias Services" for details and requirements about device alias modes.

Configuring a Virtual Device with a Static FC ID

Typically, the FC ID for the virtual device is assigned by the system. There is a virtual domain reserved and advertised for SDV devices and it is used when allocating the FC ID. The FC ID is registered with the name server and has the same FC4 properties as the primary device it represents.

When a fabric containing a virtual device configuration reboots, the virtual device's domain or FC ID may change; there is no guarantee that the virtual device FC ID will remain the same because it is not a part of the configuration. You can define the FC ID for a virtual device to be static. Configuring a device to have a static FC ID ensures that the same FC ID is used for the virtual device configuration across SAN-OS reboots, and it also provides the following benefits:

- Makes it easier for SAN administrators to plan and design the SAN (for example, administrators can assign virtual domains and FC IDs to be used by virtual devices).
- Improves monitoring and troubleshooting because the same FC ID is assigned to a virtual device on every instance.

✎

Note   This procedure is optional, but we recommend it. You can also configure a static domain ID. Only the domain part of the FC ID is static; other values are system-assigned.

If you are using SDV to virtualize devices for either AIX or HP-UX platforms, we recommend you create a static FC ID.

To configure a static FC ID when creating a virtual device, follow these steps:

	Command	Purpose
Step 1	switch# **config t** switch(config)#	Enters configuration mode.
Step 2	switch(config)# **sdv virtual-device name vdev1 vsan 2**	Configures a virtual device alias name (vdev1) and defines its primary device (vsan 2). Enters SDV manager configuration submode.
Step 3	switch(config-sdv-virt-dev)# **virtual-fcid 0x960001** switch(config-sdv-virt-dev)# **exit**	Assigns the FC ID 0x960001 to the virtual device. Exits SDV manager configuration submode.
Step 4	switch(config)# **sdv commit vsan 2**	Commits the VSAN configuration to the fabric.

Linking a Virtual Device with a Physical Device

After creating a virtual device and configuring it as part of a zone, you can define the primary device for it using the **link** command, which is also used to fail over to the secondary device.

Note When a link operation fails over to the secondary device, the virtual device is taken offline and then brought online.

Configuring LUN Zone Members for SDV Devices

You can configure LUN zone members for SDV devices. Following are the types of SDV LUN zone configurations for existing real devices and configured SDV devices:

- Real initiator (I1)
- Real target (T1) supporting LUNs from 0 to 12
- Virtual target (VT1) virtualizing the real target (T1)
- Virtual initiator (VI1) virtualizing real initiator (I1)

Real Initiator and SDV Virtual Target with LUN

In Example 20-1 a real initiator is zoned with an SDV virtual target (including the LUN).

Example 20-1   Real Initiator and SDV Virtual Target with LUN

```
zoneset name zs1 vsan 2
    zone name z1 vsan 2
        member device-alias I1
        member device-alias VT1 lun 0
        member device-alias VT2 lun 1
```

SDV Virtual Initiator and Real Target with LUN

In Example 20-2 an SDV virtual initiator is zoned with a real target (including the LUN).

Example 20-2   SDV Virtual Initiator and Real Target with LUN

```
zoneset name zs2 vsan 1
    zone name z2 vsan 1
        member device-alias VI1
        member device-alias T1 lun 0
        member device-alias T2 lun 1
```

SDV Virtual Initiator and SDV Virtual Target with LUN.

In Example 20-3 an SDV virtual initiator is zoned with an SDV virtual target (including the LUN).

Example 20-3

```
zoneset name zs3 vsan 1
    zone name z3 vsan 1
        member device-alias VI1
        member device-alias VT1 lun 0
        member device-alias VT2 lun 1
```

Resolving Fabric Merge Conflicts

Whenever two fabrics merge SDV merges its database. A merge conflict can occur when there is a run-time information conflict or configuration mismatch. Run-time conflicts can occur do to:

- Identical pWWNs being assigned to different virtual devices
- The same virtual devices are assigned different pWWNs.
- The virtual device and virtual FC ID are mismatched.

A *blank commit* is a commit operation that does not contain configuration changes, and enforces the SDV configuration of the committing switch fabric-wide. A blank commit operation resolves merge conflicts by pushing the configuration from the committing switch throughout the fabric, thereby reinitializing the conflicting virtual devices. Exercise caution while performing this operation, as it can easily take some virtual devices offline.

Merge failures resulting from a pWWN conflict can cause a failure with the device alias as well. A blank commit operation on a merge-failed VSAN within SDV should resolve the merge failure in the device alias.

You can avoid merge conflicts due to configuration mismatch by ensuring that:

- The pWWN and device alias entries for a virtual device are identical (in terms of primary and secondary).
- There are no virtual device name conflicts across VSANs in fabrics.

SDV Requirements and Guidelines

Be aware of the following requirements and guidelines as you plan and configure SDV:

- SDV should be enabled on switches where devices that are part of SDV zones are connected.
- SDV does not work for devices connected to non-MDS switches.
- Broadcast zoning is not supported for a zone with a virtual device.

- IVR and SDV cannot be used for the same device. In other words, a SDV-virtualized device cannot be part of a IVR zone or zoneset.

- Virtual device names should be unique across VSANs because they are registered with the device alias server, which is unaware of VSANs. For example, if you have enabled SDV and have registered a name, vt1 in both VSAN 1 and VSAN 2, then the device alias server cannot store both entries because they have the same name.

- You cannot specify the same primary device for different virtual devices.

- SDV does not work with soft zoning (*Soft zoning* means that zoning restrictions are applied only during interaction between the name server and the end device. If an end device somehow knows the FC ID of a device outside its zone, it can access that device), nor does it work with the **zone default-zone permit vsan** operation (which would otherwise permit or deny traffic to members in the default zone).

- If devices are not already zoned with the initiators, then you can configure SDV virtual device zones with no negative impact. If they are already zoned, then zoning changes are required.

- The real device-virtual device zone cannot coexist with the real device-real device zone. If the real devices are not already zoned together, then you can configure the real device-virtual device zone with no negative impact. If these devices are already zoned, then adding the real device-virtual device zone may cause the zone activation to fail. If this occurs, then you must delete one of the zones before activation.

  For example, a user attempts to create a configuration with zone A, which consists of I, the initiator, and T, the target (I,T), and zone B, which consists of a virtual initiator, VI, and real target, T (zone VI, T). Such a configuration would fail. Likewise, an attempt to configure zone C, which consists of an initiator, I, and target T, with zone D, which consists of an initiator, I, and virtual target, VT (zone I, VT), would also fail.

⚠️

Caution   There must be at least one SDV-enabled switch that is *not* a Cisco MDS 9124 Switch between the server and the device that are being virtualized. In other words, SDV does not work when initiators and primary devices are connected to the same Cisco MDS 9124 Switch.

Discarding Changes

At any time, you can discard the uncommitted changes to the running configuration and release the fabric lock (prior to issuing the **sdv commit** command). If you discard the pending changes, the configuration remains unaffected and the lock is released.

To discard uncommitted SDV configuration changes and release the lock follow these steps:

	Command	Purpose
Step 1	switch# **config t** switch(config)#	Enters configuration mode.
Step 2	switch(config)# **sdv abort vsan 10**	Discards the pending configuration changes.

S e n d   d o c u m e n t a t i o n   c o m m e n t s   t o   m d s f e e d b a c k - d o c @ c i s c o . c o m

Clearing SDV Changes

If you have performed a SDV task and have forgotten to release the lock by either committing or discarding the changes, an administrator can release the lock from any switch in the fabric. If the administrator performs this task, your changes to the pending database are discarded and the fabric lock is released.

Tip   The pending database is only available in the volatile directory and is subject to being discarded if the switch is restarted.

To use administrative privileges and release a locked SDV session, use the **clear sdv session** command in EXEC mode.

```
switch# clear sdv session vsan 2
```

Guidelines for Downgrading SDV

Prior to SAN-OS Release 3.1(3), SDV did not support virtual initiators and LUN zoning. Consequently, in SAN-OS Releases 3.1(3) and later, if virtual initiators are configured or SDV devices are configured as LUN-based members of a zone, a configuration check will indicate that downgrading to SAN-OS Release 3.1(2) may be disruptive and is therefore not recommended.

Downgrading With Virtual Initiators Configured

If SDV virtual initiators are configured, you will be unable to downgrade to SAN-OS release 3.1(2)..

This incompatibility is a *loose* one–it will only warn users before a downgrade. It is recommended that you remove the virtual initiator configuration or shut down the initiator port so that there are no inconsistencies in the downgraded version.

Note   We also recommend that users trigger a manual discovery on all the switches before configuring the virtual initiators; in fact, you can trigger the discovery before proceeding to the downgrade by entering the following commands.

```
switch# discover scsi-target local os all
discovery started
switch# discover scsi-target remote os all
discovery started
switch#
```

Downgrading with SDV LUN Zoning Configured

Following are the downgrade scenarios when SDV LUN zoning is configured:

- Real initiator and SDV virtual target with LUN
- SDV virtual initiator and real target with LUN
- SDV virtual initiator and SDV virtual target with LUN

In each of these cases, a configuration check is registered to prevent users from downgrading to SAN-OS Release 3.1(2). This incompatibility is a *strict* one (it will be disruptive if you proceed with the downgrade).

To avoid the configuration check, delete all the LUN zone members from SDV zones and then activate the zone set before the downgrade.

SDV Configuration Example

The following example shows all tasks (required and optional) associated with configuring SDV.

Step 1   Enter configuration mode.

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
```

Step 2   Enable SDV.

```
switch(config)# sdv enable
```

Step 3   Locate the pWWNs of the devices in the VSAN (vsan 2). Record the pWWNs, as you will need them in the next step when you create a virtual device.

```
switch(config)# do show fcns database vsan 2
VSAN 2:
--------------------------------------------------------------------------
FCID        TYPE  PWWN                    (VENDOR)       FC4-TYPE:FEATURE
--------------------------------------------------------------------------
0x9f0201    NL    21:00:00:04:cf:cf:45:40 (Seagate)      scsi-fcp
0x9f0423    NL    21:00:00:04:cf:cf:38:d6 (Seagate)      scsi-fcp

Total number of entries = 2
```

Step 4   Create a virtual device (vdev1) for the VSAN and specify both the primary and secondary pWWNs.

```
switch(config)# sdv virtual-device name vdev1 vsan 2
switch(config-sdv-virt-dev)# pwwn 21:00:00:04:cf:cf:45:40 primary
switch(config-sdv-virt-dev)# pwwn 21:00:00:04:cf:cf:38:d6
```

Step 5   Create a static FC ID for the target device.

```
switch(config-sdv-virt-dev)# virtual-fcid 0x960001
switch(config-sdv-virt-dev)# exit
```

Step 6   Commit the new virtual device configuration to the fabric.

```
switch(config)# sdv commit vsan 2
switch(config)# exit
```

Step 7   Enter the **show sdv database** command, which displays the primary and secondary virtual devices created when you configured the virtual device in the VSAN (vsan 2). Ensure that the virtual device name, pWWNs, and FC ID are correct.

```
switch# show sdv database vsan 2
virtual-device name vdev1 vsan 2
[ WWN:50:00:53:00:00:d2:e0:01 FCID:0x960001 Real-FCID:0x9f0201 ]
  virtual-fcid 0x960001
  pwwn 21:00:00:04:cf:cf:45:40 primary
  pwwn 21:00:00:04:cf:cf:38:d6
```

Step 8   Enter the **show device-alias database** command, which displays the contents of the device alias database. Ensure that the new virtual device name appears and that the name is correct.

```
switch# show device-alias database

device-alias name vdev1 pwwn 50:00:53:00:01:c9:70:01

Total number of entries = 1
```

Step 9   Create a zone member under one zone set named zs1 in the VSAN (vsan 2) and add the virtual target device to the new zone using the device alias member type. Before activating the new zone, display the zone-set information to ensure that the zone-set name, zone name, and pWWNs are correct.

```
switch# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# zoneset name zs1 vsan 2
switch(config-zoneset)# zone name zzz1
switch(config-zoneset-zone)# member device-alias vdev1
switch(config-zoneset-zone)# member pwwn 21:00:03:04:55:cf:d6:40
switch(config-zoneset-zone)# end
switch# show zoneset
zoneset name zs1 vsan 2
  zone name zzz1 vsan 2
    pwwn 50:00:53:00:01:c9:70:01 [vdev1]
    pwwn 21:00:03:04:55:cf:d6:40
```

Step 10   Activate the new zone configuration.

```
switch(config)# zoneset activate name zs1 vsan 2
Zoneset activation initiated. check zone status
switch(config)# exit
```

Step 11   Display the active zone-set to ensure the data in the new zone configuration is correct. Also confirm that the pWWNs are correct.

```
switch# show zoneset active vsan 2
zoneset name zs1 vsan 2
  zone name zzz1 vsan 2
  * fcid 0x211324 [pwwn 50:00:53:00:01:c9:70:01] [vdev1]
    pwwn 21:00:03:04:55:cf:d6:40

switch# show sdv database vsan 2
virtual-device name vdev1 vsan 2
[ WWN:50:00:53:00:00:d2:e0:01 FCID:0x960001 Real-FCID:0x9f0201 ]
  virtual-fcid 0x960001
  pwwn 21:00:00:04:cf:cf:45:40 primary
  pwwn 21:00:00:04:cf:cf:38:d6
```

Step 12   Enter the SDV manager configuration submode and display the pWWNs for the virtual devices in vsan 2. After making a note of the pWWNs, migrate the primary virtual device to the secondary device, link this new primary device to a physical device, and then commit this configuration to the fabric. Confirm that the secondary virtual device is now the primary.

```
switch# conf t

switch(config-sdv-virt-dev)# link pwwn 21:00:00:04:cf:cf:38:d6
switch(config-sdv-virt-dev)# exit
switch(config)# sdv commit vsan 2
switch(config)# do show sdv database vsan 2
virtual-device name vdev1 vsan 2
[ WWN:50:00:53:00:00:d2:e0:01 FCID:0x960001 Real-FCID:0x9f0423 ]
  virtual-fcid 0x960001
  pwwn 21:00:00:04:cf:cf:45:40
  pwwn 21:00:00:04:cf:cf:38:d6 primary
```

```
virtual-device name vdev2 vsan 2
[ WWN:50:00:53:00:00:0b:50:01 ]
```

Displaying SDV Information

To display the results of the last commit to the SDV database:

```
switch# show sdv session status vsan  1
Session Parameters for VSAN: 1
------------------------------
Last Action Time Stamp    : Fri Feb  2 10:17:20 2007
Last Action               : Commit
Last Action Result        : Success
Last Action Failure Reason : none
```

To display the results of the last CFS SDV fabric merge for a VSAN:

```
switch# show sdv merge status vsan  1
Merge Status for VSAN     : 1
----------------------------
Last Merge Time Stamp     : None
Last Merge State          : None
Last Merge Result         : SUCCESS
Last Merge Failure Reason: None [cfs_status: 0]
```

To display details about the SDV database:

```
switch# show sdv database vsan 2
virtual-device name vdev1 vsan 2
[ WWN:50:00:53:00:00:d2:e0:01 FCID:0x960001 Real-FCID:0x9f0201 ]
  virtual-fcid 0x960001
  pwwn 21:00:00:04:cf:cf:45:40 primary
  pwwn 21:00:00:04:cf:cf:38:d6
```

To display statistics about SDV for a VSAN:

```
switch# show sdv statistics vsan 1
VSAN ELS-CMD    Requests  Accepts  Rejects  Drops
-------------------------------------------------
1    ELS_PLOGI  54        54       0        0
1    ELS_RRQ     2         2       0        0
1    ELS_PRLI   54        54       0        0
```

Default Settings

Table 20-1 lists the default settings for SDV parameters.

Table 20-1   Default SDV Configuration Parameters

Parameters	Default
enable	disabled