



Distributing Device Alias Services

All switches in the Cisco MDS 9000 Family support Distributed Device Alias Services (device alias) on a per-VSAN basis and on a fabric-wide basis. Device alias distribution allows you to move host bus adapters (HBAs) between VSANs without manually reentering alias names.

This chapter includes the following sections:

- [About Device Aliases, page 5-79](#)
- [About Device Alias Modes, page 5-79](#)
- [Device Alias Databases, page 5-83](#)
- [About Legacy Zone Alias Configuration Conversion, page 5-88](#)
- [Device Alias Statistics Cleanup, page 5-90](#)
- [Device Alias Configuration Verification, page 5-90](#)
- [Default Settings, page 5-92](#)
- [Resolving Device Alias Merge Failures, page 5-93](#)

About Device Aliases

When the port WWN (pWWN) of a device must be specified to configure different features (zoning, QoS, port security) in a Cisco MDS 9000 Family switch, you must assign the correct device name each time you configure these features. An incorrect device name may cause unexpected results. You can avoid this problem if you define a user-friendly name for a port WWN and use this name in all of the configuration commands as required. These user-friendly names are referred to as *device aliases* in this chapter.

About Device Alias Modes

Device alias supports two modes: basic and enhanced mode.

- When device alias runs in the basic mode, all applications function like the applications on the 3.0 switches. When you configure the basic mode using device aliases, the application immediately expands to pWWNs. This operation continues until the mode is changed to enhanced.

- When device alias runs in the enhanced mode, all applications accept the device-alias configuration in the native format. The applications store the device alias name in the configuration and distribute it in the device alias format instead of expanding to pWWN. The applications track the device alias database changes and take actions to enforce it.

A native device-alias configuration is not accepted in the interop mode VSAN. IVR zoneset activation will fail in interop mode VSANs if the corresponding twilight zones being injected are native device alias members.

- When the device-alias is in basic mode, when you try to add a device-alias member to a zone, it will be added as a PWWN member and not as a device-alias member. Hence when you change the PWWN for device-alias entry it will not get updated. You have to manually edit the zones containing that device alias by removing the old entry and reconfiguring the zones with the same device alias and then activating it. The update happens in Enhanced device-alias mode. In this mode, since the configuration is accepted in the native form, when the pwwn for the device-alias is changed, the zones containing that device-alias are automatically updated with the new pwwn.

Changing Mode Settings

When the device alias mode is changed from basic to enhanced mode, the applications are informed about the change. The applications start accepting the device alias-based configuration in the native format.



Note

Because the device alias was previously running in the basic mode, the applications do not have any prior native device alias configuration.

The applications check for an existing device alias configuration in the native format. If the device alias is in the native format, the applications reject the request and device alias mode cannot be changed to basic.

All native device alias configurations (both on local and remote switches) must be explicitly removed, or all device alias members must be replaced with the corresponding pWWN before changing the mode back to basic.

Device Alias Mode Distribution

If the device alias distribution is turned on, it is distributed to the other switches in the network whenever there is a change in the mode. You cannot change the mode from basic to enhanced unless all the switches are upgraded to Release 3.1. The device alias enhancements will not apply unless the entire fabric is upgraded to Release 3.1.



Note

When all the switches are upgraded to Release 3.1, you cannot automatically convert to enhanced mode. You do not need to change to enhanced mode, you can continue working in the basic mode.

Merging Device Alias

If two fabrics are running different device alias modes and are joined together, the device alias merge will fail. There is no automatic conversion of one mode to the other during the merge process. You will need to resolve the issue.

**Note**

Release 3.0 switches run in basic mode.

At the application level, a merger takes place between the applications and the fabric. For example, zone merge occurs when the E port is up and the IVR/PSM/DPVM merge occurs due to CFS. This merge is completely independent of the device alias merge.

If the application running on an enhanced fabric has a native device alias configuration, the application must fail the merge. The application has to fail the merge even though the other fabric is can support the native device alias-based configuration, but is running in the basic mode. You will need to resolve the issue. Once the device alias merge issue is resolved, each application must be fixed accordingly.

The following issue occurs when there is a device-alias database mismatch in the switches that are part of the same fabric.

The device-alias associated to PWWN is present in the port security/DPVM database even if the respective device-alias member is not present in the switch. The device-alias associated to PWWN is missing in the port security/DPVM database even if device-alias member is present in the switch.

Resolving Merge and Device Alias Mode Mismatch

If two fabrics are running in different modes and the device alias merge fails between the fabrics, the conflict can be resolved by selecting one mode or the other. If you choose the enhanced mode, ensure that all the switches are running at least the Release 3.1 version. Otherwise, the enhanced mode cannot be turned on. If you choose the basic mode, the applications running on the enhanced fabric have to comply with the device alias merge.

The device alias merge fails because of mode mismatch, but the application merge succeeds if it does not have any native device alias configurations.

If the native device alias configuration is attempted on an application from a Release 3.1 switch, the commit must be rejected because of device alias mode mismatch on some of the applications.

**Note**

The applications should not accept any native device alias configuration over SNMP if the device alias is running in the basic mode on that particular switch.

**Note**

Confcheck will be added when the enhanced mode is turned on and removed when it is turned off. Applications have to add confcheck if they have a device alias configuration in the native format. They have to remove confcheck once the configuration is removed.

Device Alias Features

Device aliases have the following features:

- The device alias information is independent of your VSAN configuration.
- The device alias configuration and distribution is independent of the zone server and the zone server database.
- You can import legacy zone alias configurations without losing data.

- The device alias application uses the Cisco Fabric Services (CFS) infrastructure to enable efficient database management and distribution. Device aliases use the coordinated distribution mode and the fabric-wide distribution scope (refer to the *Cisco MDS 9000 Family NX-OS System Management Configuration Guide*).
- When you configure zones, IVR zones, or QoS features using device aliases, and if you display these configurations, you will automatically see that the device aliases are displayed along with their respective pWWNs.

Device Alias Requirements

Device aliases have the following requirements:

- You can only assign device aliases to pWWNs.
- The mapping between the pWWN and the device alias to which it is mapped must have a one-to-one relationship. A pWWN can be mapped to only one device alias and vice versa.
- A device alias name is restricted to 64 alphanumeric characters and may include one or more of the following characters:
 - a to z and A to Z
 - 1 to 9
 - - (hyphen) and _ (underscore)
 - \$ (dollar sign) and ^ (up caret)



Note

If the device-alias name is 64 characters in length, the DPVM and other application databases do not update properly. Restrict the number of characters in the device-alias name to 63.

Zone Aliases Versus Device Aliases

[Table 5-1](#) compares the configuration differences between zone-based alias configuration and device alias configuration.

Table 5-1 Comparison Between Zone Aliases and Device Aliases

Zone-Based Aliases	Device Aliases
Aliases are limited to the specified VSAN.	You can define device aliases without specifying the VSAN number. You can also use the same definition in one or more VSANs without any restrictions.
Zone aliases are part of the zoning configuration. The alias mapping cannot be used to configure other features.	Device aliases can be used with any feature that uses the pWWN.
You can use any zone member type to specify the end devices.	Only pWWNs are supported along with new device aliases such as IP addresses.
Configuration is contained within the Zone Server database and is not available to other features.	Device aliases are not restricted to zoning. Device alias configuration is available to the FCNS, zone, fcping, traceroute, and IVR applications.

Table 5-1 Comparison Between Zone Aliases and Device Aliases (continued)

Zone-Based Aliases	Device Aliases
FC aliases are not displayed with the associated WWNs in the show command outputs like show zoneset active, show flogi database, and show fcns database.	Device aliases are displayed with the associated WWNs in the show command outputs like show zoneset active, show flogi database, and show fcns database.
FC aliases are not distributed as part of active zoneset and are only distributed as part of full zone database as per the FC standards.	Device Aliases are distributed through CFS.

Device Alias Databases

The device alias feature uses two databases to accept and implement device alias configurations.

- Effective database—The database currently used by the fabric.
- Pending database—Your subsequent device alias configuration changes are stored in the pending database.

If you modify the device alias configuration, you need to commit or discard the changes as the fabric remains locked during this period.

This section includes the following sections:

- [Creating Device Aliases, page 5-83](#)
- [About Device Alias Distribution, page 5-84](#)
- [About Creating a Device Alias, page 5-84](#)
- [About Device Alias Configuration Best Practices, page 5-85](#)
- [Committing Changes, page 5-86](#)
- [Discarding Changes, page 5-87](#)
- [About Legacy Zone Alias Configuration Conversion, page 5-88](#)
- [Disabling and Enabling Device Alias Distribution, page 5-88](#)

Creating Device Aliases

To create a device alias in the pending database, follow these steps:

	Command	Purpose
Step 1	switch# config t switch(config)#	Enters configuration mode.
Step 2	switch(config)# device-alias database switch(config-device-alias-db)#	Enters the pending database configuration submode.

	Command	Purpose
Step 3	<code>switch(config-device-alias-db)# device-alias name Device1 pwn 21:01:00:e0:8b:2e:80:93</code>	Specifies a device name (Device1) for the device that is identified by its pWWN. Starts writing to the pending database and simultaneously locks the fabric as this is the first-issued device alias configuration command.
	<code>switch(config-device-alias-db)# no device-alias name Device1</code>	Removes the device name (Device1) for the device that is identified by its pWWN.
	<code>switch(config-device-alias-db)# device-alias rename Device1 Device2</code>	Renames an existing device alias (Device1) with a new name (Device2).

To display the device alias configuration, use the **show device-alias name** command.

```
switch# show device-alias name x
device-alias name x pwn 21:01:00:e0:8b:2e:80:93
```

About Device Alias Distribution

By default, device alias distribution is enabled. The device alias feature uses the coordinated distribution mechanism to distribute the modifications to all switches in a fabric.

If you have not committed the changes and you disable distribution, then a commit task will fail.

Example 5-1 Displays a Failed Status

```
switch# show device-alias status
Fabric Distribution: Disabled
Database:- Device Aliases 25
Status of the last CFS operation issued from this switch:
=====
Operation: Commit
Status: Failed (Reason: Operation is not permitted as the fabric distribution is currently disabled.)
```



Note

From the Cisco MDS NX-OS Release 6.2.9 onwards, the ASCII configuration replay takes longer time for DDAS (Distributing Device Alias Services) without the **write erase** command.

About Creating a Device Alias

When you perform the first device alias task (regardless of which device alias task), the fabric is automatically locked for the device alias feature. Once you lock the fabric, the following situations apply:

- No other user can make any configuration changes to this feature.
- A copy of the effective database is obtained and used as the pending database. Modifications from this point on are made to the pending database. The pending database remains in effect until you commit the modifications to the pending database or discard (**abort**) the changes to the pending database.

About Device Alias Configuration Best Practices

As a part of the device-alias configuration best practices, the following guidelines need to be adopted within a device-alias session:

If a device-alias name is reused while configuring a **rename** command, then the command fails and gets moved to the rejected list.

Example 5-2 Displays the rejected device-alias command

```
switch(config-device-alias-db)# device-alias name dev10 pwnn 10:10:10:10:10:10:10:10
switch(config-device-alias-db)# device-alias rename dev10 new-dev10
Command rejected. Device-alias reused in current session :dev10
Please use 'show device-alias session rejected' to display the rejected set of commands
and for the device-alias best-practices recommendation.
switch(config-device-alias-db)#
```

If a PWWN is reused while configuring an **add** or **delete** command, then the command fails and gets moved to the rejected list.

Example 5-3 Displays the rejected device-alias command

```
switch(config-device-alias-db)# device-alias name dev11 pwnn 11:11:11:11:11:11:11:11
switch(config-device-alias-db)# no device-alias name dev11
Command rejected. Pwnn reused in current session: 11:11:11:11:11:11:11:11 is mapped to
device-alias dev11
Please use 'show device-alias session rejected' to display the rejected set of commands
and for the device-alias best-practices recommendation.
switch(config-device-alias-db)#
```

If a device-alias name is reused in an **add** command which was earlier being renamed in a **rename** command, the command fails and gets moved to the rejected list.

```
switch(config-device-alias-db)# device-alias rename da3 new-da3
switch(config-device-alias-db)# device-alias name da3 pwnn 2:2:2:2:3:3:3:3
Command rejected. Device-alias name reused in current session: da3
Please use 'show device-alias session rejected' to display the rejected set of commands
and for the device-alias best-practices recommendation.
switch(config-device-alias-db)#
```

Example 5-4 Displays the rejected device-alias command

The rejected set of commands can be displayed using the **show device-alias session rejected** command.

```
switch(config-device-alias-db)# show device-alias session rejected
To avoid command rejections, within a device alias session
Do not reuse:
a) a device alias name while configuring a rename command
b) a PWWN while configuring an add or delete command
c) a device alias name already renamed while configuring add command
```

Rejected commands must be committed in a separate device alias session which may cause traffic interruption for those devices. Plan accordingly. Refer to this command in the NX-OS Command Reference Guide for more information about device alias configuration best practices

```
Rejected Command List
-----
device-alias rename dev10 new-dev10
no device-alias name dev11
```

```
device-alias name da3 pwnn 02:02:02:02:03:03:03:03
switch(config-device-alias-db) # #
```

Committing Changes

If you commit the changes made to the pending database, the following events occur:

1. The pending database contents overwrites the effective database contents.
2. The pending database is emptied of its contents.
3. The fabric lock is released for this feature.

To commit the changes, follow these steps:

	Command	Purpose
Step 1	switch# config t switch(config)#	Enters configuration mode.
Step 2	switch(config)# device-alias commit	Commits the changes made to the currently active session.

Whenever a switch in the fabric attains a lock and goes for a blank commit, the following warning is thrown out:

```
WARNING: Device-alias DB is empty in this switch.
Initiating a commit from this switch will clear [wipe out] Device-alias DB across all the
switches in the fabric, losing Device-alias full DB config permanently.
Do you want to continue? (y/n) [n]
```



Note

Once the "**device-alias commit**" is done the running configuration has been modified on all switches participating in device-alias distribution. You can then use the "**copy running-config startup-config fabric**" command to save the running-config to the startup-config on all the switches in the fabric.

Enabling the Device Alias Pending Diff Display

To enable the display of the pending-diff and the subsequent confirmation on issuing a device-alias commit, follow these steps:

	Command	Purpose
Step 1	switch# config t switch(config)#	Enters configuration mode.
Step 2	switch(config)# device-alias confirm-commit	Enables the confirm commit option for device-alias.
Step 3	switch(config)# device-alias commit The following device-alias changes are about to be committed + device-alias name Device1 pwnn 21:01:00:e0:8b:2e:80:93 Do you want to continue? (y/n) [n] y	If the device-alias confirm-commit command is enabled, on committing the pending database, the pending-diff is displayed on the console and user is prompted for Yes or No. If the device -alias confirm-commit command is disabled, the pending-diff is not displayed and the user is not prompted for Yes or No.

Discarding Changes

If you discard the changes made to the pending database, the following events occur:

1. The effective database contents remain unaffected.
2. The pending database is emptied of its contents.
3. The fabric lock is released for this feature.

To discard the device alias session, perform this task:

	Command	Purpose
Step 1	switch# config t switch(config)#	Enters configuration mode.
Step 2	switch(config)# device-alias abort	Discards the currently active session.

To display the status of the discard operation, use the show **device alias status** command.

```
switch# show device-alias status
Fabric Distribution: Enabled
Database:- Device Aliases 24
Status of the last CFS operation issued from this switch:
=====
Operation: Abort
Status: Success
```

Fabric Lock Override

If you have performed a device alias 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 changes are only available in the volatile directory and are subject to being discarded if the switch is restarted.

To clear device-alias session, use the **clear device-alias session** command in CONFIGURATION mode.

```
switch(config)# clear device-alias session
```

To verify the status of the clear operation, use the **show device-alias session status** command.

```
switch(config)# show device-alias session status
Last Action Time Stamp      : None
Last Action                  : None
Last Action Result          : None
Last Action Failure Reason  : none
```

Clearing Database Content

To clear all the database content, use the **clear device-alias database** command in CONFIGURATION mode.

```
switch(config)# clear device-alias database
```

To verify the status of the **clear device-alias database** command, use the **show device-alias database** command.

```
switch(config)# show device-alias database
```

Clearing Statistics

To clear all the statistics, use the **clear device-alias statistics** command in CONFIGURATION mode.

```
switch# clear device-alias statistics
```

Disabling and Enabling Device Alias Distribution

To disable or enable the device alias distribution, follow these steps:

	Command	Purpose
Step 1	switch# config t switch(config)#	Enters configuration mode.
Step 2	switch(config)# no device-alias distribute	Disables the distribution.
	switch(config)# device-alias distribute	Enables the distribution (default).

To display the status of device alias distribution, use the **show device-alias status** command (see [Example 5-5](#) and [Example 5-6](#)).

Example 5-5 Displays Device Alias Status When Distribution Is Enabled

```
switch# show device-alias status
Fabric Distribution: Enabled <-----Distribution is enabled
Database:-Device Aliases 24
Locked By:-User "Test" SWWN 20:00:00:0c:cf:f4:02:83<-Lock holder's user name and switch ID
Pending Database:- Device Aliases 24
Status of the last CFS operation issued from this switch:
=====
Operation: Enable Fabric Distribution
Status: Success
```

Example 5-6 Displays Device Alias Status When Distribution Is Disabled

```
switch# show device-alias status
Fabric Distribution: Disabled
Database:- Device Aliases 24
Status of the last CFS operation issued from this switch:
=====
Operation: Disable Fabric Distribution
Status: Success
```

About Legacy Zone Alias Configuration Conversion

You can import legacy zone alias configurations to use this feature without losing data, if they satisfy the following restrictions:

- Each zone alias has only one member.
- The member type is pWWN.
- The name and definition of the zone alias should not be the same as any existing device alias name.

If any name conflict exists, the zone aliases are not imported.



Tip

Ensure to copy any required zone aliases to the device alias database as required by your configuration.

When an import operation is complete, the modified alias database is distributed to all other switches in the physical fabric when you perform the **commit** operation. At this time if you do not want to distribute the configuration to other switches in the fabric, you can perform the **abort** operation and the merge changes are completely discarded.

This section includes the following topics:

- [Importing a Zone Alias, page 5-89](#)
- [Device Alias Statistics Cleanup, page 5-90](#)

Importing a Zone Alias

To import the zone alias for a specific VSAN, follow these steps:

	Command	Purpose
Step 1	switch# config t switch(config)#	Enters configuration mode.
Step 2	switch(config)# device-alias import fcalias vsan 3	Imports the fcalias information for the specified VSAN.

To display device alias information in zone sets, use the **show zoneset** command (see [Example 5-7](#) and [Example 5-8](#)).

Example 5-7 Displays the Device Aliases in the Zone Set Information

```
switch# show zoneset
zoneset name s1 vsan 1
  zone name z1 vsan 1
    pwn 21:01:00:e0:8b:2e:80:93 [x] <-----Device alias displayed for each pWWN.
    pwn 21:00:00:20:37:39:ab:5f [y]
  zone name z2 vsan 1
    pwn 21:00:00:e0:8b:0b:66:56 [SampleName]
    pwn 21:00:00:20:37:39:ac:0d [z]
```

Example 5-8 Displays the Device Aliases in the Active Zone Set

```
switch# show zoneset active
zoneset name s1 vsan 1
  zone name z1 vsan 1
    * fcid 0x670100 [pwn 21:01:00:e0:8b:2e:80:93] [x]
    pwn 21:00:00:20:37:39:ab:5f [y]

  zone name z2 vsan 1
    * fcid 0x670200 [pwn 21:00:00:e0:8b:0b:66:56] [SampleName]
```

```
pwwn 21:00:00:20:37:39:ac:0d [z]
```

Device Alias Statistics Cleanup

Use the **clear device-name statistics** command to clear device alias statistics (for debugging purposes):

```
switch# clear device-alias statistics
```

Database Merge Guidelines

For information about CFS merge support, refer to the *Cisco MDS 9000 Family NX-OS System Management Configuration Guide* for detailed concepts.

When merging two device alias databases, follow these guidelines:

- Verify that two device aliases with different names are not mapped to the same pWWN.
- Verify that two different pWWNs are not mapped to the same device aliases.
- Verify that the combined number of device aliases in both databases cannot exceed 8191 (8K) in fabrics running SAN-OS Release 3.0(x) and earlier, and 20K in fabrics running SAN-OS Release 3.1(x) and later. If the combined number of device entries exceeds the supported limit, then the merge will fail.
- Ensure the device -alias mode is similar for the both the fabrics being merged.

Device Alias Configuration Verification

You can view device alias information by using the **show device-alias** command. See Examples 5-9 to 5-20.

Example 5-9 Displays All Configured Device Aliases from the Effective Database

```
switch# show device-alias database
device-alias name SampleName pwwn 21:00:00:e0:8b:0b:66:56
device-alias name x pwwn 21:01:00:e0:8b:2e:80:93

Total number of entries = 2
```

Example 5-10 Displays the Pending Database with No Modifications

```
switch# show device-alias database pending
There are no pending changes
```

Example 5-11 Displays the Pending Database with Modifications

```
switch# show device-alias database pending
device-alias name x pwwn 21:01:00:e0:8b:2e:80:93
device-alias name SampleName pwwn 21:00:00:e0:8b:0b:66:56
device-alias name y pwwn 21:00:00:20:37:39:ab:5f
```

```
device-alias name z pwwn 21:00:00:20:37:39:ac:0d

Total number of entries = 4
```

Example 5-12 Displays the Specified Device Name in the Pending Database

```
switch# show device-alias name x pending
device-alias name x pwwn 21:01:00:e0:8b:2e:80:93
```

Example 5-13 Displays the Specified pWWN in the Pending Database

```
switch# show device-alias pwwn 21:01:00:e0:8b:2e:80:93 pending
device-alias name x pwwn 21:01:00:e0:8b:2e:80:93
```

Example 5-14 Displays the Difference Between the Pending and Effective Databases

```
switch# show device-alias database pending-diff
- device-alias name Doc pwwn 21:01:02:03:00:01:01:01
+ device-alias name SampleName pwwn 21:00:00:e0:8b:0b:66:56
```

Example 5-15 Displays the Specified pWWN

```
switch# show device-alias pwwn 21:01:01:01:01:11:01:01
device-alias name Doc pwwn 21:01:01:01:01:11:01:01
```

Example 5-16 Displays the Device Alias in the FLOGI Database

```
switch# show flogi database
-----
INTERFACE  VSAN    FCID          PORT NAME          NODE NAME
-----
fc2/9      1       0x670100     21:01:00:e0:8b:2e:80:93  20:01:00:e0:8b:2e:80:93
           [x] <-----Device alias name
fc2/12     1       0x670200     21:00:00:e0:8b:0b:66:56  20:00:00:e0:8b:0b:66:56
           [SampleName] <-----Device alias name

Total number of flogi = 2
```

Example 5-17 Displays the Device Alias in the FCNS Database

```
switch# show fcns database

VSAN 1:
-----
FCID      TYPE  PWWN          (VENDOR)          FC4-TYPE:FEATURE
-----
0x670100  N     21:01:00:e0:8b:2e:80:93 (Qlogic)          scsi-fcp:init
           [x]
0x670200  N     21:00:00:e0:8b:0b:66:56 (Qlogic)          scsi-fcp:init
           [SampleName]

Total number of entries = 2
```

Example 5-18 Displays the *fcping* Statistics for the Specified Device Alias

```
switch# fcping device-alias x vsan 1
28 bytes from 21:01:00:e0:8b:2e:80:93 time = 358 usec
28 bytes from 21:01:00:e0:8b:2e:80:93 time = 226 usec
28 bytes from 21:01:00:e0:8b:2e:80:93 time = 372 usec
```

Example 5-19 Displays the *fctrace* Information for the Specified Device Alias

```
switch# fctrace device-alias x vsan 1
Route present for : 21:01:00:e0:8b:2e:80:93
20:00:00:05:30:00:4a:e2(0xfffc67)
```

Where available, device aliases are displayed regardless of a member being configured using a **device-alias** command or a zone-specific **member pwnn** command (see [Example 5-7](#) and [Example 5-8](#)).

Example 5-20 Displays Statistics for the Device Alias Application

```
switch# show device-alias statistics
      Device Alias Statistics
=====
Lock requests sent: 2
Database update requests sent: 1
Unlock requests sent: 1
Lock requests received: 1
Database update requests received: 1
Unlock requests received: 1
Lock rejects sent: 0
Database update rejects sent: 0
Unlock rejects sent: 0
Lock rejects received: 0
Database update rejects received: 0
Unlock rejects received: 0
Merge requests received: 0
Merge request rejects sent: 0
Merge responses received: 2
Merge response rejects sent: 0
Activation requests received: 0
Activation request rejects sent: 0
Activation requests sent: 2
Activation request rejects received: 0
```

Default Settings

[Table 5-2](#) lists the default settings for device alias parameters.

Table 5-2 Default Device Alias Parameters

Parameters	Default
Database in use	Effective database.
Database to accept changes	Pending database.
Device alias fabric lock state	Locked with the first device alias task.

Resolving Device Alias Merge Failures

The most common device-alias merge failure issues occur when merging databases. When a device-alias merge fails, we recommend that you review the syslog messages on the switch in which the merge was initiated in order to identify the issues. The application server in each fabric that is responsible for the merge is indicated by the term Merge Master in the messages.

In this example, the syslog messages indicate that the merge failed as a result of a database mismatch:

```
2007 Apr 9 15:52:42 switch-1 %CFS-3-MERGE_FAILED: Merge failed for app device-alias,
local switch wwn 20:00:00:0d:ec:2f:c1:40, ip 172.20.150.38, remote switch wwn
20:00:00:0d:ec:04:99:40, ip 172.20.150.30
2007 Apr 9 15:52:42 switch-1 %DEVICE-ALIAS-3-MERGE_FAILED: Databases could not be merged
due to mismatch.
```

**Note**

Use the **device-alias distribute** command to initiate a merge or remerge of device-alias databases. Use the **device-alias commit** command to *push* a switch's device-alias database to all the other switches in a fabric. If the switches whose device-alias databases are not merged (more than one merge master is shown in the output of the **show cfs merge status name device-alias** command), then the **device-alias commit** command causes the device-alias databases that are not merged to be overwritten.

Device Alias Best Practices

This section lists the best practices that you should follow when creating and using device aliases:

- Device aliases should be used to simplify the management of world wide names (WWNs) whenever possible. It is easier to identify devices with aliases rather than with WWNs. Hence, you should assign aliases to WWNs to easily identify the WWNs.
- Device-alias names are case-sensitive.
- Operate device aliases in Enhanced mode whenever possible. In Enhanced mode, applications accept a device-alias name in its *native* format, rather than expanding the alias to a port world wide name (pWWN). Because applications such as zone server, Inter-VSAN Routing (IVR), Port Security Manager (PSM), and Dynamic Port VSAN Membership automatically track and enforce device-alias membership changes, you have a single point of change.

**Note**

Interop mode VSANs do not accept Enhanced mode configurations.

- Preplan device-alias configurations and implement a consistent naming convention.
- Keep documented backups of all device-alias configurations.
- Plan for what the final device-alias database should be after the merge, before attempting to resolve merge failures. This can prevent traffic disruptions caused by accidentally overwriting device-alias entries.

**Caution**

Avoid performing a *blank commit* to resolve Cisco Fabric Services (CFS) merge failures. A blank commit overwrites the device-alias databases on all the switches with the device-alias database on the local switch.



Note A blank commit is a device-alias commit that is used when there are no changes (including mode changes), or when it is okay to overwrite the device-alias databases on the remote switches with the local switch's device-alias database.

Device alias mismatches might occur because of the following reasons:

- Duplicate Device-Alias Names-Same device-alias name, but different pWWNs. In such a scenario, the **show device-alias merge status** command displays the reason for the merge failure as "Reason: Another device-alias already present with the same name."
- Duplicate pWWNs-Different device-alias names, but same pWWN. In such a scenario, the **show device-alias merge status** command displays the reason for the merge failure as "Reason: Another device-alias already present with the same pwwn."



Note Each time device-alias changes are committed, the running configuration should be copied to the startup configuration on all the switches that were updated. Use the **copy running-config startup-config fabric** command to copy the running configuration to the startup configuration for all the switches in the fabric. If you do not copy the running configuration to the startup configuration after the device-alias changes are committed, and if the switch reloads, or loses power and restarts, the startup configuration will not have the correct device-alias database and merge failure will occur.

Resolving Device Alias Mismatches

If a switch with an existing device-alias database is being added to an existing fabric, conflicts might arise because of the following reasons:

- The same device-alias name is used, but with different pWWNs.
- The same pWWN is used, but with different device-alias names.

To resolve duplicate device-alias names, perform these steps:

Step 1 Run the **show cfs merge status name device-alias** command to review the CFS or device-alias merge failure syslogs to confirm that the merge failed:

```
switch-1# show cfs merge status name device-alias

Physical-fc Merge Status: Failed

[Sun Sep 25 14:45:55 2016]

Failure Reason: Another device-alias already present with the same pwwn

Local Fabric

-----
Switch WWN                IP Address
-----
20:00:54:7f:ee:1b:0e:b0  10.127.103.211      [Merge Master] <<< Merge Master#1
                        [switch-1]

Total number of switches = 1
```

```

Remote Fabric
-----
Switch WWN                IP Address
-----
20:00:54:7f:ee:1b:0e:50  10.197.111.54      [Merge Master] <<< Merge Master#2
Total number of switches = 1

```



Note A properly merged device-alias application should only show a single merge master. If there is more than one merge master, as shown in the above example, it indicates that the device-alias databases are not merged.

Step 2 Use the **no device-alias distribute** command on the switch in which the merge failure occurred in order to disable the device-alias distribution:

```
switch-1(config)# no device-alias distribute
```

Step 3 Resolve merge failure on the switch. See “[Resolving Merge Failures, page 5-95](#)” section.

Resolving Merge Failures

This section provides information about how to resolve merge failures.

Resolving Duplicate Device Alias Names (Same Device Alias Name, Different pWWNs)



Note A device-alias name is considered to be duplicate when the same device-alias name is used to point to different pWWNs.

To verify if a duplicate device-alias name exists in fabrics, perform these steps:

Step 1 Run the **show device-alias merge status** command to identify if the reason for the merge failure is a database mismatch:

```

switch# show device-alias merge status
Result: Failure
Reason: Another device-alias already present with the same name

```

Step 2 Review the CFS or the device-alias merge failure syslog to confirm that the merge failed. Alternatively, run the **show cfs merge status name device-alias** command to view the status of the merge.

```

switch# show cfs merge status name device-alias
Physical-fc Merge Status: Failed [ Mon Apr 9 15:57:58 2007 ] <===Merge status
Local Fabric
-----
Switch WWN                IP Address
-----
20:00:00:0d:ec:2f:c1:40  172.20.150.38      [Merge Master] <<< Merge Master#1
switch-1
Total number of switches = 1

```

```

Remote Fabric
-----
Switch WWN                IP Address
-----
20:00:00:0d:ec:04:99:40 172.20.150.30      [Merge Master] <<< Merge Master#2
                        switch-2
Total number of switches = 1

```

- Step 3** Compare the device-alias databases manually to identify the duplicate device-alias names.

In the following example, the same device-alias name, A1, is assigned to two different pWWNs—a pWWN on a local switch and a pWWN on a peer switch.

From merge master#1:

```

switch-1# show device-alias database
...output trimmed to show only mismatched device-alias
device-alias name A1 pwn 21:01:01:01:01:01:02

switch-2# show device-alias database
...output trimmed to show only mismatched device-alias
device-alias name A1 pwn 21:01:01:01:01:01:03

```

- Step 4** Run the **device-alias name name pwn id** command to change the pWWN on one of the switches to match the pWWN on the other switch.



Note Perform this step after device-alias distribution is disabled by running the **no device-alias distribute** command.

In the following example, the pWWN 21:01:01:01:01:01:02 on switch-1 is changed to match the pWWN 21:01:01:01:01:01:03 on switch-2:

```

switch-1# configure
Enter configuration commands, one per line. End with CNTL/Z.
switch-1(config)# device-alias database
switch-1(config-device-alias-db)# no device-alias name A1
switch-1(config-device-alias-db)# show device-alias database | i A1
switch-1(config-device-alias-db)# device-alias name A1 pwn 21:01:01:01:01:01:03
switch-1(config-device-alias-db)# show device-alias database | i A1
device-alias name A1 pwn 21:01:01:01:01:01:03

```

- Step 5** If there are more duplicate device-alias names, perform step 3 and step 4 to resolve the duplicate device-alias names issue.
- Step 6** Use the device-alias distribute command to enable the device-alias distribution and initiate a merge:
- ```
switch-1(config)# device-alias distribute
```
- Step 7** Use the **show cfs merge status name device-alias** command to verify in the output if the merge was successful.

## Resolving Duplicate pWWNs (Different Device Alias Names, Same pWWN)

To verify that the same pWWN is mapped to different device-alias names in fabrics, perform these steps:

- Step 1** Run the `show device-alias merge status` command to identify the reason for the merge failure:
- ```
switch# show device-alias merge status
Result: Failure
Reason: Another device-alias already present with the same pwwn.
```
- Step 2** Review the CFS or device-alias merge failure syslog to confirm that the merge failed. Alternatively, run the `show cfs merge status name device-alias` command to view the status of the merge.
- ```
switch# show cfs merge status name device-alias
Physical-fc Merge Status: Failed [Mon Apr 9 15:57:58 2007] <===Merge status
Local Fabric

Switch WWN IP Address

20:00:00:0d:ec:2f:c1:40 172.20.150.38 [Merge Master] <<< Merge Master#1
switch-1
Total number of switches = 1
Remote Fabric

Switch WWN IP Address

20:00:00:0d:ec:04:99:40 172.20.150.30 [Merge Master] <<< Merge Master#2
switch-2
Total number of switches = 1
```
- Step 3** Compare the device-alias databases manually to identify the duplicate pWWNs. On the switches where the merge failed in step 1, use the `show device-alias database` command to identify if a pWWN that is mapped to two different device-alias names exists.
- In this example, the pWWN 21:01:01:01:01:01:02 is mapped to the device-alias A3 on switch-1 and to the device-alias A1 on switch-2:
- ```
switch-1# show device-alias database
device-alias name A3 pwwn 21:01:01:01:01:01:02
Total number of entries = 1
switch-2# show device-alias database
device-alias name A1 pwwn 21:01:01:01:01:01:02
```
- Step 4** Use the `device-alias name name pwwn id` command to change the device-alias name on one of the switches to match the device-alias name on the other switch.



Note Perform this step after device-alias distribution is disabled by using the `no device-alias distribute` command.

In the following example, the device-alias name A3 on switch-1 is changed to match the device-alias name A1 on switch-2:

```
switch-1# configure
Enter configuration commands, one per line. End with CNTL/Z.
switch-1(config)# device-alias database
switch-1(config-device-alias-db)# no device-alias name A3
switch-1(config-device-alias-db)# device-alias name A1 pwwn 21:01:01:01:01:01:02
```

- Step 5** If there are more duplicate device-alias names, perform step 3 and step 4 to resolve the device-alias names.
- Step 6** Use the **device-alias distribute** command to enable the device-alias distribution and initiate a merge:
- ```
switch-1(config)# device-alias distribute
```
- Step 7** Use the **show cfs merge status name device-alias** command to verify in the output if the merge was successful.

## Resolving Mode Mismatch

The Device Alias feature can operate in either Basic or Enhanced mode. If the modes are different in two fabrics, CFS merge between the fabrics fails.

To verify that the device-alias mode is different in two fabrics, perform these steps:

- Step 1** Review the CFS or device-alias merge failure syslog to confirm that the merge failed. Alternatively, run the **show cfs merge status name device-alias** command to view the status of the merge.

```
switch# show cfs merge status name device-alias
Physical-fc Merge Status: Failed [Mon Apr 9 15:57:58 2007] <===Merge status
Local Fabric

Switch WWN IP Address

20:00:00:0d:ec:2f:c1:40 172.20.150.38 [Merge Master] <<< Merge Master#1
 switch-1
Total number of switches = 1
Remote Fabric

Switch WWN IP Address

20:00:00:0d:ec:04:99:40 172.20.150.30 [Merge Master] <<< Merge Master#2
 switch-2
Total number of switches = 1
```

- Step 2** Use the **show device-alias merge status** command to verify that the reason for the merge failure is a mode mismatch. If there is a mode mismatch, the reason that is displayed in the output is either "Databases could not be merged due to mode mismatch" or "One of the merging fabrics cannot support device-alias Enhanced mode."

```
switch# show device-alias merge status
Result: Failure
Reason: Databases could not be merged due to mode mismatch.
```

- Step 3** Use the `show device-alias status` command to verify the device-alias mode for each of the fabric. In this example, switch-1 is running in Enhanced mode, while switch-2 is running in Basic mode:

```
switch-1# show device-alias status
Fabric Distribution: Enabled
Database:- Device Aliases 2 Mode: Enhanced
switch-2# show device-alias status
Fabric Distribution: Enabled
Database:- Device Aliases 2 Mode: Basic
```

- Step 4** Use the `no device-alias distribute` command to disable device-alias distribution after you detect mismatched device-alias modes.
- Step 5** Depending on the mode you want to change in the switch, use either the `device-alias mode enhanced` command to change the switch mode to Enhanced, or use the `no device-alias mode enhanced` command to change the switch mode to Basic mode (default mode).




---

**Note** If you want to change the device-alias mode from Enhanced to Basic, but an application contains a device-alias configuration in the native format, the device-alias mode cannot be changed until you explicitly remove all the native device-alias configurations or replace all the device-alias members with the corresponding pWWNs.

---

- Step 6** Use the `device-alias distribute` command to enable the device-alias distribution and initiate a merge.

## Resolving a Validation Failure

If the merger of device aliases takes place without any conflicts, the resultant device-alias database is validated with the registered applications on each switch in both the fabrics being merged. If an application fails the validation of the merged database for any reason, the device-alias merge fails.

To verify that the device-alias database merge is failing because of an application-validation failure, perform these steps:

- Step 1** Review the CFS or device-alias merge failure syslog to confirm that the merge failed. Alternatively, use the `show cfs merge status name device-alias` command to view the status of the merge.
- Step 2** Use the `show device-alias merge status` command to verify that the reason for the merge failure is an application-validation failure:

```
switch# show device-alias merge status
Result: Failure
Reason: This is a non device-alias error.
```

- Step 3** Examine the syslog messages. The syslog for the switch in which the validation is rejected and the syslog for the switch managing the merge show relevant error messages.

This example shows a sample message on a switch in which the validation is rejected:

```
2007 Apr 10 00:00:06 switch-2 %DEVICE-ALIAS-3-MERGE_VALIDATION_REJECTED:
Failed SAP: 110 Reason: inter-VSAN zone member cannot be in more than one
VSAN Expln:
```

This example shows the syslog message on a switch that is managing the merge, and in which the validation is rejected:

```
2007 Apr 9 16:41:22 switch-1 %DEVICE-ALIAS-3-MERGE_VALIDATION_FAILED: Failed
```

```
SWWN: 20:00:00:0d:ec:04:99:40 Failed SAP: 110 Reason: inter-VSAN zone member cannot
be in more than one VSAN Expln:
```

- Step 4** Use the **show device-alias internal validation-info** command on the switch managing the merge, and examine the output.

This example shows that SAP 110 on switch 20:00:00:0d:ec:04:99:40 (switch-2) rejected the validation. The status message shows the reason for the failure along with the system application number:

```
switch# show device-alias internal validation-info
Validation timer: 0s
Per SAP Info Table:
=====
SAPS: 0
MTS Buffer Array Details:
=====
Buffers: 0
Local Status:
=====
Num Reqs Sent: 0 20:00:00:0d:ec:04:99:40
Num SAPs Done: 0
Failed SAP : 0 Status: success Expln:
Remote Status:
=====
CFS Resp Rcvd: TRUE
Failed SWWN : 20:00:00:0d:ec:04:99:40
SAP : 110 Status: inter-VSAN zone member cannot be in more than one VSAN <=== Status
Expln:
```

- Step 5** Use the **show system internal mts sup sap number description** command to find the application that rejected the configuration on the switch that rejected the validation.

In this example, the application that rejected the device-alias validation was the IVR process.

```
switch# show system internal mts sup sap 110 description
IVR-SAP
```

- Step 6** Analyze the device-alias validation failure. This analysis is dependent on the application that failed the validation as well as the device-alias database configuration.

In this example, IVR is failing the validation. To troubleshoot this problem, begin by reviewing the device-alias databases that are being merged. Use the **show device-alias database** command from the switch managing the merge for each fabric.

```
switch# show device-alias database
device-alias name A1 pwnn 21:01:01:01:01:01:01:01
device-alias name A2 pwnn 21:01:01:01:01:01:01:02 => Pre-merge: A2 defined on switch-1
Total number of entries = 2
switch# show device-alias database
device-alias name A1 pwnn 21:01:01:01:01:01:01:01 => Pre-merge: A2 not defined on
switch-2
Total number of entries = 1
Because IVR is enabled on switch-2, review the IVR zone set.
switch# show ivr zoneset
```

```

zoneset name s1
 zone name z1
 pwn 21:01:01:01:01:01:01:02 vsan 1 autonomous-fabric-id 1
 device-alias A2 vsan 2 autonomous-fabric-id 1

```

Prior to the database merge, device-alias A2 is not defined on switch-2. Because of the merge between switch-1 and switch-2, device-alias A2 becomes available on switch-2, and A2 is mapped to pWWN 21:01:01:01:01:01:01:02.

The device alias-based member A2 in the IVR zone z1 is resolved and mapped to pWWN 21:01:01:01:01:01:01:02, and is a member of VSAN 2. However, pWWN 21:01:01:01:01:01:01:02 is already a member of VSAN 1. The mapping that occurs because of the device-alias merge makes the IVR configuration illegal. The same pWWN cannot be a member of multiple VSANs.

In the case when IVR configuration is illegal, the pWWN in VSAN 2 is defined using the device alias (A2), while the member in VSAN 1 is defined using the actual pWWN. The IVR detects this situation and rejects the device-alias validation. As a result, the device-alias merge fails.

## Resolving Database Conflicts

If an entry in the device-alias database conflicts with the configuration of a registered application, the device-alias database commit fails the validation process. Correct either the device-alias database or the application configuration.

To determine the application that failed the validation and the reason for the failure, perform these steps:

**Step 1** Use the device-alias commit command to view the output.

This example shows that the commit failed because there is a conflict between the device-alias database and an application configuration:

```

switch# configure
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# device-alias commit
inter-VSAN zone member cannot be in more than one VSAN ==> reason for commit failure

```

**Step 2** Determine which application configuration is in conflict with the device-alias database by reviewing the syslogs for the switch in which the commit was issued.

This example shows that SAP 110 (IVR) on swwn 20:00:00:0d:ec:04:99:40 (switch-2) has rejected the validation, and therefore, the device-alias commit has failed:

```

2007 Apr 10 11:54:24 switch-1 %DEVICE-ALIAS-3-VALIDATION_FAILED: Failed=>Validation
Status

Swwn: 20:00:00:0d:ec:04:99:40 Failed SAP: 110 Reason: inter-VSAN zone ==>Switch and
SAP member cannot be in more than one VSAN Expln: ==>Reason

2007 Apr 10 11:54:24 switch-1 %DEVICE-ALIAS-3-COMMIT_FAILED: Failed to ==>Commit
status commit the pending database: inter-VSAN zone member cannot be in more ==>Reason
than one VSAN

```

**Step 3** Review the syslog on the switch in which the validation is rejected.

This example shows that the following syslog is printed on switch-2:

```

2007 Apr 10 19:13:08 switch-2 %DEVICE-ALIAS-3-VALIDATION_REJECTED: Failed

```

```
SAP: 110 Reason: inter-VSAN zone member cannot be in more than one VSAN ==>SAP and
reason
```

- Step 4** Compare the existing device-alias database (including the desired changes) and the application configuration to find the conflict.

This example uses the **show device-alias database** and **show ivr zoneset** commands along with the console logs of the device-alias database changes made prior to the commit. The comparison shows that the definition of the new device-alias A2 results in the resolution of the enhanced device-alias member A2 in the IVR zone z1 to pWWN 21:01:01:01:01:01:02, which is already a member of zone z1. The pWWN is directly defined as a member of VSAN 1, while the enhanced device-alias A2 is defined as a member of VSAN 2. This configuration is not allowed in the IVR. The IVR detects the configuration problem and rejects the device-alias database validation.

```
switch# show device-alias database ==> existing device alias database
device-alias name A1 pwnn 21:01:01:01:01:01:01
Total number of entries = 1
switch# show ivr zoneset ==> display existing IVR zone set
zoneset name s1
zone name z1
pwnn 21:01:01:01:01:01:02 vsan 1 autonomous-fabric-id 1
 device-alias A2 vsan 2 autonomous-fabric-id 1
switch# configure
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# device-alias database
switch(config-device-alias-db)# device-alias name A2 pwnn 21:01:01:01:01:01:02
switch(config-device-alias-db)# exit
switch(config)# device-alias commit
inter-VSAN zone member cannot be in more than one VSAN
```

- Step 5** Correct the conflict by making adjustments to the application configuration, or by making changes to the device-alias database, and running the **device-alias commit** command again.

## Verifying the Device-Alias Database Status

This section provides information about verifying the device-alias database status.

| Command Name                                      | Description                                                                                  |
|---------------------------------------------------|----------------------------------------------------------------------------------------------|
| <b>show cfs merge status name device-alias</b>    | Displays information about the status of the CFS merge for the device-alias database.        |
| <b>show device-alias database</b>                 | Displays the entire device-alias database.                                                   |
| <b>show device-alias internal validation info</b> | Displays information about the status of the validation process (part of a commit or merge). |
| <b>show device-alias merge status</b>             | Displays the result of the device-alias merge operation and the reason for the result.       |

| Command Name                            | Description                                                                                                                                                                                                                   |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>show device-alias session status</b> | Returns the status of the last CFS command, such as <b>clear</b> , <b>commit</b> , or <b>terminate</b> . The results of the last used CFS command and reason fields help identify the reason for the failure.                 |
| <b>show device-alias status</b>         | Displays configuration information for the device-alias service, including whether fabric distribution is enabled, the number of device aliases in the database, lock information, and the database mode (Basic or Enhanced). |

