

SNMP Operations

1 Background

The Net-SNMP tools are useful particularly since they implement aspects of the protocol in a simple, understandable way, and are ideal for learning about the SNMP protocol. They are also useful troubleshooting tools. The tool `snmpget` implements the `get-request` part of the SNMP protocol. Read the man page for `snmpcmd` and make sure you understand what you are doing.

1.1 The `get-request` Operation

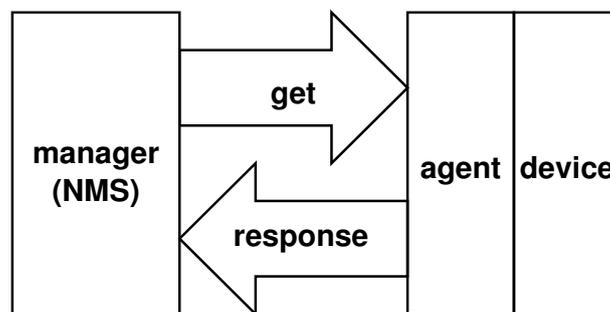


Figure 1: A manager sends an SNMP `get-request` to an agent, which sends a `response` to the manager.

Figure 1 shows a *manager* sending an SNMP `get-request` to an *agent* running on a managed network device, such as a network switch or network server. The `get-request` contains one or more OIDs. The agent responds by sending back the data associated with each OID.

1.2 Instance Numbers

Managed objects come in two main types:

- scalars, and
- tables.

The OID of each managed object ends with an *instance number*. The instance number is zero for a scalar, and is equal to the index for a table entry. We will look at SNMP tables in detail later. The instance numbers for table entries start from the value 1.

2 Procedure

1. The `snmpget` command can be used to retrieve data from a remote host given its host name, authentication information and an OID. As a simple example:

```
$ snmpget -v 1 -c public nicku.org system.sysUpTime.0
```

In the above example, `nicku.org` is the host name we wanted to talk to, using the SNMP community string `public` and we requested the value of the OID `system.sysUpTime.0`, using SNMP protocol version 1.

2. We used SNMPv1 here. SNMPv2c, which is similar in nature to SNMPv1 with small modifications, still uses clear-text community names as “passwords” to authenticate the sender of the command. The result from a command using the SNMPv2c version is the same:

```
$ snmpget -v 2c -c public nicku.org system.sysUpTime.0
```

3. All of the utilities allow abbreviation of the OIDs and do random searches by default, (like `snmptranslate` with its `-IR` option), and so if you like, you can specify a small part of the OID:

```
$ snmpget -v 2c -c public nicku.org sysUpTime.0
```

4. A common mistake when using the `snmpget` command is to leave off the *instance number* of the data you’re looking for. See section 1.2 on the previous page. In the above commands, the variable requested by the OID is a scalar and the instance number for scalars is always a simple ‘0’ (zero), hence the trailing ‘.0’ in all the oids above. If you had left it off, you would have gotten an error. Note that the errors differ slightly between SNMPv1 and SNMPv2c:

```
$ snmpget -v 1 -c public nicku.org sysUpTime
Error in packet
Reason: (noSuchName) There is no such variable name in this MIB.
Failed object: system.sysUpTime
```

```
$ snmpget -v 2c -c public nicku.org sysUpTime
system.sysUpTime = No Such Instance currently exists
```

5. Multiple variables can be retrieved in one transaction as well:

```
$ snmpget -v 2c -c public nicku.org sysUpTime.0 sysContact.0
```

6. Do you think that the tool `snmpget` functions in the role of an *agent* or in the role of a *manager*?

7. Now use `snmptranslate -Tp` and the documentation for the tools in the software package `net-snmp-utils` to determine the load average on `nicku.org`. Compare the readings you get with:

```
$ ssh ictlab uptime
10:01am up 14 days, 26 min, 9 users, load average: 0.29, 0.24, 0.16
```

Hint: try piping the output of `snmptranslate -Tp` through `grep -i load`. The option `-C5` to `grep` may help. This prints 5 lines before and after the matching line.