

# Note technique de dépannage SNMP

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Conventions](#)

[Gestionnaires et agents](#)

[MIB, ID d'objet et instances](#)

[Applications](#)

[MIB](#)

[Conseils](#)

[RFC](#)

[Informations connexes](#)

## [Introduction](#)

Ce document présente brièvement le protocole SNMP (Simple Network Management Protocol) et explique comment tester la fonctionnalité SNMP sur un serveur rack Cisco Unified Computing System (UCS) série C. Vous pouvez également l'utiliser comme référence pour le dépannage SNMP de base.

SNMP est une norme IETF (Internet Engineering Task Force) définie par plusieurs requêtes de commentaires (RFC). SNMP est utilisé dans les systèmes de gestion de réseau (NMS) pour surveiller les périphériques réseau. SNMP fonctionne au niveau de la couche application. Le principal objectif du protocole SNMP est d'activer les NMS afin d'obtenir des informations à partir de périphériques gérés.

## [Conditions préalables](#)

### [Conditions requises](#)

Ce document suppose que le NMS et tous les périphériques réseau ont été configurés correctement et ont fonctionné correctement dans le passé.

### [Components Used](#)

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- SNMP

- Serveurs rack Cisco UCS série C

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, make sure that you understand the potential impact of any command.

## Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

## Gestionnaires et agents

Le gestionnaire est le NMS et l'agent est le logiciel qui s'exécute sur le périphérique géré. SNMP est utilisé pour transporter les messages entre les gestionnaires et les agents.

## MIB, ID d'objet et instances

Une base d'informations de gestion (MIB) est un ensemble d'objets. Il existe deux types d'objets : scalaire (objet unique) et tabulaire (objets multiples). Un objet est une ou plusieurs caractéristiques ou variables d'un périphérique géré. Ces caractéristiques sont identifiées par un identificateur d'objet (OID). Un OID est une représentation numérique, séparée par des points, du nom de l'objet et de l'ID de l'instance. Les objets scalaires, tels que sysDesc, ont toujours une instance de 0. Si l'objet est une colonne dans une table, l'ID d'instance est généralement l'index de la colonne. Les MIB sont structurées dans une arborescence et les ID d'objet sont dérivés de cette arborescence.

**Remarque :** L'exemple de sortie de ce document a été capturé à partir d'un serveur rack Cisco UCS série C.

Cet exemple montre comment examiner la définition de CISCO-UNIFIED-COMPUTING-PROCESSOR-MIB pour cucsProcessorUnitEntry.

```
cucsProcessorUnitEntry OBJECT-TYPE
    SYNTAX          CucsProcessorUnitEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Entry for the cucsProcessorUnitTable table."
    INDEX { cucsProcessorUnitInstanceId }
    ::= { cucsProcessorUnitTable 1 }

CucsProcessorUnitEntry ::= SEQUENCE {
    cucsProcessorUnitInstanceId          CucsManagedObjectId,
    cucsProcessorUnitDn                  CucsManagedObjectDn,
    cucsProcessorUnitRn                  SnmpAdminString,
    cucsProcessorUnitArch                CucsProcessorUnitArch,
    cucsProcessorUnitCores               Gauge32,
    cucsProcessorUnitCoresEnabled        Gauge32,
    cucsProcessorUnitId                  Gauge32,
    cucsProcessorUnitModel                SnmpAdminString,
    cucsProcessorUnitOperState            CucsEquipmentOperability,
    cucsProcessorUnitOperability          CucsEquipmentOperability,
```

```

cucsProcessorUnitPerf
CucsEquipmentSensorThresholdStatus,
cucsProcessorUnitPower                CucsEquipmentPowerState,
cucsProcessorUnitPresence              CucsEquipmentPresence,
cucsProcessorUnitRevision              SnmpAdminString,
cucsProcessorUnitSerial                 SnmpAdminString,
cucsProcessorUnitSocketDesignation     SnmpAdminString,
cucsProcessorUnitSpeed                  INTEGER,
cucsProcessorUnitStepping               Gauge32,
cucsProcessorUnitThermal
CucsEquipmentSensorThresholdStatus,
cucsProcessorUnitThreads                Gauge32,
cucsProcessorUnitVendor                 SnmpAdminString,
cucsProcessorUnitVoltage
CucsEquipmentSensorThresholdStatus,
cucsProcessorUnitVisibility             CucsMemoryVisibility,
cucsProcessorUnitOperQualifierReason    SnmpAdminString

```

`cucsProcessorUnitEntry` n'est pas accessible, car il s'agit d'une séquence ou d'un parent et n'est pas lié à une instance spécifique. Les objets sont répertoriés sous la séquence ou le parent ; il peut être utile de les voir comme des lignes dans un tableau. Les objets correspondent à des OID spécifiques (nom numérique de l'objet + instance), qui peuvent être sollicités individuellement pour information. Si vous utilisez la commande `snmpwalk` sur la séquence ou le parent, vous recevez des réponses pour tous les objets sous-jacents. Cependant, vous devez utiliser un outil de traduction afin de mapper le nom de l'objet aux OID.

#### Note:

- La clause Syntax définit le format de l'entier. Par exemple, les compteurs 32, 64, 32, IP address, etc.
- Max-access définit la lecture-écriture, la lecture-crédation ou l'absence d'accès.
- La description est explicite.
- Index identifie l'ID de table.

Par exemple, si vous utilisez la commande `snmpwalk` sur `cucsProcessorUnitOperStateObject`, les résultats suivants s'affichent :

```

[root@localhost snmp]# snmpwalk -v2c -c public 14.17.2.121
.1.3.6.1.4.1.9.9.719.1.41.9.1.9
SNMPv2-SMI::enterprises.9.9.719.1.41.9.1.9.1 = INTEGER: 1
SNMPv2-SMI::enterprises.9.9.719.1.41.9.1.9.2 = INTEGER: 1

```

Il y a deux réponses à cette demande, car il y a deux processeurs dans le serveur. Les réponses reflètent également les deux instances pour `cucsProcessorUnitOperState`, une pour chaque processeur. Les informations OID sont extraites pour tous les objets de ce type. Dans cet exemple, la commande `snmpwalk` a utilisé le chemin dans l'arborescence MIB. Afin de traduire ceci, vous pouvez charger les MIB localement ou rechercher un outil qui traduit les objets MIB SNMP et les OID.

Cet exemple utilise un outil de traduction afin de traduire l'OID :

```

.1.3.6.1.4.1.9.9.719.1.41.9.1.9 (path through the tree)
cucsProcessorUnitOperState OBJECT-TYPE
    -- FROM CISCO-UNIFIED-COMPUTING-PROCESSOR-MIB
    -- TEXTUAL CONVENTION CucsEquipmentOperability
    SYNTAX Integer { unknown(0), operable(1), inoperable(2),

```

```
degraded(3),poweredOff(4), powerProblem(5), removed(6), voltageProblem(7),
thermalProblem(8), performanceProblem(9), accessibilityProblem(10),
identityUnestablishable(11), biosPostTimeout(12), disabled(13),
fabricConnProblem(51), fabricUnsupportedConn(52), config(81),
equipmentProblem(82), decomissioning(83), chassisLimitExceeded(84),
notSupported(100), discovery(101), discoveryFailed(102), identify(103),
postFailure(104), upgradeProblem(105), peerCommProblem(106),
autoUpgrade(107) }
    MAX-ACCESS read-only
    STATUS Current
    DESCRIPTION "Cisco UCS processor:Unit:operState managed object property"
 ::= { iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) cisco(9)
ciscoMgmt(9) ciscoUnifiedComputingMIB(719) ciscoUnifiedComputingMIBObjects(1)
cucsProcessorObjects(41) cucsProcessorUnitTable(9) cucsProcessorUnitEntry(1) 9 }
```

Au-dessus du type d'objet, vous pouvez voir le chemin à travers l'arborescence MIB qui a été utilisé par la commande **snmpwalk**. La réponse de 1 indique que les processeurs sont en état de fonctionnement.

## Applications

Pour utiliser la commande **snmpwalk** sur un PC, vous devrez peut-être rechercher et installer une application. Vous pouvez également rechercher des applications SNMP de l'interface utilisateur graphique Windows.

Un Mac fonctionne nativement.

Il y a des paquets disponibles pour Linux. Pour Centos ou Redhat, Net-SNMP est une application de ligne de commande qui teste la fonctionnalité SNMP. Afin d'installer Net-SNMP sur Centos ou Redhat, utilisez cette commande :

```
yum install net-snmp-utils
```

**Remarque** : si vous prévoyez de charger les MIB afin d'effectuer des traductions localement, créez un fichier `snmp.conf` et incluez les MIB souhaités.

## MIB

- [Liste de prise en charge MIB du gestionnaire Cisco UCS série C](#)
- [Liste d'assistance MIB de la gamme Cisco UCS B](#)

## Conseils

- Assurez-vous que la MIB particulière est prise en charge dans la version de code applicable.
- Vérifiez que l'adresse IP du périphérique géré est accessible.
- Assurez-vous que les ports sont ouverts sur les pare-feu.
- Utiliser les traces Wireshark ou TCPDump pour les problèmes de réponse et de paquets mal formés ; filtre sur l'adresse IP et le port (port UDP 161 pour SNMP, port UDP 162 pour les interruptions de rappel).

## RFC

Il existe de nombreuses RFC associées au protocole SNMP ; voici une liste partielle :

- 1155 Structure et identification des informations de gestion pour les interréseaux TCP/IP
- 1156 base d'informations de gestion pour la gestion réseau des interréseaux TCP/IP (MIB I)
- 1157 protocole de gestion de réseau simple
- 1213 base d'informations de gestion pour la gestion réseau des interréseaux TCP/IP (MIB II)
- 1441 Introduction à la version 2 du cadre de gestion de réseau standard Internet
- 1452 Coexistence entre la version 1 et la version 2 du cadre de gestion de réseau standard Internet
- 2578 Structure de l'information de gestion Version 2
- 3414 Modèle de sécurité basé sur utilisateur pour la version 3 du protocole de gestion de réseau simple
- 3584 Coexistence entre les versions 1, 2 et 3 du protocole de gestion de réseau standard Internet

## [Informations connexes](#)

- [Support et documentation techniques - Cisco Systems](#)