

This section describes how to troubleshoot and resolve problems that might occur when using device aliases.

## Contents

- [1 Overview](#)
- [2 Initial Troubleshooting Checklist](#)
  - ◆ [2.1 Merge Failure Messages](#)
  - ◆ [2.2 Merge Validation Failure Messages](#)
  - ◆ [2.3 Commit Failure Messages](#)
  - ◆ [2.4 Verifying Device Alias Database Status Using the CLI](#)
  - ◆ [2.5 Limitations and Restrictions](#)
- [3 Merge Failure Issues](#)
  - ◆ [3.1 Resolving Duplicate Device Alias Names](#)
  - ◆ [3.2 Resolving Mapping a pWWN to Different Device Alias Names](#)
  - ◆ [3.3 Resolving Mode Mismatch](#)
  - ◆ [3.4 Resolving Merge Failures in Mixed Fabric](#)
  - ◆ [3.5 Resolving a Validation Failure](#)
  - ◆ [3.6 Resolving Merge in Progress Issues](#)
- [4 Validation and Commit Failure Issues](#)
  - ◆ [4.1 Resolving Database Conflicts](#)
  - ◆ [4.2 Resolving Application Busy Situations](#)
  - ◆ [4.3 Resolving Database Size Issues](#)
  - ◆ [4.4 Resolving Mode Issues](#)

## Overview

When you configure different features for a Cisco MDS 9000 Family switch, such as zoning, DPVM, or port security, you must specify the correct device name each time. An inaccurate device name can cause unexpected results. You can circumvent this problem by defining and using device aliases.

A device alias is a user-friendly name for a port WWN (pWWN) that can be used in all configuration commands as required. All switches in the Cisco MDS 9000 Family support Distributed Device Alias Services (device aliases).

## Initial Troubleshooting Checklist

Device alias problems can result in merge, commit, or validation failures, and resolution requires gathering information about the specific failure and the fabric configuration. Begin troubleshooting device alias issues by checking the following items:

| Checklist   | Check off |
|---|-----------|
| Verify the device alias mode (basic or enhanced). |           |
| Verify that device alias distribution is enabled. |           |
| Verify that the fabric is not locked.             |           |

---

**Note:**



See the "System Messages" section on page 1-10 for more information about using system (syslog) messages for troubleshooting purposes.

## Merge Failure Messages

When a device alias merge fails, review the syslog messages on the switch where the merge was issued (the merge manager). In Example 15-1, the syslog messages indicate that the merge failed as a result of a database mismatch.

### Example 15-1 Merge Failure Syslog Messages

```
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.
```

## Merge Validation Failure Messages

When a device alias database fails validation during the merge process, review the syslog messages on the switch managing the merge and on the switch where the validation was rejected. In Example 15-2, the syslog message indicates that the validation failed on switch sWWN 20:00:00:0d:ec:04:99:40. Example 15-3 shows the syslog message received on the switch that rejected the validation. Both messages indicate that the merge validation failed because the merge would result in the definition of the same zone member in multiple VSANs.

### Example 15-2 Message on Switch Managing the Merge

```
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:
```

### Example 15-3 Message on Switch Rejecting the Validation

```
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:
```

## Commit Failure Messages

When a device alias commit fails, review the syslog messages for the switch where the command was issued. Example 15-4 shows that SAP 110 (IVR) failed validation on switch sWWN 20:00:00:0d:ec:04:99:40, which caused the commit to fail. Example 15-5 shows the failure message.

### Example 15-4 Commit Validation Failure Message

```
2007 Apr 10 11:54:24 switch-1 %DEVICE-ALIAS-3-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:
```

### Example 15-5 Commit Failure Message

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

## Verifying Device Alias Database Status Using the CLI

Table 15-1 lists several commands that can be used to verify the device alias database status.

**Table 15-1 CLI Commands for Troubleshooting Device Alias Issues**

CLI Command

Description

**show cfs merge status name device-alias**

Displays information about the status of the CFS merge for the device alias database.

**show device-alias database**

Displays the entire device alias database.

**show device-alias internal validation info**

Displays information about the status of the validation process (part of a commit or merge).

**show device-alias merge status**

Displays the result of the device alias merge operation, and the reason for the result.

**show device-alias session status**

Returns the status of the last CFS command, such as clear, commit or abort. The last action result and reason fields help identify the reason for a failure.

**show device-alias status**

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).

## Limitations and Restrictions

The following limitations and restrictions are associated with the use of device aliases:

- Enhanced mode operation is only supported on switches running SAN-OS Release 3.1(1) or later.
- Interop mode VSANs do not accept enhanced mode (native alias-based) configurations.
- When merging device alias databases, the combined number of device aliases in both databases cannot exceed 8000 (8191 device aliases) in fabrics running SAN-OS Release 3.0(x) and earlier, and 20000 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.

## Merge Failure Issues

The most common device alias problems occur when merging databases. This section describes the steps to take when a merge of two fabrics fails because the merge of the device alias databases fails. See Table 15-2.

**Symptom** Merge failed.

**Table 15-2 Device Alias Database Merge Fails**

Symptom

Possible Cause

Solution

Merge fails for device alias.

A device alias is configured with the same name but with a different pWWN on either side of the fabric.

Change the conflicting device alias name on one fabric. See the "Resolving Duplicate Device Alias Names" section.

The same pWWN is mapped to different device alias names on either side of the fabric.

Change the mapping so the pWWN is mapped to the same device alias name on both sides of the fabric. See the "Resolving Mapping a pWWN to Different Device Alias Names" section.

The device alias operation mode (basic or enhanced) is not the same for both fabrics.

Check the device alias mode. Both fabrics must be operating in the same device alias mode. See the "Resolving Mode Mismatch" section.

Note Device alias enhanced mode is only supported on switches running SAN-OS Release 3.1(1) and later.

The size of the device alias database exceeds the limit supported by some switches in the fabric.

Upgrade the SAN-OS release for all switches to 3.1(1) or later. See the "Resolving Merge Failures in Mixed Fabric" section.

An application fails the validation of the merge database.

Analyze the problem from the application perspective and make the appropriate corrections. See the "Resolving a Validation Failure" section.

Merge remains in progress and does not complete.

A user is editing the application database.

Determine which application database is locked, and take the appropriate steps to unlock it. When the application database is unlocked, the merge should be automatically retried. See the "Resolving Merge in Progress Issues" section.

A user edited the application database and forgot to release the lock.

The application is busy handling its own merge.

The application database is locked due to a defect in the application.

## Resolving Duplicate Device Alias Names

To verify that a duplicate device alias name exists in the fabrics, follow these steps:

1. Review the CFS or device alias merge failure syslogs to confirm that the merge failed. Optionally, you can issue the following CLI command to view the status of the merge:

```
switch-1# 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]
                           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]
                           switch-2
Total number of switches = 1
```

2. Verify that the reason for the merge failure is a database mismatch by using the **show device-alias merge status** command.

```
switch-1# show device-alias merge status
  Result: Failure
  Reason: Databases could not be merged due to mismatch.
```

3. Identify the duplicate device alias name by using the **show device-alias database** command from the appropriate switch in each of the two fabrics. In this example, both databases have the same device alias, A1, mapped to different pWWNs.

```
switch-1# show device-alias database
  device-alias name A1 pwn 21:01:01:01:01:01:01:01
  Total number of entries = 1
switch-2# show device-alias database
  device-alias name A1 pwn 21:01:01:01:01:01:01:02
  Total number of entries = 1
```

4. Make the appropriate changes to the device alias database for one of the fabrics. Refer to the *Cisco MDS 9000 Family CLI Configuration Guide* for details.

---

## Resolving Mapping a pWWN to Different Device Alias Names

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

1. Review the CFS or device alias merge failure syslogs to confirm that the merge failed. Optionally, you can issue the **show cfs merge status name device-alias** command to view the status of the merge.
2. Verify the reason for the merge failure by using the **show device-alias merge status** command.

```
switch-1# show device-alias merge status
  Result: Failure
  Reason: Another device-alias already present with the same pwn.
```

3. Identify the pWWN that is mapped to two different device alias names by using the **show device-alias database** command from the appropriate switch in each of the fabrics. In this example, the pWWN 21:01:01:01:01:02 is mapped to device alias A3 on switch switch-1, and to device alias A1 on switch switch-2.

```
switch-1# show device-alias database
  device-alias name A3 pwnn 21:01:01:01:01:02
  Total number of entries = 1
switch-2# show device-alias database
  device-alias name A1 pwnn 21:01:01:01:01:02
  Total number of entries = 1
```

4. Make the appropriate changes to the device alias database for one of the fabrics. Refer to the *Cisco MDS 9000 Family CLI Configuration Guide* for details.

---

## Resolving Mode Mismatch

The device alias feature can operate in either basic or enhanced mode. If the modes are different in the two fabrics, the CFS merge between those fabrics will fail.

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

---

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

2. Verify that the reason for the merge failure is a mode mismatch by using the **show device-alias merge status** command. The reason is either "Databases could not be merged due to mode mismatch" or "One of the merging fabrics cannot support device-alias enhanced mode."

```
switch-1# show device-alias merge status
  Result: Failure
  Reason: Databases could not be merged due to mode mismatch.
```



**Note:** All switches in the fabric must be running SAN-OS Release 3.1(1) or later to support device alias enhanced mode operation.

---

3. Verify the device alias mode for each of the fabrics by using the **show device-alias status** command. 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
```

4. Make the appropriate change to the device alias mode for one of the fabrics. Try doing a blank commit to change the mode. A blank commit occurs when you lock the fabric, but do not make any database definition changes, and then perform a commit. This results in the distribution of the effective database of the local switch to all other switches in the fabric. In this case, change the device alias mode only, and then commit that change.

---

**Note:**



If you need to change the device alias mode from enhanced to basic, and any application contains a device alias configuration in 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. Refer to the *Cisco MDS 9000 Family CLI Configuration Guide* for details.

## Resolving Merge Failures in Mixed Fabric

A mixed fabric is a fabric consisting of switches running different software versions. Prior to SAN-OS Release 3.1(x), the limit for device aliases in a merged fabric was 8000. In Release 3.1(1) and later, this limit has been expanded to 20000. If you attempt to merge a fabric that includes switches running Release 3.0(x) or earlier with a fabric that includes switches running Release 3.1(x) or later, and the merged device alias database would exceed 8000, the merge will fail.

To verify that the mixed fabric merge is failing because of database size, follow these steps:

1. Review the CFS or device alias merge failure syslogs to confirm that the merge failed, or issue the **show cfs merge status name device-alias** command to view the status of the merge.
2. Verify that the reason for the merge failure is incompatible capabilities or configurations by using the **show device-alias merge status** command.

```
switch-1# show device-alias merge status
Result: Failure
Reason: The merging fabrics have incompatible capabilities or configurations.
```

3. Upgrade all switches in the problem fabric to SAN-OS Release 3.1(x) or later, and then attempt the merge again. Otherwise, if you must merge the fabrics, you will have to reduce the number of entries in the combined device alias database.

## Resolving a Validation Failure

If the device alias databases can be merged without any conflicts, the resultant device alias database is validated with the registered applications on each switch in both 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, follow these steps:

1. Review the CFS or device alias merge failure syslogs to confirm that the merge failed, or issue the **show cfs merge status name device-alias** command to view the status of the merge.
2. Verify that the reason for the merge failure is an application validation failure by using the **show device-alias merge status** command.

```
switch-1# show device-alias merge status
Result: Failure
Reason: This is a non device-alias error.
```

3. Examine syslog messages. The syslog for the switch where the validation is rejected and the syslog for the switch managing the merge show relevant error messages.

Example 15-6 shows a sample message on the switch where the validation is rejected.

### Example 15-6 Validation Rejection Syslog Message

```
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:
```

Example 15-7 shows the syslog message on the switch managing the merge for the same merge failure.

### Example 15-7 Validation Failure Syslog Message

```
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:
```

4. Examine the output of the **show device-alias internal validation-info** command, issued from the switch managing the merge. In this example, 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 (SAP) number.

```
switch-1# 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:
```

5. Find the application that has rejected the configuration by using the **show system internal mts sup sap number description** command on the switch that rejected the validation. In this example, the application that rejected the device-alias validation was the IVR process.

```
switch-2# show system internal mts sup sap 110 description
IVR-SAP
```

6. Analyze the device alias validation failure. This process 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-1# 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-2# 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-2# show ivr zoneset
zoneset name s1
  zone name z1
    pwnn 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.

When this occurs, 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 this case, the pWWN in VSAN 2 is defined using the device alias (A2), while the member in VSAN 1 is defined using the actual pWWN. IVR detects this situation and rejects the device alias validation and the device alias merge fails.

---

## Resolving Merge in Progress Issues

When a merge occurs, the merged device alias database is validated with the registered applications on each switch in both of the fabrics. During this process, an application database may be in a locked state. Common causes are listed in Table 15-2.

Because the application is busy, it cannot validate the device alias database, and it returns a busy response to the device alias process. This information is propagated back to CFS. CFS then waits for the length of the defined retry time, and then automatically retries the device alias merge. During the defined retry time, the device alias merge status is shown as in progress. (See Example 15-8).

### Example 15-8 Displaying Merge Status

```
switch-1# show cfs merge status na device-alias
Physical-fc Merge Status: In Progress ==> Status shows merge is still in progress
```

If the application lock is released within the defined CFS retry time, the device alias validation can be processed. Otherwise, the application returns a busy response to the process again, and CFS again waits for the length of the defined retry time before retrying the device alias validation again. This situation is repeated until the application lock is released.

This is not an uncommon situation. When the ISL comes up between two fabrics, it typically triggers a CFS merge of multiple applications. By the time the device alias merge moves to validation stage, other applications may be busy handling their own merge. This is a transient state, and the device alias merge would be reversed and retired after the defined retry time.

If for any reason the application database remains locked, the device alias merge remains in progress and cannot complete.

To verify a locked application database and to determine how to proceed, follow these steps:

1. Use the **show cfs merge status na device-alias** command to see the merge status.

```
switch-1# show cfs merge status na device-alias
Physical-fc Merge Status: In Progress ==>The merge status shows in progress
Local Fabric
-----
Switch WWN                IP Address
-----
20:00:00:0d:ec:2f:c1:40  172.20.150.38      [Merge Master]
                        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]
Total number of switches = 1
```

2. Obtain more information about the progress of the merge by using the **show device-alias internal validation-info** command. In the following example, SAP 110 (IVR) on SWWN 20:00:00:0d:ec:04:99:40 (switch-2) has rejected the validation. The explanation indicates that the IVR application has rejected the validation because it is busy.

```
switch-1# 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
  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 ==> switch switch-2
SAP    : 110   Status: Some of the registered modules are busy
handling other requests. Please retry the command after some time.
Expln: there is a pending action. Check ivr status
```

3. Determine the IVR status by using the **show cfs lock** command from the switch-2 switch. In this example, the IVR database is locked by the user admin, and the device alias merge remains in progress.

```
switch-2# show cfs lock
Application: ivr
Scope      : Physical-fc
-----
Switch WWN                IP Address                User Name  User Type
-----
20:00:00:0d:ec:04:99:40  172.20.150.30            admin     CLI/SNMP v3
                        switch-2
Total number of entries = 1
Application: device-alias
Scope      : Physical-fc
-----
```

| Switch WWN              | IP Address    | User Name | User Type |
|-------------------------|---------------|-----------|-----------|
| 20:00:00:0d:ec:2f:c1:40 | 172.20.150.38 | CFS       | Merge     |

Total number of entries = 1

4. Resolve the problem by unlocking the application database. In this example, when the user admin completes the edits and commits the changes, or otherwise unlocks the IVR database (using an **ivr abort** or **ivr clear** command), the device alias merge validation will proceed.

---

## Validation and Commit Failure Issues

After making changes to the device alias database, you must enter a **commit** command to commit the changes. The commit process includes validation of the database change by all registered applications on all switches in the fabric. This section identifies the common problems that might cause validation and commit failures. See Table 15-3.

**Symptom** Commit or configuration failed.

**Table 15-3 Device Alias Database Commit Fails**

Symptom

Possible Cause

Solution

Commit fails due to a validation failure

The device alias database conflicts with the application configuration.

Tune the device alias database or the application configuration to eliminate the conflict. See the "Resolving Database Conflicts" section.

The application is busy.

Verify that the application that failed the validation is no longer busy, and then retry the commit. See the "Resolving Application Busy Situations" section.

Commit or configuration fails.

The size of the device alias database exceeds the supported limit.

Reduce the number of entries in the device alias database, or upgrade the SAN-OS release for all switches to Release 3.1(1) or later. See the "Resolving Database Size Issues" section.

Enhanced mode is not supported on some switches in the mixed fabric.

Upgrade the SAN-OS release for all switches in the fabric to Release 3.1(1) or later, or change the device alias mode to basic. See the "Resolving Mode Issues" section.

## 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. You must either correct the device alias database or the

application configuration.

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

1. View the output from the **device-alias commit** command. In this example, the commit fails because there is a conflict between the device alias database and an application configuration.

```
switch-1(config)# device-alias commit
inter-VSAN zone member cannot be in more than one VSAN ==> reason for commit failure
```

2. Determine which application configuration is in conflict with the device alias database by reviewing the syslog for the switch where the commit was issued. In this example, 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
```

3. Review the syslog on the switch where the validation is rejected. In this example, the following syslog is printed on the switch 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
Expln:
```

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 would result in the resolution of enhanced device alias member A2 in 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 IVR. IVR detects the configuration problem and rejects the device alias database validation.

```
switch-1# show device-alias database ==> existing device alias database
device-alias name A1 pwn 21:01:01:01:01:01:01
Total number of entries = 1
```

```
switch-2# show ivr zoneset ==> display existing IVR zone set

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

```
switch-1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch-1(config)# device-alias database
switch-1(config-device-alias-db)# device-alias name A2 pwn 21:01:01:01:01:01:02
switch-1(config-device-alias-db)# exit
switch-1(config)# device-alias commit
```

inter-VSAN zone member cannot be in more than one VSAN

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

---

## Resolving Application Busy Situations

A device alias commit can fail because an application is busy and cannot perform the requested validation. If that occurs, the device alias CFS lock is not released and the pending database contents remain unchanged. When the application is no longer busy, retry the commit request.

Example 15-9 shows the console output that occurs when an application busy situation causes the commit to fail.

### Example 15-9 Application Busy Console Output

```
switch-2(config)# device-alias commit
Some of the registered modules are busy handling other requests. Please
retry the command after some time.
```

Review the syslogs (Example 15-10) and use the **show device-alias** command (Example 15-11) to gather additional details about the application that returned busy.

### Example 15-10 Application Busy Syslog Message

```
2007 Apr 10 02:30:55 switch-2 %DEVICE-ALIAS-3-VALIDATION_FAILED: Failed SAP: ==>Status
110 Reason: Some of the registered modules are busy handling other requests. ==>Reason
Please retry the command after some time. Expln: there is a pending action. ==>Resolution
Check ivr status.
```

### Example 15-11 show device-alias Command Output

```
switch-2# show device-alias internal validation-info
Validation timer:    0s

Per SAP Info Table:
=====
SAPS:  2
  SAP:  18      Num Resp Rcvd:  1
  SAP:  110     Num Resp Rcvd:  1

MTS Buffer Array Details:
=====
Buffers:  0

Local Status:
=====
Num Reqs Sent:  1
Num SAPs Done:  2
Failed SAP   : 110      Status: Some of the registered modules are busy ==>Status and reason
handling other requests. Please retry the command after some time.
Expln: there is a pending action. Check ivr status

Remote Status:
=====
CFS Resp Rcvd: FALSE
Failed SWWN   : 00:00:00:00:00:00:00:00
  SAP        :  0      Status: success      Expln:
```

The failed SAP in Example 15-11 is 110 (IVR). Unless the IVR lock is released, you cannot commit the device alias database changes.

## Resolving Database Size Issues

The maximum size of the device alias database increased from 8000 to 20000 for switches running SAN-OS Release 3.1(1) or later. If the fabric includes switches running SAN-OS Release 3.0(x) or earlier and switches running Release 3.1(1) or later, the maximum size of the device alias database is limited to 8000. This is the limit imposed on the switches running Release 3.0(x) or earlier. Unless all of the switches are upgraded to Release 3.1(1) or later, you cannot configure more than 8000 device aliases.

If you attempt to configure or commit a device alias database that exceeds the maximum size supported in the fabric, the configuration or the commit will fail. Example 15-12 shows the commit failure and Example 15-13 shows the configuration failure.

### Example 15-12 Commit Failure Due to Database Size Limits

```
switch-1(config)# device-alias commit
Some of the switches in the fabric are running a version of software which
cannot support either the issued command or maximum device-alias limits.
Please fix those switches and retry the command
```

### Example 15-13 Configuration Failed Due to Database Size Limits

```
switch-1(config)# device-alias name test8193 pwwn 21:01:01:01:01:01:01:01
Some of the switches in the fabric are running a version of software which
cannot support either the issued command or maximum device-alias limits.
Please fix those switches and retry the command.
```

## Resolving Mode Issues

Device alias enhanced mode is available in switches running SAN-OS Release 3.1(1) and later. In a mixed fabric (a fabric that includes switches running Release 3.1(1) and switches running Release 3.0(x) or earlier), an attempt to enable device alias enhanced mode fails. Unless all of the switches in the fabric can be upgraded to Release 3.1(1) or later, you cannot enable device alias enhanced mode.

If you attempt to enable enhanced mode in a successfully merged mixed fabric, or to commit a device alias database that enables enhanced mode to resolve a merge failure in a mixed fabric, the configuration or the commit will fail. Example 15-14 shows the commit failure and Example 15-15 shows the configuration failure.

### Example 15-14 Commit Failure Due to Mode Conflicts

```
switch-1(config)# device-alias commit
Some of the switches in the fabric are running a version of software which
cannot support either the issued command or maximum device-alias limits.
Please fix those switches and retry the command.
```

### Example 15-15 Configuration Failure Due to Mode Conflicts

```
switch-1(config)# device-alias mode enhanced
Some of the switches in the fabric are running a version of software which
cannot support enhanced mode. Please fix those switches and retry the
command.
```

**Back to Main Page:** [Cisco MDS SAN-OS Troubleshooting Guide](#)

---