3.9 Test Tools

Three programs in the Development Kit distribution are for testing and exercising individual components of the SNMP library. They are coded to run in a BSD 4.3 UNIX environment.

Sources for these commands reside in the `avltest`, `asntest`, and `smptest` subdirectories of the Development Kit distribution hierarchy. The shell scripts that reside in the `scripts` subdirectory are useful ways of invoking the smptest and asntest tools.

3.8 The Snmptrapd Command

The snmptrapd command listens for SNMP trap messages arriving from the network and displays them as they arrive. It is coded to run in a BSD 4.3 UNIX environment.

The code for the snmptrapd command illustrates the use of the Development Kit to accelerate development of network management applications as well as the use of the SNMP protocol *trap* operation.

Sources for the snmptrapd command reside in the `snmptrapd` subdirectory of the Development Kit distribution hierarchy.

It is invoked with the syntax:

snmptrapd [**-p** *remotePort*] [**-c** *communityName*]

If the **-p** flag is present, then the program will listen for trap messages at the UDP port specified as *remotePort,* instead of that assigned to the "snmptrap" service in the `/etc/services` database.

If the **-c** flag is present, then the program will generate and accept trap messages associated with the community name specified as *communityName,* instead of using the community name "public."

```
Integer
Counter
Guage
OctetString
ObjectId
IPAddr
TimeTicks
```

3.7 The Snmptrap Command

The snmptrap command directs an SNMP trap message at a specified remote network management application. It is coded to run in a BSD 4.3 UNIX environment.

The code for the snmptrap command illustrates the use of the Development Kit to accelerate development of network management applications as well as the use of the SNMP protocol *trap* operation.

Sources for the snmptrap command reside in the `snmptrap` subdirectory of the Development Kit distribution hierarchy.

It is invoked with the syntax:

snmptrap [**-h** *remoteHost*] [**-p** *remotePort*] [**-c** *communityName*] *enterprise agent-address generic-trap specific-trap timestamp* [*name kind value*] ...

If the **-h** flag is present, then the program will send its trap message to the IP address specified as *remoteHost.*

If the **-p** flag is present, then the program will send its trap message to the UDP port specified as *remotePort,* instead of that assigned to the "snmptrap" service in the `/etc/services` database.

If the **-c** flag is present, then the program will generate and accept management requests associated with the community name specified as *communityName,* instead of using the community name "public."

For example, the command

snmptrap -h anyhost 1.3.6.1.4.1.42.1.1 18.26.0.115 4 0 999 1.3.6.1.2.1.1.1.0 OctetString "Fred Router by Flintstones Inc"

has the effect of sending to a management application at host *anyhost* an SNMP trap message. The originator of the message is identified by its enterprise value, 1.3.6.1.4.1.42.1.1 and its IP address, 18.26.0.115. The message is an authenticationFailure trap, type 4. The time associated with this trap message is 999 time ticks, and a standard MIB sysDescr value has been included as additional information.

The kinds of values that are recognized as additional items in a trap message are

ues for the host named *anyhost*. The former is altered to the `OctetString` value "hello," whereas the latter is altered to the `ObjectId` value 1.2.3.4.5.6.7.

The kinds of values recognized are those enumerated in RFC 1065:

```
Integer
Counter
Guage
OctetString
ObjectId
IPAddr
TimeTicks
```

3.6 The Snmpset Command

The snmpset command initiates a network management request to remote management agent and displays the response. It is coded to run in a BSD 4.3 UNIX environment.

The code for the snmpset command illustrates the use of the Development Kit to accelerate development of network management applications as well as the use of the SNMP protocol *set* operation.

Sources for the snmpset command reside in the `snmpset` subdirectory of the Development Kit distribution hierarchy.

The snmpset command attempts to alter the items of management information named according to the conventions of RFC 1067 to have values specified by the user. It is invoked with the syntax:

snmpset [**-h** *remoteHost*] [**-p** *remotePort*] [**-c** *communityName*] [**-t** *timeout*] [**-i** *requestId*] [*name kind value*] ...

If the **-h** flag is present, then the program will send its management request to the IP address specified as *remoteHost.*

If the **-p** flag is present, then the program will send its management request to the UDP port specified as *remotePort,* instead of that assigned to the "snmp" service in the `/etc/services` database.

If the **-c** flag is present, then the program will generate and accept management requests associated with the community name specified as *communityName,* instead of using the community name "public."

If the **-i** flag is present, then the program will identify its management request by the number specified as *requestId,* instead of using the value zero.

If the **-t** flag is present, then the program will terminate after the number of seconds specified as *timeout,* instead of waiting forever for a response from the remote agent.

For example, the command

snmpset -h anyhost 1.3.6.1.2.1.1.1.0 OctetString "hello" 1.3.6.1.2.1.1.2.0 ObjectId 1.2.3.4.5.6.7

has the (hypothetical) effect of altering the `sysDescr` and `sysObjectId` val-

3.5 The Snmpnext Command

The snmpnext command attempts to retrieve and display specified subtrees of the MIB. It is coded to run in a BSD 4.3 UNIX environment.

The code for the snmpnext command illustrates the use of the Development Kit to accelerate development of network management applications as well as the use of the SNMP protocol *next* operation.

Sources for the snmpnext command reside in the `snmpnext` subdirectory of the Development Kit distribution hierarchy.

It is invoked with the syntax:

snmpnext [**-h** *remoteHost*] [**-p** *remotePort*] [**-c** *communityName*] [**-t** *timeout*] [**-i** *requestId*] [*name*] ...

If the **-h** flag is present, then the program will send its management request to the IP address specified as *remoteHost.*

If the **-p** flag is present, then the program will send its management request to the UDP port specified as *remotePort,* instead of that assigned to the "snmp" service in the `/etc/services` database.

If the **-c** flag is present, then the program will generate and accept management requests associated with the community name specified as *communityName,* instead of using the community name "public."

If the **-i** flag is present, then the program will identify its first management request by the number specified as *requestId,* instead of using the value zero.

If the **-t** flag is present, then the program will terminate after the number of seconds specified as *timeout,* instead of waiting forever for a response from the remote agent.

For example, the command

snmpnext -h anyhost 1.3.6.1.2.1.4.21.1

has the effect of retrieving the IP routing table for the host named *anyhost.* The command

snmpnext -h anyhost 1.3.6.1.2.1.5

has the effect of retrieving the all ICMP protocol statistics for the host named *anyhost.*

3.4 The Snmpget Command

The snmpget command initiates a network management query to remote management agent and displays the response. It is coded to run in a BSD 4.3 UNIX environment.

The code for the snmpget command illustrates the use of the Development Kit to accelerate development of network management applications as well as the use of the SNMP protocol *get* operation.

Sources for the snmpget command reside in the `snmpget` subdirectory of the Development Kit distribution hierarchy.

The snmpget command attempts to retrieve and display the items of management information named on the command line according to the conventions of RFC 1067. It is invoked with the syntax:

snmpget [**-h** *remoteHost*] [**-p** *remotePort*] [**-c** *communityName*] [**-t** *timeout*] [**-i** *requestId*] [*name*] ...

If the **-h** flag is present, then the program will send its management request to the IP address specified as *remoteHost.*

If the **-p** flag is present, then the program will send its management request to the UDP port specified as *remotePort,* instead of that assigned to the "snmp" service in the `/etc/services` database.

If the **-c** flag is present, then the program will generate and accept management requests associated with the community name specified as *communityName,* instead of using the community name "public."

If the **-i** flag is present, then the program will identify its management request by the number specified as *requestId,* instead of using the value zero.

If the **-t** flag is present, then the program will terminate after the number of seconds specified as *timeout,* instead of waiting forever for a response from the remote agent.

For example, the command

snmpget -h anyhost 1.3.6.1.2.1.1.1.0 1.3.6.1.2.1.1.2.0

has the effect of displaying the `sysDescr` and `sysObjectId` values for the host named *anyhost.*

3.3 The Snmpd Command

The snmpd command realizes the function of a network management agent in a BSD 4.3 UNIX environment. The code for the snmpd command illustrates the use of the Development Kit facilities to accelerate development of an agent that can service network management requests. The set of management information supported by this code does not fully conform to RFC 1066, and, accordingly, this application should not be deployed for any but experimental purposes.

Sources for the snmpd command reside in the `snmpd` subdirectory of the Development Kit distribution hierarchy.

The snmpd command is invoked with the syntax:

snmpd [**-h** *localHost*] [**-p** *localPort*] [**-c** *communityName*]

If the **-h** flag is present, then the daemon will attempt to listen for management requests at the IP address specified as *localHost,* instead of merely binding to the wildcard address.

If the **-p** flag is present, then the daemon will attempt to listen for management requests at the UDP port specified as *localPort,* instead of that assigned to the "snmp" service in the `/etc/services` database.

If the **-c** flag is present, then the daemon will accept management requests associated with the community name specified as *communityName,* instead of using the community name "public."

3.2 The BSD Library

The BSD library realizes some of the support necessary to construct network management applications in a BSD 4.3 UNIX environment. Sources for the BSD library reside in the `bsd` subdirectory of the Development Kit distribution hierarchy.

One sort of support afforded by the BSD library is provision of the relatively trivial transport services required for the operation of the SNMP protocol.

Another sort of support afforded by the BSD library is that of accumulating from the operating environment those items of management information identified in RFC 1066 as part of the Internet standard MIB. The BSD library code that realizes this function is provided only to illustrate how the Development Kit can be used to accelerate the construction of network management agents. The particular items of management information addressed in the BSD library code are selected for their value in illustrating some aspect of using the Development Kit; no claims are made that the MIB subset supported by this code is one particularly well-suited to effective network management. Nor should the strategies by which this code extracts management information from the UNIX environment be regarded as necessarily optimal.

A network management agent based exclusively on the support of the Development Kit BSD library in its current form is not fully compliant with RFC 1066, and, accordingly, should not be deployed for purposes other than experimentation.

and generation of the SNMP protocol response is provided by the SNMP library. Insertion, removal, and retrieval of objects in the MIB namespace is realized by algorithms of logarithmic complexity. All management information types identified in RFC 1065 are supported.

The SNMP library also includes limited support for developing human interfaces to network management applications in the form of functions that translate between ASN.1-derived values and human-readable text.

The use of the SNMP library is illustrated by the application programs comprised by the Development Kit distribution.

3.1 The SNMP Library

The SNMP library is a body of software from which network management applications may be quickly constructed with a minimum of additional code. As well as support for building user-oriented, SNMP client applications, significant support is afforded for rapid construction of SNMP agents to which management queries may be directed.

Sources for SNMP library code reside in the `snmp` subdirectory of the distribution directory hierarchy. The code is cast entirely in terms of abstract types defined in `ctypes.h`, so that the object code produced from the library source may be easily tuned to suit the hardware data types and instruction set of the target machine.

The services that the library requires of its run-time environment are entirely represented by the seven functions listed in file `local.h`:

```
char    *       malloc ();
int             free ();
int             bzero ();
int             bcopy ();
int             strcmp ();
char    *       strcpy ();
int             strlen ();
```

The interface and semantics expected by the library for these functions is that attributed to them in the BSD 4.3 *UNIX Programmer's Manual Reference Guide.* These functions are familiar to many C language programmers and are easily implemented in environments other than UNIX.

The SNMP library provides for generation and recognition of those ASN.1 data types required for SNMP network management. Consistent with the robustness principle of protocol engineering, the generation of ASN.1 values complies fully with RFC 1067, while the recognition of ASN.1 values received from the network is less restrictive.

The library provides facilities whereby data structures in user-written code may be conveniently exported as MIB objects for interrogation and manipulation by remote network management applications. Support for processing of SNMP protocol requests, retrieval of the appropriate exported MIB items,

3 SNMP Development Kit Overview

The SNMP Development Kit is a body of software designed to make the fabrication of network management applications as easy as possible; it is not a complete suite of ready-to-use network management tools.

A few, simplistic management tools are provided, and these are not entirely without use. However, their function is more one of illustrating the use of the Development Kit libraries and verifying its behavior: they represent none of the sophistication of the applications that the Development Kit is designed to support.

C Language source code is provided for all components of the Development Kit. That code compiles and runs successfully on DEC MicroVAX [2] machines running BSD 4.3 UNIX. However, the library that realizes the greatest part of the Development Kit functionality depends exclusively on materials supplied in the distribution, and so it will most likely compile in any environment.

The behavior of applications built from the Development Kit is believed in good faith to conform to RFC 1067. Applications built from the Development Kit successfully interoperate with at least two other, independent SNMP implementations.

The several parts of the SNMP Development Kit are described below.

[2] *DEC* and *MicroVAX* are trademarks of Digital Equipment Corporation.

2 SNMP Development Kit Goals

The foremost goal of the SNMP Development Kit is to foster serious as well as playful exploration of current and novel techniques for network management. This kit is intended as an aid for those who seek practical experience in both the use of network management information and the engineering issues involved in its acquisition and manipulation.

The design of the SNMP Development Kit is consistent with its goal:

- The SNMP Development Kit code is designed to be relatively portable in order that use of the package not be restricted to particular hardware or software environments.
- The SNMP Development Kit code is designed to be highly modular in order that individual components of the package may be easily refined, modified, or reused to construct novel applications.
- The SNMP Development Kit code is designed around sharply defined interfaces, so that use of the package or any of its components requires no detailed knowledge of its internals.
- The SNMP Development Kit is deliberately and flagrantly unoptimized in order that neither the scope of its application nor its value as an example implementation is diminished by parochial or obscure coding strategies.

While the deliberate inattention to performance issues implicit in the Development Kit code distinguishes it from more finely crafted, commercial SNMP implementations, this difference increases its value as a common reference point from which (long overdue) discussion of management protocol performance can proceed.

snmp-dk@allspice.lcs.mit.edu

are welcome and may be answered as time permits.

1 SNMP Development Kit Distribution

The SNMP Development Kit is produced in the Advanced Network Architecture group at the MIT Laboratory for Computer Science. The copyright for all materials in this distribution is held by the Institute. They are made available for use by others according to the following terms:

Copyright 1988, 1989,
Massachusetts Institute of Technology

Permission to use, copy, modify, and distribute this software for any purpose and without fee is hereby granted, provided that this copyright and permission notice appear on all copies and supporting documentation, the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the program without specific prior permission, and notice be given in supporting documentation that copying and distribution is by permission of M.I.T. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

The SNMP Development Kit distribution is available via anonymous FTP in the form of a UNIX [1] tar file named

`pub/snmp/snmp.tar`

on host

allspice.lcs.mit.edu

The SNMP Development Kit is distributed with absolutely NO warranty or support. However, bug fixes, bug reports, questions, comments, or flames pertaining to the Development Kit sent via electronic mail to

[1] *UNIX* is a trademark of AT&T Bell Laboratories.

The SNMP Development Kit

Release Notes

James R. Davin

Laboratory for Computer Science
Massachusetts Institute of Technology
1 January 1989