



# Consistency Checker

---

- [Limitations for Consistency Checker, on page 1](#)
- [Information about Consistency Checker, on page 1](#)
- [Running the Consistency Checker, on page 3](#)
- [Output Examples for Consistency Checker, on page 3](#)
- [Feature History for Consistency Checker, on page 7](#)

## Limitations for Consistency Checker

The Consistency Checker has the following limitations:

- Consistency Checkers are CPU intensive. It is not recommended to run the checkers at very short intervals.
- Legacy Consistency Checkers do not have support for snapshot. So, the previous runs cannot be displayed.
- There is no command to stop/abort the already running Consistency Checkers.
- Forwarding Engine hardware entry validations are partially implemented. Only programming failures can be detected and reported.
- Layer2 MAC Consistency Checker can validate the MAC address in hardware with software copy.
- Consistency checker is designed to reduce false positives in all cases. However, there could be rare cases of reporting a false positive in the following scenarios:
  - Large table state changes (i.e clear, relearn etc).
  - Under very high CPU usage due to any other feature while a consistency checker running. The consistency checker may report inconsistency in processes where CPU usage is high.

## Information about Consistency Checker

### Overview of Consistency Checker

The Consistency Checker collects information on various table states within the software and the hardware. It compares the software state with the hardware state. If there is any inconsistency, it flags the issue immediately. This helps to reduce increased troubleshooting time at a later period. The consistency checker

supplements basic troubleshooting and helps to identify scenarios where inconsistent states between software and hardware tables are causing issues in the network, thereby reducing the mean time to resolve the issue.

There are two types of consistency checker implementation available:

- Legacy Consistency Checker - supports validating the entry from control plane to the forwarding engine (or hardware copy).
- End-to-End Consistency Checker - supports validating the software entry from control plane to all processes involved in distributing and handling the entry, as well as the forwarding engine's hardware copy.

### End-to-End Consistency Checker

End-to-End (E2E) Consistency Checker supports full scan and single entry and should be started manually or run via gold diagnostic. The consistency checker can be started for a single entry using the command which helps to isolate the issue at which forwarding process entry is not consistent and helps speed up the debugging.

Every time the consistency checker is started, a runID is provided. Using the runID, its status, summary, details can be viewed. The last 5 snapshots are available any time for you to check the previous run's result.

E2E consistency checker performs the following functions:

- Validates the IOS entry to software tables/processes (Forwarding manger-RP, Forwarding manager-FP and FED) for all modules.
- Reports various inconsistencies (entry inconsistent, entry missing, stale entry) and sends a syslog to alert the administrator.
- Helps to speed up the fault isolation.
- Records any inconsistent entry with relevant data.
- Consistency checker supports the recursive single entry check which can validate the dependent objects along with the actual entry. (i.e, A Layer 3 Multicast with N outgoing interfaces can be validated for multicast entries along with OIFs programming, OIF's Adjacency validation, etc)
- Constant memory usages irrespective of total entries in a table.



---

**Note** The consistency checker is bound to CPU utilization and can not exceed the configured value while validating the tables across processes.

---

### Features Supported in Consistency Checker

The following features are supported in consistency checker:

- Legacy Consistency Checker
  - **Layer2 MAC Consistency Checker:** This consistency checker validates the IOS entry to FED software entry. It also validates the MAC address into hardware tables.
  - **Layer3 FMANFP Entry Consistency Checker:** This consistency checker validates the Layer 2, Layer 3, and multicast objects status in the Forwarding Manager-FP process. This includes stale objects and long pending objects.

- E2E Consistency Checker
  - **Layer2 Multicast Consistency Checker:** This consistency checker validates the IOS Layer 2 multicast IGMP/MLD VLAN, the group entry to Forwarding Manager-FP software entry, FED software entry, and FED hardware programming errors.

## Running the Consistency Checker

The table shown below lists the commands to run the various consistency checkers:

Command	Purpose
<b>show consistency-checker l2</b>	Runs the consistency-checker on the Layer 2 forwarding tables.
<b>show consistency-checker l3</b>	Runs the consistency-checker on the Layer 3 forwarding tables.
<b>show consistency-checker mcast l2m</b>	Runs the consistency-checker on the Layer 2 multicast forwarding tables.
<b>show consistency-checker objects</b>	Runs the End-to-End consistency-checker on objects.
<b>show consistency-checker run-id</b> <i>run-id</i>	Runs the End-to-End consistency-checker by run ID.
<b>show consistency-checker switch</b>	Runs the consistency-checker on the specified switch.

## Output Examples for Consistency Checker

The following is a sample output for the **show consistency-checker mcast l2m** command where the consistency checker runs a full scan:

```
Device# show consistency-checker mcast l2m start all
L2 multicast Full scan started. Run_id: 2
Use 'show consistency-checker run-id 2 status' for completion status.

Device#
*Feb 17 06:19:14.889: %FED_CCK_ERRMSG-4-INCONSISTENCY_FOUND: F0/0: fed: Consistency
Checker(CCK) detected inconsistency for l2m_vlan. Check 'show consistency run-id 2 detail'.
*Feb 17 06:19:14.890: %FED_CCK_ERRMSG-4-INCONSISTENCY_FOUND: F0/0: fed: Consistency
Checker(CCK) detected inconsistency for l2m_group. Check 'show consistency run-id 2 detail'.
Device#
*Feb 17 06:19:19.432: %IOSXE_FMANRP_CCK-6-FMANRP_COMPLETED: Consistency Check for Run-Id 2
is completed. Check 'show consistency-checker run-id 2'.
Device#
Device# show consistency-checker run-id 2 status
Process: IOSD
  Object-Type      Status           Time(sec)      Exceptions
  l2m_vlan         Completed        13             No
  l2m_group        Completed        13             No

Process: FMAN-FP
  Object-Type      Status           Time(sec)      State
  l2m_vlan         Completed        9              Consistent
```

```

l2m_group          Completed          9          Consistent

Process: FED
  Object-Type      Status          Time(sec)   State
  l2m_vlan         Completed      9           Inconsistent
  l2m_group        Completed      9           Inconsistent

Device#
Device# show consistency-checker run-id 2
Process: IOSD
  Object-Type      Start-time          Entries      Exceptions
  l2m_vlan         2021/02/17 06:19:05    22          0
  l2m_group        2021/02/17 06:19:05    24          0

Process: FMAN-FP
  *Statistics(A/I/M/S/Oth): Actual/Inherited/Missing/Stale/Others

  Object-Type      Start-time          State          A/  I/  M/  S/Oth
  l2m_vlan         2021/02/17 06:19:05    Consistent    0/  0/  0/  0/  0
  l2m_group        2021/02/17 06:19:05    Consistent    0/  0/  0/  0/  0

Process: FED
  *Statistics(A/I/M/S/HW/Oth): Actual/Inherited/Missing/Stale/Hardware/Others

  Object-Type      Start-time          State          A/  I/  M/  S/  HW/Oth
  l2m_vlan         2021/02/17 06:19:05    Inconsistent  1/  0/  0/168/  0/  0
  l2m_group        2021/02/17 06:19:05    Inconsistent  4/  0/  2/  0/  0/  0

Device#
Device# show consistency-checker run-id 2 detail
Process: IOSD

Process: FMAN-FP

Process: FED
  Object-Type:l2m_vlan  Start-time:2021/02/17 06:19:05
  Status:Completed  State:Inconsistent
  Key/data          Reason
  (Ipv4, vlan: 768)      Stale
  snoop:off stp_tcn:off flood:off pimsn:off
  (Ipv4, vlan: 769)      Stale
  snoop:off stp_tcn:off flood:off pimsn:off
  (Ipv6, vlan: 900)      Inconsistent
  snoop:on stp_tcn:on flood:on pimsn:off
  (Ipv6, vlan: 767)      Stale
  snoop:off stp_tcn:off flood:off pimsn:off

  Object-Type:l2m_group  Start-time:2021/02/17 06:19:05
  Status:Completed  State:Inconsistent
  Key/data          Reason
  (Ipv4, vlan:100 (*,227.0.0.0))      Inconsistent
  Group ports: total entries: 0
  (Ipv4, vlan:100 (*,227.1.0.0))      Missing

Device#

```

The following is a sample output for the **show consistency-checker mcast l2m** command where the consistency checker runs a recursive single-entry scan:

```

Device# show consistency-checker mcast l2m start vlan 900 229.1.1.1 recursive
Single entry scan started with Run_id: 2

*Feb 17 06:54:09.880: %IOSXE_FMANRP_CCK-6-FMANRP_COMPLETED: Consistency Check for Run-Id 2

```

```

is completed.
Check 'show consistency-checker run-id 2'.
Device#
Device# show consistency-checker run-id 2
Process: IOSD
  Object-Type      Start-time          Entries      Exceptions
  l2m_vlan         2021/02/17 06:54:01      1            0
  l2m_group        2021/02/17 06:54:01      1            0

Process: FMAN-FP
  *Statistics(A/I/M/S/O): Actual/Inherited/Missing/Stale/Others

  Object-Type      Start-time          State          A / I / M / S / O
  l2m_vlan         1970/01/01 00:10:03      Consistent    0/ 0/ 0/ 0/ 0
  l2m_group        1970/01/01 00:10:03      Consistent    0/ 0/ 0/ 0/ 0

Process: FED
  *Statistics(A/I/M/S/HW/O): Actual/Inherited/Missing/Stale/Hardware/Others

  Object-Type      Start-time          State          A / I / M / S / HW/ O
  l2m_vlan         2021/02/17 06:54:01      Inconsistent  1/ 0/ 0/ 0/ 0/ 0
  l2m_group        2021/02/17 06:54:01      Inconsistent  0/ 1/ 0/ 0/ 0/ 0

Device#
Device# show consistency-checker run-id 2 detail
Process: IOSD
  Object-Type:l2m_vlan  Start-time:2021/02/17 06:54:01
  Key/data              Reason
  (Ipv4, vlan:900)      Success
  snoop:on stp_tcn:off flood:off pimsn:off

  Object-Type:l2m_group  Start-time:2021/02/17 06:54:01
  Key/data              Reason
  (Ipv4, vlan:900, (*,229.1.1.1))  Success
  Twel/0/5

Process: FMAN-FP

Process: FED
  Object-Type:l2m_group  Start-time:2021/02/17 06:54:01
  Status:Completed      State:Inconsistent
  Key/data              Reason
  (Ipv4, vlan:900 (*,229.1.1.1))  Inherited
  Group ports: total entries: 1
  TwentyFiveGigE1/0/5

  -----Recursion-level-1-----
  Object-Type:l2m_vlan  Start-time:2021/02/17 06:54:01
  Status:Completed      State:Inconsistent
  Key/data              Reason
  (Ipv4, vlan: 900)      Inconsistent
  snoop:on stp_tcn:off flood:on pimsn:off

Device#

```

The following is a sample output for the **show consistency-checker objects** command where the consistency checker runs a scan on objects:

```

Device# show consistency-checker objects l2m_group
Process: IOSD
  Run-id      Start-time          Exception
  1           2021/02/17 05:20:42      0

```

```
2          2021/02/17 06:19:05      0
```

```
Process: FMAN-FP
```

```
*Statistics(A/I/M/S/Oth): Actual/Inherited/Missing/Stale/Others
```

Run-id	Start-time	State	A/	I/	M/	S/Oth
1	2021/02/17 05:20:42	Consistent	0/	0/	0/	0/ 0
2	2021/02/17 06:19:05	Consistent	0/	0/	0/	0/ 0

```
Process: FED
```

```
*Statistics(A/I/M/S/HW/Oth): Actual/Inherited/Missing/Stale/Hardware/Others
```

Run-id	Start-time	State	A/	I/	M/	S/	HW/Oth
1	2021/02/17 05:20:42	Consistent	0/	0/	0/	0/	0/ 0
2	2021/02/17 06:19:05	Inconsistent	4/	0/	2/	0/	0/ 0

```
Device#
```

```
Stark#sh consistency-checker run 2 detail
```

```
Process: IOSD
```

```
Object-Type:l2m_vlan Start-time:2021/02/17 06:54:01
Key/data Reason
(Ipv4, vlan:900) Success
snoop:on stp_tcn:off flood:off pimsn:off
```

```
Object-Type:l2m_group Start-time:2021/02/17 06:54:01
Key/data Reason
(Ipv4, vlan:900, (*,229.1.1.1)) Success
Twel/0/5
```

```
Process: FMAN-FP
```

```
Process: FED
```

```
Object-Type:l2m_group Start-time:2021/02/17 06:54:01
Status:Completed State:Inconsistent
Key/data Reason
(Ipv4, vlan:900 (*,229.1.1.1)) Inherited
Group ports: total entries: 1
TwentyFiveGigE1/0/5
```

```
-----Recursion-level-1-----
```

```
Object-Type:l2m_vlan Start-time:2021/02/17 06:54:01
Status:Completed State:Inconsistent
Key/data Reason
(Ipv4, vlan: 900) Inconsistent
snoop:on stp_tcn:off flood:on pimsn:off
```

```
Device# show consistency-checker objects l2m_group 2 detail
```

```
Process: IOSD
```

```
Process: FMAN-FP
```

```
Process: FED
```

```
Object-Type:l2m_group Start-time:2021/02/17 06:19:05
Status:Completed State:Inconsistent
Key/data Reason
(Ipv4, vlan:100 (*,227.0.0.0)) Inconsistent
Group ports: total entries: 0
(Ipv4, vlan:100 (*,227.1.0.0)) Missing
(Ipv4, vlan:100 (*,227.0.0.1)) Inconsistent
Group ports: total entries: 0
(Ipv4, vlan:100 (*,227.1.0.1)) Missing
(Ipv4, vlan:100 (*,227.0.0.2)) Inconsistent
Group ports: total entries: 0
```

```
(Ipv4, vlan:100 (*,227.0.0.3))      Inconsistent
Group ports: total entries: 0
```

Device#

## Feature History for Consistency Checker

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.3.1	Consistency Checker	The Consistency Checker collects information on various table states within the software and the hardware and flags any inconsistency it finds immediately. It supplements basic troubleshooting and helps to identify scenarios where inconsistent states between software and hardware tables are causing issues in the network, thereby reducing the mean time to resolve the issue.
Cisco IOS XE Bengaluru 17.6.1	Consistency Checker	This feature was enhanced and the multicast consistency checkers were introduced. The following keywords were added to the <b>show consistency-checker</b> command: <b>mcast</b> , <b>objects</b> , and <b>run-id</b> .

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>

<http://www.cisco.com/go/cfn>.

