

SNMP BORNE WIFI ZABBIX

Version	Date
1.0	04/01/2023

Table des matières

PROJET	3
PROJET 1.....	4
Utilisation de iReasoning MIB Browser.....	4
Site de référencement : mibs.observium.org.....	5
Création d'un rôle de découverte sous Zabbix :	5
Création d'un item sous Zabbix :.....	6
PROJET 2.....	7
Clonage de la Template :.....	7
Création des items :	7
Item prototype « CPU utilization » :	8
Item prototype « Model name » :	8
Item prototype « serial number » :	8
Affectation de la Template au contrôleurs virtuels (VC) :.....	9
Récupération de données via la Template :	10

PROJET

Sous projet 1

Une update est demandée sur Zabbix concernant l'ajout d'un rôle de découverte « Discovery » permettant d'identifier les points d'accès sur un secteur (Niort pour les tests).

Un item sera ajouté suite à la création du rôle de découverte, consistant à recupérer le numéro de série (depuis une OID) pour identifier les composants dans le software, afin d'être plus réactif sur les demandes d'informations pour remplacer un équipement si une panne a lieu.

Le but demandé est donc de trouver le numéro de série de l'OID de ces objets.

Cependant, un contrôleur virtuel est mis en place au-dessus des points d'accès qui va générer des doublons voir de tripler sur la requête (Discovery). L'objectif est de récupérer un numéro et d'éviter les doublons (pas optimisé).

Les logiciels permettant l'identification de l'OID se fera avec « iReasoning MIB Browser », la communauté «XXXXXX», l'opération en « Walk » et une adresse IP d'un point d'accès pour les tests sur l'équipement :« X.X.X.X » situé à **Niort**.

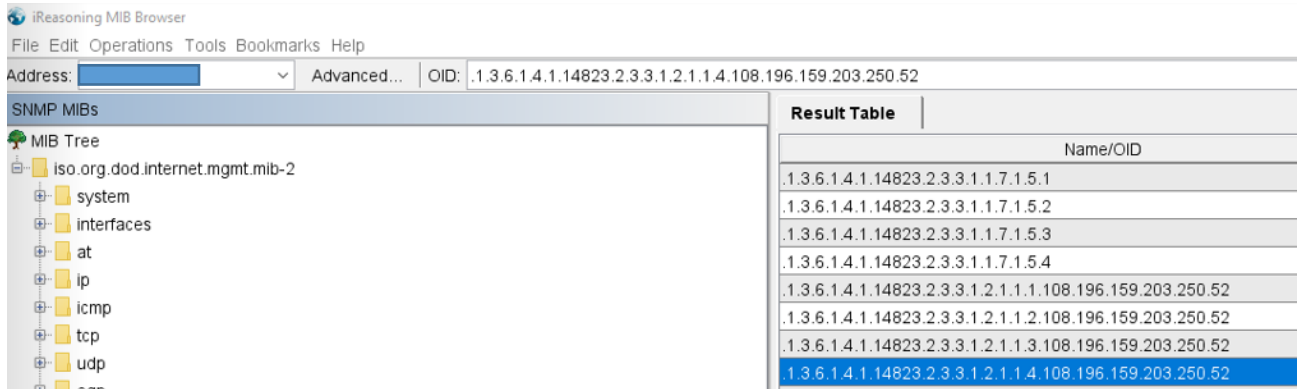
Des sites web de référencements à disposition sur Internet pour l'approfondissement de la recherche de la requête : « <https://oidref.com> » ; <https://mibs.observium.org/> et www.oid-info.com

Sous projet 2

La deuxième partie consiste donc de régler le problème de doublons par la création d'un nouveau Template qui se nommera « aruba-vc-snmpv2 » et, qui aura pour principe de regrouper les contrôleurs virtuels du groupe « wifi-vc ». De plus, mettre en place des items pour la collecte d'informations complémentaires comme le numéro de série, le nom du model, l'utilisation du CPU ainsi que le nom de l'équipement.

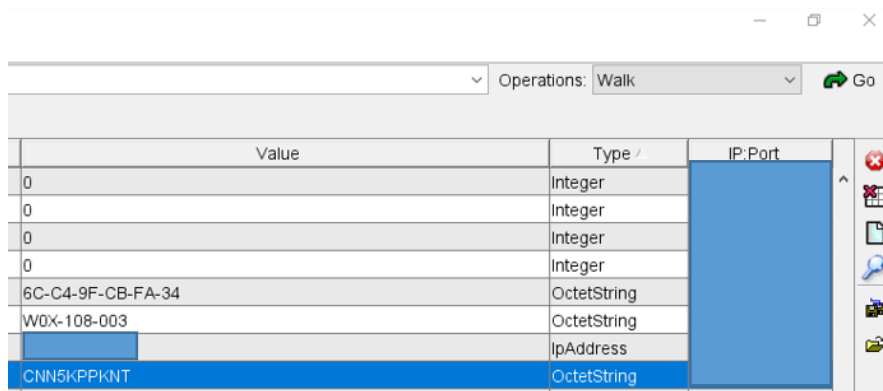
PROJET 1

Utilisation de iReasoning MIB Browser



The screenshot shows the iReasoning MIB Browser interface. The 'Address' field is set to 'Advanced...' and the 'OID' field contains '.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.4.108.196.159.203.250.52'. The 'MIB Tree' on the left shows a hierarchy starting from 'iso.org.dod.internet.mgmt.mib-2' down to 'ip'. The 'Result Table' on the right displays a list of Name/OID pairs. The last entry, '.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.4.108.196.159.203.250.52', is highlighted in blue.

Name/OID
.1.3.6.1.4.1.14823.2.3.3.1.1.7.1.5.1
.1.3.6.1.4.1.14823.2.3.3.1.1.7.1.5.2
.1.3.6.1.4.1.14823.2.3.3.1.1.7.1.5.3
.1.3.6.1.4.1.14823.2.3.3.1.1.7.1.5.4
.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.1.108.196.159.203.250.52
.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.2.108.196.159.203.250.52
.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.3.108.196.159.203.250.52
.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.4.108.196.159.203.250.52



The screenshot shows the 'Operations' window in iReasoning MIB Browser. The 'Operations' dropdown is set to 'Walk' and the 'Go' button is visible. The table below shows the results of the walk operation.

Value	Type /	IP:Port
0	Integer	
0	Integer	
0	Integer	
0	Integer	
6C-C4-9F-CB-FA-34	OctetString	
W0X-108-003	OctetString	
	IpAddress	
CNN5KPPKNT	OctetString	

La valeur récupérée de l'équipement du numéro série est « **CNN5KPPKNT** » du point d'accès X.X.X.X avec comme valeur : 1.3.6.1.4.1.14823.2.3.3.1.2.1.1.4.{#SNMPINDEX}

Name/OID:	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.4.108.196.159.203.250.52;	Value	(OctetString):
CNN5KPPKNT			

La partie grise correspond à toutes les bornes Wi-Fi qu'elles ont en commun d'où le 4 à la fin sur le numéro de série.

Site de référencement : mibs.observium.org

MIB	OID
aiInterAPType	.1.3.6.1.4.1.14823.2.3.3.1.1.8.1.8
aiStateGroup	.1.3.6.1.4.1.14823.2.3.3.1.2
aiAccessPointTable	.1.3.6.1.4.1.14823.2.3.3.1.2.1
aiAccessPointEntry	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1
aiAPMACAdresse	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.1
aiAPTotalMemory	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.10
aiAPStatus	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.11
aiAPHwopmode	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.12
aiAPRôle	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.13
aiAPName	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.12
aiAPIPAddress	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.13
aiAPSerialNum	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.14
aiAPModel	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.15
aiAPModelName	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.16
aiAPCPUUtilization	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.17
aiAPMémoireLibre	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.18
aiAPUptime	.1.3.6.1.4.1.14823.2.3.3.1.2.1.1.19

Le résultat fait référence au numéro de série de la borne Wi-Fi est :

aiAPSerialNum == (MACRO) .1.3.6.1.4.1.14823.2.3.3.1.2.1.1.14 == (OID)

Un **Discovery** sera effectué car nul ne sait combien il y a de bornes Wi-Fi en place et actifs. Cela permet aussi de ne pas en oublier.

Création d'un rôle de découverte sous Zabbix :

Discovery rule configuration:

- Parent discovery rules: **aruba:ap:snmpv2**
- Name: **Discovery serial ap**
- Type: **SNMP agent**
- Key: **sn.ap.discovery**
- Host interface: **[selected]**
- SNMP OID: **discovery[["aiAPSerialNum"], 1.3.6.1.4.1.14823.2.3.3.1.2.1.1.14]**
- Update interval: **1d**
- Custom intervals:

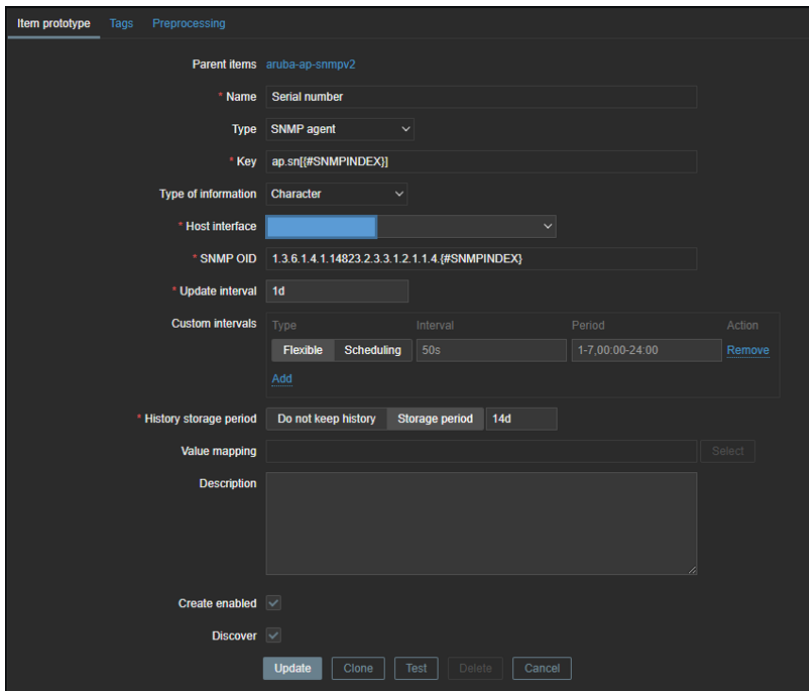
Type	Interval	Period	Action
Flexible	Scheduling	50s	1-7,00:00-24:00
- Keep lost resources period: **30d**
- Description: **[empty]**
- Enabled: ☒

Requête retournée :

```
[{"#AIAPSERIALNUM":"CNN5KPPKY4"},
{"#AIAPSERIALNUM":"CNN5KPPL01"},
{"#AIAPSERIALNUM":"CNN5KPPKSB"},
{"#AIAPSERIALNUM":"CNN5KPPKNT"},
{"#AIAPSERIALNUM":"CNN8KWC1C8"},
{"#AIAPSERIALNUM":"CNN8KWC1BK"}]
```

La demande effectuée est un « Discovery » en mode agent SNMP sur l'équipement (borne Wi-Fi) demandant le numéro de série qui sera actualisé tous les jours et si l'équipement est un jour changé il gardera dans son historique 30 jours. Le problème avec cette requête est qu'il va relever les numéros de séries de chaque points d'accès sans pour autant associer au bon équipement d'où l'intervention d'un item.

Création d'un item sous Zabbix :



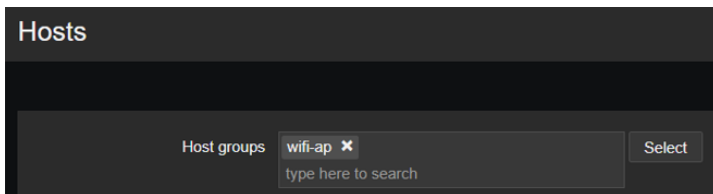
The screenshot shows the 'Item prototype' form in Zabbix. The 'Parent items' are 'aruba-ap-snmpv2'. The 'Name' is 'Serial number'. The 'Type' is 'SNMP agent'. The 'Key' is 'ap.sn[{#SNMPINDEX}]'. The 'Type of information' is 'Character'. The 'Host interface' is selected. The 'SNMP OID' is '1.3.6.1.4.1.14823.2.3.3.1.2.1.1.4.{#SNMPINDEX}'. The 'Update interval' is '1d'. The 'Custom intervals' table shows 'Flexible' type, 'Scheduling' interval, '50s' interval, and '1-7,00:00-24:00' period. The 'History storage period' is 'Do not keep history'. The 'Value mapping' is empty. The 'Description' is empty. The 'Create enabled' and 'Discover' checkboxes are checked. The 'Update' button is highlighted.

Requête retourné pour une borne :

```
[{"{#SNMPINDEX}":"108.196.159.203.250.52", "{#AIAPSERIALNUM}":"CNN5KPPKNT"}]
```

L'item mis en place va permettre d'associer le numéro de trouver série depuis le rôle de découverte au bon équipement qui sera demander tous les jours.

La requête : 1.3.6.1.4.1.14823.2.3.3.1.2.1.1.4.{#SNMPINDEX}



The screenshot shows the 'Hosts' form in Zabbix. The 'Host groups' field is populated with 'wifi-ap' and a search box is visible below it.

L'application de la Template « aruba-ap-snmpv2 » au groupe « wifi-ap » (borne wifi) va pour pouvoir relever les numéros de séries de chaque équipement.

Cependant cette méthode pour récupérer ces données va être doublée voire triplée car la requête passée par le contrôleur virtuel dont lui aussi va dire combien il a de point d'accès en sa possession et ainsi les points d'accès vont à leur tour répondre à la requête demander.

On s'aperçoit que si la requête passe par le contrôleur virtuel va faire remonter tous les équipements avec le numéro de série qui peut joindre. Donc le but va être de cloner cette Template avec son/ses item(s) et de le placer sur les contrôleurs virtuels (wifi-vc), tout en supprime celle faite avant.

PROJET 2

Clonage de la Template :

Avant de supprimer la Template, effectuer un clonage en permettant de récupérer le « Discovery » ; et « l'item » ayant pour but de relever les numéros de série depuis les VC qui seront sous le nom d'aruba-vc-snmpv2. Et de changer les éléments qui ont été clonés précédemment.

Discovery rules

All templates / **aruba-vc-snmpv2** / Discovery list / Discovery slave ap / Item prototypes 3 / Trigger prototypes / Graph prototypes / Host p

Discovery rule / Preprocessing / LLD macros / Filters / Overrides

Name : Discovery slave ap

Type : SNMP agent

Key : sl.ap.discovery

SNMP OID : discovery[#{AIAPNAME},1.3.6.1.4.1.14823.2.3.3.1.2.1.1.2]

Update interval : 1d

Custom intervals

Type	Interval	Period	Action
Flexible	Scheduling	50s	1-7:00:00-24:00

Keep lost resources period : 30d

Description

Enabled ☒

Buttons : Update Clone Test Delete Cancel

Requête retourné pour une borne :

```
{#{SNMPINDEX}:"108.196.159.203.226.74",
#{AIAPNAME}:"W0X-108-001"},

{#{SNMPINDEX}:"108.196.159.203.232.152",
#{AIAPNAME}:"W0X-108-002"},

{#{SNMPINDEX}:"108.196.159.203.232.252",
#{AIAPNAME}:"W0X-108-004"},

{#{SNMPINDEX}:"108.196.159.203.250.52",
#{AIAPNAME}:"W0X-108-003"},

{#{SNMPINDEX}:"168.91.247.201.50.126",#{
AIAPNAME}:"W0X-108-005"},

{#{SNMPINDEX}:"168.91.247.201.55.104",#{
AIAPNAME}:"W0X-108-006"}
```

Le « Discovery » sera effectuer sur le nom des équipements pour une lecture de données cohérent :

discovery[#{AIAPNAME},1.3.6.1.4.1.14823.2.3.3.1.2.1.1.2]

Création des items :

Item prototypes Create item prototype

All templates / aruba-vc-snmpv2 / Discovery list / Discovery slave ap / **Item prototypes 3** / Trigger prototypes / Graph prototypes / Host prototypes

	Name	Key	Interval	History	Trends	Type	Create enabled	Discover	Tags
...	{#{AIAPNAME}} CPU utilization	system.cpu.utilization.[#{SNMPINDEX}]	1m	90d		SNMP agent	Yes	Yes	
...	{#{AIAPNAME}} Model name	system.model.[#{SNMPINDEX}]	1d	90d		SNMP agent	Yes	Yes	
...	{#{AIAPNAME}} System serial number	system.serial.number.[#{SNMPINDEX}]	1d	14d		SNMP agent	Yes	Yes	

Displaying 3 of 3 found

Les items vont pouvoir récupérer l'élément qui leur correspond au sein de la « Discovery » et qu'il soit bien attribué au bon équipement via le « SNMPINDEX ».

Item prototype « CPU utilization » :

Item prototype Tags Preprocessing

* Name

Type

* Key

Type of information

* SNMP OID

* Update interval

Custom intervals

Type	Interval	Period	Action
Flexible	Scheduling	50s	1-7,00 00-24.00

[Add](#) [Remove](#)

* History storage period

Value mapping

Description

Create enabled ☒

Discover ☒

Requête retourné sur le VC (W0X-108-002) :

W0X-108-002 CPU utilization	55s	4
-----------------------------	-----	---

Item prototype « Model name » :

Requête retourné sur le VC (W0X-108-002) :

W0X-108-002 Model name	3h 14m 12s	505
------------------------	------------	-----

Item prototype Tags Preprocessing

* Name

Type

* Key

Type of information

* SNMP OID

* Update interval

Custom intervals

Type	Interval	Period	Action
Flexible	Scheduling	50s	1-7,00 00-24.00

[Add](#) [Remove](#)

* History storage period

Value mapping

Description

Create enabled ☒

Discover ☒

Item prototype « serial number » :

Item prototype Tags Preprocessing

* Name

Type

* Key

Type of information

* SNMP OID

* Update interval

Custom intervals

Type	Interval	Period	Action
Flexible	Scheduling	50s	1-7,00 00-24.00

[Add](#) [Remove](#)

* History storage period

Value mapping

Description

Create enabled ☒

Discover ☒

Requête retourné sur le VC (W0X-108-002) :

W0X-108-002 System serial number	3h 14m 42s	CNN5KPPL01
----------------------------------	------------	------------

Affectation de la Template au contrôleurs virtuels (VC) :

L'affectation de cette Template sur les contrôleurs virtuels va pouvoir optimiser la recherche des équipements et de faire l'inventaire sur tout le parc (Ouest). En sélectionnant tous les contrôleurs virtuels.

✓ Name ▲	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
✓ VC-ANGERS	Items 46	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-ANGOULEME	Items 40	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-BREST	Items 43	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-BRIVE	Items 52	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-LE-MANS	Items 43	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-LES-HERBIERS2 ⚡	Items 9	Triggers 7	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-LIMOGE	Items 43	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-NIORT	Items 40	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-ORLEANS	Items 40	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-ORLEANS-ISTELI	Items 34	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-QUIMPER	Items 37	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-RENNES	Items 58	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-ST-BRIEUC	Items 34	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		
✓ VC-VANNES	Items 40	Triggers 12	Graphs	Discovery 3	Web			aruba-vc-snmpv2 generic-snmp, interfaces-snmp-ap	Enabled	SNMP	None		

Dans le champ « Replace » définir la Template qui a été créée précédemment :

Mass update

Host

IPMI

Tags

Macros

Inventory

Encryption

Value mapping

Link templates

Link

Replace

Unlink

aruba-vc-snmpv2

type here to search

Select

Clear when unlinking

Host groups

Original

Description

Original

Monitored by proxy

Original

Status

Original

Update

Cancel

Récupération de données via la Template :

Latest data

< 

Host groups:

Tags:

Name:

Show tags
Tag display priority
Show details

Subfilter affects only filtered data

HOSTS
VC-ORLEANS 6

DATA
☒ With data ☐ Without data

Host	Name	Last check	Last value
VC-ORLEANS	W0X-096-001 CPU utilization	26s	5
VC-ORLEANS	W0X-096-002 CPU utilization	26s	3
VC-ORLEANS	W0X-096-003 CPU utilization	26s	3
VC-ORLEANS	W0X-096-004 CPU utilization	26s	3
VC-ORLEANS	W0X-096-005 CPU utilization	26s	3
VC-ORLEANS	W0X-096-006 CPU utilization	26s	4

Filtrage de données :

Depuis le groupe « wifi-vc » (là où sont regroupé tout les contrôleur virtuelle) on va demander dans le champ « Host » un équipements (VC) d'un centre et de dire l'élément que l'on veut relever (cpu).

La liste nous montre bien en « % » utilisé par les bornes wifi ainsi que leur nom.

Pour chaque centre on peut maintenant relever les valeurs de chaque borne wifi depuis le « VC » ça nous optimise bien de la place contrairement au **projet 1**. Et le résultat est plus propre car on a le secteur puis le nom de l'équipement avec les données derrière.

Voici le groupe appartenant à tous les VC :

wifi-vc Hosts 14 Templates VC-ANGERS, VC-ANGOULEME, VC-BREST, VC-BRIVE, VC-LE-MANS, VC-LES-HERBIERS2, VC-LIMOGES, VC-NIORT, VC-ORLEANS, VC-ORLEANS-ISTEL, VC-QUIMPER, VC-RENNES, VC-ST-BRIEUC, VC-VANNES

Displaying 3 of 3 found