

Template Development

Using example SNMP devices.



Lab instructions

<https://space.zabbix.com/s/85jog3Qapy2wGjX>

<https://tinyurl.com/zabbixsummit2026>



Who am I? Who are you? I'll go first

Let's meet

Michal Kudlacz, Technical Support Engineer, Zabbix



Monitoring and Integration Solutions

Search by name

Home / Product /

Propose Integration

- All Categories Official Templates Agents API Applications AWS Backups Business KPI Caching CMDB Clouds
- Containers CRM DevOps Databases ERP HA & Clusters Helpdesks Infrastructure IoT Java Logfiles Mail
- Message brokers Mobile Monitoring systems Network Notifications & Alerting Orchestration Operation Systems Printers
- Search Engines Security Services Servers Storage Telephony Virtual Machines Visualization Web

 Apache HTTP Server	 ((OTRS)) Community Edition	 3com	 3CX	 Acronis
 actidata	 Active Directory	 ActiveMQ	 Adaptec	 ADVA
 Airco	 AKCP	 Alcatel Lucent	 Alerta	 AlertOps
 All Quiet	 ALVAO	 Alvarion	 AMD	 Android

Specification & Research

Choosing what to monitor

"We want to monitor everything!"

snmpwalk: 32k lines

Finding metrics: MIB vs snmpwalk

A MIB definition is not evidence – device vendors implement subsets of their own MIBs

Defining alerting & thresholds

Key metrics such as Availability, Memory & CPU utilization, Interfaces are mostly sufficient

Convert questions into metrics and triggers

Question	Metric	Method
Is it reachable?	ICMP ping, loss, response time	simple check
Is the agent answering?	SNMP agent availability	internal
What is it?	sysDescr, serial, firmware	static, 1h
Has it rebooted?	sysUpTime	static, 1m
How much has it printed?	prtMarkerLifeCount	static, 5m
Does it need attention?	hrPrinterDetectedErrorState	static, 1m
Will a consumable run out?	Supplies level and capacity	discovery
Is the network port healthy?	ifOperStatus, octets, errors	discovery

MIB file = SNMP dictionary

What MIB gives you

**Name, numeric
OID path, data
type, allowed
values, units,
access mode,
table index and
description**

What device gives you

**Real values
returned by
snmpget /
snmpwalk for
implemented OIDs**

What Zabbix needs

**OID, value type,
preprocessing,
discovery**

MIBs - best practices

Always confirm MIB theory

Walk target device model / firmware

Prefer standard MIBs first

Use vendor MIBs for gaps such as sensors, stack, PoE, supplies

Use numeric OIDs

Safer approach

Raw MIB fragment

ifHighSpeed OBJECT-TYPE

SYNTAX Gauge32

UNITS "1,000,000 bits per second"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An estimate of the interface's current
bandwidth..."

::= { ifXEntry 15 }

The OLD tree – parent and child notation

MIBs often define objects relatively, not as full numeric OIDs. The line `::= { parent N }` means: create this object as child number N under the parent object.



Example

`interfaces OBJECT IDENTIFIER ::= { mib-2 2 }`
means:
`mib-2 = 1.3.6.1.2.1`
`interfaces = 1.3.6.1.2.1.2`

How to read parent

If you see `::= { ifEntry 8 }`, first resolve ifEntry. If ifEntry is 1.3.6.1.2.1.2.2.1, then the object is 1.3.6.1.2.1.2.2.1.8.

Scalar vs table objects

Scalar

Single value for the device. Instance usually ends in .0

Example: sysUpTime.0 = uptime value

Table column

A column definition is not yet a final value. Walk it to see rows.

Example: ifName.101 = Gi1/0/1

ifName.102 = Gi1/0/2

snmpwalk output

sysUpTime.0 = Timeticks: (123456)
0:20:34.56

ifOperStatus.101 = INTEGER: up(1)

ifOperStatus.102 = INTEGER: down(2)

Some values are OIDs, not numbers or text

Certain SNMP objects intentionally return another OID as their value. Treat the value as a pointer, then resolve it.

Common cases

sysObjectID.0 returns the vendor/model identity OID

Do not classify it as a metric

Treat like relationship data

Example

SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.1.1208

1.3.6.1.2.1.1.2.0 = Cisco enterprise branch

OBJECT IDENTIFIER and OID-valued SYNTAX

system OBJECT IDENTIFIER ::= { mib-2 1 }

sysObjectID OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprises subtree (1.3.6.1.4.1)."

::= { system 2 }

cisco OBJECT IDENTIFIER ::= { enterprises 9 }

-- vendor-specific model identifiers live below the Cisco enterprise branch

Some values are Enumerated

A device returns an INTEGER and MIB maps each number to a human-readable coded state. See cLApPowerStatus within CISCO-LWAPP-AP-MIB

Example

2 = fifteendotfour = 15.4 W from Ethernet

3 = sixteendoteight = 16.8 W from Ethernet

6 = twentyfivedotfive = 25.5 W from Ethernet

Operational meaning

This is not a counter

This is not a gauge to graph as "2"

It is a status value and should use a value map

```
1254
1255 cLApPowerStatus OBJECT-TYPE
1256     SYNTAX          INTEGER {
1257         low(1),
1258         fifteendotfour(2),
1259         sixteendoteight(3),
1260         full(4),
1261         external(5),
1262         twentyfivedotfive(6),
1263         mixedmode(7)
1264     }
1265     MAX-ACCESS       read-only
1266     STATUS           current
1267     DESCRIPTION
1268         "This object represents the power status of the AP
1269         low           - The AP is drawing low power from
1270                        Ethernet
1271         fifteendotfour - The AP is drawing 15.4 watts from
1272                        Ethernet
1273         sixteendoteight - The AP is drawing 16.8 watts from
1274                        Ethernet
1275         twentyfivedotfive - The AP is drawing 25.5 watts from
1276                        Ethernet
1277         full          - The power is high enough for
1278                        normal operation of the AP.
1279                        The power source could be
1280                        Ethernet or external
1281         external      - The power source is not from
1282                        Ethernet.
1283         mixedmode     - The power is high enough for
1284                        operation of AP. In this state
1285                        one radio will operate in full
1286                        power mode while the other radio
1287                        will be powered off or operate
1288                        at reduced capacity."
1289     ::= { cLApEntry 20 }
1290
```

cLApPowerStatus

What am I talking to?

```
$ snmpget -v2c -c public printer .1.3.6.1.2.1.1.1.0 .1.3.6.1.2.1.1.2.0
sysDescr.0   = STRING: Xerox AltaLink C8155; SS 122.010.025.20910,
              NC 122.010.20910, UI 122.010.20910, ME 032.045.000, ...
sysObjectID.0 = OID: enterprises.253.8.62.1.35.2.9.1.1
```

253 -> look it up in the IANA enterprise numbers registry

iana.org/assignments/enterprise-numbers

253 Xerox

8072 Net-SNMP

2021 UCD-SNMP (the old net-snmp registration)

2699 PWG, the Printer Working Group

9 Cisco

Network device starting points

Purpose	Object	Where / note
Interface discovery	ifName, ifDescr, ifAlias	IF-MIB ifTable/ifXTable
Port state	ifOperStatus, ifAdminStatus	1.3.6.1.2.1.2.2.1.8 /.7
Traffic	ifHCInOctets, ifHCOutOctets	64-bit counters; delta/sec x 8 = bps
Errors/discards	ifInErrors, ifOutErrors, ifInDiscards	Counters per interface
Port speed	ifHighSpeed	UNITS: 1,000,000 bits/sec
Inventory	entPhysicalSerialNum, entPhysicalName	ENTITY-MIB Indexed by entPhysicalIndex
Neighbors	lldpRemTable	LLDP-MIB
MAC/VLAN tables	dot1dTpFdbTable, dot1qTpFdbTable	BRIDGE-MIB / Q-BRIDGE-MIB

Printer starting points

Purpose	Object	Where / note
Printer-MIB root	Printer-MIB	1.3.6.1.2.1.43
Supplies table	prtMarkerSuppliesTable	1.3.6.1.2.1.43.11.1.1
Supply description	prtMarkerSuppliesDescription	...43.11.1.1.6
Supply unit	prtMarkerSuppliesSupplyUnit	...43.11.1.1.7
Supply max capacity	prtMarkerSuppliesMaxCapacity	...43.11.1.1.8
Supply level	prtMarkerSuppliesLevel	...43.11.1.1.9
Pages / marker life	prtMarkerLifeCount	...43.10.2.1.4
Xerox private branch	Xerox MIBs	1.3.6.1.4.1.253

Index values are part of the OID

Indexes are not always a single integer. Some tables use multiple indexes or encoded strings / IP addresses. Always read INDEX in the MIB.

Simple index

ifOperStatus, i.e. .8.101

```
ifOperStatus.101 = up(1)  
lldpRemSysName.0.7.3 = "switch-core-01"  
ipNetToMediaPhysAddress.2.192.0.2.10 = 00:11:22:33:44:55
```

Multiple indexes

LLDP remote tables often use several index components: time mark, local port, remote index

String / IP indexes

Some tables encode a string length + characters or IP address octets into the OID suffix.

From MIB file to OID

Translate a name to numeric OID

```
snmptranslate -m +IF-MIB -On -IR ifOperStatus
```

Show full MIB definition

```
snmptranslate -m +IF-MIB -Td -IR ifOperStatus
```

Walk a column numerically

```
snmpwalk -v2c -c public -On <device IP> 1.3.6.1.2.1.2.2.1.8
```

Resolve an OID value returned by a device

```
snmptranslate -m +ALL -Td .1.3.6.1.4.1.9.1.1208
```

```

→ ~ snmptranslate -m +IF-MIB -On -IR ifOperStatus
.1.3.6.1.2.1.2.2.1.8
→ ~
→ ~
→ ~ snmptranslate -m +IF-MIB -Td -IR ifOperStatus

IF-MIB::ifOperStatus
ifOperStatus OBJECT-TYPE
-- FROM      IF-MIB
SYNTAX       INTEGER {up(1), down(2), testing(3), unknown(4), dormant(5), notPresent(6), lowerLayerDown(7)}
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "The current operational state of the interface. The
              testing(3) state indicates that no operational packets can
              be passed. If ifAdminStatus is down(2) then ifOperStatus
              should be down(2). If ifAdminStatus is changed to up(1)
              then ifOperStatus should change to up(1) if the interface is
              ready to transmit and receive network traffic; it should
              change to dormant(5) if the interface is waiting for
              external actions (such as a serial line waiting for an
              incoming connection); it should remain in the down(2) state
              if and only if there is a fault that prevents it from going
              to the up(1) state; it should remain in the notPresent(6)
              state if the interface has missing (typically, hardware)
              components."
 ::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) interfaces(2) ifTable(2) ifEntry(1) 8 }
→ ~
→ ~
→ ~ snmpwalk -v2c -c public -On 127.0.0.1:1166 1.3.6.1.2.1.2.2.1.8
.1.3.6.1.2.1.2.2.1.8.1 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.2 = INTEGER: up(1)
.1.3.6.1.2.1.2.2.1.8.3 = INTEGER: up(1)
→ ~
→ ~
→ ~ snmptranslate -m +ALL -Td .1.3.6.1.4.1.9.1.1208
SNMPv2-SMI::enterprises.9.1.1208
enterprises OBJECT-TYPE
-- FROM      SNMPv2-SMI, RFC1155-SMI
 ::= { iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) 9 1 1208 }
→ ~
→ ~
→ ~

```

Commands in action

From RAW data to Zabbix

SNMP walk master item + LLD

Walk tables like i.e. ifName and ifDescr. Create LLD macros: {#IFNAME}, {#IFDESCR}

Item prototypes

Use column OIDs with {#SNMPINDEX}. Example:
net.if.in[ifHCInOctets.{#SNMPINDEX}]

Preprocessing

Counters: change per second

Octets to bits: multiply by 8

Preprocessing

Item prototype

Item prototype

Tags 3

Preprocessing 3

Preprocessing steps ?	Name	Parameters	Custom on fail	Actions
1:	SNMP walk value	1.3.6.1.2.1.31.1.1.1.10.1	Unchanged	☐ Test Remove
2:	Change per second		☐	Test Remove
3:	Custom multiplier	8	☐	Test Remove

[Add](#)

Type of information Numeric (unsigned)

Update

Clone

Test

Delete

Cancel

Item prototype

Item prototype

Tags 3

Preprocessing 3

Preprocessing steps ?	Name	Parameters	Custom on fail	Actions
1:	SNMP walk value	1.3.6.1.2.1.31.1.1.1.15.1	Unchanged	☐ Test Remove
2:	Custom multiplier	1000000	☐	Test Remove
3:	Discard unchanged with heartbeat	1h	☐	Test Remove

[Add](#)

Type of information Numeric (unsigned)

Update

Clone

Test

Delete

Cancel

Counters / traffic volume

Ever-growing values, i.e. interface octets

Port speed / gauge value

Current value, not an accumulated counter

Lab

Build a simple SNMP template for printers



Exercise goals:

- Verify that the devices respond over SNMP
- Create a master item that performs an SNMP walk of the supplies table
- Build a discovery rule based on the walk result
- Create an item prototype for supply levels
- Convert supply levels into percentages
- Add a trigger prototype for low supplies
- Link the same template to a second printer
- Troubleshoot unsupported items
- Add regular individual SNMP items for metrics that do not require discovery

SNMP lab devices

lab-printer-01

127.0.0.1:1165

lab-printer-02

127.0.0.1:1166

Prepared Zabbix hosts

- lab-printer-01
- lab-printer-02

Initial state

- lab-printer-01 already has an SNMP interface configured,
- lab-printer-01 has an empty template called Printer by SNMP linked to it,
- lab-printer-02 has only an SNMP interface configured,
- the Printer by SNMP template is an empty starting point.

If you get stuck, import the relevant checkpoint file:

Data collection > Templates > Import

Step 0 - Check SNMP connectivity

Before we build anything in Zabbix, we verify that the device responds over SNMP

Check the system description:

```
snmpget -v2c -c public -On 127.0.0.1:1165 .1.3.6.1.2.1.1.0
```

Check the printer supplies description column:

```
snmpwalk -v2c -c public -On 127.0.0.1:1165 .1.3.6.1.2.1.43.11.1.1.6
```

[] SNMP agent is responding

[] public community works

[] address and port are correct

[] device exposes Printer-MIB data

[] supplies table contains rows

Architecture

SNMP walk -> Master item -> Discovery -> Items -> Triggers

Pages printed -> Direct SNMP item

Printer status -> Direct SNMP item

CPU & Memory -> Direct SNMP items + calculated item

Asynchronous data collection

Instead of polling every toner or supply separately, we perform one SNMP walk of the table. Zabbix stores the result in a master item. Discovery rule and dependent items then use the same collected data.

Pros:

- fewer SNMP requests
- asynchronous data collection with walk[] item
- consistent data
- items are created automatically for discovered rows

Step 1 – Create the master walk item

Data collection > Templates > Printer by SNMP > Items > Create item

Name: Marker supplies walk

Type: SNMP agent

Key: printer.supplies.walk

Type of information: Text

Update interval: 1m

History: Do not store

Tags: component = raw

SNMP OID:

walk[1.3.6.1.2.1.43.11.1.1.4,1.3.6.1.2.1.43.11.1.1.6,1.3.6.1.2.1.43.11.1.1.7,1.3.6.1.2.1.43.11.1.1.8,1.3.6.1.2.1.43.11.1.1.9]

Test the item on the HOST (lab-printer-01)

This walk collects several columns from the printer supplies table:

What data do we get?

- supply class
- supply description
- supply unit
- maximum capacity
- current level

Step 2 – Create a discovery rule

In the template, Items, click the three dots to the left of "Marker supplies walk" and choose "Create dependent discovery rule"

Name: Marker supplies discovery

Key: printer.supplies.discovery

Preprocessing tab, step 1: SNMP walk to JSON.

{#SUPPLY.DESCR} 1.3.6.1.2.1.43.11.1.1.6 Unchanged

{#SUPPLY.CLASS} 1.3.6.1.2.1.43.11.1.1.4 Unchanged

{#SUPPLY.UNIT} 1.3.6.1.2.1.43.11.1.1.7 Unchanged

{#SUPPLY.MAX} 1.3.6.1.2.1.43.11.1.1.8 Unchanged

Step 3 – Create an item prototype

Marker supplies discovery > Item prototypes > Create item prototype

Name	Supply {#SUPPLY.DESCR}: remaining
Type	Dependent item
Key	printer.supply[{#SNMPINDEX}]
Master item	Printer by SNMP: Marker supplies walk
Type of information	Numeric (float)
Units	%
Tags	component = supply

Step 3 – Create an item prototype

Preprocessing steps

Step 1: SNMP walk value 1.3.6.1.2.1.43.11.1.1.9.{#SNMPINDEX} Unchanged

Step 2: JavaScript. Converts the level to a percentage

```
var max = Number('{#SUPPLY.MAX}'),  
    unit = Number('{#SUPPLY.UNIT}'),  
    level = Number(value);  
if (unit == 19) return level; if (max <= 0) throw 'capacity not reported'; return 100 * level / max;
```

Once added, then on the HOST, select the master item and press **Execute now**

Step 4 – Create a trigger

4a. Configure a new macro on the TEMPLATE

Data collection > Templates > Printer by SNMP > Macros tab
`{ $SUPPLY.WARN } 15`

4b. Marker supplies discovery > Trigger prototypes > Create trigger prototype

Name: Supply `{ #SUPPLY.DESCR }` is low

Severity: Warning

`last(/Printer by SNMP/printer.supply[{ #SNMPINDEX }]) < { $SUPPLY.WARN }`

See if new alarm has been raised for lab-printer-01

Step 5 – Link the template to the second printer host

Link Printer by SNMP to lab-printer-02

Check items in latest data. Three of them on lab-printer-02 are now UNSUPPORTED

Step 6 – Fix those three items

Rules: do not edit the JavaScript, do not touch the items that already work
and do not disable anything

Step 6 – Fix those three items

The answer is one preprocessing step as in stage6.yaml

Insert BEFORE the JavaScript, as step 2 of three:

In range

min 0, max empty

Custom on fail > Discard value

The second catch here is that Zabbix clears the unsupported flag when an item successfully stores a value. Since Discard value is in place, it will never reset the flag.

Select the three unsupported items on lab-printer-02, Delete, then Execute now on the master so discovery recreates them. Alternatively, you can unlink and clear template and then link it again.

Step 7 – Adding individual SNMP items (not every metric needs a walk)

Data collection > Templates > Printer by SNMP > Items > Create item

Name: Pages printed
Type: SNMP agent
Key: printer.pages
Type of information: Numeric (unsigned)
SNMP OID: get[1.3.6.1.2.1.43.10.2.1.4.1.1]
Units: pages
Update interval: 5m
Tags: component = system

Name: Printer status
Type: SNMP agent
Key: printer.status
Type of information: Numeric (unsigned)
SNMP OID: get[1.3.6.1.2.1.25.3.5.1.1.1]

Update interval: 1m
Tags: component = health

Add both, go to latest data and execute them. Verify the reported values.

Fix the units display.

Step 8 – Adding individual SNMP items – Memory and CPU

Data collection > Templates > Printer by SNMP > Items > Create item

Name: Memory used
Type: SNMP agent
Key: printer.memory.used
Type of information: Numeric (unsigned)
SNMP OID: get[1.3.6.1.2.1.25.2.3.1.6.1]
Units B
Update interval 1m
Tags component = memory

Preprocessing, step 1: Custom multiplier 1024

Name: Memory size
Type: SNMP agent
Key: printer.memory.size
Type of information: Numeric (unsigned)
SNMP OID: get[1.3.6.1.2.1.25.2.2.0]
Units B
Update interval: 1m
Tags: component = memory

Preprocessing, step 1: Custom multiplier 1024

Step 8 – Adding individual SNMP items – Memory and CPU

Data collection > Templates > Printer by SNMP > Items > Create item

Name: Memory utilization
 Type: Calculated
 Key: printer.memory.util
 Type of information: Numeric (float)
 Formula:
 100*last(//printer.memory.used)/last(//printer.memory.size)
 Units %
 Update interval 1m
 Tags component = memory

Name: CPU utilization
 Type: SNMP agent
 Key: printer.cpu.util
 Type of information: Numeric (unsigned)
 SNMP OID: walk[1.3.6.1.2.1.25.3.3.1.2]
 Units %
 Update interval: 1m
 Tags: component = cpu

 Preprocessing, step 1: Regular expression
 Parameters: INTEGER: (\d+) \1

Without preprocessing, item would be unsupported:

lab-printer-01 Value of type "string" is not suitable for value type "Numeric (unsigned)". Value ".1.3.6.1.2.1.25.3.3.1.2.8 = INTEGER: 7"

Step 9 – Adding trigger for High Memory Utilization > 80%

9a. Configure a new macro on the TEMPLATE

Data collection > Templates > Printer by SNMP > Macros tab

`{$MEMORY.UTIL.MAX}` 80

9b. Data collection > Templates > Printer by SNMP > Triggers > Create trigger

Name: Memory utilization is high (>{\$MEMORY.UTIL.MAX}%)

Severity: High

`last(/Printer by SNMP/printer.memory.util)>{$MEMORY.UTIL.MAX}`

See if new alarm has been raised for lab-printer-02

Lab completed. Q&A



THANK YOU!

Hope that SNMP templates are not that scary anymore! :)

