The Simple Times™

THE BI-MONTHLY NEWSLETTER OF SNMP TECHNOLOGY, COMMENT, AND EVENTS℠
VOLUME 2, NUMBER 2 MARCH/APRIL, 1993

The Simple Times is an openly-available publication devoted to the promotion of the Simple Network Management Protocol (SNMP). In each issue, ***The Simple Times*** presents: a refereed technical article, an industry comment, and several featured columns. In addition, some issues include brief announcements, summaries of recent publications, and an activities calendar. For information on submissions, see page 16.

In this Issue:

Technology and Commentary
Technical Article 1
Industry Comment 5

Featured Columns
Applications and Directions 5
Ask Dr. SNMP 6
Security and Protocols 7
Standards 9
Working Group Synopses 11

Miscellany
Forthcoming Publications 15
Activities Calendar 15

Publication Information **16**

The Simple Times is openly-available. You are free to copy, distribute, or cite its contents. However, any use must credit both the contributor and ***The Simple Times***. (Note that any trademarks appearing herein are the property of their respective owners.) Further, this publication is distributed on an "as is" basis, without warranty. Neither the publisher nor any contributor shall have any liability to any person or entity with respect to any liability, loss, or damage caused or alleged to be caused, directly or indirectly, by the information contained in ***The Simple Times***.

The Simple Times is available via both electronic-mail and hard-copy. For information on subscriptions, see page 16.

Technical Article

Michael L. Kornegay, VisiSoft

In this issue: *Toward Useful — and Standardized — SNMP Management Applications*

Three documents in the Request for Comments series (RFCs 1155, 1157, and 1212) describe and specify the original Internet-standard Network Management Framework, which is now referred to as SNMPv1. These documents define a communication protocol, a data encoding method, and a specification language for defining information that may be exchanged between systems. The framework allows vendors to provide an interoperable way for diverse systems to exchange information using simple primitive operations.

SNMP version 2 (SNMPv2) will be a significant, but incremental, improvement of original framework. Notable enhancements include: improved error handling and bulk retrieval features, a security framework, compliance definitions, and Manager-to-Manager functionality.

Unfortunately, these enhancements do not address all of the issues needed to provide a network management technology which allows for both interoperable — and useful — interaction between management applications and agents. This article discusses the state of today's management applications, presents a dream for the "fully generic" management applications of tomorrow, and then considers what must be in place to approach that dream.

Current Management Applications

The current state of the art in SNMP management applications is discouraging. Applications are either highly vendor-specific, or generic-but-simple.

Vendor-specific applications do a good job managing a specific vendor's network entities. They display management information in useful ways, allow control of the entity using the SNMP `set` operation, and may even assist the user in performing true network management. They do not interoperate well with MIB modules defined by other organizations. If the network has a diverse mix of network elements, different applications are usually

necessary for different devices, thereby losing many of the advantages of integrated network management.

In contrast, generic applications attempt to provide tools the user can configure to gather management information from diverse MIB modules. There are two primary forms of these generic applications: simple MIB browsers and configurable monitoring tools. Both are very limited in what they can accomplish since the really important information about the objects in a MIB module is not available in machine-understandable form.

A MIB browser simply collects data from network entities and displays it for the user. As Bob Stewart has noted:

> "The MIB browser helps because it gives you one place to be confused instead of many. It concentrates but doesn't integrate, reduce, or interpret."

The term "MIB browser" is also commonly used in a derogatory fashion when referring to limited management applications.

A configurable monitoring tool can be programmed to periodically collect user-configured MIB information, possibly perform computational manipulations on the retrieved information, and generate user-defined alarms based on the retrieved data and/or the manipulation of that retrieved data. These alarms could be based either on simple thresholds or on unconstrained assertions, such as an *if* statement in a programming language.

In addition, several network management development platforms are available that allow users to either develop desired applications in-house or purchase applications that are compatible with that platform. This approach results in systems similar to either the vendor-specific or generic-but-simple applications discussed above.

Fully generic Management Application dream?

There are those who naively believe that a fully generic management application is both possible and a reasonable expectation. Such an application could be given a new MIB module, and then automatically — without any further intervention on the part of the user — be able to fully manage a device which implements that module.

Unfortunately, this has been viewed as an "all or nothing" situation, leading some groups down the garden path of unending specification, while giving other groups an excuse not to fall into a bottomless pit. However, there can be a compromise between the generic-but-simple management applications and the dream of fully-generic management applications, and this is where effort should be focused. Although it's not clear how far we can "push the edge of the envelope", where to start is obvious.

Management Application's Needs

There are many possible areas of work to support a framework for management applications. Perhaps the most important are MIB specifications and monitoring strategies. These two issues are at the heart of the usefulness problem discussed earlier, and SNMPv2 only begins to address these issues. There are other issues which also need to be considered; these are briefly examined after we look at these two key issues.

Needed: MIB Specification

MIB specifications include the documents and/or MIB modules that define a specific area of management information.

Although there is a huge number of MIB objects available via SNMP, it is not clear how useful many of these really are. Fred Baker has noted:

> "What I have gotten frustrated with in the past is `Counters` and `Gauges` — health checks and statistical summaries — which go in because they 'might be interesting' or 'could be useful'. What the MIBs SHOULD contain is useful information..."

Specification should include fully documenting individual MIB objects, and relationships between MIB objects. Currently the `OBJECT-TYPE` macro is the method for documenting individual MIB objects, and there is no formal method for documenting relationships, although sometimes ASN.1 comments are used. An interesting case of this is the ASN.1 comments in the SNMPv2 MIB that include Case diagrams, pictorial information which shows the arithmetic relationship between the objects defined in that MIB module. The problem here is that not enough information is included, and much of the information included is natural language-based and, as such, cannot be parsed and understood by management applications.

It should be noted that these efforts have no effect at all on agent implementations, except to possibly allow the MIBs to be more easily and quickly implemented since they are more thoroughly documented and do not contain unjustified or extraneous objects. However, the use of this information will enable management applications and MIBs to better interoperate.

The `OBJECT-TYPE` macro is the mechanism used to document an individual MIB object. The original `OBJECT-TYPE` macro, defined in RFC 1155, provides

very little machine-understandable information about a object, e.g.,

```
ifEntry OBJECT-TYPE
    SYNTAX  IfEntry
    ACCESS  read-write
    STATUS  mandatory
    ::= { ifTable 1 }
```

As a result, any "interesting" information ended up in the textual commentary of the object's definition,

SNMPv1's OBJECT-TYPE macro was enhanced by RFC 1212 to convey more information in some new clauses, i.e., DESCRIPTION, REFERENCE, INDEX, and DEFVAL. The DESCRIPTION clause

> "...provides all semantic definitions necessary for implementation..."

The values of the DESCRIPTION and REFERENCE clauses are textual strings, which limit their automatic use by management applications. (The textual commentary defined separately from the original macro has just been moved to the DESCRIPTION clause.) Unfortunately, the DESCRIPTION clause is the clause of last resort where anything that is not able to be documented in other clauses ends up, e.g.,

```
ifEntry OBJECT-TYPE
    SYNTAX  IfEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
       "An interface entry containing
        objects at the subnetwork layer
        and below for a particular
        interface."
    INDEX   { ifIndex }
    ::= { ifTable 1 }
```

SNMPv2's OBJECT-TYPE macro was enhanced to include an optional UNITS clause, along with an indication as to whether a collection of objects defines a table (INDEX) or an extension to a table (AUGMENTS). The AUGMENTS clause will hopefully result in fewer enterprise-specific MIB definitions since they can now augment a standard table with their own specific columns.

Although RFC 1212 was published two years ago, very few enterprise MIBs make useof RFC 1212's OBJECT-TYPE macro. Some felt that with all the work put into many new macros in SNMPv2, an effort to get additional machine-understandable information in the OBJECT-TYPE macro, as well as the other new macros, was desirable. Not all agreed though and, as such, the evolution of the SNMPv2 macros will continue along the path taken by SNMPv1.

What should be added to an OBJECT-TYPE macro that helps support management applications? This is a difficult question, although many informal suggestions have been made. These include ranges, inter-object relationships, justification for inclusion and definition, management strategy recommendations, action parameterization, algorithm descriptions, assertions, help text, and so on.

However, either extending the OBJECT-TYPE macro or defining new macros will be quite a challenge. For example, how can we represent object relationships or assertions using ASN.1? Regardless, the main goal should be to avoid the reliance on textual clauses which are fine for humans but undecipherable by machines.

Needed: Monitoring Strategies

Monitoring strategies include techniques used to gather management information from network entities, possibly including the integration, reduction, and interpretation of that data. How MIB objects are monitored is important for several reasons, including performance, bandwidth, localization, access, data reduction, and so on.

The original monitoring strategy for SNMP is termed *trap-directed polling*, as demonstrated in the RMON MIB. However, periodic polling is currently the most common practice.

With the wide deployment of manageable devices, more distributed techniques are becoming necessary. Some use the term *delegation-based management* to refer to these distributed techniques.

The SNMPv2 Manager-to-Manager MIB is a simple example of this and an excellent beginning toward delegation-based SNMP management. An intermediate-level SNMP entity, acting in both a manager and agent role, monitors MIB objects as configured in its Manager-to-Manager MIB. When an object violates a threshold, this entity informs another management entity using SNMPv2's `inform` PDU. The advantage is that the central or top-level management applications do not have to retrieve and process all the data to participate in the management of these remotely monitored devices.

The work put into the Manager-to-Manager MIB should be extended — the threshold and delta comparisons are often too limiting. Consider the possibility of adding a third category to `snmpAlarmSampleType`, called `assertionTest`, in addition to the currently defined `absoluteValue` and `deltaValue` categories. An assertion would be a boolean expression that would declare a relationship between the components of an expression. There may be one assertion that affirms what being in an alarm state is, or two assertions, one that causes an alarm condition to be generated, and one that causes the alarm condition to be cleared.

Delegated monitoring also needs to have the capability to deal with historic data. Historic data may consist of multiple actual samples of a MIB object, or multiple smoothed samples of a MIB object. Having historic samples is useful for accounting-type applications. In addition, having smoothed historic data is useful for defining alarms based on deviation from a smoothed norm. Accounting-type applications may want historic entries for every sample, while smoothing techniques may require entries for discrete time-slots each day.

The use of delegation-based management can significantly improve the monitoring capabilities of management applications.

Needed: Standardization Efforts

It must be emphasized that standardization efforts are needed to ensure a market large enough to attract a significant number of management application developers. To move forward and provide a framework for interoperable management applications, standards activities are required in many areas. For the following suggestions, assume that SNMPv2 is the baseline.

First, the historic presumption to avoid unnecessarily burdening the *primitive* agent should be continued, although intermediate-level SNMP entities acting in a dual-role may become quite complex. Many of the specification issues, thought by many to be the most important issues, will have no effect at all on the primitive agents or the framework; however, they will have an effect on MIB specification efforts, whether standardized or vendor-specific.

The framework provides several ASN.1 macros that help specify the definition and semantics of management information. The problem is that these macros rely too much on textual clauses that management applications cannot understand, and therefore can't make use of. Further, these limitations allow MIB designers to design MIB modules which are inadequately documented — thereby placing an unnecessary burden on management applications. The author believes the historic practice of simplifying agents at the expense of adding complexity to management applications should be used as a pattern for a new practice: management applications should be simplified at the expense of adding complexity to the *definition* of managed objects. MIB designers need to be *forced* to do what they have not been able to *voluntarily* do: properly and completely document managed objects.

The framework defines the Manager-to-Manager MIB which is a start toward distributed delegation-based management, in contrast to the trap-directed polling model encouraged by SNMPv1. The definition of new MIBs supporting distributed delegation-based management are very important issues to the future of SNMP.

Management applications need to dynamically access the MIB modules that describe the MIB objects available in the agent. Defining a MIB that allows the definition of objects to be retrieved from the *non-primitive* agents, such as the dual-role delegated managers discussed earlier, could be a useful tool.

Some believe that an Application Programmer Interface (API) for SNMP on different platforms should be standardized, both for use by management applications and "extensible" agents. The IETF has avoided this issue since it considers this problem a "local matter". However, there is an industry working group that is defining an SNMP API for Microsoft Windows. (Send a note to `winsnmp-request@microdyne.com` to be added to the group's mailing list.) The group's previous effort, an event-driven sockets interface to Microsoft Windows, has been a success. Both of these efforts could and should be considered for adoption in other operating system environments.

User interface information includes user-oriented labels, formatting instructions, help text, icon definition, front-panel bitmaps, and so on. The framework does not address this sort of information. Some believe that the MIB module specification should contain this sort of information, while others believe that new macros, which reference object definitions, should be used to convey this information. Still some believe the graphical interface itself should be standardized, possibly by defining user interface guidelines similar to those defined for popular Graphical User Interface (GUI) environments.

Conclusions

The implementation and adoption of the SNMP framework has been extremely successful, as evidenced by its widespread deployment.

However, to achieve truly interoperable network management capabilities, we need to develop standards in the areas discussed above, and provide a framework for management applications, not just agents. These standards are important in order to ensure that the market is large enough to attract a significant number of application developers and to ensure that there is a framework they can rely on.

The SNMP community needs to get more management application developers involved in the IETF process. We need to accept that standardization of issues related to management applications is not a "local" issue; it is an interoperability issue.

If you are interested in these issues, be sure to join the SNMP general discussion list (see the *Working Group Synopses* column for subscription information).

Industry Comment

Marshall T. Rose

Due to external events (the INTEROP Spring conference and the next IETF meeting) this is a short issue. So once again I have an excuse for "no comment".

Applications and Directions

Steven L. Waldbusser

In this issue: *Hints on coexistence and transition from SNMP to SNMPv2*

Many networking vendors have been considering how their SNMPv2 products will interact with SNMPv1 products, and customers are beginning to explore how they will integrate SNMPv2 products amongst their SNMPv1 infrastructure. This article will explore some of the important issues in this area.

Thankfully, the SNMP framework is largely unchanged in SNMPv2. While many new features have been added, few paradigms have shifted, enabling applications to use SNMPv2 with a minimum of change. In fact, except for security parameters, SNMPv1 applications can run on top of the SNMPv2 protocol without modification. These applications can then be modified as appropriate to take advantage of SNMPv2 features, but without impact on the coexistence plan. (It is worth noting that coexistence is largely unsuccessful when dealing with two dissimilar frameworks such as SNMP and CMIP).

First, it is important to characterize the environment into which we must integrate SNMPv2. There are many SNMPv1 products fielded, both management stations and agents. Agents often reside in embedded systems where software upgrades may be unavailable or difficult to install. Management stations typically reside on workstations or PCs and are fairly easy to update with new software. It is also important to note that agents are much more numerous and diverse than management stations. For these reasons, it is sensible to plan to upgrade the management stations to SNMPv2 first, while providing coexistence with SNMPv1 agents which will be assumed to transition more slowly.

Proxy Agent

There are two transition mechanisms mentioned in the SNMPv2 coexistence document: proxy agents and bi-lingual managers. The first of these mechanisms entails the use of proxy agents to translate SNMPv2 packets from management stations into SNMPv1 packets to be sent to agents, and SNMPv1 replies from agents into SNMPv2 packets to be sent to management stations. The rules for translation in the proxy agent are exceedingly simple and consist of three trivial translation rules.

Proxy agents add complexity that is visible to a network manager. When faced with lack of response from an agent, a network manager using a proxy agent will need to determine if the proxy agent is running, rather than being able to simply trust the response from the management tool. The proxy agent also adds a level of hierarchy to the configuration of the network management system that adds complexity. If product developers need to use a proxy agent solution, they should work hard to shield this complexity from the user so that their tool is easily trusted.

Bi-lingual Manager

The bi-lingual manager solution is implemented by providing a management station with the capability to send either SNMPv1 or SNMPv2 packets. The choice of protocol is a local configuration matter on the management station, and will typically be chosen on a per-device basis.

The management station could be configured by hand to specify which protocol to use for each host, but a mechanism to automatically configure this would be necessary for all but the smallest environments. This mechanism would probably be implemented as part of the auto-discovery process, typically by recording which of the two protocols can elicit a response to a simple `get` request (e.g., an unauthenticated request for `sysObjectID`).

This bi-lingual manager approach is nearly seamless to the network management user. Most operations are performed in exactly the same way regardless of which protocol is being used. Those hosts that are configured to use SNMPv2 will perform management operations faster and have some additional features available to them.

Other than security parameters, SNMPv1 applications will not need to be modified to run on top of SNMPv2. However, in order to enable the additional features of the SNMPv2 protocol, these applications will need to be modified. For example, a routing table management application may wish to use the `get-bulk` PDU to transfer routing tables more quickly. When designing an SNMPv2-enabled application, it would be helpful to consult the rules for proxy agent behavior for ideas on how to retain coexistence with SNMPv1. For example, these rules describe how an application can translate a `get-bulk` PDU into a `get-next` PDU for transmission to an SNMPv1 agent. It will be helpful if future SNMP API's and libraries provide this translating function automatically so that applications are easier to write.

Bi-lingual Agent

A third mechanism that will be used is the bi-lingual agent. This is implemented by allowing an agent to respond to either SNMPv1 or SNMPv2 requests. It is fairly simple to implement and does not add a lot of cost to the agent. However, this approach is not explicitly part of the transition plan, as it places an additional burden on the agent.

It is expected that bi-lingual agents will nonetheless become prevalent due to their ease of implementation, simplicity in installation and configuration, and low additional cost. Product developers will often decide that the additional cost of implementing both protocols is worth the advantage of retaining backwards compatibility with older management stations. Unless facing code-size constraints, most vendors will choose to go above and beyond the call of duty and ship their systems with both protocols enabled.

Although upgrading an SNMPv1 agent to SNMPv2 is straight-forward, it is important for product developers and systems integrators to understand the security implications of implementing both protocols. Since SNMPv1 is non-secure, care must be taken that no access is given to objects via SNMPv1 that require authentication or privacy with SNMPv2. This would be like locking the doors to a convertible! SNMP agents should help prevent misconfiguration by only providing public information to SNMPv1.

Conclusions

Due to the evolutionary nature of the SNMPv2 changes, it is simple to plan for coexistence. Between bi-lingual managers and proxy agents, the transition plan provides a solution for any environment. Bi-lingual agents will also be prevalent and will make the transition even easier for network management users.

Ask Dr. SNMP

Jeffrey D. Case

Dear *Dr. SNMP*,
The business wire carried an interesting story this morning — it seems that a vendor has introduced "the only network management system supporting the full RMON Token Ring Standard." Have I missed something? To the best of my knowledge, the TR-RMON effort is still underway in the IETF, and there is no document on the standards-track.

I know IEEE takes a stand against "misuse" of its standards, but is anyone in the IETF chartered to answer these kinds of issues? We try to educate our sales people and customers about the IETF standards process and the true status of RMON MIB and other standards-track protocols, but it's tough to fight blarney like this! Any suggestions or help you can give would be much appreciated.

— *Steamed in Santa Clara*

Dear *Steamed in Santa Clara*,
Down on the farm, we have a saying:

> "If you think fishermen are the biggest liars in the world, just ask a jogger how far he runs in the morning."

I share your frustration. It would seem that the only difference between used car salesmen and people who write network management press releases is that the used car salesmen know that they are lying. I hope the cause is ignorance rather than callous disregard for the truth.

Unfortunately, there doesn't appear to be a good way to police this kind of distortion of the truth. The only suggestion I can make is that the readers of ***The Simple Times*** should encourage the editor to start a new section in this publication which awards a prize for the most flagrant lack of truth-in-advertising. There is a dairy farmer across the road who, along with his bulls, surely is able to furnish the appropriate prize material to be sent to each issue's winner(s).

In the meantime, perhaps the company who published the errant story should publish an appropriate correction and apology.

Dear *Dr. SNMP*,
Even though a system object, `sysServices` describes what protocol layers are implemented in a device, MIB-II also mandates objects at the IP layer. How, then, should the agent residing on a MAC bridge respond to requests for IP layer objects?

— *Wondering in West Perth*

Dear *Wondering in West Perth*,
Down on the farm, we have a saying:

> "Right's right and right don't wrong nobody."

The right response from the agent would return the values for the requested objects. The general rule of thumb is that if the system supports a particular layered protocol, then the protocol should be instrumented, and the network management information should then be available via the SNMP. In your example, if the bridge supports IP, then the agent should respond to queries for objects in the IP layer from properly authenticated and authorized network management stations with the

appropriate values. The rules are similar for the other layers such as ICMP, and UDP. In many protocol transparent systems, such as your example of MAC bridges, IP, ICMP, UDP and ARP will be present in support of network management. It is appropriate for these layers to be manageable even though the traffic through these layers might be limited to SNMP packets. It is expected that the SNMP will increasingly be used with other transport protocols, e.g., IPX or DDP, rather than UDP. If in those cases UDP is not present, it is unnecessary and inappropriate for the agent to support the UDP layer objects. In those cases, the agent should support the appropriate objects for the layers which are present.

Dear *Dr. SNMP*,
The people who make our management application software claim that the names used in our proprietary MIBs should be unique, not only within one MIB, but also between MIBs; i.e., it is illegal to use the name "state" for two different objects in two different MIBs. The solution should be to use a unique prefix for all object names within a MIB. However, this vendor cannot reference any RFC where this requirement is stated.

Is this true? If so, could you please point me to the RFC stating this? And if so, then how are the name prefixes allocated in order to ensure global uniqueness?

— *Debilitated in Denmark*

Dear *Debilitated in Denmark*,
Down on the farm, we have a saying:

> "Believe nothing of what you hear and only half of what you see."

In the Internet-standard Network Management Framework, the Structure of Management Information (SMI) defines the mechanisms used for describing and naming objects for the purpose of management. According to that document (RFC 1155), objects are uniquely and unambiguously named by `OBJECT IDENTIFIER`s. The `OBJECT IDENTIFIER` associated with a particular MIB object is directly related to its position in the global naming tree, and thereby also directly related to the authority which defined it. This yields globally unique names, since no single position in the global tree can be occupied by more than one MIB object, i.e., MIB object names are unique when specified in `OBJECT IDENTIFIER` form.

In addition, each object in the MIB is also given a textual name, called a descriptor. This "user-friendly" string is a mnemonic, printable string which promotes a common language for humans to use when discussing the MIB and also facilitates simple table mappings for user interfaces. However, these descriptors are unique only within the Internet-standard MIB — they are not guaranteed to be unique in other sections of the MIB, i.e., across the various proprietary MIBs.

What this means is that it is not illegal according to the SMI to use the name "state" for two different objects in two different MIB documents. (Please note the subtle deviation from your text here. *Dr. SNMP* believes there is only one MIB, even though it is defined in many MIB documents.) However, it may still be illegal according to a "higher" authority, such as a vendor's representative.

Now, having said all that, *Dr. SNMP* must return to pragmatics. In the state where I learned to drive an automobile, pedestrians always have the right-of-way. Consequently, if one chooses to step in front of a fast-moving large truck, then the driver MUST stop his or her vehicle. If they fail to stop and hit you, since you have the right-of-way, you would be in the right. You would be dead, but you would be right.

The situation is often similar in network management. You can insist on having different objects in different MIB documents having the same descriptor. You'd be right. However, if management stations cannot accommodate that (and many cannot), then you'd also be dead.

Security and Protocols

Keith McCloghrie

SNMP version 2 incorporates the work on SNMP security which was published last summer, but with some changes. Agreement on the last set of these changes, in the IETF's SNMP Security working group, was reached at the beginning of this year; these aspects were the last part of SNMPv2 to become stable. In this issue, we'll look at this final set of changes.

The Introduction of Context

One issue raised during the IETF working group's deliberations was the so-called "party proliferation" problem. This problem occurred since SNMP parties (at that time) identified not only particular entities and security properties but also a local MIB view or a particular proxy relationship. Thus, multiple parties with the same security properties were required for each manager with access to multiple views/proxy relationships. Each such additional party increased the chore of initializing and maintaining the party clocks and secrets.

To avoid this, SNMPv2 separates the identification needed for security from the identification of the management information context. A SNMPv2 party identifies only a particular entity and its security properties. For

the other identification needed, SNMPv2 introduces the notion of a *context*.

A SNMPv2 context identifies the context in which the names of specific management variables are to be interpreted. For example, if one of the variables named in a SNMPv2 PDU is the MIB-II variable, `sysUpTime.0`, then the specification of a context determines which `sysUpTime.0` is referenced. So, in order to interpret the variables named in a SNMPv2 PDU, a context is specified in the message "wrapper" around the PDU. This specification of a context is in addition to the specification of a message's source and destination parties which now identify just the sending and receiving SNMP entities and their security properties.

Three kinds of contexts are defined: a local MIB view; a local entity's MIB view; and, a proxy relationship. Simple agents will only have contexts of the first kind, with one context for each of their defined MIB views. Agents which provide access to the management information of multiple devices (e.g., a single agent for multiple bridges or an agent for multiple repeaters) will also have contexts of the second kind, identifying the bridge, repeater, or other entity to which specific MIB objects refer. Contexts of the third kind will be used by proxy agents to identify the real agent to which proxy requests are to be forwarded. (In fact, such a context specifies the source party, destination party, and context to be used in forwarding the proxy request to the real agent.)

Access Control

Access control in SNMPv2 determines which entities have what kind of access to which management information and specifies the level of security at which that access must take place. With the introduction of context, authorization is now based on the combination of the source party, the destination party, and the context specified in a message's wrapper. That is, a particular combination of source party, destination party and context is authorized to use particular PDU types. This authorization information is stored as access control entries in an SNMP entity's MIB (in the `aclTable`).

Identifying Temporal Semantics

As well as identifying a particular MIB view or proxy relationship, a SNMPv2 context also identifies the "temporal" semantics of the referenced MIB objects. For the present, three distinctions in temporal semantics are defined:

- the current values of referenced MIB objects;
- the values at the next re-initialization/reboot of the agent; and,
- "cached" values of referenced MIB objects.

We normally think of SNMP requests as requests for the current values. However, some systems have parameters which are best (or can only be) changed at the next system restart (e.g., the system's IP address). When a SNMP set operation changes such a value, the new value is stored in non-volatile storage until the next restart. With SNMPv1, there was no totally satisfactory answer to the question of whether the response to a subsequent retrieval of the changed parameter returned the current value or the new value. Different vendors had solved this in different ways. With the inclusion of temporal semantics as part of the definition of a context, this ambiguity is removed.

The idea behind "cached" values is to allow an agent which maintains its management information in a cache, to know how old the management information requested by a retrieval operation can be. For example, if the context of a retrieval request has temporal semantics of `cachedTime.30`, then the values returned must be no more than 30 seconds old.

In contrast to other aspects of SNMPv2, there is little or no implementation experience with the use of temporal semantics other than `currentTime`. However, a context's temporal semantics are identified using `OBJECT IDENTIFIER`s. Thus, if experience proves their worth, other types of temporal semantics can be added in the future, as required.

Identification of Contexts

Contexts are identified in the same way as parties, i.e., by globally unique `OBJECT IDENTIFIER`s. Also in a similar manner to parties, "initial context identifiers" are defined within a specific branch of the `OBJECT IDENTIFIER` tree, by using the address (e.g., the IP address) of the agent. By this means, an agent can automatically configure itself at installation time with a default set of parties and contexts with appropriate definitions and privileges.

In addition to solving the party proliferation problem, the introduction of context also has a layering advantage which will allow the network operator's interface on a management station to be simplified. Specifically, operators (or applications) can select the object resources that they want to manage (i.e., a context) and indicate whatever communication requirements they have (i.e., authentication and privacy), and the next layer down selects the parties, and so on. This means that users can now ignore parties and deal solely with contexts. For

management stations trying to hide complexity, this is a big win.

The Use of DES Becomes Optional

Another of the final set of changes was the removal of any need for using DES in the creation and maintenance of the security aspects of an SNMPv2 agent. An earlier change had specified that DES was no longer required for the changing of party secrets. With one further change, which allows new parties to be created using SNMPv2 without the use of DES, the implementation of DES becomes optional. This is achieved through having the secrets of new parties be initialized as copies of the secrets of an existing party. These copies must be changed before use, in order to avoid a security loophole, but that can also be done without using DES.

DES is still useful. For example, if a SNMPv2 `set` operation is used to change a user's password in a terminal server, then the use of DES to encrypt the SNMP message prevents eavesdroppers from obtaining the password through inspection of the message. So, some agents may still wish to implement DES. However, those agents for which the restrictions on exporting DES is a problem, no longer need to support it.

Reduced requirement for NV Storage

Finally, another of the final set of changes was the addition of *storage type* as a property of each party, context, MIB view, and access control entry. Three values of storage type are defined: `non-volatile`, `volatile`, and, `permanent`. Parties, contexts, views, and access control entries of type `non-volatile` can be created and deleted, and are retained across reboots; thus, they must be kept in NV storage. Those of type `volatile` can also be created and deleted, but disappear at the next reboot of the agent; thus, they do not need to be kept in NV storage. Those of type `permanent` cannot be created or deleted, and are typically stored in ROM.

The use of volatile and permanent storage types provides for a significant reduction in the amount of non-volatile storage needed to implement SNMPv2 by agents with limited resources.

Standards

David T. Perkins

In March, the SNMP over CLTS, SNMP over DDP, and SNMP over IPX documents were finally published as RFCs. They allow network elements to be managed with SNMP, if they are in networks which are based on OSI, AppleTalk, or Netware, It should be noted that if a network element supports multiple protocol stacks, then UDP is the preferred transport protocol to use.

Several new SNMP documents are currently in the pipeline for publication as RFCs. The ones that are most anticipated are those defining the second version of the Internet-standard Network Management Framework, SNMPv2. Most likely, they will be published the first week in April (right after the publication of this issue of ***The Simple Times***).

Recently Published RFCs

RFC 1418 - SNMP over OSI (Proposed Standard)
This document defines the mappings so that SNMP can be run over OSI's connectionless-mode transport service (CLTS). It replaces an earlier experimental version defined in RFC 1283, and previously RFC 1161.

RFC 1419 - SNMP over AppleTalk (Proposed Standard)
This document defines the mappings so that SNMP can be run over AppleTalk's DDP.

RFC 1420 - SNMP over IPX (Proposed Standard)
This document defines the mappings so that SNMP can be run over Novell's IPX. It replaces an earlier informational version defined in RFC 1298.

Profile of a Leader of MIB Development

Who are the authors (or editors) of the SNMP MIBs in the standards-track? A quick listing below shows the number of MIBs for each person:

- McCloghrie: 1213, 1229, 1230, 1231, 1286, 1353, 1368
- Baker: 1253, 1315, 1354, 1381, 1389, 1406
- Stewart: 1316, 1317, 1318
- Cox: 1304, 1407
- Fox: 1230, 1231
- Rose: 1213, 1414
- Tesink: 1304, 1407
- Throop: 1381, 1382
- Waldbusser: 1243, 1271
- Brown: 1315
- Carvalho: 1315
- Case: 1285
- Coltun: 1253

- Davin: 1353
- Decker: 1286
- Galvin: 1353
- Kastenholz: 1398
- Langille: 1286
- Malkin: 1389
- McMaster: 1368
- Rijsinghani: 1286
- Saperia: 1289
- St. Johns: 1414
- Watt: 1406
- Willis: 1269

The person with the second highest count may be a surprise to some. Fred Baker, a relative newcomer to the SNMP community has, since his participation starting in 1990, authored, co-authored, or edited six SNMP MIB documents. Fred has kept a low profile, but has pushed hard enough to get those MIBs out. Starting from a position at Control Data Corporation working on communications technology connecting mainframes, Fred next moved to Vitalink Communications. In the years that he was there, from 1983 to 1990, working on bridging and multi-protocol bridge/router technology, Fred was awarded two patents. From there, Fred moved to ACC where he continues to work on bridging and routing. The MIBs he has authored or co-authored reflect these areas.

According to Fred,

> "You're successful (as a working group chair) if you produce an output that solves the intended problems and the WG members still like each other when they're done. That often means a lot of behind the scenes work, pinging the authors to do what they volunteered to do, holding private conversations with different factions to help them find common ground, etc."

One characteristic that assists Fred in his success is his broad understanding and experience in each technology in which he is working. This means both implementing the networking protocols (i.e., writing the code), and deploying the products for real users to get a feel for the design of the MIB for managing the protocols. From this implementation and deployment experience, Fred looks for common ground with other engineers when putting together a MIB proposal for standardization. For him,

> "There is often more than one 'right' solution, and the way to get the best one is to figure out what problem you're trying to solve before you solve it, get all the possible solutions on the table, and then make the best choice you can with the constraints you've got. You've got to find common ground, perhaps create common ground."

Fred sees finding candidates for leadership positions as a process of looking for "folks who exhibit maturity" and have some amount of "prior success" in other endeavors. Those who are "narrow-minded" or "given to flame wars" would not be recommended by Fred. Another important characteristic that Fred recommends for IETF leaders is demonstrated success in leading their families at home.

In closing out the interview with Fred, I asked him if he had any suggestions for advancing the current MIBs along the standards-track. He drew a blank on specific suggestions. He noted that the current process seems to have too many delays built into it. Any change that accelerated the process would be appreciated.

As of this writing, the position of area director for the network management area is open. Given Fred's comments and the huge list of outstanding MIBs at the proposed standard level, it looks like the best candidate for the area director position would be someone who is well-organized to track in parallel the status of all the MIBs and one who can persuade companies to work together to interoperate their implementations and share their deployment experience. Looking down the road, this same set of skills is needed to get SNMPv2 moving along the standards-track.

In the next issue, all of the SNMPv2 documents will be described and a road map given to navigate through the 12 documents and over 400 pages of specifications.

Summary of Standards

Full Standards:

- 1155 - Structure of Management Information (SMI);
- 1157 - Simple Network Management Protocol (SNMP);
- 1212 - Concise MIB definitions; and,
- 1213 - Management Information Base (MIB-II).

Draft Standards:

- 1398 - Ether-Like Interface Type MIB.

Proposed Standards:

- 1229 - Extensions to the generic-interface MIB;

- 1230 - IEEE 802.4 Token Bus Interface Type MIB;
- 1231 - IEEE 802.5 Token Ring Interface Type MIB;
- 1239 - Reassignment of experimental MIBs to standard MIBs;
- 1243 - AppleTalk MIB;
- 1253 - OSPF version 2 MIB;
- 1269 - BGP version 3 MIB;
- 1271 - Remote LAN Monitoring MIB;
- 1285 - FDDI Interface Type MIB;
- 1286 - Bridge MIB;
- 1289 - DECnet phase IV MIB;
- 1304 - SMDS Interface Protocol (SIP) Interface Type MIB;
- 1315 - Frame Relay DTE Interface Type MIB;
- 1316 - Character Device MIB;
- 1317 - RS-232 Interface Type MIB;
- 1318 - Parallel Printer Interface Type MIB;
- 1351 - SNMP Administrative Model;
- 1352 - SNMP Security Protocols;
- 1353 - SNMP Party MIB;
- 1354 - SNMP IP Forwarding Table MIB;
- 1368 - IEEE 802.3 Repeater MIB;
- 1381 - X.25 LAPB MIB;
- 1382 - X.25 PLP MIB;
- 1389 - RIPv2 MIB;
- 1406 - DS1/E1 Interface Type MIB;
- 1407 - DS3/E3 Interface Type MIB;
- 1414 - Identification MIB;
- 1418 - SNMP over OSI;
- 1419 - SNMP over AppleTalk; and,
- 1420 - SNMP over IPX.

Experimental:

- 1187 - Bulk table retrieval with the SNMP;
- 1224 - Techniques for managing asynchronously generated alerts;
- 1227 - SNMP MUX protocol and MIB;
- 1228 - SNMP Distributed Program Interface (SNMP-DPI); and,
- 1238 - CLNS MIB.

Informational:

- 1147 - A network management tool catalog;
- 1215 - A convention for defining traps for use with the SNMP;
- 1270 - SNMP communication services;
- 1303 - A convention for describing SNMP-based agents; and,
- 1321 - MD5 message-digest algorithm.

Historical:

- 1156 - Management Information Base (MIB-I)
- 1161 - SNMP over OSI;
- 1232 - DS1 Interface Type MIB;
- 1233 - DS3 Interface Type MIB;
- 1283 - SNMP over OSI;
- 1284 - Ether-Like Interface Type; and,
- 1298 - SNMP over IPX.

Working Group Synopses

Frank J. Kastenholz

This column is a summary of activities. There is no substitute for actually participating in a working group. Even if you cannot go to the meetings, you can subscribe to the mailing lists. Included in each working group's summary is the address of the group's mailing list. To subscribe, simply append "`-request`" on to the local-part of the address. For example, the submission address for the SNMP general discussion list is

```
snmp@psi.net
```

so to subscribe, you'd send a message to

```
snmp-request@psi.net
```

If you are interested in a group's activities and do not subscribe to the mailing list, you should!

SNMP General Discussion

Submissions: snmp@psi.net

Someone asked whether the syntax

```
INDEX { INTEGER }
```

was valid. This syntax is allowed by RFC 1212; however, this does not convey as much information as, e.g.,

```
INDEX { ifIndex }
```

therefore, use of this syntax is discouraged. (Note, however, that in SNMPv2, this usage is not allowed.)

A brief discussion was held as to the best way to implement a "ping" MIB. One commercial implementation was examined, which overloaded the instance identification of a MIB object with all of the parameters needed to perform the ping. Others suggested including all of the parameters in separate entries in a table.

One person asked if there was a terminal server MIB, and was directed to RFCs 1316, 1317, and 1318.

A plea was made for putting useful values in the system group objects. The message pointed out that "Rhett Butler" is not a useful value for sysContact, and "Hometown, USA" in sysLocation is meaningless. One response suggested that NMSs can't rely on these fields since they are not guaranteed to be accurate, and that the telephone was the preferred medium to deal with the problem; further, since this information can't be relied on, there is no incentive to configure the fields correctly! This columnist finds such arguments to be silly — people really should configure things properly.

A question was asked concerning the use, meaning, and content of the enterprise field of the Trap-PDU. After much discussion, all agreed that this field represents either the authority under which the trap was defined, or the identity of whatever is issuing the trap; however, this field does not necessarily have any relationship to the enterprises branch in the MIB tree.

A question was asked about how to process a get-request which asks for several variables, when one or more of the variables are not supported. The agent must return a response, identical to the request, except that the error-status is set to noSuchName and the error-index indicates which variable is not supported. The question also evoked many examples of incorrect behavior in existing implementations. This, in turn, evoked more calls for clearer specifications and more testing.

A question was asked regarding how to represent floating point numbers. This might be a problem since there is no such data type in the SMI. Some suggestions included encapsulating them directly in an OCTET STRING, or as text in a DisplayString. Another suggestion was to change the units being reported (e.g., instead of reporting a time value in seconds and then having to figure out how to represent .02 seconds, change the units to milliseconds and return 20 for .02 seconds).

Someone asked how a TimeTicks value that has the high-bit set should be encoded. It must be encoded in 5 octets, where the first octet is zero-valued. Once again, a discussion ensued which pointed out that many people do this wrong; this was followed by the usual calls for more testing.

A question of how to deal with multiple logical interfaces per physical interface (and vice-versa) was raised. It was pointed out that this is not handled well by the current interfaces group and the issue should be addressed in the future.

A question was asked of how to handle a PDU with no variable bindings, assuming all other parts of the PDU are correct. The correct answer is to simply return an error-status of noError since no error was detected in processing the PDU. Once again, this brought out a call for clearer specifications.

An announcement was posted that Bellcore is in the process of defining: an ATM-based PVC service, and an associated Customer Network Management Service, both likely to depend on SNMP. A birds-of-a-feather meeting has been scheduled for the Columbus IETF meeting.

A person asked what variables to include in the authenticationFailure trap and how to report which node sent the mechanism that caused the failure. The answer to the first question is that RFC 1157 says no variable bindings need be present; the answer to the second question is that proprietary MIB variables can be used to convey this information.

Someone asked whether it is possible to, in a single set-request, set variables defined in different MIB modules or groups. The answer is yes.

Someone asked how X.25 addresses are represented in the PhysAddress textual convention. The address should be in binary coded decimal.

Appletalk/IP Working Group

Submissions: apple-ip@cayman.com

A change to the address encodings for SNMPv2 over DDP was made. This was also noted on the SNMPv2 mailing list.

A question on the meaning of the phrase "the configuration status of this port" in the DESCRIPTION clause of the atportNetConfig object in RFC 1243 was posted. There were no responses.

BGP Working Group

Submissions: iwg@ans.net

No traffic to report.

Bridge MIB Working Group

Submissions: bridge-mib@decwrl.dec.com

There was a brief discussion as to whether the dot1dTpPortInFrames and dot1dTpPortOutFrames objects should include "bridge management frames" (the MIB does count these frames). The discussion started with the point that the IEEE 802.1d specification indicates that these counts are to be used to help calculate the forwarding rates of the bridge. A response indicated that the intent has always been to include all frames destined to the bridge, rather than to the forwarding process. The assumption is that the number of management frames is negligible compared to the total traffic.

Character MIB Working Group

Submissions: char-mib@decwrl.dec.com

No traffic to report.

Chassis MIB Working Group

Submissions: chassismib@cs.utk.edu

A question was posted how to represent a built-in agent in the chasSlotTable. One suggested solution was to represent the agent as a "virtual" slot, specifically, chasNumSlots+1. Another suggestion was to change the slot table indexing with a type and location rather than just a slot number.

A discussion of how to represent per-slot environmental sensors ensued. The MIB currently contains only chassis-wide sensors. A proposal was made to extend the chasEnvironTable to include a slot index, allowing both per-slot and chassis-wide sensors. (Slot 0 would refer to the entire chassis).

A new Internet-Draft of the Chassis MIB was posted.

DECnet Phase IV MIB Working Group

Submissions: phiv-mib@jove.pa.dec.com

No traffic to report.

Ethernet MIB Working Group

Submissions: enet_mib@ftp.com

Some errors in the ASN.1 and the descriptive text were pointed out. These errors will probably be fixed when the MIB is reviewed for promotion to full-Internet Standard.

There is a specific problem that readers should be aware of: specifically, the comments associated with dot3TestTdr contain a typo — they should reference ifExtnsTestCode, not ifExtnsTestResult.

FDDI MIB Working Group

Submissions: fddi-mib@cs.utk.edu

There was an extensive discussion on mechanisms to map FDDI MACs to the interfaces group. A number of problems were pointed out. No solution was apparent from the mailing list traffic. This thread also digressed into a general "ghost resources not present" discussion. The latter discussion was successfully concluded by adding hardware-present indications to the MAC and PORT groups. Thus, when something is not present, there will be an indication of that in the MIB.

A list of the changes to the MIB for SMT v7.3 was posted. There were some small changes and clarifications suggested and adopted.

Host MIB Working Group

Submissions: hostmib@andrew.cmu.edu

Some minor editorial and technical glitches in the document were fixed.

Additions were proposed in the general area of printer status.

A suggestion was made to change hrProcessorIdle to show the amount of time that the process has been in idle state.

Some problems in hrSWRunID and hrSWRunType were pointed out. Others said that they were not problems and that you just had to work a little harder to get the information. A rejoinder said that not all agent systems support the underlying information and that this object does not meet the typical criteria for inclusion in a MIB. The discussion went round and round, raising issues of unambiguous identification of software, registration at installation, classification of software. There was no clear resolution of the issue.

Hub MIB Working Group

Submissions: hubmib@synoptics.com

The minutes of the meeting at the Washington DC IETF meeting were published.

Discussion started on the MAU MIB. The main items were rpMauJabbers and rpMauLostMedias. These objects are not countable for all MAU types. So the description of the objects has been changed accordingly and their names have been changed to rpMauJabberStateChanges and rpMauMediaAvailableChanges, respectively.

There was discussion on what value the MIB object rptrAddrTrackLastSourceAddress should return when no source addresses had been seen. The consensus of the group seemed to be to report an address of all zeros to indicate the condition. An alternative, of reporting a zero-length string, is also being considered.

There was discussion of whether an HUB agent is connected to the network via a virtual/implicit repeater port. This is important in that it affects the number of repeater ports that are reported for a hub, either "N" or "N+1" (with the "+1" being an implicit repeater port). There was no clear resolution of the issue.

IDPR Working Group

Submissions: idpr-wg@bbn.com

A revised Internet-Draft of the IDPR MIB was posted.

IDRP for IP Working Group

Submissions: idrp-for-ip@merit.edu

No traffic to report.

IPLPDN Working Group

Submissions: iplpdn@nri.reston.va.us

Several bugs in the Frame Relay DTE MIB were pointed out and fixed. Some additional variables were also proposed. Most of them were rejected due to a lack of operational experience and being of no obvious utility.

IS-IS Working Group

Submissions: isis@merit.edu

No traffic to report.

NOCtools Working Group

Submissions: noctools@merit.edu

On January 11, 1993, an announcement was posted that the IESG had approved the latest NOCTools document for publication as an Informational RFC.

OSPF Working Group

Submissions: ospfigp@gated.cornell.edu

No traffic to report.

PPP Working Group

Submissions: ietf-ppp@ucdavis.edu

No traffic to report.

RIP Working Group

Submissions: ietf-rip@xylogics.com

No traffic to report.

Remote Monitoring (RMON) MIB Working Group

Submissions: rmonmib@jarthur.claremont.edu

A bug in the statement of the filter matching algorithm was pointed out.

A brief discussion on the maximum size of the token ring RIF field was held. The maximum is 32 bits, though some systems arbitrarily limit it to a smaller number (e.g., 18 bits).

The notInRingPoll object was deleted from the MIB as it is apparently not implementable in a reasonable fashion.

Questions were raised about the algorithms used to implement ringStationOrder. A simpler algorithm was proposed. Some problems with the simpler algorithm were pointed out. A counter-proposal was made to nuke the table entirely. There was no clear resolution of the issue.

A clarification was made on the net as to why the hostInPkts and hostInOctets counters exclude error packets and octets respectively while the hostOutPkts and hostOutOctets include error packets and octets, respectively (to track down which host is putting the errors on the network).

One person asked, for purposes of adding hosts to the host table, whether frames that were too short or long are to be considered "good" (hosts may be added only

as a result of "good" frames). These frames should be considered "bad".

TCP Client Identity Protocol

Submissions: ident@nri.reston.va.us

No traffic to report.

SNMP Security Working Group

Submissions: snmp-sec-dev@tis.com

A question was asked as to why initial contexts use IP addresses. The answer is that it is a convention that allows unique initial contexts to be algorithmically determined without requiring manual configuration.

The working group chairs, in conjunction with the SNMPv2 working group chair, issued a call for consensus on the documents. There were no objections.

SNMPv2 Working Group

Submissions: snmp2@thumper.bellcore.com

There was considerable discussion about process issues — whether the SNMPv2 effort caused important ideas to be squashed, whether SNMPv2 was forced down the throats of the community, and so on. In response, the IESG undertook an investigation of the SNMPv2 process, inviting comments from the community, which would be held in confidence by the IESG. At the conclusion of the investigation, the IESG exonerated the SNMPv2 process with two minor, obligatory faults found:

- it was suggested that the design team rejected suggestions too quickly — having already thought of the idea and then discarding it (some months earlier during the design process) — and that perhaps it would be better for members of the design team to wait a day before responding to a suggestion, regardless of the technical merit of the suggestion; and,
- the schedule was deemed to be overly aggressive — which, at every opportunity, everyone claimed they wanted.

This discussion, due to massive cross-posting, was carried out simultaneously on the SNMPv2, SNMP, and SNMP Security mailing lists. After the investigation concluded, on March 15, 1993, a last call was issued by the IESG.

This discussion also detoured into a thread of how NMS developers are supposed to learn how to use the various MIB modules which are defined. This discussion also digressed into the "we need real network management applications, not more MIB-browsers" theme. When someone claimed that very few of the companies producing NMS products can afford to have enough expertise in-house to develop these types of applications, this columnist decided that the discussion had become silly.

Trunk MIB Working Group

Submissions: trunk-mib@saffron.acc.com

Announcements were posted that RFCs 1406 and 1407 on managed objects for DS1/E1 and DS3/E3 interfaces, respectively, were published.

Although the working group has formally concluded, the mailing list will continue as a forum for implementors.

UPS MIB Working Group

Submissions: ups-mib@cs.utk.edu

The main topic of discussion was the battery group. A straw-man proposal was posted and discussion then turned to the individual objects in the group.

X.25 MIB Working Group

Submissions: x25mib@dg-rtp.dg.com

No traffic to report.

Forthcoming Publications

SNMP, SNMPv2, and CMIP: The Practical Guide to Network Management Standards
William Stallings, Addison-Wesley, 1993.
ISBN 0–201–63331–0 (to appear in April, 1993).

The Simple Book: An Introduction to Internet Management, 2nd edition
Marshall T. Rose, Prentice Hall, 1993.
ISBN 0-13-177254-6 (to appear in August, 1993).

Activities Calendar

- IFIP Symposium on Network Management
  April 18–23, San Francisco, CA
  For information: +1 415–512–1316

Publication Information

The Simple Times is published with a lot of help from the SNMP community.

Publication Staff

Coordinating Editor:	
Dr. Marshall T. Rose	Dover Beach Consulting, Inc.
Featured Columnists:	
Dr. Jeffrey D. Case	SNMP Research, Inc. University of Tennessee
Frank J. Kastenholz	FTP Software, Inc.
Keith McCloghrie	Hughes LAN Systems, Inc.
David T. Perkins	SynOptics Communications, Inc.
Steven L. Waldbusser	Carnegie Mellon University

Contact Information

Postal:	***The Simple Times*** c/o Dover Beach Consulting, Inc. 420 Whisman Court Mountain View, CA 94043–2186
Tel:	+1 415–968–1052
Fax:	+1 415–968–2510
E-mail:	`st-editorial@simple-times.org`
ISSN:	1060–6068

Submissions

The Simple Times solicits high-quality articles of technology and comment. Technical articles are refereed to ensure that the content is marketing-free. By definition, commentaries reflect opinion and, as such, are reviewed only to the extent required to ensure commonly-accepted publication norms.

The Simple Times also solicits terse announcements of products and services, publications, and events. These contributions are reviewed only to the extent required to ensure commonly-accepted publication norms.

Submissions are accepted only in electronic form. A submission consists of ASCII text. (Technical articles are also allowed to reference encapsulated PostScript figures.) Submissions may be sent to the contact address above, either via electronic-mail or via magnetic media (using either 8-mm `tar` tape, $\frac{1}{4}$-in `tar` cartridge-tape, or $3\frac{1}{2}$-in MS-DOS floppy-diskette).

Each submission must include the author's full name, title, affiliation, postal and electronic mail addresses, telephone, and fax numbers. Note that by initiating this process, the submitting party agrees to place the contribution into the public domain.

Subscriptions

The Simple Times is available via electronic-mail in three editions: *PostScript*, *MIME* (the multi-media 822 mail format), and *richtext* (a simple page description language). For more information, send a message to

```
st-subscriptions@simple-times.org
```

with a `Subject` line of

```
help
```

In addition, ***The Simple Times*** has numerous hard-copy distribution outlets. Contact your favorite SNMP vendor and see if they carry it. If not, contact the publisher and ask for a list. (Communications via e-mail or fax are preferred).