tation can either happen at installation time or during runtime. As an example, consider an SNMP agent running on a Linux system. Some capabilities of this agent may depend on the configuration options used by the system administrator when installing the agent. Other capabilities may depend on the presence of certain operating system features which are determined during runtime.

As noted before, agent capabilities specifications are relatively static. Thus, the question is whether an agent capabilities statement indicates exactly what an agent actually supports at any moment or whether an agent capabilities statement defines the maximally possible support. Either interpretation has its problems. The result is that agent capabilities statements are useful in some but not necessarily all environments.

What should MIB compilers do with agent capabilities specifications?

First, lets define the term MIB compiler. The term is simple, but there seem to be many interpretations. A MIB compiler should validate that a collection of MIB modules is syntactically valid, and check that it is semantically valid as far as possible that can be done by a computer program. (Note that much of the semantics associated with a MIB module are specified in text for humans and not for computer programs.) After validation, a MIB compiler generates output in another format that is directly usable or is input to another program. For example, a MIB compiler could create data structure definitions and code stubs to assist in creating an SNMP agent. Note that the data structures and stub code for one agent implementation would probably be of little use for other agent implementations. Alternatively, a MIB compiler could output information that assists a program that graphs values of objects. There are many uses.

So what should a MIB compiler do with an agent capabilities specification? Of course, a MIB compiler should validate agent capabilities specifications like all other SMIv2 constructs. Unfortunately, many MIB compilers "skip over" agent capabilities specifications without any checking to see if they are valid. They do this because implementors could not figure out what to do with the agent capabilities specifications. Other MIB compilers try to use agent capabilities specifications to drive how data structures and stub code is generated for agents. My experience and belief is the opposite. That is, the control mechanisms that are used to specify the implementation characteristics of an agent and used to drive a code generator, should be used to drive the creation of agent capabilities specifications.

What should a MIB browser do with agent capabilities specifications?

A MIB browser is a program that allows a user to examine and change the values of MIB objects. It should also allow the creation of new instances of objects. It is possible for a MIB browser to give hints to its user about which objects can be accessed based on information from an agent capabilities specification. However, the benefit seems to be little. I am not aware of any MIB browser that uses agent capabilities specifications.

I have not seen any agent capabilities specifications from vendors, why not?

There are three primary reasons why equipment vendors do not ship agent capabilities specifications with their devices:

1. It is difficult to get agent implementors to completely document their implementation. It is "extra" work to do after they have finished. And it is often almost impossible to create agent capabilities specifications for old agents without re-engineering the agent implementation.

2. The product marketing managers do not like to publish specifications that appear to look like bug lists, even when an agent conforms to compliance specifications.

3. Since the agent capabilities are identified by OID values, it is difficult for users to find the appropriate MIB modules. One of the most common requests on the SNMP news group is for MIB modules from vendors that define a given OID value.

Book Reviews

The reviews published in this column represent the opinion of the author(s). Please contact the author(s) directly if you want to share your comments. Please contact the editors of *The Simple Times* if you are interested to publish your own book review in this column.

Red Hat Linux Network Management Tools
Reviewed by Aiko Pras, University of Twente

The book starts with the usual explanation of TCP/IP and SNMP. Although the book was published in April 2000, the explanation focuses on SNMPv1 and spends only a few words on SNMPv3. This is a missed opportunity, since the main tools discussed in the book already

support SNMPv3 and many readers may have questions related to that version.

After the introduction the book continues with some of the core Linux utilities and tools, such as `arp`, `ifconfig`, `netstat`, `ping`, `tcpdump` and `traceroute`. For each of these tools there are about 10 pages of text; this text may be quite useful for many readers. The following section discusses additional system utilities and tools, such as `arpwatch`, `ethereal`, `fping`, `nmap` and `xtraceroute`. Again, the text is useful, although some of the examples could be shorter without loss of information.

The remaining two third of the book focuses on SNMP. First there is a discussion of 80 pages on MIB-II; unfortunately this discussion is very similar to the text of RFC1213 and similar information is already available from many other sources. After MIB-II, the book continues with a discussion of UCD SNMP and SUN's Solstice Enterprise Agents. Although many readers may find the UCD-SNMP text valuable, it is unclear why a book with the title "Red Hat Linux Network Management Tools" discusses tools for a Solaris platform. Fortunately, the total text on UCD-SNMP is much larger than the text on the SUN agent.

After the discussion on agents, the book continues with some of the most popular Web-enabled tools: `mrtg` and `ntop`. Although there is already many documentation on the Web related to these two packages, the book is still interesting to read. The last chapters of the book concentrate on the Linux `control panel` and on `scotty`'s `tkined`. It is unclear why the discussion on the Linux `control panel` was postponed until the end of the book; it would have been more logical to discuss it earlier. The chapter on `tkined` is quite useful; there are about 50 pages of information which can not easily be found elsewhere. It is not clear, however, why the `Tnm` part of the `scotty` package has not been discussed. It would have been a good replacement for the sections on the SUN agent.

The book is surely a valuable source of information for people relatively new to Linux based network management. What is absolutely missing, however, are references to sources elsewhere. Although all packages that were discussed are contained on the two CDs that accompany the book, the book should at least have included the URLs from where the readers can download the latest versions of these packages. In fact, except for a few examples like UCD-SNMP, the book does not even mention the source and authors of these packages (the publisher even claims that "the software and information on the diskette are the property of The McGraw-Hill Companies"!).

Standards Summary

Please consult the latest version of *Internet Official Protocol Standards*. As of this writing, the latest version is RFC 2700.

SMIv1 Data Definition Language

Full Standards:

- RFC 1155 - Structure of Management Information
- RFC 1212 - Concise MIB Definitions

Informational:

- RFC 1215 - A Convention for Defining Traps

SMIv2 Data Definition Language

Full Standards:

- RFC 2578 - Structure of Management Information
- RFC 2579 - Textual Conventions
- RFC 2580 - Conformance Statements

SNMPv1 Protocol

Full Standards:

- RFC 1157 - Simple Network Management Protocol

Proposed Standards:

- RFC 1418 - SNMP over OSI
- RFC 1419 - SNMP over AppleTalk
- RFC 1420 - SNMP over IPX

SNMPv2 Protocol

Draft Standards:

- RFC 1905 - Protocol Operations for SNMPv2
- RFC 1906 - Transport Mappings for SNMPv2
- RFC 1907 - MIB for SNMPv2

Experimental:

- RFC 1901 - Community-based SNMPv2
- RFC 1909 - Administrative Infrastructure
- RFC 1910 - User-based Security Model