

A Convention for Defining Traps  
for use with the SNMP

Status of this Memo

This memo suggests a straight-forward approach towards defining traps used with the SNMP. Readers should note that the use of traps in the Internet-standard network management framework is controversial. As such, this memo is being put forward for information purposes. Network management practitioners who employ traps are encouraged to make use of this document. Practitioners who do not employ traps can safely ignore this document.

This memo provides information for the Internet community. It does not specify any standard. Distribution of this memo is unlimited.

Table of Contents

1. Historical Perspective .....	1
2. Defining Traps .....	2
2.1 Mapping of the TRAP-TYPE macro .....	3
2.1.1 Mapping of the ENTERPRISE clause .....	3
2.1.2 Mapping of the VARIABLES clause .....	4
2.1.3 Mapping of the DESCRIPTION clause .....	4
2.1.4 Mapping of the REFERENCE clause .....	4
2.1.5 Mapping of the TRAP-TYPE value .....	4
2.2 Usage Examples .....	5
2.2.1 Enterprise-specific Trap .....	5
2.2.2 Generic-Traps for use with the SNMP .....	5
3. Acknowledgements .....	7
4. References .....	9
5. Security Considerations.....	9
6. Author's Address.....	9

1. Historical Perspective

As reported in RFC 1052, IAB Recommendations for the Development of Internet Network Management Standards [1], a two-prong strategy for network management of TCP/IP-based internets was undertaken. In the short-term, the Simple Network Management Protocol (SNMP), defined in RFC 1067, was to be used to manage nodes in the Internet community. In the long-term, the use of the OSI network management framework was to be examined. Two documents were produced to define the management

information: RFC 1065, which defined the Structure of Management Information (SMI), and RFC 1066, which defined the Management Information Base (MIB). Both of these documents were designed so as to be compatible with both the SNMP and the OSI network management framework.

This strategy was quite successful in the short-term: Internet-based network management technology was fielded, by both the research and commercial communities, within a few months. As a result of this, portions of the Internet community became network manageable in a timely fashion.

As reported in RFC 1109, Report of the Second Ad Hoc Network Management Review Group [2], the requirements of the SNMP and the OSI network management frameworks were more different than anticipated. As such, the requirement for compatibility between the SMI/MIB and both frameworks was suspended. This action permitted the operational network management framework, based on the SNMP, to respond to new operational needs in the Internet community by producing MIB-II.

In May of 1990, the core documents were elevated to "Standard Protocols" with "Recommended" status. As such, the Internet-standard network management framework consists of: Structure and Identification of Management Information for TCP/IP-based internets, RFC 1155 [3], which describes how managed objects contained in the MIB are defined; Management Information Base for Network Management of TCP/IP-based internets, which describes the managed objects contained in the MIB, RFC 1156 [4]; and, the Simple Network Management Protocol, RFC 1157 [5], which defines the protocol used to manage these objects.

## 2. Defining Traps

Due to its initial requirement to be protocol-independent, the Internet-standard SMI does not provide a means for defining traps. Instead, the SNMP defines a few standardized traps and provides a means for management enterprises to transmit enterprise-specific traps.

However, with the introduction of experimental MIBs, some of which have a need to define experiment-specific traps, a convenient means of defining traps is desirable. The TRAP-TYPE macro is suggested for this purpose:

```
IMPORTS
    ObjectName
    FROM RFC1155-SMI;
```

```
TRAP-TYPE MACRO ::=
BEGIN
    TYPE NOTATION ::= "ENTERPRISE" value
                        (enterprise OBJECT IDENTIFIER)
                        VarPart
                        DescrPart
                        ReferPart
    VALUE NOTATION ::= value (VALUE INTEGER)

    VarPart ::=
        "VARIABLES" "{" VarTypes "}"
        | empty
    VarTypes ::=
        VarType | VarTypes "," VarType
    VarType ::=
        value (vartype ObjectName)

    DescrPart ::=
        "DESCRIPTION" value (description DisplayString)
        | empty
    ReferPart ::=
        "REFERENCE" value (reference DisplayString)
        | empty

END
```

It must be emphasized however, that the use of traps is **STRONGLY** discouraged in the Internet-standard Network Management Framework. The TRAP-TYPE macro is intended to allow concise definitions of existing traps, not to spur the definition of new traps.

## 2.1. Mapping of the TRAP-TYPE macro

It should be noted that the expansion of the TRAP-TYPE macro is something which conceptually happens during implementation and not during run-time.

### 2.1.1. Mapping of the ENTERPRISE clause

The ENTERPRISE clause, which must be present, defines the management enterprise under whose registration authority this trap is defined (for a discussion on delegation of registration authority, see the SMI [3]). This value is placed inside the enterprise field of the SNMP Trap-PDU.

By convention, if the value of the ENTERPRISE clause is

```
snmp    OBJECT IDENTIFIER ::= { mib-2 11 }
```

as defined in MIB-II [7], then instead of using this value, the value of sysObjectID is placed in the enterprise field of the SNMP Trap-PDU. This provides a simple means of using the TRAP-TYPE macro to represent the existing standard SNMP traps; it is not intended to provide a means to define additional standard SNMP traps.

#### 2.1.2. Mapping of the VARIABLES clause

The VARIABLES clause, which need not be present, defines the ordered sequence of MIB objects which are contained within every instance of the trap type. Each variable is placed, in order, inside the variable-bindings field of the SNMP Trap-PDU. Note that at the option of the agent, additional variables may follow in the variable-bindings field.

However, if the value of the ENTERPRISE clause is

```
snmp    OBJECT IDENTIFIER ::= { mib-2 11 }
```

as defined in MIB-II [7], then the introduction of additional variables must not result in the serialized SNMP Message being larger than 484 octets.

#### 2.1.3. Mapping of the DESCRIPTION clause

The DESCRIPTION clause, which need not be present, contains a textual definition of the trap type. Note that in order to conform to the ASN.1 syntax, the entire value of this clause must be enclosed in double quotation marks, although the value may be multi-line.

Further, note that if the MIB module does not contain a textual description of the trap elsewhere then the DESCRIPTION clause must be present.

#### 2.1.4. Mapping of the REFERENCE clause

The REFERENCE clause, which need not be present, contains a textual cross-reference to a trap, event, or alarm, defined in some other MIB module. This is useful when de-serializing a MIB produced by some other organization.

#### 2.1.5. Mapping of the TRAP-TYPE value

The value of an invocation of the TRAP-TYPE macro is the (integer) number which is uniquely assigned to the trap by the registration authority indicated by the ENTERPRISE clause. This value is placed

inside the specific-trap field of the SNMP Trap-PDU, and the generic-trap field is set to "enterpriseSpecific(6)".

By convention, if the value of the ENTERPRISE clause is

```
snmp    OBJECT IDENTIFIER ::= { mib-2 11 }
```

as defined in MIB-II [7], then the value of an invocation of the TRAP-TYPE macro is placed inside the generic-trap field of the SNMP Trap-PDU, and the specific-trap field is set to 0. This provides a simple means of using the TRAP-TYPE macro to represent the existing standard SNMP traps; it is not intended to provide a means to define additional standard SNMP traps.

## 2.2. Usage Examples

### 2.2.1. Enterprise-specific Trap

Consider a simple example of an enterprise-specific trap that is sent when a communication link failure is encountered:

```
myEnterprise OBJECT IDENTIFIER ::= { enterprises 9999 }

myLinkDown TRAP-TYPE
    ENTERPRISE    myEnterprise
    VARIABLES     { ifIndex }
    DESCRIPTION   "A myLinkDown trap signifies that the sending
                  SNMP application entity recognizes a failure
                  in one of the communications links represented
                  in the agent's configuration."
    ::= 2
```

### 2.2.2. Generic-Traps for use with the SNMP

Consider how the standard SNMP traps might be defined:

```
coldStart TRAP-TYPE
    ENTERPRISE    snmp
    DESCRIPTION   "A coldStart trap signifies that the sending
                  protocol entity is reinitializing itself such
                  that the agent's configuration or the rotocol
                  entity implementation may be altered."
    ::= 0

warmStart TRAP-TYPE
    ENTERPRISE    snmp
```

## DESCRIPTION

"A warmStart trap signifies that the sending protocol entity is reinitializing itself such that neither the agent configuration nor the protocol entity implementation is altered."

::= 1

## linkDown TRAP-TYPE

ENTERPRISE snmp

VARIABLES { ifIndex }

## DESCRIPTION

"A linkDown trap signifies that the sending protocol entity recognizes a failure in one of the communication links represented in the agent's configuration."

::= 2

## linkUp TRAP-TYPE

ENTERPRISE snmp

VARIABLES { ifIndex }

## DESCRIPTION

"A linkUp trap signifies that the sending protocol entity recognizes that one of the communication links represented in the agent's configuration has come up."

::= 3

## authenticationFailure TRAP-TYPE

ENTERPRISE snmp

## DESCRIPTION

"An authenticationFailure trap signifies that the sending protocol entity is the addressee of a protocol message that is not properly authenticated. While implementations of the SNMP must be capable of generating this trap, they must also be capable of suppressing the emission of such traps via an implementation-specific mechanism."

::= 4

```
egpNeighborLoss TRAP-TYPE
    ENTERPRISE    snmp
    VARIABLES     { egpNeighAddr }
    DESCRIPTION   "An egpNeighborLoss trap signifies that an EGP
                  neighbor for whom the sending protocol entity
                  was an EGP peer has been marked down and the
                  peer relationship no longer obtains."

 ::= 5
```

### 3. Acknowledgements

This document was produced by the SNMP Working Group:

Anne Ambler, Spider  
Karl Auerbach, Sun  
Fred Baker, ACC  
Ken Brinkerhoff  
Ron Broersma, NOSC  
Jack Brown, US Army  
Theodore Brunner, Bellcore  
Jeffrey Buffum, HP  
John Burrell, Wellfleet  
Jeffrey D. Case, University of Tennessee at Knoxville  
Chris Chiptasso, Spartacus  
Paul Ciarfella, DEC  
Bob Collet  
John Cook, Chipcom  
Tracy Cox, Bellcore  
James R. Davin, MIT-LCS  
Eric Decker, cisco  
Kurt Dobbins, Cabletron  
Nadya El-Afandi, Network Systems  
Gary Ellis, HP  
Fred Engle  
Mike Erlinger  
Mark S. Fedor, PSI  
Richard Fox, Synoptics  
Karen Frisa, CMU  
Chris Gunner, DEC  
Fred Harris, University of Tennessee at Knoxville  
Ken Hibbard, Xylogics  
Ole Jacobsen, Interop  
Ken Jones  
Satish Joshi, Synoptics  
Frank Kastenholz, Racal-Interlan  
Shimshon Kaufman, Spartacus  
Ken Key, University of Tennessee at Knoxville

Jim Kinder, Fibercom  
Alex Koifman, BBN  
Christopher Kolb, PSI  
Cheryl Krupczak, NCR  
Paul Langille, DEC  
Peter Lin, Vitalink  
John Lunny, TWG  
Carl Malamud  
Randy Mayhew, University of Tennessee at Knoxville  
Keith McCloghrie, Hughes LAN Systems  
Donna McMaster, David Systems  
Lynn Monsanto, Sun  
Dave Perkins, 3COM  
Jim Reinstedler, Ungerman Bass  
Anil Rijasinghani, DEC  
Kathy Rinehart, Arnold AFB  
Kary Robertson  
Marshall T. Rose, PSI (chair)  
L. Michael Sabo, NCSC  
Jon Saperia, DEC  
Greg Satz, cisco  
Martin Schoffstall, PSI  
John Seligson  
Steve Sherry, Xyplex  
Fei Shu, NEC  
Sam Sjogren, TGV  
Mark Sleeper, Sparta  
Lance Sprung  
Mike St.Johns  
Bob Stewart, Xyplex  
Emil Sturniold  
Kaj Tesink, Bellcore  
Dean Throop, Data General  
Bill Townsend, Xylogics  
Maurice Turcotte, Racal-Milgo  
Kannan Varadhou  
Sudhanshu Verma, HP  
Bill Versteeg, Network Research Corporation  
Warren Vik, Interactive Systems  
David Waitzman, BBN  
Steve Waldbusser, CMU  
Dan Wintringhan  
David Wood  
Wengyik Yeong, PSI  
Jeff Young, Cray Research



#### 4. References

- [1] Cerf, V., "IAB Recommendations for the Development of Internet Network Management Standards", RFC 1052, NRI, April 1988.
- [2] Cerf, V., "Report of the Second Ad Hoc Network Management Review Group", RFC 1109, NRI, August 1989.
- [3] Rose M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", RFC 1155, Performance Systems International, Hughes LAN Systems, May 1990.
- [4] McCloghrie K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets", RFC 1156, Hughes LAN Systems, Performance Systems International, May 1990.
- [5] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.
- [6] Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), International Organization for Standardization International Standard 8824, December 1987.
- [7] Rose M., Editor, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", RFC 1213, Performance Systems International, March 1991.

#### 5. Security Considerations

Security issues are not discussed in this memo.

#### 6. Author's Address

Marshall T. Rose  
Performance Systems International  
5201 Great America Parkway  
Suite 3106  
Santa Clara, CA 95054

Phone: +1 408 562 6222

EMail: mrose@psi.com  
X.500: rose, psi, us