DPVM

- Information About DPVM, on page 1
- Default Settings, on page 3
- Guidelines and Limitations for DPVM, on page 3
- Configuring DPVM, on page 3
- Verifying the DPVM Configuration, on page 10
- DPVM Example Configuration, on page 11
- Feature History, on page 13

Information About DPVM

You can use Dynamic Port VSAN Membership (DPVM) to dynamically assign VSAN membership to ports by assigning VSANs based on the device WWN. DPVM eliminates the need to reconfigure the port VSAN membership to maintain fabric topology when a host or storage device connection is moved between two Cisco SAN switches or two ports within a switch. It retains the configured VSAN regardless of where a device is connected or moved.

DPVM assignment is based on the port world wide name (pWWN) and node world wide name (nWWN). A DPVM database contains mapping information for each device pWWN/nWWN assignment and the corresponding VSAN. Cisco NX-OS checks the database during a device FLOGI and obtains the required VSAN details.

The pWWN identifies the host or device and the nWWN identifies a node that consists of multiple devices. You can assign any one of these identifiers or any combination of these identifiers to configure DPVM mapping. If you assign a combination, preference is given to the pWWN.

DPVM uses the Cisco Fabric Services (CFS) infrastructure to allow efficient database management and distribution.

DPVM Databases

The DPVM database consists of a series of device mapping entries. Each entry consists of a device pWWN or nWWN assignment along with the dynamic VSAN assigned. You can configure a maximum of 16,000 DPVM entries in the DPVM database. This database is global to the whole switch (and fabric) and is not maintained for each VSAN.

DPVM uses the following three databases:

Configuration (config) database

Stores all configuration changes when CFS distribution is disabled. Changes to this database are reflected in the active DPVM database when you activate the DPVM config database.

Active database

Represents the DPVM configuration that is currently active in the fabric.

Pending database

Stores all configuration changes when CFS distribution is enabled. Changes to this database are reflected in the config or active DPVM database when you commit the DPVM pending database.

Related Topics

Activating the DPVM Config Database, on page 6
Verifying the DPVM Configuration, on page 10

DPVM Database Distribution

DPVM can use CFS to distribute the database to all switches in the fabric to allow devices to move anywhere and keep the same VSAN membership.

Note You should enable CFS distribution on all switches in the fabric.

Using the CFS infrastructure, each DPVM server learns the DPVM database from each of its neighboring switches during the ISL bring-up process. If you change the database locally, the DPVM server notifies its neighboring switches, and that database is updated by all switches in the fabric.

When you enable CFS distribution for DPVM, the DPVM configuration database is copied into the DPVM pending database. All changes to the DPVM configuration are now stored in the DPVM pending database and the feature is locked (that is, no other switch can make changes to the DPVM database until you commit the changes or discard the changes and free the CFS lock).

The DPVM pending database includes the following changes:

- Adding, deleting, or modifying database entries.
- Activating, deactivating, or deleting the configuration database.
- Enabling or disabling autolearning.

CFS distributes these changes to all switches in a fabric when you commit the changes. You can also discard (abort) the changes at this point.

Database Merge

When you merge two independent fabrics into one fabric, DPVM attempts to merge the DPVM database (the configuration database and static (unlearned) entries in the active DPVM database). To ensure a successful database merge, follow these guidelines:

- Verify that the activation status and the auto-learn status is the same for both fabrics.
- Verify that the combined number of device entries in each database does not exceed 16000 entries.

Note If you do not follow these two conditions, the merge will fail. The next CFS distribution will forcefully synchronize the databases and the activation states in the fabric.

Related Topics

Displaying DPVM Database Merge Results, on page 9

Default Settings

Table 1: Default DPVM Parameter Settings

Parameters	Default
DPVM feature	Disabled
DPVM CFS distribution	Enabled
Autolearning	Disabled

Guidelines and Limitations for DPVM

DPVM has the following guidelines and limitations:

- You should enable DPVM CFS distribution for all switches in your fabric.
- Connect the dynamic device to an F-port on the switch.
- Verify that the static port VSAN of the F port is valid (not isolated, not suspended, and in existence).
- Verify that the dynamic VSAN configured for the device in the DPVM database is valid (not isolated, not suspended, and in existence).
- DPVM supports MAC-based device mapping for FCoE devices. DPVM does not support pWWN mapping for FCoE devices.

Note DPVM overrides any existing static port VSAN membership configuration. If the VSAN that corresponds to the dynamic port is deleted or suspended, the port is shut down.

Configuring DPVM

Enabling the DPVM Feature

You must enable the DPVM feature before you can configure DPVM.

SUMMARY STEPS

1. **config t**
2. **feature dpvm**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**feature dpvm** **Example:** `switch(config)# feature dpvm`	Enables the DPVM feature.
Step 3	(Optional) **show feature** **Example:** `switch(config)# show feature`	Displays the enabled or disabled state for each feature.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Adding Entries into the DPVM Database

You can manually add entries into the config and pending DPVM databases.

Note The DPVM pending database is stored in volatile memory. Changes are lost if the switch reboots. You should commit changes as soon as possible.

Before you begin

- Ensure that you have enabled the DPVM feature.
- Ensure that you have configured device aliases in enhanced mode if you want to configure device aliases in the DPVM database.

SUMMARY STEPS

1. **config t**
2. **dpvm database**
3. **pwwn** *pwwn* **vsan** *vsan-id*
4. **nwwn** *nwwn* **vsan** *vsan-id*
5. **device-alias** *alias* **vsan** *vsan-id*
6. **exit**
7. (Optional) **show dpvm pending-diff**
8. **dpvm commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	Required: **dpvm database**
	Example:
	`switch(config)# dpvm database`
	`switch(config-dpvm-db)#`
		Creates the DPVM config database and enters database configuration mode.

Step 3	Required: **pwwn** *pwwn* **vsan** *vsan-id* **Example:** `switch(config-dpvm-db)# pwwn 12:33:56:78:90:12:34:56 vsan 100`	Maps the configured pWWN to the VSAN. The *pwwn* is in pWWN dotted notation. The *vsan-id* range is from 1 to 4093.
Step 4	Required: **nwwn** *nwwn* **vsan** *vsan-id* **Example:** `switch(config-dpvm-db)# nwwn 14:21:30:12:63:39:72:81 vsan 101`	Maps the configured nWWN to the VSAN. The *nwwn* is in nWWN dotted notation. The *vsan-id* range is from 1 to 4093.
Step 5	Required: **device-alias** *alias* **vsan** *vsan-id* **Example:** `switch(config-dpvm-db)# device-alias device1 vsan 102`	Maps the configured device alias to the VSAN. The *alias* is any case-sensitive alphanumeric string up to 64 characters. The *vsan-id* range is from 1 to 4093.
Step 6	Required: **exit** **Example:** `switch(config-dpvm-db)# exit`	Exits DPVM database configuration mode.

	Command or Action	Purpose
Step 7	Required: (Optional) **show dpvm pending-diff** **Example:** `switch(config)# show dpvm pending`	(Optional) Displays the differences between the pending database and the config database. You can optionally discard these changes using the **dpvm abort** command.
Step 8	Required: **dpvm commit** **Example:** `switch(config)# dpvm commit`	Commits the DPVM pending database to the config database. This step is required to release the CFS lock on the DPVM configuration and to distribute this change across the fabric. You can optionally use the **dpvm abort** command to discard these changes and release the CFS lock.

What to do next

You should compare the DPVM config database to the active database and activate these changes.

Activating the DPVM Config Database

You can activate the DPVM config database to make it the active database. Activation might fail if conflicting entries are found between the DPVM config database and the currently active DPVM database. However, you can force activation to override conflicting entries.

To disable DPVM, you must explicitly deactivate the currently active DPVM database by entering the **no dpvm activate** command.

Before you begin

- Ensure that you have enabled the DPVM feature.

SUMMARY STEPS

1. **config t**
2. (Optional) **dpvm database diff config**
3. **dpvm activate**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	(Optional) **dpvm database diff config** **Example:**	Compares the DPVM config database to the current DPVM active database. Use this output to verify your config database changes before activating them. You can optionally

	Command or Action	Purpose
	`switch(config)# dpvm database diff config`	use the **dpvm database copy active** command to copy the active database into the config database to discard the old config database.
Step 3	Required: **dpvm activate** **Example:** `switch(config)# dpvm activate`	Copies the DPVM config database into the DPVM active database.

Related Topics

DPVM Databases, on page 1
Verifying the DPVM Configuration, on page 10

Clearing the DPVM CFS Session Lock

If you have performed a DPVM 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 DPVM pending database are discarded and the fabric lock is released.

SUMMARY STEPS

1. **clear dpvm session**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	Required: **clear dpvm session** **Example:** `switch# clear dpvm session`	Discards the DPVM pending database and releases the CFS lock.

Enabling Autolearning

You can configure the DPVM database to automatically learn (autolearn) about new devices within each VSAN. Autolearning is a two-part process. When you enable autolearning, DPVM creates learned entries by populating device pWWNs and VSANs in the active DPVM database. DPVM learns currently logged in devices as well as any new devices that log in while autolearning is enabled. These learned entries become permanent in the active DPVM database when you disable autolearning.

The following conditions apply to autolearning:

- If a device logs out while autolearn is enabled, that entry is automatically deleted from the active DPVM database.
- If the same device logs multiple times into the switch through different ports, the VSAN that corresponds to last login is remembered

- Learned entries do not override previously configured and activated entries.

Before you begin

- Ensure that the active DPVM database is already available.

SUMMARY STEPS

1. **configure terminal**
2. **dpvm auto-learn**
3. (Optional) **show dpvm ports** [**vsan** *vsan-id*]
4. **no dpvm auto-learn**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**dpvm auto-learn** **Example:** `switch(config)# dpvm auto-learn`	Enables autolearning. Disable autolearning after a period of time to make the learned entries permanent in the active DPVM database.
Step 3	(Optional) **show dpvm ports** [**vsan** *vsan-id*] **Example:** `switch(config)# show dpvm ports vsan 3`	Displays dynamic (autolearned) entries.
Step 4	**no dpvm auto-learn** **Example:** `switch(config)# no dpvm auto-learn`	Disables autolearning. Any learned entries become permanent in the active DPVM database.

Clearing Autolearned Entries

If DPVM autolearning is enabled, you can clear any or all learned entries from the active DPVM database.

Note Clearing autolearned entries does not initiate a CFS session and can only be configured on the local switch.

Before you begin

- Ensure that DPVM autolearning is enabled.

SUMMARY STEPS

1. **clear dpvm auto-learn pwwn** *pwwn*
2. **clear dpvm auto-learn**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**clear dpvm auto-learn pwwn** *pwwn* **Example:** `switch# clear dpvm auto-learn pwwn 55:22:33:44:55:66:77:88`	Clears an individual autolearned entry.
Step 2	**clear dpvm auto-learn** **Example:** `switch# clear dpvm auto-learn 8`	Clears all autolearned entries.

Displaying DPVM Database Merge Results

When you merge two independent fabrics, DPVM attempts to merge the associated DPVM databases. You can review the results of this database merge to determine if it succeeded or failed.

SUMMARY STEPS

1. **show dpvm merge status**
2. **show dpvm merge statistics**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	Required: **show dpvm merge status** **Example:** `switch(config)# show dpvm merge status`	Displays information about the last DPVM database merge event.
Step 2	Required: **show dpvm merge statistics** **Example:** `switch(config)# show dpvm merge statistics`	Displays statistics about the last DPVM database merge.

Example

The following example shows conflicts in the DPVM database merge:

```
switch# show dpvm merge status
Last Merge Time Stamp     : Fri March 25 15:46:36 2011
Last Merge State          : Fail
Last Merge Result         : Fail
Last Merge Failure Reason : DPVM DB conflict found during merge [cfs_status: 76]
              Last Merge Failure Details: DPVM merge failed due to database conflict
Local Switch WWN          : 20:00:00:0d:ec:24:e5:00
Remote Switch WWN         : 20:00:00:0d:ec:09:d5:c0

------------------------------------------------------------------------
        Conflicting DPVM member(s)                    Loc VSAN   Rem VSAN
------------------------------------------------------------------------
dev-alias dpvm_dev_alias_1 [21:00:00:04:cf:cf:45:ba]    1313       1414
dev-alias dpvm_dev_alias_2 [21:00:00:04:cf:cf:45:bb]    1313       1414
dev-alias dpvm_dev_alias_3 [21:00:00:04:cf:cf:45:bc]    1313       1414
[Total 3 conflict(s)]
switch#
```

Related Topics

Database Merge, on page 2

Verifying the DPVM Configuration

To display the DPVM configuration, perform one of the following tasks:

Command	Purpose
show dpvm status	Displays the status for the DPVM configuration.
show dpvm database [**active**]	Displays information about DPVM databases.
show dpvm merge {**status** | **statistics**}	Displays information the last DPVM merge event.
show dpvm pending [**activation**]	Displays information about the DPVM pending database.
show dpvm pending-diff	Displays the differences between the pending database and the config database.
show dpvm ports [**vsan** *vsan-id*]	Displays information about the dynamic ports associated with a VSAN.
show dpvm session status	Displays information about DPVM CFS session.

Related Topics

DPVM Databases, on page 1
Activating the DPVM Config Database, on page 6

DPVM Example Configuration

This example shows how to configure a basic DPVM configuration.

SUMMARY STEPS

1. Enable DPVM and DPVM CFS distribution.
2. Activate the DPVM database.
3. Enable autolearning.
4. Access other switches in the fabric to verify the DPVM configuration.
5. Disable autolearning.
6. Access other switches in the fabric to verify the DPVM configuration.

DETAILED STEPS

Procedure

Step 1 Enable DPVM and DPVM CFS distribution.

Example:

```
switch1# config t
Enter configuration commands, one per line.  End with CNTL/Z.
switch1(config)# feature dpvm
switch1(config)# show dpvm status
No active DB, auto-learn is off, distribution is enabled,
Duplicated pwwn will be  Rejected.
```

DPVM is enabled but the active database is empty.

Step 2 Activate the DPVM database.

Example:

```
 switch1(config)# dpvm activate
switch1(config)# dpvm commit
switch1(config)# show dpvm database active
switch1(config)#
```

DPVM is enabled but the active database is empty.

Step 3 Enable autolearning.

Example:

```
 switch1(config)# dpvm auto-learn
switch1(config)# dpvm commit
switch1(config)# show dpvm database active
pwwn 21:00:00:e0:8b:0e:74:8a vsan 4(*)
pwwn 21:01:00:e0:8b:2e:87:8a vsan 5(*)
[Total 2 entries]
* is auto-learnt entry

switch1# show dpvm ports
```

```
-------------------------------------------------------------
Interface   Vsan        Device pWWN       Device nWWN
-------------------------------------------------------------
fc1/24       4   21:00:00:e0:8b:0e:74:8a  20:00:00:e0:8b:0e:74:8a
fc1/27       5   21:01:00:e0:8b:2e:87:8a  20:01:00:e0:8b:2e:87:8a
switch1# show flogi database
--------------------------------------------------------------------------------
INTERFACE  VSAN    FCID            PORT NAME               NODE NAME
--------------------------------------------------------------------------------
fc1/24     4     0xe70100  21:00:00:e0:8b:0e:74:8a  20:00:00:e0:8b:0e:74:8a
fc1/27     5     0xe80100  21:01:00:e0:8b:2e:87:8a  20:01:00:e0:8b:2e:87:8a

Total number of flogi = 2.

switch1# show dpvm status
DB is activated successfully, auto-learn is on
```

The currently logged in devices (and their current VSAN assignment) populate the active DPVM database. However, these autolearned entries are not permanent in the active DPVM database.

The output of the **show dpvm ports** and the **show flogi database** commands display two other devices that have logged in (referred to as switch9 and switch3 in this sample configuration).

Step 4 Access other switches in the fabric to verify the DPVM configuration.

Example:

```
 switch9# show dpvm database active
pwwn 21:00:00:e0:8b:0e:87:8a vsan 1(*)
pwwn 21:01:00:e0:8b:2e:74:8a vsan 1(*)
[Total 2 entries]
* is auto-learnt entry

switch9# show dpvm status
DB is activated successfully, auto-learn is on


switch3# show dpvm database active
pwwn 21:00:00:e0:8b:0e:76:8a vsan 1(*)
pwwn 21:01:00:e0:8b:2e:76:8a vsan 1(*)
[Total 2 entries]
* is auto-learnt entry

switch3# show dpvm status
DB is activated successfully, auto-learn is on
```

The autolearned entries show up in the active database for other switches in the fabric.

Step 5 Disable autolearning.

Example:

```
 switch1(config)# no dpvm auto-learn
switch1(config)# dpvm commit

switch1# show dpvm status
DB is activated successfully, auto-learn is off
switch1# show dpvm database active
pwwn 21:00:00:e0:8b:0e:74:8a vsan 4
pwwn 21:01:00:e0:8b:2e:87:8a vsan 5
```

```
pwwn 21:00:00:e0:8b:0e:87:8a vsan 1
pwwn 21:01:00:e0:8b:2e:74:8a vsan 1
pwwn 21:00:00:e0:8b:0e:76:8a vsan 1
pwwn 21:01:00:e0:8b:2e:76:8a vsan 1
[Total 6 entries]
* is auto-learnt entry

switch1# show dpvm status
DB is activated successfully, auto-learn is off
```

The autolearned entries are now permanent in the active DPVM database.

Step 6 Access other switches in the fabric to verify the DPVM configuration.

Example:

```
 switch9# show dpvm database active
pwwn 21:00:00:e0:8b:0e:87:8a vsan 1
pwwn 21:01:00:e0:8b:2e:74:8a vsan 1
pwwn 21:00:00:e0:8b:0e:76:8a vsan 1
pwwn 21:01:00:e0:8b:2e:76:8a vsan 1
pwwn 21:00:00:e0:8b:0e:74:8a vsan 4
pwwn 21:01:00:e0:8b:2e:87:8a vsan 5
[Total 6 entries]
* is auto-learnt entry

switch9# show dpvm status
DB is activated successfully, auto-learn is off

switch3# show dpvm database active
pwwn 21:00:00:e0:8b:0e:76:8a vsan 1
pwwn 21:01:00:e0:8b:2e:76:8a vsan 1
pwwn 21:00:00:e0:8b:0e:87:8a vsan 1
pwwn 21:01:00:e0:8b:2e:74:8a vsan 1
pwwn 21:00:00:e0:8b:0e:74:8a vsan 4
pwwn 21:01:00:e0:8b:2e:87:8a vsan 5
[Total 6 entries]
* is auto-learnt entry

switch3# show dpvm status
DB is activated successfully, auto-learn is off
```

The autolearned entries show up in the active database for other switches in the fabric.

Feature History

Table 2: Feature History for DPVM

Feature Name	Releases	Feature Information
DPVM	5.2(1)	This feature was introduced.