The Simple Times™

THE QUARTERLY NEWSLETTER OF SNMP TECHNOLOGY, COMMENT, AND EVENTS
VOLUME 9, NUMBER 1 DECEMBER, 2001

The Simple Times is an openly-available publication devoted to the promotion of the Simple Network Management Protocol. In each issue, *The Simple Times* presents technical articles and featured columns, along with a standards summary and a list of Internet resources. In addition, some issues contain summaries of recent publications and upcoming events.

In this Issue:

The Simple Times is available as an online journal in HTML, PDF and PostScript. New issues are announced via an electronic mailing list. For information on subscriptions, see page 20.

Editorial

Aiko Pras, University of Twente
Jürgen Schönwälder, University of Osnabrück

The year 2001 has been one of the more silent years in the SNMP history. But sometimes silence is a good thing, especially if it is not caused by a lack of activity. Quite some work has been done in 2001 to progress core technology documents and to complete or revise several MIB modules.

Work on SNMPv3 is reaching completion as the specifications are currently progressing through the IESG approval process to become a full Internet Standard. The publication of the SNMPv3 specifications as Internet Standards is expected to happen in 2002 and it might go hand in hand with an action to classify SNMPv1 as Historic. Of course, such an IESG action does not immediately impact all the widely deployed SNMPv1 implementations. SNMPv1 implementations will stay with us for many more years. But still, classifying SNMPv1 Historic sets a clear signal that SNMPv3 has become the stable SNMP version of the future and people are safe in deploying SNMPv3 implementations in large scales.

The IETF standard for extensible agents (RFC 2741, RFC 2742) has been elevated to Draft Standard in December 2001. The AgentX specifications are some of the few documents that manage to progress in the IETF standardization process without any changes to the RFCs. This is even more impressive since the working group collected 13 implementation and interoperability reports.

The MIB module for Differentiated Services, another milestone in the set of IETF MIBs, has been approved for publication as Proposed Standard in November 2001. Work on this MIB involved the cooperation of several subject matter experts since the automated configuration of routers supporting the differentiated services architecture is frequently used to demonstrate policy-based configuration management schemes.

It is good to see that the core SNMP technology has reached stability again after a lengthy period of several competing versions in the 1990s. The SMIv2 data definition language is a full Internet Standard since 1999 and it is expected that the SNMPv3 protocol is published as a full Internet Standard in 2002. This gives the industry a stable technology to create new products and network operators a stable base for their future deployment decisions. On the other hand, every successful living technology also needs a controlled evolutionary path forward to address new requirements and to adapt to changes in the environment. It is thus not surprising that the IETF has started two new working groups:

- The *Evolution of SNMP* working group (EOS) focuses on the evolution of the protocol operations.
- The *Next Generation Structure of Management Information* working group (SMIng) focuses on the evolution of the data definition language.

The SMIng working group has produced a document which discusses the objectives of the SMIng work. This document was published as RFC 3216 in December.

SNMP Set: Can it be saved?

Andy Bierman, Cisco Systems

There are many factors preventing the SNMP command responder application interface from displacing the command line interface (CLI) as the primary configuration mechanism for network devices. Some of the issues have been raised on the SNMP WG mailing list in the past, but were largely ignored at the time. Not all of the issues are directly related to SNMP or the SMI, and not surprisingly, the most important factor is money. SNMP agent code costs too much to develop, test, and maintain, compared to the equivalent CLI code. The complex SNMP `set` state machine and the overhead of lexicographic sorting are the worst problem areas, usually much more (five to ten times?) expensive to develop than CLI code. In order for SNMP to succeed as a configuration management interface, development costs must be competitive with the CLI, while offering significantly more value to developers than the CLI. This should be possible, since the CLI is primarily a human interface and does not provide many important features of a programmatic interface, such as stability, detailed command semantics, or compliance and conformance information.

The problem starts with the API definition itself – the SMIv2 information module. CLI specifications are easier to define than MIBs because there are not nearly as many documentation requirements and modification rules. SMIv2 syntax is not straightforward, widely understood, or easy to learn. Furthermore, SNMP has no real transaction semantics, which means input and output parameters to commands are modelled as data elements (i.e. MIB objects).

So why are SNMP configuration objects so expensive to develop compared to the equivalent CLI configuration commands? Simply put, the CLI accepts input at an appropriate granularity (one command at a time) while SNMP accepts input at an inappropriate granularity (one parameter at a time). This applies to row creation as well as row modification.

SNMP `set` PDUs may contain an arbitrary number of (potentially unrelated) arbitrarily partial commands, and the agent is expected to accept individual parameters, not complete commands. The partial “commands-in-progress” have to be processed “best-effort”, as if an entire set of parameters existed, and then saved as MIB objects, so these parameters can be retrieved by an NMS in subsequent read operations.

The agent cannot simply store partial commands until they are complete, because some parameters have “act now” semantics and others have “act on activation” semantics (e.g. `bufferControlMaxOctetsRequested` from the RMON-MIB (RFC 2819) is an “act-now” parameter while `bufferControlChannelIndex` is an “act-on-activation” parameter, both defined in the same table).

The requirement to accept partial input is most complicated if some parameters are inter-related, which is usually the case. The `set` PDU logic for such MIB objects can be quite complicated, as well as the selection of appropriate default values for missing parameters. For example, if an agent supporting the `DISMAN-PING-MIB` (RFC 2925) received a `set` PDU containing only one varbind for a `pingCtlTargetAddressType` object (syntax `InetAddressType`) set to `dns(16)`, then the agent has to select an arbitrary value for the `pingCtlTargetAddress` object, which is supposed to be a DNS name.

One work-around for this problem is for the MIB designer to place as many essential parameters as possible in the `INDEX` clause. This is problematic from a MIB design perspective, since `OBJECT IDENTIFIER`s are limited to 128 sub-identifiers and the `INDEX` clause implies uniqueness across all its parameters. From an agent implementation perspective, this technique simply trades `set` PDU complexity for lexicographic sorting complexity. For an management system developer, editing an existing row becomes more complex because parameters in the `INDEX` clause cannot simply be changed, but rather the entire row must be deleted and re-created with a new index value. This can sometimes cause disruptive behavior on the device as well.

The CLI allows only one command at a time to be input, instead of an arbitrary mix of commands. If essential parameters are missing, the entire input is rejected as a syntax error, rather than accepted as partial input. These two simple restrictions make the development and testing of CLI code relatively trivial,