Key Criteria for Evaluating a
Portable Embedded SNMP Agent Solution

By DMH Software (http://dmhsoftware.com)

Executive Summary

As embedded real-time projects becomes more complex, portable SNMP solution has been widely adopted as a viable management solution in the data networking, telecom, transportation, government, industry automation and many other fields. SNMP is a mandatory requirement for many brand new projects in the government space, cable-TV, transportation, communication, and many other vertical fields. It is definitely a requirement for other embedded devices which must integrate with existing systems that already use SNMP.

Developing an SNMP agent solution with all its environment is not an easy task. Some of the established vendors that provide such a solution have invested many person-years to overcome the complexity of the protocol and the lexicographical presentation requirements. When developers need to make an SNMP agent solution selection they face strict requirements and constraints.

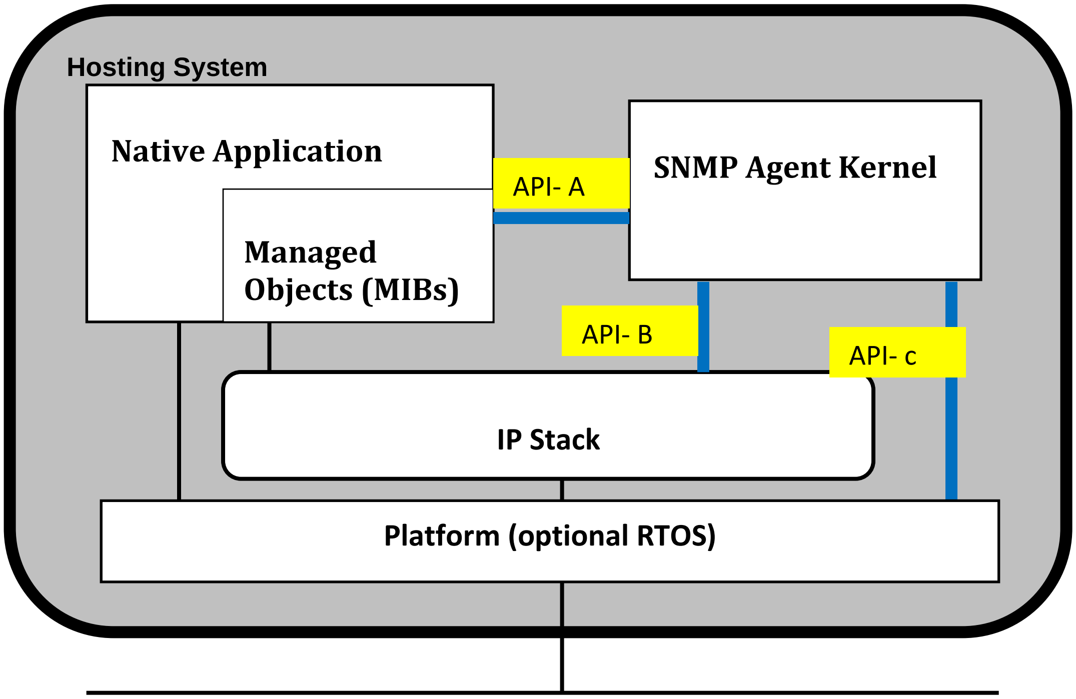
This paper discusses the key criteria for evaluating a portable embedded SNMP Agent solution. It explains the constraints imposed on SNMP Agent solution's operation by a real-life embedded environment; furthermore, it covers topics such as development, integration and maintenance of such a solution. It also discusses the key components of an effective portable SNMP Agent technology. This paper is not an in-depth SNMP technical presentation and the reader is encouraged to be familiar through sources such as Stallings (1999), DMH (2008) and other sources.

The following table summarizes the key selection criteria for a portable and embedded SNMP agent System. The columns' headers provide the key selection criteria while the rows' headers provide the key components in the evaluated System. The + sign in the table's cells indicates that a key product component influences a key selection criterion.

	Integration effort and TTM	MIBs Develop-ment effort	Platform & project Constraints	Upgrade and Support	Total Cost of ownership (TCO)
Completeness	+	+			+
Design	+	+		+	
Run-time performance/ Size			+		
Documentation	+	+		+	+
Platform support	+		+		+
Quality and inter-operability	+			+	
Product Upgrades				+	+

SNMP Solution Components

The following block diagram provides the architecture and basic elements of a portable SNMP Agent and consists of SNMP Agent Kernel, MIBs implementations, Native application, IP Stack, Platform, optional Real Time Operating System (RTOS), and various interfaces between the components. In addition, a MIB compiler is used during development time.



- Native application – the user/device application which can include several MIBs' implementations.
- IP Stack – an SNMP solution requires transport services (typically datagram) to receive and send SNMP messages via API B.
- Platform – the run-time environment which include the physical hardware and basic support system and optionally an RTOS. The platform provides basic services such as memory allocation to the SNMP kernel via API C.
- Managed object – an implementation of a MIB that manage the device. The MIB interfaces with the SNMP Kernel using API-A. The MIB can access the IP Stack and optionally and platform to perform its tasks.
- SNMP Agent kernel – this is the heart of SNMP Agent solution. It processes incoming requests and eventually interacts with the MIB modules to retrieve or modify the value of a requested object. The kernel interacts with the underlying transport (IP stack) subsystem and communicates with the SNMP manager.
- A MIB compiler is an essential component in any SNMP solution. The compiler translate Abstract Syntax Notation One (ASN.1, a set of formal rules for describing the structure of the managed objects that is shared between the SNMP manger and agent) into source code files

Evaluation of an SNMP Portable System

The evaluation team should consider the following criteria as a basis for a solution selection:

Integration effort and Time to Market (TTM)

The time to complete the project is composed of the SNMP learning curve, the kernel porting effort, MIBs development, testing integration with the rest of the system, quality and performance test/improvements. The following list of SNMP solution's product features influence on the TTM:

- Updated and complete code. The product should include the most recent RFC and/or DOCSIS standards; for example it should include support for Advanced Encryption System – AES (RFC 3826, 2004).
- Interoperability. The required SNMP solution should have been tested for interoperability with the most rigorous industry tests including DOCSIS; in addition field proven operation with most available SNMP management systems is a major plus.
- Fast integration time depends on open and flexible software engineering design. Such a design support smooth updates, changes and custom additions.
- The documentation is a critical factor in accelerating the TTM. Usage of tools such as Doxygen can significantly enhance the learning curve, integration and testing.

MIBs Development Effort

MIBs' development may be the area where programmers spend most of their time along the SNMP project. Below are several factors that influence such an effort:

- The MIB compiler. The product should have SMIv2 MIB-compiler that generates C and Java codes. The created code should clearly mark the areas where users need to add system code and extensions.

- The solution should have several built-in MIBs that will speed up the development process. An example for such a MIB is IP-MIB, which interacts with the IP protocol.

- The documentation is crucial to make sure software engineers get fast up to speed. The MIB API (API-A) should be well documented

Platform and Project Constraints

Each SNMP project is part of a larger system and must operate under strict real-time constraints. Such requirements include run-time performance, code size and support for specific features. The following list outlines the several important such considerations and their implications:

- The SNMP solution should be platform independent, while allows for a smooth platform support. The platform support area should be clear and the product should have gone through numerous platforms' integrations processes.

- Code size – the SNMP kernel should be small enough to fit with small platforms such as those based on 8051.

- Run-time performance and CPU load is another crucial element since both SNMP protocol and the lexicographical support may impose high CPU load

- The SNMP solution should be field proven - deployed in many commercial products and integrated in many types of systems and variations, for example working with both IPv4, IPv6 and dual protocol stacks. Another example is working with multiple SNMP managers to make sure it is interoperable.

Upgrade and Support

An SNMP project is usually a live project and changes are introduced during the project lifetime. Such changes can require the development of new MIBs, an upgrade to SNMPv3, support of a different platform, adding of new features and more. The following list outlines several such topics and their implication

- Upgrades between different environments are crucial for the project team. For example, a development team can start off with a DOS SNMPv2c, later upgrade to DOS SNMPv3, and finally, upgrade to Windows platform. It is crucial that the kernel source will not change along such project transitions.

- The quality of the code determines the support response time. Products which do not meet advanced software engineering design criteria, will have support issues and suffer from delayed response.
- The maintenance is crucially important further down the road if no engineer from the original engineering team stayed in his/her position.

Total Cost of Ownership (TCO)

Sometimes simple product price comparison does not reflect the true cost of ownership which should be evaluated along the development and maintenance project lifetime. The project TCO includes the following components:

- Product price is the first cost topic the evaluation team faces; however, sometimes, this price is a small part of the TCO as discussed below.

- Kernel integration effort, including internal and outsourced software engineering to support the platform and its environment. When using a well designed product, such integration can last few days only.

- MIBs development effort, including internal and outsourced software engineering. The MIB compiler, API-A and MIB extension should provide a smooth MIB development and testing. Such tools should hide the internal complexity of the lexicographical presentation and allows developers to focus on their mission.

- Testing. Some commercial products allow testing on standard platform such as Linux or Windows providing a rapid development and testing environment; such an environment provides an excellent reference in case of an issue with the operational system's hardware or software component.

- Support and upgrades. Once the Kernel is integrated, future upgrade can become a smooth process. For example, when the AES encryption was added by IETF, some project teams requested an upgrade for SNMPv3 to support an enhanced security. With such smooth upgrade feature, projects could be upgraded with minimum operational disruption and almost no engineering effort.

TCO of an SNMP project can stack up to 1 person-year which is significantly more expensive than most of the commercial products

About DMH software

For the last 15 years, DMH Software is a recognized global leader in SNMP Agent solutions with over 50 customers worldwide. It provides field proven portable, real-time and extensible C and Java implementations of SNMP Agents (SNMPv1, SNMPv2c, SNMPv3). DMH Software's SDK includes a SMIv2 MIB-Compiler for rapid MIB development. The agent can be used in a wide range of platforms - from very small embedded systems such as 8bit 8051, and up to 64bit systems. In addition, the agent can fit proprietary RTOS, Standard RTOS or no RTOS. Since DMH's portable software is highly portable, it offers a free platform integration (if such integration is required).

DMH Software is recognized for its high-quality software products, open architecture, highly professional coding and documentation and small code size which is ideal for small secured mission-critical systems. DMH Software is proud of its fast project customization and urgent customer response time.

DMH Software is a private company which operates out of Acton, Massachusetts and includes an experienced team in embedded software, communications, and SNMP. For more information please visit us on our web site: http://dmhsoftware.com

Acronyms

ACSE – Association Control Service Element

ASN.1 – Abstract Syntax Notation One

IETF – Internet Engineering Task Force

MIB – Management Information Base

PDU – Protocol Data Unit

RFC – Request for Comment

SMI – Structure of Management Information

SNMP – Simple Network management protocol

UDP – User Datagram Protocol

References

DMH. (2008). *DMH Software*. Retrieved April 20, 2008, from htt://dmhsoftware.com

RFC 3826. (2004). *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*. Retrieved April 22, 2008, from http://www.rfc-archive.org

Stallings, J. (1999). *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Reading, MA: Addison Wesley Longman, Inc.