



Redundancy Management Interface

- [Redundancy management interfaces, on page 1](#)

Redundancy management interfaces

A redundancy management interface is a high availability feature that

- enables pairing scenarios and gateway monitoring for Cisco Catalyst 9800 Series Wireless Controllers
- supports both IPv4 and IPv6 dual stack environments, and
- facilitates dynamic pairing, upgrade/downgrade behaviors, ARP handling, and AAA integration.

Gateway monitoring

Gateway monitoring is a network management capability that

- selects the gateway IP based on the static routes that match the RMI subnet using the broadest mask and least gateway IP
- provides enhanced visibility and statistics related to gateway reachability and RMI state, and
- simplifies troubleshooting and control over high availability (HA) and RMI functionality by offering detailed diagnostics.

Key aspects

Gateway IP selection:

- From Cisco IOS XE 17.2.1 onwards, the **ip default-gateway gateway-ip** command is deprecated for RMI gateway configuration.
- The system automatically chooses a gateway IP by evaluating static routes and selecting the gateway that:
 - matches the RMI subnet,
 - uses the broadest subnet mask,
 - and has the lowest IP address.

- If no matching static route exists, gateway failover does not operate—even if management gateway-failover is enabled.

Enhanced statistics and transparency:

- From Cisco IOS XE 17.18.1 onwards, gateway monitoring provides improved visibility into gateway reachability, RMI state, and detailed diagnostics.
- **Benefits include:**
 - Enhanced visibility into gateway health and RMI state.
 - Simplified troubleshooting through detailed statistics and state information.
 - Greater transparency and control over HA and RMI, improving reliability and diagnostics.

Refer to the [Gateway statistics monitoring, on page 4](#) for gateway monitoring commands.

Airport's air traffic control system

Gateway monitoring works like an airport's air traffic control system. Instead of a pilot manually selecting a runway (like using the old **ip default-gateway** command), a central system automatically identifies the best runway based on current conditions (route matching, subnet mask, and lowest IP address). It then tracks the success or failure of each takeoff and landing (probe statistics and diagnostics), making sure the whole airport runs smoothly and efficiently while providing real-time updates and troubleshooting support when issues arise.

Configure gateway monitoring interval (CLI)

Set the interval at which the management gateway failover feature monitors gateway availability on a switch.

Adjusting the gateway monitoring interval determines how frequently the device checks gateway status to trigger failover if necessary. This procedure is performed using command-line interface (CLI) commands.

Before you begin

- Ensure you have access to the device CLI in privileged EXEC mode.
- Confirm that you have the required permissions to enter configuration mode and modify gateway monitoring settings.

Follow these steps to configure the gateway monitoring interval using the CLI:

Procedure

Step 1 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 2 **management gateway-failover interval *interval-value***

Example:

```
Device(config)# management gateway-failover interval 6
```

Configures the gateway monitoring interval.

interval-value - Refers to the gateway monitoring interval. The valid range is from 6 to 12. Default value is 8.

Step 3 **end**

Example:

```
Device(config)# end
```

Saves the configuration and exits configuration mode and returns to privileged EXEC mode.

The system monitors the gateway at the specified interval. Your configuration is saved, and the device returns to privileged EXEC mode.

Configure gateway monitoring (CLI)

Enable and configure gateway monitoring on a device.

This task is typically performed to ensure that network devices can effectively monitor and manage gateway connections, facilitating smooth network operations and management. Gateway monitoring involves setting up a default gateway and enabling monitoring features.

Follow these steps to configure gateway monitoring (CLI):

Procedure

Step 1 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 2 **[no] management gateway-failover enable**

Example:

```
Device(config)# management gateway-failover enable
```

Enables gateway monitoring. (Use the **no** form of this command to disable gateway monitoring.)

Step 3 **end**

Example:

```
Device(config)# end
```

Returns to privileged EXEC mode.

The gateway monitoring is enabled and configured, and the configuration changes are active.

What to do next

To save the configuration, use the **write memory** command.

Gateway statistics monitoring

To display detailed gateway statistics, use the **show platform software rif-mgr chassis {active | standby } r0 gateway-statistics** command and to display gateway status information, use the **show platform software rif-mgr chassis { active | standby } r0 resource-status** command.

```
Device# show platform software rif-mgr chassis active r0 gateway-statistics
```

```
Number of L2 (ARP/ND) probes sent      : 164890
Number of ICMP probes sent             : 20
Number of L2 (ARP/ND) responses received : 164878
Number of ICMP responses received      : 0
Number of L2 (ARP/ND) responses failed : 12
Number of ICMP responses failed        : 20
Last fail count                        : 32   at (05/26/25 18:21:33)
Longest fail count                     : 32   at (05/26/25 18:21:33)
```

```
Device# show platform software rif-mgr chassis active r0 resource-status
```

```
RIF Resource Status
```

```
RP Status   : Up
RMI Status  : Up
Current Chassis State : Active
Peer Chassis State    : Standby
Gateway Monitoring Configured : True
Gateway Monitoring Interval   : 10
Gateway Status           : Up
Peer Gateway Status       : Up
```

Gateway Reachability Detection

Gateway reachability detection

Gateway reachability detection is a feature that

- minimizes the downtime on APs and clients when the gateway reachability is lost on the active controller
- enables both active and standby controllers to keep track of gateway reachability by sending Internet Control Message Protocol (ICMP) and ARP requests periodically to the gateway, and
- detects gateway failures within approximately 8 seconds using consecutive failure monitoring.

Gateway reachability detection behavior

Both active and standby controllers use the RMI IP as the source IP. The messages are sent at 1 second interval. If it takes 8 (or configured value) consecutive failures in reaching the gateway, the controller declares the gateway as non-reachable. It takes approximately 8 seconds to detect if a controller has lost gateway reachability.

Gateway monitoring with native IPv6 uses ICMP Neighbor Discovery protocols and ICMPv6 ECHO to check gateway reachability.

Therefore, you can monitor only the IPv6 gateway when RMI IPv6 is configured.

This means that only one IPv4 or IPv6 gateways can be monitored.



Note If the standby controller loses gateway, the standby moves to the standby recovery mode.
If the active controller loses gateway, the active reloads and standby becomes active.

Requirement: proper configuration of default gateway

To ensure the gateway reachability check functions correctly, configure the default gateway next hop in the same subnet as the WMI, which also serves the RMI.

If you configure the default gateway with a next hop IP address outside the WMI subnet, the controller does not send packets for the gateway reachability check. This configuration can cause failover or redundancy issues.

Configure system redundancy workflow

Summary

The key components involved in the configuration workflow are:

- Redundancy Management Interface (RMI): Provides redundancy capabilities through GUI or CLI configuration
- Controllers: Require reloading for RMI configuration to take effect
- IPv6 Static Route: Enables network routing configuration
- Gateway Monitoring Interval: Monitors gateway connectivity at specified intervals

Workflow

The configuration workflow involves these stages:

1. Configure Redundancy Management Interface using either GUI or CLI method

For more information, see [Configure redundancy management interface \(GUI\), on page 26](#).



Note For RMI configuration to take effect, ensure that you reload your controllers.

2. Configure IPv6 Static Route to establish network routing

For information, see [Gateway monitoring, on page 1](#).

3. Configure Gateway Monitoring Interval using CLI to set monitoring parameters

For more information, see [Configure gateway monitoring interval \(CLI\), on page 2](#).

Migrate to RMI IPv6

You can migrate to RMI IPv6 from two different configurations: from RMI IPv4 or from HA pairing without RMI. The migration process varies depending on your current configuration.

Before you begin

Follow these steps to migrate to RMI IPv6:

Procedure

Determine your current configuration and follow the appropriate migration path.

- If migrating from RMI IPv4, continue with the following substeps:

- a. Unconfigure the RMI IPv4 using the following CLIs:

```
Device# conf t
Device(config)# no redun-management interface <vlan_name> chassis 1 address
<ip_address1> chassis 2 address <ip_address2>
```

Note

This CLI unconfigures RMI on both the controllers.

- b. **Note**

Take a backup of the running config on active before you reload the controller.

Reload the controller.

- c. Copy the backed up config to the running config on the box which would have lost all the config.
- d. Configure the RMI IPv6 on both the controllers.
- e. Reload the controller.

- If migrating from HA Pairing (without RMI), for information on HA pairing, see [Configuring Redundancy Management Interface \(GUI\)](#).
-

Standby monitoring

Standby monitoring is a system health feature that

- monitors the health of a system on a standby controller using programmatic interfaces and commands
- allows monitoring of parameters such as CPU, memory, interface status, power supply, fan failure, and the system temperature
- is enabled when Redundancy Management Interface (RMI) is configured, and
- cannot be dynamically enabled or disabled.

Requirements and configuration

Standby Monitoring is enabled when Redundancy Management Interface (RMI) is configured, no other configuration is required. The RMI itself is used to connect to the standby and perform standby monitoring.



Note The active controller uses the management or RMI IP to initiate AAA requests. Whereas, the standby controller uses the RMI IP to initiate AAA requests. Thus, the RMI IPs must be added in AAA servers for a seamless client authentication and standby monitoring.

To enable standby console, ensure that the following configuration is in place:

```
redundancy
main-cpu
secondary console enable
```



Note The Standby Monitoring feature is not supported on a controller in the active-recovery and the standby-recovery modes.

The Standby Monitoring feature supports only the following traffic on the RMI interface of the standby controller:

- Address Resolution Protocol (ARP)
- Internet Control Message Protocol (ICMP)
- TCP Traffic (to or from) ports: 22, 443, 830, and 3200
- UDP RADIUS ports: 1645 and 1646
- UDP Extended RADIUS ports: 21645 to 21844

Feature scenarios include:

- Monitoring the health of the standby directly from the standby controller using Standby RMI IP
- Getting syslogs from the standby controller using the Standby RMI IP

Use cases include:

- Enabling SNMP agent and programmatic interfaces on the standby controller: You can directly perform an SNMP query or programmatic interface query to the standby's RMI IP and active controller
- Enabling syslogs on the standby controller: You can directly get the standby syslogs from the standby controller

RADIUS accounting support:

Whenever you log in to a standby device, the RADIUS start record must be sent to the external RADIUS server. Similarly, when you log out of a device, the RADIUS stop record must be sent to the external RADIUS server.

TACACS+ authentication support:

Users are authenticated through the RMI using the external TACACS+ server. The username and password are evaluated in the TACACS+ server. Depending on the response received from the server, a user will be able to log in to the standby device.

TACACS+ accounting support:

Whenever you log in to the standby device, the TACACS+ accounting start record must be sent to the external TACACS+ server. Similarly, when you log out of a device, the TACACS+ accounting stop record must be sent to the external TACACS+ server.



Note This configuration must be in place to configure AAA to send the accounting packets:

```
aaa accounting exec {default | named-list} start-stop group {RAD | tac-group-name}
```



Note The TACACS+ login to the standby device is not supported when TACACS+ server is configured with hostname.

Monitoring the Health of Standby Parameters Using SNMP

Standby monitoring using standby RMI IP

Standby monitoring using standby RMI IP is a network management capability that

- enables direct SNMP queries to the standby controller's RMI IP address
- supports IF-MIB monitoring for interface statistics on standby controllers from Release 17.5 onwards, and
- automatically enables SNMP on the standby controller when SNMP agent is enabled on the active controller.

Supported mibs and monitoring capabilities

When an SNMP agent is enabled on the standby controller, you can directly perform an SNMP query to the standby's RMI IP. From Release 17.5 onwards, you can query the following MIB on the standby controller:

Table 1: MIB name and notes

MIB Name	Notes
IF-MIB	This MIB is used to monitor the interface statistics of the standby controller using the standby RMI IP address.



Note If an SNMP agent is enabled on the active controller, by default, the SNMP is enabled on the standby controller.

Standby monitoring using the active controller

Monitor the health parameters of the standby controller, including memory, CPU, port status, power statistics, and peer gateway latencies, using various MIB objects through the active controller.

CISCO-LWAPP-HA-MIB

The CISCO-LWAPP-HA-MIB monitors the health parameters of the standby controller, that is, memory, CPU, port status, power statistics, peer gateway latencies, and so on.

You can query these MIB objects of CISCO-LWAPP-HA-MIB.

Table 2: MIB objects and notes

MIB Objects	Notes
cLHaPeerHotStandbyEvent	This object can be used to check if the standby controller has turned hot-standby or not.
cLHaBulkSyncCompleteEvent	This object represents the time at which the bulksync is completed.

CISCO-PROCESS-MIB

The CISCO-PROCESS-MIB monitors CPU and process statistics. Use it to monitor CPU-related or memory-related BINOS processes. The standby CISCO-PROCESS-MIB can be monitored using the active controller.

ENTITY-MIB

The ENTITY-MIB is used to monitor hardware details of the active and standby controllers using the active controller.



Note The standby Route Processor (RP) sensors are appended in the active RP sensors.

Standby IOS linux syslogs

Standby IOS Linux syslogs are a logging mechanism that

- relay logs using the same method as on the active Cisco IOS for wireless controllers
- enable external logging from the standby IOS starting from Release 17.5 onwards, and
- forward all syslogs generated on the standby controller to the configured external server through BINOS processes.

Standby syslog behavior

From Release 17.5 onwards, external logging of syslogs from the standby IOS is enabled. As BINOS processes on standby also forwards the syslogs to Cisco IOS, all the syslogs generated on the standby controller is forwarded to the configured external server.



Note RMI IP address is used for logging purpose.

The standby controller tries to send syslogs to the IPv4 server because logging is only configured on IPv4 even though IPv4 is not supported by standby. This is the expected behavior when an HA pair is configured with the RMI IPv6 address, the active controller has dual stack, and logging is configured on the IPv4 address

Standby interface status using active SNMP

The standby interface information is sent to the active controller using IPC in these scenarios:

- When there is a change in the interface status.
- When a new interface is added or deleted on the standby controller.

When the active controller receives the interface information from the standby controller, the active controller's database is populated with the standby interface information.

When an SNMP query is received for the standby interface information, the SNMP handlers corresponding to the CISCO-LWAPP-HA-MIB reads them from the standby interface database on the active and populates the MIB objects in CISCO-LWAPP-HA-MIB.

You can query the following MIB objects of CISCO-LWAPP-HA-MIB.

Table 3: MIB objects of CISCO-LWAPP-HA-MIB

MIB Object	Notes
stbyIfIndex	This is a unique value (greater than zero) for each interface of the standby controller.
stbyIfName	This is the name of the standby interface.
stbyIfPhysAddress	This is the interface address of the standby controller in the protocol sublayer.
stbyifOperStatus	This is the current operational state of the interface in the standby controller.
stbyifAdminStatus	This is the desired state of the interface of the standby controller.

To verify the logging on the active when the standby fails to send interface statistics, use this command:

```
Device# debug snmp ha-chkpt
Device# debug snmp ha-intf_db
```

Standby controller health monitoring using programmatic interfaces

You can monitor parameters such as CPU, memory, sensors, and interface status on a standby controller using programmatic interfaces such as NETCONF and RESTCONF. The RMI IP of the standby controller can be used for access to the following operational models:

The models can be accessed through:

- Cisco-IOS-XE-device-hardware-oper.yang
- Cisco-IOS-XE-process-cpu-oper.yang
- Cisco-IOS-XE-platform-software-oper.yang
- Cisco-IOS-XE-process-memory-oper.yang
- Cisco-IOS-XE-interfaces-oper.yang

For more information on the YANG models, see the *Programmability Configuration Guide, Cisco IOS XE 17.3.x*.

Standby controller health monitoring (CLI)

This section describes the different commands that can be used to monitor the standby device.

You can connect to the standby controller through SSH using the RMI IP of the standby controller. The user credentials must have been configured already. Both local authentication and RADIUS authentication are supported.



Note The **redun-management** command needs to be configured on both the controllers, primary and standby, prior to high availability (HA) pairing.

Port state monitoring

This is a sample output of the **show interfaces interface-name** command:

```
Device-standby# show interfaces GigabitEthernet1

GigabitEthernet1 is down, line protocol is down
Shadow state is up, true line protocol is up
  Hardware is CSR vNIC, address is 000c.2909.33c2 (bia 000c.2909.33c2)
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full Duplex, 1000Mbps, link type is force-up, media type is Virtual
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:06, output 00:00:24, output hang never
  Last clearing of "show interface" counters never
  Input queue: 30/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 389000 bits/sec, 410 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    3696382 packets input, 392617128 bytes, 0 no buffer
    Received 0 broadcasts (0 multicasts)
    0 runs, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog, 0 multicast, 0 pause input
    18832 packets output, 1218862 bytes, 0 underruns
    Output 0 broadcasts (0 multicasts)
    0 output errors, 0 collisions, 2 interface resets
```

```

3 unknown protocol drops
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier, 0 pause output
0 output buffer failures, 0 output buffers swapped out

```

This is a sample output of the **show ip interface brief** command:

Device# **show ip interface brief**

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet1	unassigned	YES	unset	down	down
GigabitEthernet0	unassigned	YES	NVRAM	administratively down	down
Capwap1	unassigned	YES	unset	up	up
Capwap2	unassigned	YES	unset	up	up
Capwap3	unassigned	YES	unset	up	up
Capwap10	unassigned	YES	unset	up	up
Vlan1	unassigned	YES	NVRAM	down	down
Vlan56	unassigned	YES	unset	down	down
Vlan111	111.1.1.85	YES	NVRAM	up	up

CPU or memory monitoring

This is a sample output of the **show process cpu sorted 5sec** command:

Device-standby# **show process cpu sorted 5sec**

```

CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%
PID Runtime(ms)   Invoked    uSecs   5Sec   1Min   5Min TTY Process
  10    1576556      281188    5606   0.15%  0.05%  0.05%  0 Check heaps
 232    845057      54261160   15    0.07%  0.05%  0.06%  0 IPAM Manager
 595      177        300        590   0.07%  0.02%  0.01%  2 Virtual Exec
 138   1685973   108085955   15    0.07%  0.08%  0.08%  0 L2 LISP Punt Pro
 193    19644      348767     56    0.07%  0.00%  0.00%  0 DTP Protocol
   5         0         1         0    0.00%  0.00%  0.00%  0 CTS SGACL db cor
   4        24        15    1600   0.00%  0.00%  0.00%  0 RF Slave Main Th
   6         0         1         0    0.00%  0.00%  0.00%  0 Retransmission o
   7         0         1         0    0.00%  0.00%  0.00%  0 IPC ISSU Dispatc
   2    117631    348801    337   0.00%  0.00%  0.00%  0 Load Meter
   8         0         1         0    0.00%  0.00%  0.00%  0 EDDRI_MAIN

```

To check CPU and memory utilization of binOS processes, run this command:

Device-standby# **show platform software process slot chassis standby R0 monitor**

```

top - 23:24:14 up 8 days, 3:38, 0 users, load average: 0.69, 0.79, 0.81
Tasks: 433 total, 1 running, 431 sleeping, 1 stopped, 0 zombie
%Cpu(s): 1.7 us, 2.8 sy, 0.0 ni, 95.6 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 32059.2 total, 21953.7 free, 4896.8 used, 5208.6 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 26304.6 avail Mem

```

```

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
23565 root 20 0 2347004829924 100840 S 6.7 2.5 550:29 btrace_rotator
23595 root 20 0 1820072 334636 100840 S 3.3 1.0 179:05 wcm
23597 root 20 0 1820072 334636 100840 S 3.3 1.0 179:03 wcm
23599 root 20 0 1820072 334636 100840 S 3.3 1.0 179:04 wcm
23601 root 20 0 1820072 334636 100840 S 3.3 1.0 179:05 wcm
23603 root 20 0 1820072 334636 100840 S 3.3 1.0 179:04 wcm

```

To check the environment temperature and fan status, run this command:

Device-standby# **show environment summary**

Switch	FAN (Intake)	FAN (Exhaust)	POWER SUPPLY 1	POWER SUPPLY 2
R0	Normal	Normal	Normal	Normal
Switch	Inlet Temp (C)	Outlet Temp (C)		
R0	22	28		
R0	Stby Temp: DDC INormal		22	Celsius (55 ,65 ,75 ,80) (Celsius)
R0	Stby Temp: DDC ONormal		28	Celsius (75 ,85 ,95 ,100) (Celsius)



Note The command displays both active and standby hardware details.



Note The **show environment summary** command displays data only for physical appliances such as Cisco Catalyst 9800-80 Wireless Controller, Cisco Catalyst 9800-40 Wireless Controller, Cisco Catalyst 9800-L Wireless Controller, and Cisco Catalyst 9800 Embedded Wireless Controller for Switch. The command does not display data for Cisco Catalyst 9800 Wireless Controller for Cloud.

Gateway-monitoring configuration verification

Verify the status of the gateway-monitoring configuration on active and standby controllers using specific show commands.

To verify the status of the gateway-monitoring configuration on an active controller, run this command:

Device# **show redundancy states**

```
my state = 13 -ACTIVE
peer state = 8 -STANDBY HOT
Mode = Duplex
Unit = Primary
Unit ID = 1

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured) = sso
Redundancy State = sso
Maintenance Mode = Disabled
Manual Swact = enabled
Communications = Up

client count = 129
client_notification_TMR = 30000 milliseconds
RF debug mask = 0x0
Gateway Monitoring = Disabled
Gateway monitoring interval = 8 secs
```

To verify the status of the gateway-monitoring configuration on a standby controller, run this command:

Device-stby# **show redundancy states**

```
my state = 8 -STANDBY HOT
peer state = 13 -ACTIVE
```

```

Mode = Duplex
Unit = Primary
Unit ID = 2

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured) = sso
Redundancy State = sso
Maintenance Mode = Disabled
Manual Swact = cannot be initiated from this the standby unit
Communications = Up

client count = 129
client_notification_TMR = 30000 milliseconds
RF debug mask = 0x0
Gateway Monitoring = Disabled
Gateway monitoring interval = 8 secs

```

RMI IPv4 configuration verification

Verify the RMI IPv4 configuration.

```
Device# show running-config interface vlan management-vlan
```

Building configuration...

```

Current configuration : 109 bytes
!
interface Vlan90
ip address 9.10.90.147 255.255.255.0 secondary
ip address 9.10.90.41 255.255.255.0
end

```

To verify the interface configuration for a standby controller, use this command:

```
Device-stby# show running-config interface vlan 90
```

Building configuration...

```

Current configuration : 62 bytes
!
interface Vlan90
ip address 9.10.90.149 255.255.255.0
end

```

To verify the chassis redundancy management interface configuration for an active controller, use this command:

```
Device# show chassis rmi
```

```

Chassis/Stack Mac Address : 000c.2964.1eb6 - Local Mac Address
Mac persistency wait time: Indefinite

```

H/W Current								
Chassis#	Role	Mac Address	Priority	Version	State	IP		RMI-IP
*1	Active	000c.2964.1eb6	1	V02	Ready	169.254.90.147	9.10.90.147	
2	Standby	000c.2975.3aa6	1	V02	Ready	169.254.90.149	9.10.90.149	

To verify the chassis redundancy management interface configuration for a standby controller, use this command:

```
Device-stby# show chassis rmi
```

Chassis/Stack Mac Address : 000c.2964.1eb6 - Local Mac Address
 Mac persistency wait time: Indefinite

Chassis#	Role	Mac Address	Priority	H/W Version	Current State	IP	RMI-IP
1	Active	000c.2964.1eb6	1	V02	Ready	169.254.90.147	9.10.90.147
*2	Standby	000c.2975.3aa6	1	V02	Ready	169.254.90.149	9.10.90.149

To verify the ROMMON variables on an active controller, use this command:

Device# **show romvar | include RMI**

RMI_INTERFACE_NAME = Vlan90
 RMI_CHASSIS_LOCAL_IP = 9.10.90.147
 RMI_CHASSIS_REMOTE_IP = 9.10.90.149

To verify the ROMMON variables on a standby controller, use this command:

Device-stby# **show romvar | include RMI**

RMI_INTERFACE_NAME = Vlan90
 RMI_CHASSIS_LOCAL_IP = 9.10.90.149
 RMI_CHASSIS_REMOTE_IP = 9.10.90.147

To verify the switchover reason, use this command:

Device# **show redundancy switchover history**

Index	Previous active	Current active	Switchover reason	Switchover time
-----	-----	-----	-----	-----
1	2	1	Active lost GW	17:02:29 UTC Mon Feb 3 2020

RMI IPv6 configuration verification

Verify the RMI IPv6 configuration.

To verify the chassis redundancy management interface configuration for both active and standby controllers, run this command:

Device# **show chassis rmi**

Chassis/Stack Mac Address : 00a3.8e23.a540 - Local Mac Address
 Mac persistency wait time: Indefinite
 Local Redundancy Port Type: Twisted Pair

Chassis#	Role	Mac Address	Priority	H/W Version	Current State	IP	RMI-IP
-----	-----	-----	-----	-----	-----	-----	-----
1	Standby	706d.1536.23c0	1	V02	Ready	169.254.254.17	2020:0:0:1::211
*2	Active	00a3.8e23.a540	1	V02	Ready	169.254.254.18	2020:0:0:1::212

To verify the RMI related ROMMON variables for both active and standby controllers, run this command

Device# **show romvar | i RMI**

RMI_INTERFACE_NAME = Vlan52
 RMI_CHASSIS_LOCAL_IPV6 = 2020:0:0:1::212
 RMI_CHASSIS_REMOTE_IPV6 = 2020:0:0:1::211

Redundancy port interface configuration verification

Use these commands to verify redundancy port interface configuration for RMI and RP connections in both active and standby instances.

RMI link verification commands

To verify the RMI link re-establishment count and the time since the RMI link is Up in the active instance, run these command:

```
Device# show platform software rif-mgr chassis active R0 rmi-connection-details
RMI Connection Details
  RMI Link re-establish count : 2
  RMI Link Uptime             : 21 hours 8 minutes 43 seconds
  RMI Link Upsince            : 08/05/2021 13:46:01
```

To verify the RMI link re-establishment count and the time since the RMI link is Down in the active instance, run these command:

```
Device# show platform software rif-mgr chassis active R0 rmi-connection-details
RMI Connection Details
  RMI Link re-establish count : 1
  RMI Link Downtime           : 28 seconds
  RMI Link Downsince          : 07/16/2021 03:19:11
```

To verify the RMI link re-establishment count and the time since the RMI link is Up in the standby instance, run these command:

```
Device# show platform software rif-mgr chassis standby R0 rmi-connection-details
RMI Connection Details
  RMI Link re-establish count : 1
  RMI Link Uptime             : 1 hour 39 minute 9 seconds
  RMI Link Upsince            : 07/16/2021 01:31:41
```

To verify the RMI link re-establishment count and the time since the RMI link is Down in the standby instance, run these command:

```
Device# show platform software rif-mgr chassis standby R0 rmi-connection-details
RMI Connection Details
  RMI Link re-establish count : 1
  RMI Link Downtime           : 22 seconds
  RMI Link Downsince          : 07/16/2021 03:19:17
```

RP link verification commands

To verify the RP link re-establishment count and the time since the RP link is UP for days in the active instance, run the following command:

```
Device# show platform software rif-mgr chassis active R0 rp-connection-details
RP Connection Details
  RP Connection Uptime : 12 days 17 hours 1 minute 39 seconds
  RP Connection Upsince : 07/03/2021 07:06:20
```

To verify the RP link re-establishment count and the time since the RP link is Down in the active instance, run the following command:

```
Device# show platform software rif-mgr chassis active R0 rp-connection-details
RP Connection Details
  RP Connection Downtime : 4 seconds
  RP Connection Downsince : 07/16/2021 03:33:04
```

To verify the RP link re-establishment count and the time since the RP link is UP in the standby instance, run the following command:

```
Device# show platform software rif-mgr chassis standby R0 rp-connection-details
RP Connection Details
  RP Connection Uptime   : 12 days 17 hours 2 minutes 1 second
  RP Connection Upsince  : 07/03/2021 07:05:58
```

To verify the RP link re-establishment count and the time since the RP link is Down in the standby instance, run the following command:

```
Device# show platform software rif-mgr chassis standby R0 rp-connection-details
RP Connection Details
  RP Connection Downtime   : 22 seconds
  RP Connection Downsince  : 07/16/2021 03:19:17
```

RIF and stack manager internal statistics verification

To verify the RIF and stack manager internal statistics in the active instance, run the following command:

```
Device# show platform software rif-mgr chassis active R0 rif-stk-internal-stats
RIF Stack Manager internal stats

  Stack-mgr reported RP down           : False
  DAD link status reported to Stack-Mgr : True
```

To verify the RIF and stack manager internal statistics in the standby instance, run the following command:

```
Device# show platform software rif-mgr chassis standby R0 rif-stk-internal-stats
RIF Stack Manager internal stats

  Stack-mgr reported RP down           : False
  DAD link status reported to Stack-Mgr : True
```

LMP statistics verification

To verify the number of packets sent or received for each type in the active instance, run the following command:

```
Device# show platform software rif-mgr chassis active R0 lmp-statistics
LMP Statistics

Info Type Sent           : 6
Solicit Info Type Sent   : 0
Unsolicit Info Type Sent : 6
Reload Type Sent         : 0
Recovery Type Sent       : 1
Gateway Info Type Sent   : 0
Enquiry Type Sent        : 0
Solicit Enquiry Type Sent : 0
Unsolicit Enquiry Type Sent : 0

Info Type Received       : 5
Solicit Info Type Received : 2
Unsolicit Info Type Received : 3
Reload Type Received     : 0
Recovery Type Received   : 0
Gateway Info Type Received : 4
Enquiry Type Received    : 0
Solicit Enquiry Type Received : 0
Unsolicit Enquiry Type Received : 0
```

To verify the number of packets sent or received for each type in the standby instance, run the following command:

```
Device# show platform software rif-mgr chassis standby R0 lmp-statistics
LMP Statistics
```

```

Info Type Sent : 6
Solicit Info Type Sent : 0
Unsolicit Info Type Sent : 6
Reload Type Sent : 0
Recovery Type Sent : 0
Gateway Info Type Sent : 4
Enquiry Type Sent : 0
Solicit Enquiry Type Sent : 0
Unsolicit Enquiry Type Sent : 0

Info Type Received : 5
Solicit Info Type Received : 3
Unsolicit Info Type Received : 2
Reload Type Received : 0
Recovery Type Received : 1
Gateway Info Type Received : 0
Enquiry Type Received : 0
Solicit Enquiry Type Received : 0
Unsolicit Enquiry Type Received : 0

```

Redundancy management interfaces

A redundancy management interface is a network interface that

- acts as a secondary link between active and standby wireless controllers,
- enables resource health information exchange (such as gateway reachability), and
- assists in the detection of dual-active controller conditions to maintain high availability.

The RMI might trigger a switchover based on the gateway status of the active controller.

Subdefinitions:

- **Active controller:** Uses the management IP as the primary address and RMI as the secondary IPv4 address on the management VLAN. RMI configuration is automatic.
Analogy: Like a city's current mayor who's in charge, but always ready to hand over leadership if needed.
- **Standby controller:** Has the RMI IP as the primary address; upon switchover, roles and addresses are swapped.
Analogy: Like a vice-mayor who takes the mayor's seat (with all responsibilities and keys) when the mayor is away.
- **RP (Redundancy Port):** The main dedicated physical link for state and configuration synchronization between active and standby controllers; loss of both RP and RMI links results in high availability (HA) failures.
(Analogy: Like the main road connecting two city offices, critical for daily business and coordination).
- **WMI (Wireless Management Interface):** The main management interface for controller operations and communications; shares its subnet with the RMI and may serve as a source address for certain types of traffic such as AAA packets.
Analogy: Like a city's public headquarters, used for both regular operations and official correspondence.
- **RMI (Redundancy Management Interface):** A dedicated network interface that serves as a secondary communication path between controllers for exchanging resource health information, detecting dual-active

conditions, and monitoring gateway reachability; shares the same subnet as the Wireless Management Interface (WMI).

Analogy: Like an emergency side road connecting two city halls, used for urgent official communication.

- **HA (High Availability):** A deployment setup where controllers operate as an active-standby pair to ensure uninterrupted wireless services; relies on RP and RMI links for failover and role management.
- **ARP table:** A database on a network device such as a switch that maintains mappings between IP addresses and MAC addresses; determines how to forward traffic within the local network.

(Analogy: Like a city map book held by delivery drivers in neighboring towns, showing them which mayor is at which city hall, so the right deliveries go to the right address.)

- **GARP (Gratuitous ARP):** A type of ARP (Address Resolution Protocol) packet broadcast by a controller, typically after a switchover, to update the ARP tables in connected network switches with the correct IP-to-MAC address mappings.

(Analogy: Like sending out an urgent memo to all nearby towns, telling them who the new mayor is so they update their city maps immediately after a leadership change.)

- **ARP cache timeout:** The duration for which an entry in the ARP table is considered valid before it must be refreshed; reducing this value helps the network recover quickly after role or address changes in the controllers.

(Analogy: Like setting a regular schedule for when all delivery drivers check for updates to the city map books, ensuring they quickly recognize changes in city leadership or addresses.)

- **SGACL (Security Group Access Control List):** A policy configuration that determines which types of network traffic (e.g., ICMP, ARP) are permitted between specific interfaces or devices, such as the RMI addresses of controllers.

(Analogy: Like special city rules that decide which types of emergency vehicles are allowed to travel the emergency road between city halls.)

- **ICMP (Internet Control Message Protocol):** A network protocol used for sending error messages and operational information between network devices; essential for controllers to monitor each other's status over the RMI.

(Analogy: Like sending quick bike couriers to report on the status of the roads or to alert if there's trouble between cities.)

- **AAA (Authentication, Authorization, and Accounting):** A framework for controlling user access to network resources and tracking user activity; controllers may send AAA-related packets from either the WMI IP or the RMI IP, so the AAA server must recognize both as valid.

(Analogy: Like having a security checkpoint that logs who enters or exits city buildings, whether they come from Main Street (WMI) or the emergency road (RMI).)

- **SGT (Security Group Tag):** An identification label assigned to network devices for policy enforcement with Cisco TrustSec; mapping is applied for both RMI and WMI addresses when device SGTs are used.

Analogy: Like giving every city vehicle a badge, so officials can enforce special rules for each group whether they use Main Street or the emergency road.

- **Cisco TrustSec:** A security architecture that uses SGTs to enforce network segmentation and access control policies; not supported on the RMI interface.

Analogy: Like a city's security zone system—effective on main city roads, but not supported on the emergency side road.

Limitations for RMI

- Cisco TrustSec is not supported on the RMI.
- From the Cisco IOS XE 17.14.1 release onwards, RP-only SSO is not supported for CW9800H1, CW9800H2, or CW9800M Wireless Controllers. These controllers support RP+RMI SSO deployment only. In contrast, Cisco Catalyst 9800 Wireless Controllers support both RP-only SSO and RP+RMI SSO.

Best practices for RMI

- Ensure that the SGACL is defined appropriately to allow ICMP and ARP traffic between the active and standby RMI addresses when device SGT is used, since the IP-SGT mapping applies to both the RMI and WMI addresses.
- Configure the AAA server to recognize both the WMI IP and the RMI IP as valid source addresses, because AAA packets from the controller may originate from either.
- RMI-based "RP" High Availability is mandatory in the Cisco Catalyst CW9800H1 Wireless Controller, Cisco Catalyst CW9800H2 Wireless Controller and Cisco Catalyst CW9800M Wireless Controller.

Important: gateway monitoring interval and detection time

- When gateway reachability is enabled, both the active and standby controllers check gateway status through the RMI interface.
- It takes approximately the configured gateway monitoring interval to detect when a controller has lost gateway reachability.
- The default gateway monitoring interval is eight seconds, so the minimum detection time is about eight seconds unless you configure a different value.

Recommendation: set ARP cache timeout to ensure fast HA recovery:

When both the RP and RMI links are down, the HA setup breaks and both controllers become active, causing an IP conflict in the network. The HA setup is restored once the RP link is up. Depending on the external switch, its ARP table may correctly update to the new active controller or remain stale if the switch ignores GARP packets, potentially prolonging the conflict.

- We recommend setting the ARP cache timeout to a low value. This practice enables faster recovery from multiple fault scenarios.

You should choose a timeout value that does not negatively affect network traffic. For example, 30 minutes is a suitable interval.

Analogy

Think of redundancy management interfaces (RMI) as an emergency backup road that connects two cities (the active and standby controllers). The main highway (the redundancy port, RP) handles most of the day-to-day

traffic and coordination. If the main highway is blocked or damaged, the emergency road (RMI) ensures that vital information—such as each city's health, status, and road conditions—can still be exchanged quickly.

Just as emergency vehicles and communication must be allowed to travel the backup road (analogous to allowing ICMP and ARP traffic via SGACL), the RMI helps both cities detect if both try to take charge at the same time (dual-active condition) and exchange important information about the area's main gateway (gateway status). If both the highway and the emergency road are inaccessible, each city might assume it's in charge, resulting in confusion (an IP conflict). The ARP table is like city maps in other towns (switches) that may not immediately recognize which city is currently in charge after both reconnect—setting these maps to update frequently (low ARP cache timeout) allows for faster recovery from major outages.

Controller operations with Redundant Management Interface

A controller operation with redundant management interface is a high-availability configuration that

- assigns the management IP address as the primary address on the active controller
- uses the secondary IPv4 address on the management VLAN as the RMI IP address for the active controller, and
- configures the RMI IP address as the primary IP address on the standby controller.

IP address assignment in controller operations

The active controller assigns IP addresses as follows:

- The primary address is the management IP address.
- The secondary IPv4 address on the management VLAN is the RMI (Redundant Management Interface) IP address for the active controller.

The standby controller manages IP addresses in a high-availability setup as follows:

- It does not have the wireless management IP configured.
- The RMI IP address is configured as the primary IP address on the standby controller.
- When the standby controller becomes active, the management IP address becomes the primary IP address, and the RMI IP address becomes the secondary IP address.
- If the interface on the active controller is administratively down, the same state is reflected on the standby controller.



Note Do not configure the secondary IPv4 address explicitly. RMI automatically configures a single secondary IPv4 address under the RMI.

Dual stack configurations on management vlans with RMI

A dual stack configuration is a network interface setup that

- allows both IPv4 and IPv6 addresses to be configured on the wireless management interface,

- permits monitoring only of the gateway that matches the configured RMI address family (IPv4 or IPv6), and
- restricts the visibility of the alternate family's management address on the standby controller.

Expanded explanation

- Dual stack refers to the fact that the wireless management interface can be configured with IPv4 and IPv6 addresses. If an RMI IPv4 address is configured along with an IPv4 management IP address, you can additionally configure an IPv6 management address on the wireless management interface. This IPv6 management IP address will not be visible on the standby controller.
- If an RMI IPv6 address is configured along with an IPv6 management IP address, you can additionally configure an IPv4 management address on the wireless management interface. This IPv4 management IP address will not be visible on the standby controller.
- Therefore, you can monitor only the IPv6 gateway when the RMI IPv6 address is configured, or only the IPv4 gateway when the RMI IPv4 address is configured.



Note The RMI feature supports the RMI IPv4 or IPv6 addresses.

RMI-based high-availability pairings

A RMI-based high-availability pair is a controller deployment configuration that

- uses Remote Machine Interface (RMI) to synchronize two controllers,
- provides redundancy by designating active and standby roles, and
- ensures failover and persistent state during controller reloads or outages.

RMI-based high-availability pairing scenarios and device support

You should consider RMI-based high-availability pairs in the following scenarios:

- Fresh installation: Configure high availability during the initial setup of controllers.
- Already paired controllers: Adjust or reconfigure pairing for controllers that are already part of a high-availability pair.
- Upgrade scenario: Maintain or update the pair relationship during software or hardware upgrades.
- Downgrade scenario: Ensure pairing remains stable and functional during downgrades.

Dynamic high-availability (HA) pairing requires both the active and standby controllers to reload. In practice, on the Cisco Catalyst 9800-L, 9800-40, and 9800-80 Wireless Controllers, dynamic pairing occurs when one controller reloads and becomes the standby member of the pair.



Note Unique chassis numbers must be configured for each controller before forming an HA pair, as these numbers identify the controllers within the pair.

HA pairing without previous configuration

A high-availability (HA) pairing without previous configuration is a deployment scenario for wireless controllers that

- initiates the HA setup on devices without existing ROMMON variables for RP (Route Processor) IP addresses
- allows selection between the soon-to-be-deprecated privileged EXEC mode RP-based commands and the newer RMI IP-based mechanisms, and
- derives RP IPs from RMI IPs after forming the HA pair, with restrictions on method transitions.

Command usage and method selection

When HA pairing is performed for the first time (without previous setup), devices do not have ROMMON variables for RP IP addresses.

After RMI-based HA pairing on a brand-new system:

- RP IPs are derived from RMI IPs and used in HA pairing.
- Privileged EXEC mode RP-based CLIs method of clearing and forming an HA pair is not allowed.
- To view the ROMMON variables, use the **show romvars** command.



Caution Privileged EXEC mode RP-based commands are deprecated and will be blocked after choosing the RMI-based HA pairing.

Method selection considerations:

- You can still choose from the existing privileged EXEC mode RP-based commands or the RMI IP-based mechanisms. However, the privileged EXEC mode RP-based commands are deprecated.
- If you use Cisco Catalyst Center, you can choose the privileged EXEC mode RP-based CLI mechanism till the Cisco Catalyst Center migrates to support the RMI.
- If you choose privileged EXEC RP-based CLI mechanism, the RP IPs are configured the same way as in the 16.12 release.

Use the RMI IP-based mechanism for fresh installations, even though both RP-based and RMI methods may initially be available.

Software version requirements:

- The RMI migration is supported from Cisco Catalyst Center, 2.3.3.x release version.
- RMI-based High Availability requires Cisco IOS XE release version 17.3 or above.

Cisco Catalyst Center interoperability:

- If you use Cisco Catalyst Center, you can choose the privileged EXEC mode RP-based CLI mechanism till the Cisco Catalyst Center migrates to support the RMI.
- The RMI migration is supported from Cisco Catalyst Center, 2.3.3.x release version.

Negative cases where RMI migration fails include:

- Devices are not reachable.
- Non-Cisco Catalyst 9800 Series Wireless Controllers are in use.
- Controller is running Cisco IOS XE 17.3 or below
- High Availability is not configured.
- High Availability RMI is already configured.
- Attempting upgrade to an already failed High Availability paired controller.

**Caution**

The controller GUI prohibits applying RMI migration configuration to High Availability failed devices.

Security system installation analogy

Establishing HA pairing without previous configuration is like setting up a new security system in a building that has never had one before.

At first, you have the choice between using an old key-based system (privileged EXEC mode RP-based commands), which will soon be phased out, or installing a new state-of-the-art digital access system (RMI IP-based mechanism).

Once you install the modern digital system and program it to generate security codes based on the latest technology (RP IPs derived from RMI IPs), the use of physical keys becomes unavailable.

If you later decide to upgrade, you can only add new features to the digital system; you cannot go back to using the old keys because the doors no longer have compatible locks.

This analogy illustrates how choosing the RMI-based approach in HA pairing establishes a new baseline that does not allow reverting to the older method.

Paired controllers

A paired controller is a high availability (HA) infrastructure configuration that

- links two controllers to operate jointly for redundancy and failover,
- allows seamless migration from traditional EXEC mode RP-based commands to RMI-based HA pairing, and
- ensures controller identity and connectivity are maintained even when core pairing mechanisms are updated or reloaded.

Expanded explanation

When controllers are already in an HA pair, they continue to use existing EXEC mode RP-based commands unless Remote Management Interface (RMI) is enabled. Enabling RMI migrates the system to use RMI-derived HA pairing, overwriting any existing RP IPs with those derived from the RMI configuration. The HA pair remains stable immediately after this change, but the controllers only adopt the new IPs following their next reload.

RMI requires controllers to be reloaded for the changes to take effect. Once both controllers restart, they reestablish the HA pair using the new RMI-derived RP IPs. After pairing through RMI, EXEC mode RP-based commands are blocked, preventing configuration conflicts.

Examples

- Two controllers configured as a high availability pair, where enabling RMI changes the way their active-standby relationship is managed and what IPs are used for internal communication.
- An active and standby controller pair that continues functioning during migration from legacy RP-IP pairing to RMI, without disruption until reload.

Counter-examples

- Two standalone controllers operating independently without HA pairing cannot be considered paired controllers.
- A controller pair where RMI is never configured and all management remains through EXEC mode RP-based commands does not benefit from RMI-derived features.

Analogy

Imagine a paired controller setup as two co-pilots flying an airplane together (the airplane represents your network environment). Traditionally, they use walkie-talkies (EXEC mode RP-based commands) to coordinate their flying activities. If you upgrade their communication system to headsets (RMI-based pairing), the co-pilots continue flying the plane using walkie-talkies until they both put on the new headsets (after a "reload" or restart).

From that point onward, all their coordination happens via the more reliable and advanced headsets. The co-pilots' ability to work together—their partnership—remains unbroken throughout; it is only how they communicate and identify each other's messages that changes, and only becomes effective after both are using the new headsets.

Upgrade from Cisco IOS XE 16.1.x to a later release

When upgrading a system, you have these options:

- Migrate while retaining the existing RP IP configuration: In this scenario, the current RP IP configuration remains unchanged, and future modifications will utilize EXEC mode RP-based commands.
- Migrate after clearing the HA configuration: Here, you have the choice to use either the traditional EXEC mode RP-based commands or adopt the new RMI-based RP configuration. If the previous configuration is preserved, RMI will update the RP IPs with those derived from the RMI IPs.

Downgrade scenario

Important The downgrade scenario given below is not applicable for Cisco IOS XE 17.1.x.

In a downgrade scenario, only EXEC mode RP-based commands are available. The downgrade process may follow one of these paths:

- If the upgraded system used the RMI-based RP configuration.
- If the upgraded system continued to use the EXEC mode RP-based commands.

In the above cases, the downgraded system uses the EXEC mode RP-based commands to modify the configuration. However, the downgraded system will continue to use the new derived RP IPs.

In both of these cases, the system will revert to EXEC mode RP-based commands for configuration alterations, yet will still utilize the newly derived RP IPs.



Note When you downgrade the Cisco Catalyst 9800 Series Wireless Controller to any version below Cisco IOS XE 17.1 and if the mDNS gateway is enabled on the WLAN/RLAN/GLAN interfaces, the mdns-sd-interface gateway goes down after the downgrade.

To enable the mDNS gateway on the WLAN/RLAN/GLAN interfaces in Cisco IOS XE 16.12 and earlier versions, use these commands:

wlan test 1 test

mdns-sd gateway

To enable the mDNS gateway on the WLAN/RLAN/GLAN interfaces from version Cisco IOS XE 17.1 onwards, use these command:

mdns-sd-interface gateway

Configure redundancy management interface (GUI)

Enable redundancy management for Cisco Catalyst 9800 Series Wireless Controllers using the graphical user interface (GUI).

Use this task to configure the redundancy management interface (RMI) and set up either RMI+RP or RP redundancy pairing on Cisco Catalyst 9800 Series Wireless Controllers. Configuring redundancy improves system availability and failover capabilities.

Before you begin

Ensure that Wireless Management Interface (WMI) is available before configuring RMI + RP using the GUI.

Follow these steps to configure redundancy management interface using GUI:

Procedure

Step 1 In the **Administration > Device > Redundancy** window, perform the following:

- a. Set the **Redundancy Configuration** toggle button to **Enabled** to activate redundancy configuration.
- b. In the **Redundancy Pairing Type** field, select **RMI+RP** to perform RMI+RP redundancy pairing as follows:
 - In the **RMI IP for Chassis 1** field, enter the RMI IP address for chassis 1.
 - In the **RMI IP for Chassis 2** field, enter the RMI IP address for chassis 2.
 - From the **HA Interface** drop-down list, choose one of the HA interfaces.
You can select the HA interface only for Cisco Catalyst 9800 Series Wireless Controllers.
 - Set the **Management Gateway Failover** toggle button to **Enabled** to activate management gateway failover.
 - In the **Gateway Failure Interval** field, enter an appropriate value. The valid range is 6–12 seconds (default 8 seconds).
- c. In the **Redundancy Pairing Type** field, select **RP** to perform RP redundancy pairing as follows:
 - In the **Local IP** field, enter an IP address for the local chassis.
 - In the **Netmask** field, enter the subnet mask assigned to all wireless clients.
 - From the **HA Interface** drop-down list, choose one of the HA interfaces.
You can select the HA interface only for Cisco Catalyst 9800 Series Wireless Controllers.
 - In the **Remote IP** field, enter an IP address for the remote chassis.
- d. In the **Keep Alive Timer** field, enter an appropriate timer value (1–10 ×100 milliseconds).
- e. In the **Keep Alive Retries** field, enter an appropriate retry value (3–10 seconds).
- f. In the **Active Chassis Priority** field, enter a value.

Step 2 Click **Apply** and reload controllers.

The redundancy management interface is configured, and redundancy pairing is established based on your chosen method. The controller is now set up for improved high availability and failover.

Configure redundancy management interface (CLI)

Configure a Redundancy Management Interface (RMI) on Cisco Catalyst 9800 controllers using CLI commands to support high availability (HA) between two devices.

Use this task when you want to set up high availability and redundancy between two Catalyst 9800 series controllers. The RMI coordinates HA communication and failover, ensuring service continuity in case of device failure.

Before you begin

- Ensure both controllers are cabled and powered on.
- Verify you have administrator access to both devices via CLI.
- Gather the following information:

- Chassis number (1 or 2 for each controller)
- Desired chassis priority for HA (if overriding default)
- A dedicated GigabitEthernet interface for HA communication (required for 9800-CL controllers)
- Management VLAN and corresponding IP addresses for each chassis

Procedure

Step 1 (Optional) Configure the priority of the specified device.

Example:

```
Device# chassis chassis-num priority chassis-priority
```

Example:

```
Device# chassis 1 priority 1
```

From Cisco IOS XE Gibraltar 16.12.x onwards, device reload is not required for the chassis priority to become effective.

- *chassis-num*—Enter the chassis number. The range is from 1 to 2.
- *chassis-priority*—Enter the chassis priority. The range is from 1 to 2. The default value is 1.

When both the devices boot up at the same time, the device with higher priority becomes active, and the other one becomes standby. If both the devices are configured with the same priority value, the one with the smaller MAC address acts as active and its peer acts as standby.

Step 2 Create an HA interface for your controller.

Example:

```
Device# chassis redundancy ha-interface GigabitEthernet interface-number
```

Example:

```
Device# chassis redundancy ha-interface  
GigabitEthernet 3
```

- *interface-number*: GigabitEthernet interface number. The range is from 1 to 32.

This step is applicable only for Cisco Catalyst 9800-CL Series Wireless Controllers. The chosen interface is used as the dedicated interface for HA communication between the 2 controllers.

Step 3 Enter global configuration mode.

Example:

```
Device# configure terminal
```

Step 4 Configure Redundancy Management Interface.

Example:

```
Device(config)# redun-management interface vlan vlan-interface-number chassis chassis-number  
address ip-address chassis chassis-number address ip-address
```

Example:

```
Device(config)# redun-management interface
Vlan 200 chassis 1 address 9.10.90.147
chassis 2 address 9.10.90.149
```

- *vlan-interface-number*: VLAN interface number. The valid range is from 1 to 4094.

Here, the *vlan-interface-number* is the same VLAN as the Management VLAN. That is, both must be on the same subnet.

- *chassis-number*: Chassis number. The valid range is from 1 to 2.
- *ip-address*: Redundancy Management Interface IP address.

Each controller must have a unique chassis number for RMI to form the HA pair. The chassis number can be observed as SWITCH_NUMBER in the output of **show romvar** command. Modification of SWITCH_NUMBER is currently not available through the web UI.

To disable the HA pair, use the **no redun-management interface vlan chassis** command.

Step 5 Return to privileged EXEC mode.

Example:

```
Device(config)# end
```

Step 6 Save the configuration.

Example:

```
Device# write memory
```

Step 7 Reload the controllers.

Example:

```
Device# reload
```

When the RMI configuration is done, you must reload the controllers for the configuration to take effect.

For Cisco Catalyst 9800-CL Wireless Controller VM, both the active and standby controllers reload automatically. In the case of hardware platforms, you should reload the active controller manually, as only standby the controller reloads automatically.

The redundancy management interface is configured. After reload, an HA pair is established between the two controllers, enabling redundancy and failover support.

