



Troubleshooting Config-Sync Issues

This chapter describes how to identify and resolve problems that can occur with config-sync in the Cisco Nexus 5000 Series switch.

This chapter includes the following sections:

- [Commit Failure](#)
- [Import Failure](#)
- [Merge Failure](#)
- [Switch-profile Deletion Failure](#)
- [Verify Failure](#)

Commit Failure

Use the **show switch-profile status commit** command to view commit status.

Commit failure has many possible causes:

- [Command Parsing Failed](#)
- [Verify failed](#)
- [Commands that failed commit](#)
- [Another session in progress](#)



Note

When a commit fails, commands that were entered under SP are still stored in the SP buffer. Do not configure these commands under SP again. After correcting the cause of the failure, only the commit needs to be executed.

Command Parsing Failed

Possible Cause

Appropriate conditional feature(s) are not enabled.

Solution

Ensure that appropriate conditional feature(s) are enabled.

This error message indicates that some feature commands have not been configured. Feature commands are not allowed to be configured within SP and have to be configured on BOTH peers from conf-t.

Verify failed

Possible Cause

The commands listed failed mutual-exclusion checks. These commands have already been configured under conf-t.

Solution

If you do not want these commands synchronized, remove the commands from conf-t.

Alternatively, delete these commands from the switch-profile buffer and reissue the commit.

To delete commands from the switch-profile buffer, perform the following:

- View commands in the SP buffer using the **show switch-profile buffer** command.
- Delete commands indicated by the sequence numbers with the **buffer delete <range>** command.
- Use the **buffer-move <seq id> <seq id>** command to rearrange commands in the buffer.

This is command is useful when commands in the buffer are not ordered correctly.

Commands that failed commit

Possible Cause

Commands failed during commit.

Solution

Correct the reason for the failure and re-issue the commit.

If the commit continues to fail, issue the same command from conf-t. If it succeeds from conf-t, check for any errors relating to the command using the **show system internal csm info trace** command.

For every command executed from config-sync, there is a `csm_cmd_status[0x0]` line in the trace log that indicates that the command was successful.

Another session in progress

Possible Cause

Conflict occurs if conf-t or config-sync has taken a lock.

Solution

Compare the vPC domain IDs of the two switches and ensure that they match.

Use the **show system internal csm global info** command to check whether conf-t or config-sync has taken a lock.

- If conf-t has taken a lock and not released it, command output, similar to the following example, is displayed.



Note The client type should be set to 2 as shown in the example.

Example:

```
No of sessions: 1 (Max: 32)
Total number of commands: 0 (Max: 102400)
Session Database Lock Info: Locked
Client: 2 (1: SSN, 2: CONF-T)
Ref count: zero-based ref-count
Lock acquired for cmd : some-command
```

- Identify the command that acquired the lock using the **show accounting log** command.
- After identifying the command, check for its SUCCESS/FAILURE status.
- If the command did not return a status, then config-sync would not release the lock on conf-t.
- Use the **test csm ssn-db-lock reset conf-t** command to reset the lock.
- If switch-profile has taken the lock, the client id is reported as 1 in the **show system internal csm global info** command.
 - Use the **show switch-profile status** command to determine if a merge is in progress.

A merge is indicated by pending_merge:1 /rcvd_merge:1.
 - If a merge/verify/commit session is already in progress, then sp ssn-db is locked.

Wait for the current session to complete and try again.
 - If the lock is not released, use the **show cfs lock** command to determine if the CFS fabric is locked.

Identify the application that locked CFS. If the application is session-manager, then the CFS lock was taken by config-sync.

Analyze the output from the **show system internal csm info trace**, **show cfs internal notification log name session-mgr**, and **show cfs** commands.
- Use the **show system internal csm info trace** command to view the events, trace, or error debug traces.

Import Failure

Use the **show switch-profile status** command to view import status.

Import failure has many possible causes:

- [Failed to collect running-config](#)
- [Command does not exist in global-db](#)
- [Mutual exclusion check failed on peer](#)

The following describes import options and best practices.

Table 1-1 *Import Options*

| Type | Description |
|---|---|
| import | Enables import mode. Manually enter configuration and then commit to move the configuration into SP. |
| import running-config | Imports all SP-aware configurations from the running-config into SP. Use the buffer-move and buffer-delete commands to remove commands not to be synced and then commit the configurations. |
| import interface <interface-range> | Imports running configurations for specified interfaces. Used to only import configurations of interfaces and not global configurations. |
| import running-config exclude-phy-interface | Imports only global and logical interfaces, not physical interface configurations. |

Import best practices

- The import option is used when the system is already configured and you want to bring in an existing configuration within the switch-profile and sync it with the peer.
- When VPC peer switches are already configured, the import operation is performed on both switches independently. The user must verify that the configurations are the same under SP on peer switches and then add the sync-peer command to the configuration.

If the configurations are different within the SP when the sync-peer command is added, a merge failure occurs. Use the **show switch-profile status** command and determine which configurations failed to merge.

- If one of a pair of VPC switches was previously configured and the other switch was RMA'ed, then the switch-profile is created on both switches using the sync-peer command.

This means that the configuration from the previously configured switch is imported and committed on the other switch. Also the configurations from the previously configured switch are moved from the global-db to the database of the switch-profile.

Failed to collect running-config

Possible Cause

Failure occurs if the system is too busy and the **show running** command did not complete.

Solution

Determine if a system resource utilization problem exists. Correct the problem and retry the operation.

Command does not exist in global-db

Possible Cause

Command is missing from the global-db.

Solution

Use the **show system internal csm info global-db cmd-tbl** command to determine if the command exists in the global_db.

- If the command exists in the global_db, it is possible that there is not enough space in the show run for the command. Ensure that there are no trailing space/tabs in show running config generation.
- If the command does not exist in the global_db, use the **show accounting log** command to determine if the command was configured and to display the status of the command.
 - If the command status was a failure, then the command should not be displayed in show running.
 - If the command is displayed, then the application should correct it.
- If the command was configured before reload/issu, add the command back. If the accounting log shows the command's retval as success, determine if the command is getting added to the global-db.
 - If the command was added correctly, copy r s, check global-db reload, and check if the command exists in the global-db.
 - If the command does not exist in the global-db, then the issue might be that the command is not showing up in show running on boot up.
 - If the command does not exist in the global_db, investigate the csm_save_global_command function. The csm_save_global_command function is where the command gets added to the global_db

Mutual exclusion check failed on peer

Possible Cause

The imported configuration is sent to the peer. However, if the configuration is already configured on the peer outside of SP, then the import fails the mutual exclusion check on the peer.

Solution

Remove the failed commands from conf-t on the peer and then retry import verify/commit.

Use the **show system internal csm info trace** command for further investigation to look at events, trace, or error messages.

Merge Failure

A merge between peers happens when a peer becomes reachable.

A merge is initiated when CFS sends a peer add for the peer or if the peer is already reachable. Configuring the sync-peer command starts the merge session.



Note

For a merge to succeed, the configuration in the switch-profile on both peers must match exactly.

Merge failure has many possible causes:

- [First time merge failure](#)
- [Merge after peers that were in sync previously](#)
- [Merge after reload](#)

**Note**

Use the **show system internal csm info trace** command to view events, trace, and error messages.

First time merge failure

Possible Cause

When peer switches are trying to synchronize configurations, the merge might fail when validating received configurations.

Solution

Use the **show switch-profile status** command to view which commands failed validation.

This implies that the commands on both the switches are configured differently.

Perform the following to correct the configurations:

- Remove the **sync-peers destination** command from the switch-profile.
- Use the **show running switch-profile** command on both peers to ensure that the configuration is exactly the same under switch-profile.
- Add back the **sync-peers destination** command to the switch-profile.
- Reissue the commit.

Merge after peers that were in sync previously

Possible Cause

If peers were in sync and connectivity was lost, and conflicting configuration changes were made on the switches, then the merge would fail.

Solution

Use the **show switch-profile status** command to view which commands failed the merge.

Correct the configurations and reissue the commit from the peer with the corrected configuration.

Merge after reload

Possible Cause

After a switch is reloaded, it sends its switch-profile configuration to the peer. If there was a configuration change done under SP for the peer that was not reloaded, then the merge fails.

Solution

Use the **show switch-profile status** command to view which commands failed the merge.

Correct the configurations and reissue the commit.

Switch-profile Deletion Failure

A rollback is used to delete the configurations during a switch-profile deletion.

**Note**

To check for commands that failed deletion, use the **show switch-profile status commit** command to view the status. Alternatively, use the **show switch-profile session-history** command by matching the session based on the timestamp/session type.

Switch-profile deletion failure has many possible causes:

- [Application failure](#)
- [Failure from dependent commands](#)
- [Application does not respond](#)

Other known switch-profile deletion issues:

- Rollback fails with "Deletion of switch profile failed" message (CSCti97003)
- Port-channel interface not deleted on switch-profile delete (CSCtf17697)

Application failure

Possible Cause

Switch-profile deletion failure might be that the application failed the command. It is possible that the configuration is deleted out of order.

The switch-profile does not order configurations as displayed in the show run output. There might be out of sequence issues that occur during the deletion of the switch-profile.

Solution

Use the **resequence-database** command in the conf-sync mode to resequence the commands in SP in the order that the commands appear in show running. After resequencing the commands, reissue the delete.

Failure from dependent commands

Possible Cause

Switch-profile deletion failure results from dependent commands in conf-t mode.

If a command inside SP is referenced by another command outside of SP and the first command inside SP is deleted, then failure occurs because the command outside of SP still references it.

Solution

Correct the commands, references, and reissue the delete.

Application does not respond

Possible Cause

The deletion fails because the application does not respond due to the application owning the command.

Solution

Correct the commands and reissue the delete.

Verify Failure

Verify failure has many possible causes:

- [Mutual exclusion check under local information](#)
- [Mutual exclusion check under peer information](#)
- [Rollback/ISSU in progress](#)
- [Global-db modification in progress](#)
- [Peer unable to accept lock request](#)



Note

Use the **show switch-profile status** command to view messages about the failure.



Note

Determine if the failure is on local/peer side by looking at whether the error is listed under local error(s)/peer error(s) or both.



Note

Use the **show system internal csm info trace** command to view events, trace, and error messages.

Mutual exclusion check under local information

Possible Cause

Command failed mutual-exclusion check under local information because the command has already been configured from conf-t.

Solution

Delete the command from conf-t mode and run verify from config-sync mode.

Mutual exclusion check under peer information

Possible Cause

Command failed mutual-exclusion check under peer information because the command has already been configured from conf-t on the peer.

Solution

Delete the command from conf-t mode on the peer and run verify from config-sync mode.

Rollback/ISSU in progress

Possible Cause

Verify cannot be performed when rollback/ISSU is in progress.

Solution

Stop rollback or wait for it to complete and run verify.

Global-db modification in progress

Possible Cause

Verify cannot be performed when global-db is being updated on the local/peer side.

Solution

Wait for the update to complete and run verify.

Peer unable to accept lock request

Possible Cause

Verify cannot be performed when the peer is unable to accept the lock request.

Solution

The peer is handling a transaction and cannot accept a lock request. Run verify at a later time.

Use the **show switch-profile status** command to determine if there is an ongoing transaction.

If the peer remains in the same state for a long time, use the **show cfs lock** command to determine if the CFS fabric has been locked.

Also check the application that has taken the CFS lock. If the application is ssnmgr, use the **show cfs internal session-history name session-mgr** command and the **show cfs internal notification log name session-mgr** command to view information about when a lock was acquired or released. It can also show the mapping to the csm transactions displayed with the **show switch-profile session-history** command.

■ Verify Failure