

Monitor Catalyst 9800 WLC via SNMP with OIDs

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[SNMP vs Telemetry](#)

[Configure SNMP on WLC](#)

[Through Web Interface](#)

[Through Command Line](#)

[Object Names and Object IDs \(OIDs\)](#)

[What Are Object Names and OIDs?](#)

[MIBs And List of All Object Names And IDs on Cisco WLCs](#)

[Use OIDs to Monitor the State of WLC](#)

[Monitor via Snmpwalk](#)

[Monitor via Python3 And pysnmp Library](#)

[Integration With Third Party Software \(Grafana+Prometheus/PRTG Network Monitor/SolarWinds\)](#)

[Integration With CUCM](#)

[Table of Most Commonly Monitored OIDs](#)

[Monitor Standby WLC in HA](#)

[Monitor Standby WLC Directly](#)

[Monitor Standby WLC Through Active WLC](#)

Introduction

This document describes how to configure Simple Network Management Protocol (SNMP) to monitor Cisco 9800 Wireless LAN Controller (WLC).

Prerequisites

Requirements

- Basic knowledge of 9800 WLC and SNMP protocol
- SNMP server/tool

Components Used

All tests were performed on MacOS 10.14 and a 9800-CL WLC with image version 17.5.1. Some of the OIDs mentioned in this article do not exist on older image versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

SNMP vs Telemetry

Older AireOS WLCs rely on SNMP as the main protocol for monitor. Majority of the relevant information like client count, number of joined access points, processor and memory usage can be obtained via SNMP query from the tool that monitors, to the WLC.

With 9800 WLC, the focus has been put on telemetry. Telemetry works in a "push" model where WLC sends out relevant information to the server without the need to be queried. Catalyst 9800 still offers SNMP for legacy purposes. Some information can be exclusive to telemetry and some of the OIDs previously available on AireOS are not yet available on 9800.


Configure SNMP on WLC

From Cisco IOS XE Bengaluru 17.6.1, Ethernet Service Port (Management Interface VRF/GigabitEthernet 0) is supported in Cisco Catalyst 9800 Series Wireless Controller.

Prior to this release the Catalyst 9800 WLC could only be monitored with SNMP via its Wireless Management Interface or via Redundancy Management Interface (in case of a standby WLC in HA cluster on versions 17.5.1 and higher).

Through Web Interface

SNMPv2c is a community-based version of SNMP and all communication between the devices is in clear text. SNMPv3 is the most secure version which offers message integrity checks, authentication and encryption of the packets. SNMPv1 is extremely outdated, but still exists to provide legacy software compatibility. It is not mentioned in this article.

 **Important:** SNMPv2c is enabled by default with community “private” with read+write privileges and community “public” with read-only privileges. It is recommended to remove them and create a new community with a different name.

Log into the web interface of the 9800 WLC. Under **Administration** > **Management** > **SNMP** ensure that SNMP is globally enabled. Under **Community Strings** all currently configured communities and their permission level are displayed:

[Administration](#) > [Management](#) > [SNMP](#)

SNMP Mode

ENABLED 

[General](#) [SNMP Views](#) [Community Strings](#) [V3 User Groups](#) [V3 Users](#) [Hosts](#)

[+ Add](#)

[X Delete](#)

	Community Name	Access Mode
<input type="checkbox"/>	private	Read/Write
<input type="checkbox"/>	public	Read Only

1 10 items per page 1 - 2 of 2 items

Before SNMP V3 user is created, it is required to define an SNMP V3 group. To create a group of users

with read+write permission, set the **Read View** and **Write View** to **v1default**. Read only group needs to have **Write View** empty

V3 User Groups ✕

Group Name*	<input type="text" value="readwritegroup"/>
Security Level*	<input type="text" value="Auth"/>
Read View	<input type="text" value="v1default"/> i
Write View	<input type="text" value="v1default"/>

Under the SNMP V3 Users tab, you can see all the configured users, their privileges and protocols used for authentication and encryption. Button **New** allows a creation of a new user.

There are 3 security modes available:

1. AuthPriv = Messages are authenticated and encrypted
2. AuthNoPriv = Messages are authenticated but no encrypted
3. NoAuthNoPriv = No security applied to messages

Select SHA as authentication protocol and at least AES-128 as privacy protocol is recommended.

V3 Users ✕

User Name*	<input type="text" value="snmpAdmin"/>
Group Name*	<input type="text" value="ReadWriteGroup (..."/> +
Security Mode*	<input type="text" value="AuthPriv"/> i
Authentication Protocol	<input type="text" value="SHA"/>
Authentication Password*	<input type="password" value="....."/>
Privacy Protocol	<input type="text" value="AES128"/>
Privacy Password*	<input type="password" value="....."/>

Through Command Line


SNMP can also be configured through Command Line Interface (CLI). CLI offers additional configuration parameters, like the ability to assign an access list to the v2 community or v3 user.

Example configuration of v2 read+write community, v3 read+write group and v3 user that belongs to this group:

```
snmp-server manager
snmp-server community <community-name> RW <access-list-number>
snmp-server community <community-name> RO <access-list-number>
snmp-server group <v3-group-name> v3 auth write v1default
snmp-server user <v3-user-name> <v3-group-name> v3 auth sha <auth-password> priv aes 128 <encryption-pa
```

Example of an access list that only allows the device on IP address 192.168.10.10 to query the WLC v2 community named "ReadWriteCommunity":

```
ip access-list standard 50
 10 permit 192.168.10.10
 20 deny any
snmp-server manager
snmp-server community ReadWriteCommunity RW 50
```

 **Note:** At the time of the write of this document, only standard ACLs are supported. Extended ACLs can be assigned, but they do not work.

Object Names and Object IDs (OIDs)

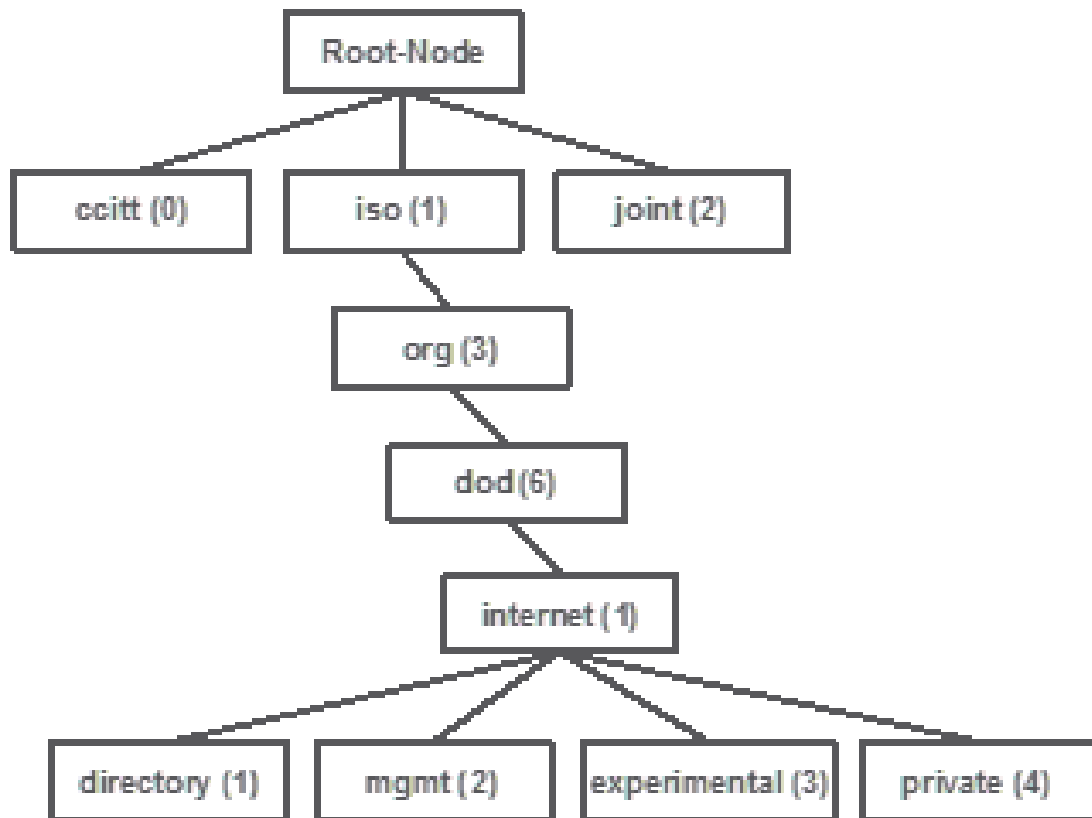
What Are Object Names and OIDs?

Object IDs, or OIDs for short, are unique identifiers that represent a certain variable or object. For example, the current processor usage is considered variable which values can be retrieved with the call upon their object ID. Each OID is unique and no two can be the same across the world, quite similar to a MAC address.

These identifiers follow a tree hierarchy, and each OID can be tracked down back to its root. Each vendor has its own branch with a common root.

An analogy could be a home address, where the root would be the country or state, followed by a city zip code, the street, and finally the home number.

The numbers followed by a dot represent each step it takes to get to a certain point in that tree or branch.

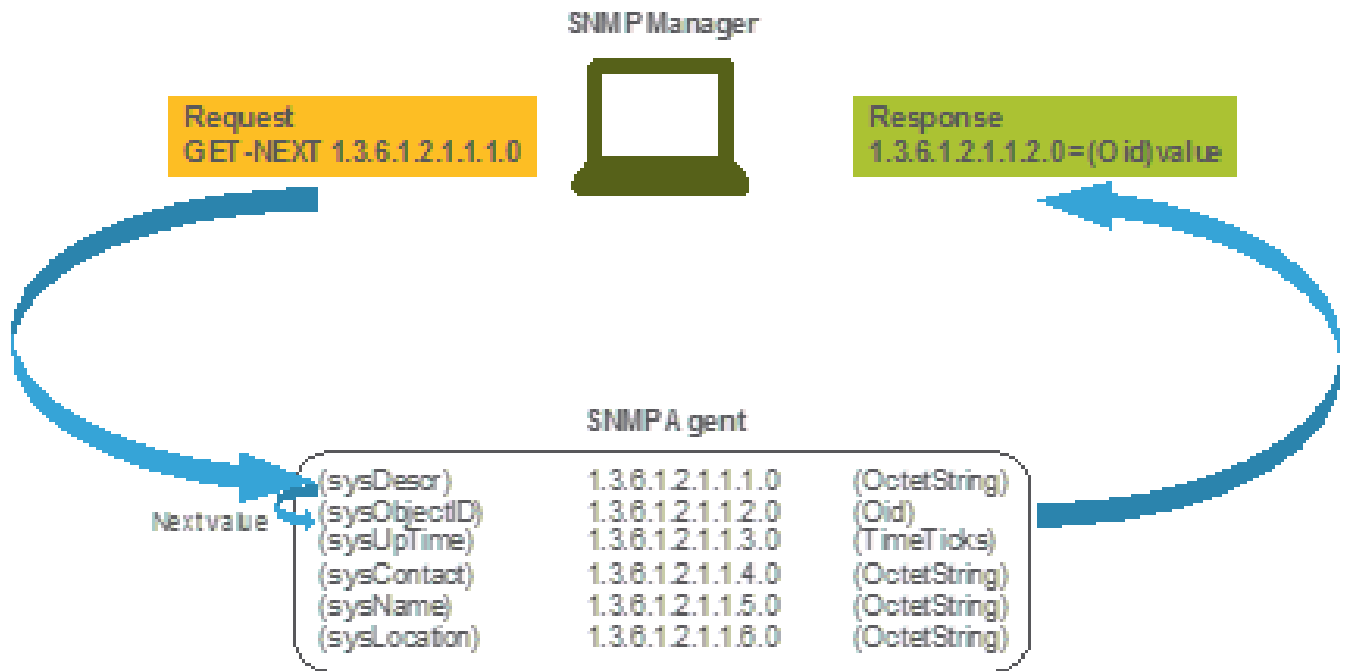


All these values are stored in a Management Information Base, or MIB for short, in each network device. Each identifier has name and a definition (range of possible values, type..).

To load a MIB on your SNMP monitor tool is not required in order to use SNMP and to query a device.

As long as a valid OID is known, the device responds with the value that is stored in the variable that the OID represents. However, if you load the MIB on your query tool, it provides the benefit of translate the object names to their IDs and allows to know their description.

In this example, the SNMP tool queries the SNMP agent of a device for its system description with the use of the OID 1.3.6.1.2.1.1.1.0.



MIBs And List of All Object Names And IDs on Cisco WLCs

Cisco offers Management Information Base (MIBs) for 9800 WLCs. It is not easily readable, but the MIB contains all available object names and their description.

All 9800 models (9800-80, 9800-40, 9800-L, 9800-CL, EWC) use the same MIB which can be downloaded here: <https://software.cisco.com/download/home/286322605/type/280775088/release/>.

The most up-to date one is the one with the most recent date, not the one with the higher code version name.

Downloaded archive file contains multiple .my text files that can either be imported into any third-party SNMP server or simply opened with a text editor. In order to find the OID of a specific object name, you first need to locate the exact file that contains it.

For example, all objects related to monitor the device physical state (like CPU and memory) are located inside a MIB called CISCO-PROCESS-MIB.my.

Here, "cpmCPUMemoryUsed" is the object name which is used to provide the amount of memory used by the WLC in bytes. MIB files all follow similar syntax. Information about the used memory object looks like this:

```
cpmCPUMemoryUsed OBJECT-TYPE
    SYNTAX          Gauge32
    UNITS           "kilo-bytes"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The overall CPU wide system memory which is currently
        under use."
    ::= { cpmCPUTotalEntry 12 }
```

Most of the third-party software to monitor relies on OIDs, and not object names. Translation between object name and object ID can be done with the usage of [Cisco SNMP object navigator tool](#).

Enter the object name into the search bar. Output provides the OID and a short description. Additionally, same tool can be used to find the object name of the provided OID.

Tools & Resources

SNMP Object Navigator

HOME | SUPPORT | TOOLS & RESOURCES | **SNMP Object Navigator**

TRANSLATE/BROWSE | SEARCH | DOWNLOAD MIBS | MIB SUPPORT - SW | Help | Feedback

Translate | [Browse The Object Tree](#) | **Related Tools** | [Support Case Manager](#) | [Cisco Community](#) | [MIB Locator](#)

Translate OID into object name or object name into OID to receive object details

Enter OID or object name: examples -
OID: 1.3.6.1.4.1.9.9.27
Object Name: ifIndex

Object Information

Specific Object Information	
Object	cpmCPUMemoryUsed
OID	1.3.6.1.4.1.9.9.109.1.1.1.1.12
Type	Gauge32
Permission	read-only
Status	current
Units	kilo-bytes
MIB	CISCO-PROCESS-MIB ; - View Supporting Images
Description	"The overall CPU wide system memory which is currently under use."

OID Tree

You are currently viewing your object with levels of hierarchy above your object.

[.iso\(1\)](#), [.org\(3\)](#), [.dod\(6\)](#), [.internet\(1\)](#), [.private\(4\)](#), [.enterprises\(1\)](#), [.cisco\(9\)](#), [.ciscoMgmt\(9\)](#), [.ciscoProcessMIB\(109\)](#), [.ciscoProcessMIBObjects\(1\)](#), [.cpmCPU\(1\)](#)

- |
- |-- [cpmCPUTotalTable\(1\)](#)
- |
- |-- [cpmCPUTotalEntry\(1\)](#)
- |
- |-- [cpmCPUTotalIndex\(1\)](#)
- |
- |-- [cpmCPUTotalPhysicalIndex\(2\)](#)
- |
- |-- [cpmCPUTotal5sec\(3\)](#)
- |
- |-- [cpmCPUTotal1min\(4\)](#)
- |
- |-- [cpmCPUTotal5min\(5\)](#)
- |
- |-- [cpmCPUTotal5secRev\(6\)](#)
- |
- |-- [cpmCPUTotal1minRev\(7\)](#)
- |
- |-- [cpmCPUTotal5minRev\(8\)](#)
- |
- |-- [cpmCPUMonInterval\(9\)](#)
- |
- |-- [cpmCPUTotalMonIntervalValue\(10\)](#)
- |
- |-- [cpmCPUInterruptMonIntervalValue\(11\)](#)
- |
- |-- [cpmCPUMemoryUsed\(12\)](#) [object Details](#)

Use OIDs to Monitor the State of WLC

After the acquisition the OID of the object that needs to be monitored, first SNMP query can be executed.

Examples in this chapter showcases how to acquire a WLC free memory (OID = 1.3.6.1.4.1.9.9.48.1.1.1.5) for SNMPv2 community private and SNMPv3 user snmpadmin with SHA Auth password **Cisco123#** and AES Privacy password set to **Cisco123#**. Controller management interface is located on 10.48.39.133.

Monitor via Snmpwalk

Snmpwalk is an SNMP application that uses SNMP GETNEXT requests to query a network entity for a tree of information. It is present by default on MacOS and most Linux distributions. For SNMPv2c, the command follows the syntax:

```
snmpwalk -v2c -c <community_name> <WLC_management_interface_ip> <OID>
```

Example:

```
<#root>
```

```
VAPEROVI:~ vaperovi$
```

```
snmpwalk -v2c -c private 10.48.39.133 1.3.6.1.4.1.9.9.109.1.1.1.12
```

```
SNMPv2-SMI::enterprises.9.9.109.1.1.1.12.2 =
```

```
3783236
```

```
<----- Free Memory in Bytes
```

If SNMPv3 is used, the command follows the syntax:

```
snmpwalk -v3 -l authPriv -u <username> -a [MD5|SHA] -A <auth_password> -x [AES|DES] -X <priv_password>
```

Select MD5/SHA and AES/DES based on how you created the SNMPv3 user on the controller.

Example:

```
<#root>
```

```
VAPEROVI:~ vaperovi$ snmpwalk -v3 -l authPriv -u snmpadmin -a SHA -A Cisco123# -x AES -X Cisco123# 10.48.39.133.1.3.6.1.4.1.9.9.109.1.1.1.12
```

```
3783236
```

```
<----- Free Memory in Bytes
```


#snmpwalk output still shows v2 even though v3 is used

Monitor via Python3 And pysnmp Library

Code snippets are written for Python 3.9 and utilize **pysnmp** module (pip install pysnmp) to make SNMP queries for memory utilization of Catalyst 9800-CL WLC. These examples use the same SNMPv2 community and SNMPv3 user created in one of the previous chapters. Simply replace the variable values and integrate the code within your own custom scripts.

SNMPv2 example:

```
from pysnmp.hlapi import *

communityName = 'private'
ipAddress = '10.48.39.133'
OID = '1.3.6.1.4.1.9.9.109.1.1.1.1.12'

for (errorIndication,
     errorStatus,
     errorIndex,
     varBinds) in nextCmd(SnmpEngine(),
                          CommunityData(communityName),
                          UdpTransportTarget((ipAddress, 161)),
                          ContextData(),
                          ObjectType(ObjectIdentity(OID)),
                          lexigraphicMode=False):

    if errorIndication:
        print(errorIndication)
    elif errorStatus:
        print('%s at %s' % (errorStatus.prettyPrint(),
                            errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))
    else:
        for varBind in varBinds:
            print(' = '.join([x.prettyPrint() for x in varBind]))
```

Output prints out:

```
SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.12.2 = 3783236
```

SNMPv3 example:

```
from pysnmp.hlapi import *

username = 'snmpadmin'
ipAddress = '10.48.39.133'
OID = '1.3.6.1.4.1.9.9.109.1.1.1.1.12'
authKey = 'Cisco123#'
privKey = 'Cisco123#'
```

```

for (errorIndication,
    errorStatus,
    errorIndex,
    varBinds) in nextCmd(SnmpEngine(),
                        UsmUserData(username, authKey, privKey,
                                    authProtocol=usmHMACSHAAuthProtocol,
                                    privProtocol=usmAesCfb128Protocol),
                        UdpTransportTarget((ipAddress, 161)),
                        ContextData(),
                        ObjectType(ObjectIdentity(OID)),
                        LexicographicMode=False):

    if errorIndication:
        print(errorIndication)
    elif errorStatus:
        print('%s at %s' % (errorStatus.prettyPrint(),
                            errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))
    else:
        for varBind in varBinds:
            print(' = '.join([x.prettyPrint() for x in varBind]))

```

Integration With Third Party Software (Grafana+Prometheus/PRTG Network Monitor/SolarWinds)

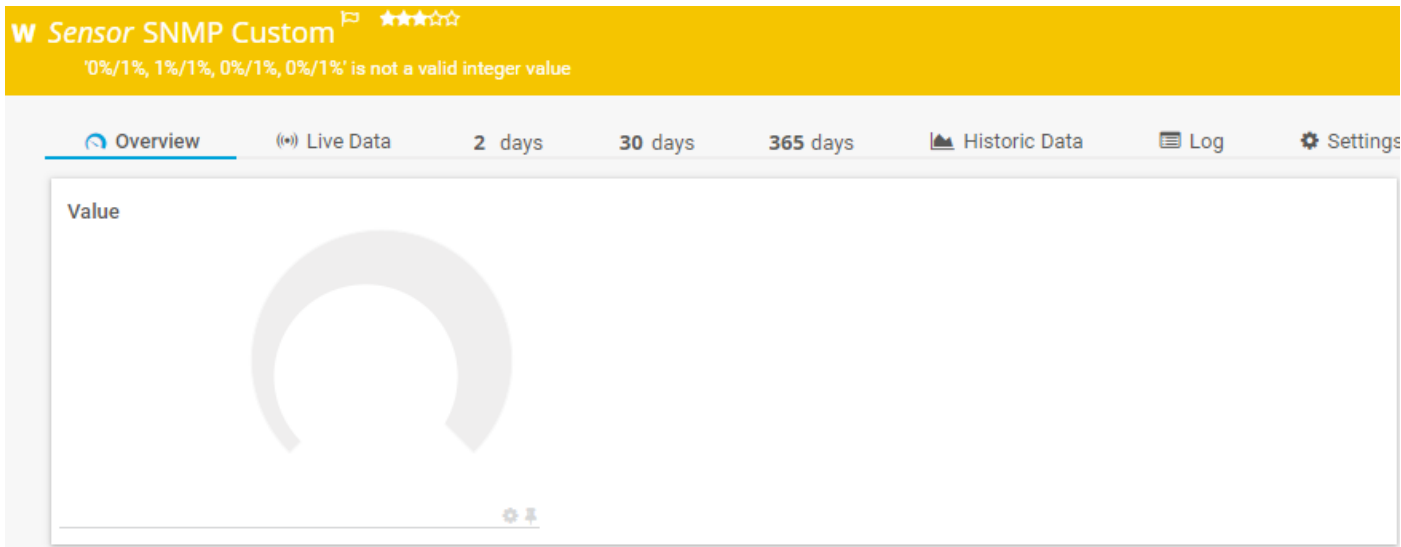
Cisco Prime Infrastructure offers the ability to easily monitor and configure multiple network devices, which includes wireless controllers.

Prime Infrastructure comes preloaded with all the OIDs and integration with WLC simply consists in the addition of the WLC credentials to Prime. With 9800 WLCs, Prime mostly relies on Telemetry to collect majority of the details from the WLC, while the small portion of the information is obtained through SNMP.

On the other hand, Cisco WLC can also be integrated with multiple 3rd party solutions for monitor, as long as the OIDs are known.

Programs like Grafana+Prometheus, PRTG Network monitor and SolarWinds server allow the MIBs or OIDs to be imported and values to be displayed in a user-friendly graph.

This integration can require some tweaks on SNMP server side. In this example, the PRTG monitor server is provided with the per core CPU utilization OID which returns the string “0%/1%, 1%/1%, 0%/1%, 0%/1%”. PRTG expects an integer value and raises an error.



Integration With CUCM

Cisco Unified Communications Manager (CUCM) has a Wireless Endpoint Tracking Feature that allows it to approximately track client location based on the AP that the client is connected to. For this feature to work, the CUCM has to pull information from the WLC via SNMP queries.

Important: Many CUCM releases are affected by the Cisco bug ID [CSCvv07486](#) - Unable to sync Access Points in WLC due to SNMP requests too large. This issue is triggered in situations where CUCM runs an affected release and WLC has more than 10 Access Points. Due to the incorrect way that CUCM queries large amount of OIDs in a single bulk request, the WLC either refuses to answer or answers with a **tooBig** response. The tooBig response is not always sent out immediately and can be delayed. SNMP debugs on WLC print "SNMP: Packet received via UDP from x.x.x.x on VlanXXSrParseV1SnmpMessage: packet is too large SrDoSnmp: ASN Parse Error".

Table of Most Commonly Monitored OIDs

The table includes some of the most common object names and their OIDs, with the consideration that MIBs present the data in non-user-friendly syntax:

Note: Command "show snmp mib | in <Object name>" can be used to verify if a certain object name is available on 9800 WLC.

Description	Object name	OID	Expected response
Overall CPU usage in % last 5 sec (equivalent to "show process cpu platform")	cpmCPUTotal5sec	1.3.6.1.4.1.9.9.109.1.1.1.1.3	INTEGER: 5
Overall CPU usage in % last 1	cpmCPUTotal1min	1.3.6.1.4.1.9.9.109.1.1.1.1.4	INTEGER: 5

min			
Overall CPU usage in % last 5 min	cpmCPUTotal5min	1.3.6.1.4.1.9.9.109.1.1.1.1.5	INTEGER: 5
CPU usage in % last 5 min, per core	cpmCore5min	.1.3.6.1.4.1.9.9.109.1.1.2.1.5.5 (.0 to .X depending on number of cores)	INTEGER: 5
Current used memory in bytes	cpmCPUMemoryUsed	1.3.6.1.4.1.9.9.109.1.1.1.1.12	INTEGER: 3783236
Current free memory in bytes	cpmCPUMemoryFree	1.3.6.1.4.1.9.9.109.1.1.1.1.13	INTEGER: 4263578
Lowest amount of free memory since last boot in bytes	cpmCPUMemoryLowest	1.3.6.1.4.1.9.9.109.1.1.1.1.15	INTEGER: 4251212
Last reload reason	whyReload	1.3.6.1.4.1.9.2.1.2	STRING: "reload"
Software image of all joined APs	bsnAPSoftwareVersion	1.3.6.1.4.1.14179.2.2.1.1.8	STRING: "17.5.1.12"
Model number of all joined APs	bsnAPModel	1.3.6.1.4.1.14179.2.2.1.1.16	STRING: "AIR-AP1840I-E-K9"
Number of clients	X	X	X
Number of joined APs	cLApGlobalAPConnectCount	1.3.6.1.4.1.9.9.513.1.3.35	Gauge32: 1
Power Supply Unit State	X	X	X
Fan State	X	X	X

Currently there are enhancement requests open to support OIDs of the total number of clients and number of joined access points:

Cisco bug ID [CSCvu26309](#) - SNMP OID for client count not present on 9800

State of the Power Supply Unit (PSU) and fan state are not supported at the time of writing of this article. Enhancement request has been opened:

Cisco bug ID [CSCwa23598](#) - 9800 WLC Enhancement / Support for PSU and fan state SNMP OID (1.3.6.1.4.1.9.9.13)

Monitor Standby WLC in HA


To monitor a standby WLC in High Availability cluster is only possible from [the release 17.5.1](#). Standby WLC can either be monitored directly through RMI or with the query of the active WLC.

Monitor Standby WLC Directly

Standby WLC can be monitored directly only if WLCs that run in RMI + RP HA type. It is done through the standby Redundancy Management Interface (RMI) IP address of the standby WLC.

In this scenario, only OIDs from **IF-MIB** are officially supported, which makes it possible only to monitor the state of all the interfaces on the standby WLC. Example output from 9800-CL WLC:

Description	Object name	OID	Expected response
Interface name	ifDescr	1.3.6.1.2.1.2.2.1.2	SNMPv2-SMI::mib-2.2.2.1.2.1 = GigabitEthernet1 SNMPv2-SMI::mib-2.2.2.1.2.2 = GigabitEthernet2 SNMPv2-SMI::mib-2.2.2.1.2.3 = GigabitEthernet3 SNMPv2-SMI::mib-2.2.2.1.2.4 = VoIP-Null0 SNMPv2-SMI::mib-2.2.2.1.2.5 = Null0 SNMPv2-SMI::mib-2.2.2.1.2.6 = Vlan1 SNMPv2-SMI::mib-2.2.2.1.2.7 = Vlan39
Interface operational state (1=up, 2=down)	ifOperStatus	1.3.6.1.2.1.2.2.1.8	SNMPv2-SMI::mib-2.2.2.1.8.1 = 2 SNMPv2-SMI::mib-2.2.2.1.8.2 = 2 SNMPv2-SMI::mib-2.2.2.1.8.3 = 1 SNMPv2-SMI::mib-2.2.2.1.8.4 = 2 SNMPv2-SMI::mib-2.2.2.1.8.5 = 2 SNMPv2-SMI::mib-2.2.2.1.8.6 = 1

 **Note:** It is expected that the standby WLC Gig 2 port (trunk port used for traffic switching) is in shutdown state. Once failover occurs, the Gig 2 port on the standby WLC comes up. Similar thing occurs with TenGigabit ports on 9800 physical appliances (9800-80, 9800-40 and 9800-CL).

Monitor Standby WLC Through Active WLC

Standby WLC state can also be monitored with the query to the active WLC. Only **CISCO-LWAPP-HA-MIB** and **CISCO-PROCESS-MIB** MIBs are officially supported. When active WLC in HA is queried, the first response represents the value of the active WLC, while the second response represents the value of the standby WLC.

Description	Object name	OID	Expected response
Overall CPU usage in % last 5 sec	cpmCPUTotal5sec	1.3.6.1.4.1.9.9.109.1.1.1.1.3	SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.3.5 = 3 SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.3.6 = 7
Overall CPU usage in % last 1 min	cpmCPUTotal1min	1.3.6.1.4.1.9.9.109.1.1.1.1.4	SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.4.5 = 8 SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.4.6 = 6
Overall CPU usage in % last 5 min	cpmCPUTotal5min	1.3.6.1.4.1.9.9.109.1.1.1.1.5	SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.5.5 = 10 SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.5.6 = 15
Current used memory in bytes	cpmCPUMemoryUsed	1.3.6.1.4.1.9.9.109.1.1.1.1.12	SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.12.5 = 4318980 SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.12.6 = 3950332
Current free memory in bytes	cpmCPUMemoryFree	1.3.6.1.4.1.9.9.109.1.1.1.1.13	SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.12.5 = 4318739 SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.12.6 = 3950738

<p>Lowest amount of free memory since last boot in bytes</p>	<p>cpmCPUMemoryLowest</p>	<p>3.6.1.4.1.9.9.109.1.1.1.1.15</p>	<p>SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.15.5 = 3763868 SNMPv2-SMI::enterprises.9.9.109.1.1.1.1.15.6 = 4132588</p>
<p>State of the standby WLC (1=up, 0=down)</p>	<p>cLHaPeerHotStandbyEvent</p>	<p>3.6.1.4.1.9.9.843.1.3.4</p>	<p>SNMPv2-SMI::enterprises.9.9.843.1.3.4.0 = 1</p>