Remote Monitoring MIB Extensions
for Identifying Application Protocol Verbs

<draft-ietf-rmonmib-appverbs-00.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [RFC2026].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Distribution of this document is unlimited. Please send comments to the RMONMIB WG mailing list <rmonmib@ietf.org>.

1. Copyright Notice
2. Abstract

This memo defines an experimental portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes the algorithms required to identify protocol operations (verbs) within the protocol encapsulations managed with the Remote Network Monitoring MIB Version 2 [RFC2021].

3. Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Copyright Notice</td>
<td>2</td>
</tr>
<tr>
<td>2 Abstract</td>
<td>2</td>
</tr>
<tr>
<td>3 Table of Contents</td>
<td>2</td>
</tr>
<tr>
<td>4 The SNMP Network Management Framework</td>
<td>3</td>
</tr>
<tr>
<td>5 Overview</td>
<td>4</td>
</tr>
<tr>
<td>5.1 Protocol Identifier Framework</td>
<td>4</td>
</tr>
<tr>
<td>5.2 Protocol Identifier Extensions for Application Verbs</td>
<td>4</td>
</tr>
<tr>
<td>5.3 Terms</td>
<td>5</td>
</tr>
<tr>
<td>5.3.1 Application Verb</td>
<td>5</td>
</tr>
<tr>
<td>5.3.2 Parent Application</td>
<td>5</td>
</tr>
<tr>
<td>5.3.3 Verb Layer</td>
<td>5</td>
</tr>
<tr>
<td>5.3.4 Verb Set</td>
<td>5</td>
</tr>
<tr>
<td>5.4 Relationship to the RMON-2 MIB</td>
<td>5</td>
</tr>
<tr>
<td>5.5 Relationship to the RMON MIB Protocol Identifier Reference</td>
<td>6</td>
</tr>
<tr>
<td>6 Verb Identifier Macro Format</td>
<td>6</td>
</tr>
<tr>
<td>6.1 Lexical Conventions</td>
<td>6</td>
</tr>
<tr>
<td>6.2 Extended Grammar for the PI Language</td>
<td>6</td>
</tr>
<tr>
<td>6.3 Mapping of the Parent Protocol Name</td>
<td>7</td>
</tr>
<tr>
<td>6.4 Mapping of the DESCRIPTION Clause</td>
<td>7</td>
</tr>
<tr>
<td>6.5 Mapping of the REFERENCE Clause</td>
<td>8</td>
</tr>
<tr>
<td>6.6 Mapping of the Verb List Clause</td>
<td>8</td>
</tr>
<tr>
<td>6.6.1 Mapping of the Verb Name Field</td>
<td>8</td>
</tr>
<tr>
<td>6.6.2 Mapping of the Verb Enum Field</td>
<td>8</td>
</tr>
<tr>
<td>7 Verb Identifiers in the protocolDirTable</td>
<td>8</td>
</tr>
<tr>
<td>7.1 Definition of the Verb Layer Numbering Space</td>
<td>10</td>
</tr>
<tr>
<td>7.2 Mapping of the ProtocolDirID object</td>
<td>10</td>
</tr>
<tr>
<td>7.3 Mapping of the ProtocolDirParameters object</td>
<td>11</td>
</tr>
<tr>
<td>7.4 Mapping of the ProtocolDirLocalIndex object</td>
<td>11</td>
</tr>
<tr>
<td>7.5 Mapping of the protocolDirDescr object</td>
<td>11</td>
</tr>
<tr>
<td>7.6 Mapping of the protocolDirType object</td>
<td>11</td>
</tr>
<tr>
<td>7.7 Mapping of the protocolDirAddressMapConfig object</td>
<td>11</td>
</tr>
</tbody>
</table>
4. The SNMP Network Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [RFC2571].

- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in RFC 1155 [RFC1155], RFC 1212 [RFC1212] and RFC 1215 [RFC1215]. The second version, called SMIv2, is described in RFC 2578 [RFC2578], RFC 2579 [RFC2579] and RFC 2580 [RFC2580].

- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [RFC1157]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [RFC1901] and RFC 1906 [RFC1906]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [RFC1906], RFC 2572 [RFC2572] and RFC 2574 [RFC2574].

- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in RFC 1157 [RFC1157]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [RFC1905].
A set of fundamental applications described in RFC 2573 [RFC2573] and the view-based access control mechanism described in RFC 2575 [RFC2575].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [RFC2570].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo does not specify a MIB module.

5. Overview

There is a need for a standardized way of identifying the protocol operations defined for particular application protocols. Different protocol operations can have very different performance characteristics, and it is desirable to collect certain metrics at this level of granularity. This memo defines extensions to the existing protocol identifier structure [RFC2074], and is intended to update, not obsolete, the existing protocol identifier encoding rules.

5.1. Protocol Identifier Framework

The RMON Protocol Identifier (PI) structure [RFC2074] allows for a variable number of layer identifiers. Each layer contributes 4 octets to the protocolDirID OCTET STRING and one octet to the protocolDirParameters OCTET STRING. These two MIB objects comprise the index into the protocolDirTable [RFC2021], and represent a globally unique identifier for a particular protocol encapsulation (or set of encapsulations if the wildcard base layer is used).

5.2. Protocol Identifier Extensions for Application Verbs

The existing RMON protocol identifier architecture requires that an application verb be represented by one additional protocol layer, appended to the protocol identifier for the parent application. Since some application verbs are defined as strings which can exceed 4 octets in length, an integer mapping must be provided for each string. This memo specifies how the verb layer is structured, as well as a verb identifier macro syntax for specification of verb name to integer mappings.
5.3. Terms

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. [RFC2119]

This document uses some terms defined in the RMON Protocol Identifier Reference document [PIREF], and some new terms that need introduction here.

5.3.1. Application Verb

Also called simply 'verb'. Refers to one of potentially many protocol operations that are defined by a particular application protocol.

Note that an application verb is not equivalent to an application protocol sub-command or opcode within a packet containing a PDU for the application. An application verb is a transaction type, and may involve several PDU types within the application protocol (e.g., SNMP Get-PDU and Response-PDU). In some applications, a verb may encompass protocol operations pertaining to more than one protocol entry in the protocol directory (e.g., ftp and ftp-data).

5.3.2. Parent Application

One of potentially many protocol encapsulations which identifies a particular application protocol. This term refers generically to any or all such encapsulations for a given set of application verbs.

5.3.3. Verb Layer

The portion of the protocol identifier octet string which identifies the application verb.

5.3.4. Verb Set

The group of verbs enumerated for a particular application protocol. The list of verb strings within a particular verb-identifier macro invocation is also called the verb set for that verb identifier.

5.4. Relationship to the RMON-2 MIB

The RMON-2 MIB [RFC2021] contains the protocolDirTable MIB objects used to identify all protocol encapsulations that can be monitored by a particular RMON agent.
This memo describes how these MIB objects are mapped by an implementation, for entries which identify application verbs. This document does not define any new MIB objects to identify application verbs.

5.5. Relationship to the RMON MIB Protocol Identifier Reference

The RMON MIB Protocol Identifier Reference [PIREF] defines the RMON Protocol Identifier Macro Specification Language, as well as the encoding rules for the ProtocolDirID and protocolDirParameters OCTET STRINGs.

This memo defines extensions to the Protocol Identifier Reference document for the identification of application verb information. It does not obsolete any portion of the Protocol Identifier Reference document.

6. Verb Identifier Macro Format

The following example is meant to introduce the verb-identifier macro. This macro-like construct is used to represent protocol verbs for a specific parent application.

6.1. Lexical Conventions

The following keyword is added to the PI language:

VERB-IDENTIFIER

6.2. Extended Grammar for the PI Language

The following is the extended BNF notation for the grammar with starting symbol <piFile>, for representing verb identifier macros. Note that only the term <piFile> is actually modified from the definition in [PIREF]. The <piDefinition> syntax is not reproduced here, since this memo is intended to extend that definition, not replace it.

```
-- a file containing one or more Protocol Identifier (PI) definitions
<piFile> = [ <piDefinition> | <piVerbDefinition> ]...

-- a PI definition
<piVerbDefinition> =
  <parentProtoName> "VERB-IDENTIFIER"
  "DESCRIPTION" string
  [ "REFERENCE" string ]
```
"::=" "{" <verbList> "}""

-- a list of verb identifier string
<verbList> = <verbId> [ <wspace> "," <wspace> <verbId> ]...

-- a verb identifier string
<verbId> = <verbName> [ <wspace> ] "(" [ <wspace> ]
            <verbEnum> [ <wspace> ] ")" [ <wspace> ]

-- a verb name
<verbName> = lcname

-- a verb enumeration
<verbEnum> = <posNum>

-- a positive integer
<posNum> = any integer value greater than zero and
         less than 16,777,216

-- <piDefinition> syntax is defined in [PIREF]
-- <wspace> syntax is defined in [PIREF]
-- lcname syntax is defined in [PIREF]

6.3. Mapping of the Parent Protocol Name

The "parentProtoName" value, called the "parent protocol name" shall be
an ASCII string consisting of one up to 64 characters. The encoding
rules are exactly as specified in section 6.2.4 of [PIREF], for the
mapping of the protocol name field. If a <protoName> and a
<parentProtoName> field contain the same value, then they refer to the
same protocol.

A protocol identifier macro SHOULD exist in the <piFile> for at least
one encapsulation of the parent application protocol, if any verb
identifier macros referencing that parent application are present in the
<piFile>.

6.4. Mapping of the DESCRIPTION Clause

The DESCRIPTION clause provides a textual description of the protocol
verb set identified by this macro. Notice that it SHOULD NOT contain
details about items covered by the DECODING and REFERENCE clauses.
The DESCRIPTION clause MUST be present in all verb-identifier macro declarations.

6.5. Mapping of the REFERENCE Clause

If a publicly available reference document exists for this set of application protocol verbs, it SHOULD be listed here. Typically this will be a URL if possible; if not then it will be the name and address of the controlling body.

The REFERENCE clause is optional, but SHOULD be implemented if an authoritative reference exists which specifies the application protocol verbs defined in the <verbList> section of this macro.

6.6. Mapping of the Verb List Clause

The verb list clause MUST be present, and is used to identify a list of application verb names, and associate a numeric constant with each verb name. At least one verb MUST be specified, and a maximum of 16,777,215 (2\(^{24} - 1\)) verbs MAY be specified. This enumerated list SHOULD be densely numbered (i.e., valued from ‘1’ to ‘N’, where ‘N’ is the total number of verbs defined in the macro).

6.6.1. Mapping of the Verb Name Field

The <verbName> field is case-sensitive, and SHOULD be set to the most appropriate string name for each application verb. If a readable string is defined in an authoritative document, then that exact string SHOULD be used. If no such string exists, then an appropriate but arbitrary string should be selected for this value.

Verb names MUST be unique for a particular parent application.

6.6.2. Mapping of the Verb Enum Field

The <verbEnum> field MUST be unique for all verbs associated with a particular parent application. This field MUST contain a value between ‘1’ and ‘16,777,215’ inclusive.

7. Verb Identifiers in the protocolDirTable

This section describes how the protocolDirTable should be populated for an application verb identified with a verb-identifier macro.
An agent MUST implement all applicable protocolDirTable MIB objects on behalf of each supported application verb.
7.1. Definition of the Verb Layer Numbering Space

The verb layer consists of the 4 octets within the protocolDirID INDEX field which identify a particular application verb.

Figure 1
Verb Layer Format
--------------------

protocolDirID string fragment
-------------------------------
| resrvd |                          |
| .. set to | verb enumeration value |
| zero   | (a) (b) (c)             |
----------------------------------- octet
| 1 | 3 | count

The first octet is a reserved field and MUST be set to zero.

The next three octets identify the <verbEnum> field used to enumerate the particular application verb represented by the <verbName> field. This field is a 24-bit unsigned integer, encoded in network byte order.

7.2. Mapping of the ProtocolDirID object

The protocolDirID OCTET STRING value for a particular application verb is represented by the protocolDirID value for the parent application, appended with the verb’s layer identifier value.

Figure 2
ProtocolDirID Format for Verbs
-------------------------------

protocolDirID string
----------------------
| parent | verb |
| protocolDirID | layer |
| string | value |
----------------------------------- octet
| length of parent ID | 4 | count

The protocolDirID object is encoded as the protocolDirID value of the parent application, followed by four additional octets representing the
verb layer. The verb layer value is encoded as [0.a.b.c] where ‘a’ is
the high order byte, ‘b’ is the middle order byte, and ‘c’ is the low
order byte of the <verbEnum> field for the specific application verb
value.

7.3. Mapping of the ProtocolDirParameters object

The protocolDirParameters OCTET STRING value for a particular
application verb is represented by the protocolDirParameters value for
the parent application, appended with one octet containing the value
zero.

7.4. Mapping of the ProtocolDirLocalIndex object

The agent MUST assign an appropriate protocolDirLocalIndex value for
each application verb, according to the encoding rules defined for this
object in [RFC2021] and [PIREF].

7.5. Mapping of the protocolDirDescr object

The agent MUST convey the <verbName> value for a particular application
verb in the protocolDirDescr object. This object SHOULD be encoded as
the protocolDirDescr value for the parent application, appended with a
‘dot’ character, followed by the exact text contained in the <verbName>
field.

7.6. Mapping of the protocolDirType object

The agent MUST set the protocolDirType object for each application verb
to the value representing the empty bit set ( {} )

7.7. Mapping of the protocolDirAddressMapConfig object

The agent MUST set the protocolDirAddressMapConfig object for each
application verb to the value ‘notSupported(1)’.

7.8. Mapping of the protocolDirHostConfig object

The agent MUST set the protocolDirHostConfig object for each application
verb, according to the monitoring capabilities for the parent
application. The agent SHOULD set this object to the same value as
configured in the parent application protocolDirHostConfig object. The
agent MAY choose to transition this object from the value
‘supportedOn(2)’ to ‘supportedOff(3)’, if the parent application
protocolDirHostConfig object first transitions from ‘supportedOn(2)’ to
`supportedOff(3)`.

**7.9. Mapping of the protocolDirMatrixConfig object**

The agent MUST set the protocolDirMatrixConfig object for each application verb, according to the monitoring capabilities for the parent application. The agent SHOULD set this object to the same value as configured in the parent application protocolDirMatrixConfig object. The agent MAY choose to transition this object from the value `supportedOn(2)` to `supportedOff(3)`, if the parent application protocolDirMatrixConfig object first transitions from `supportedOn(2)` to `supportedOff(3)`.

**7.10. Mapping of the protocolDirOwner object**

This object is encoded exactly the same for application verbs as for other protocolDirTable entries, according to the rules specified in the RMON-2 MIB [RFC2021].

**7.11. Mapping of the protocolDirStatus object**

This object is encoded exactly the same for application verbs as for other protocolDirTable entries, according to the rules specified in RMON-2 MIB [RFC2021].
8. Appendix A: Usage Examples

The following examples are listed to demonstrate how RMON verb identifiers are declared.

[ed. the WG needs to decide if verb macros should be declared in a separate RFC, the way the PI macros are split out from the PI reference document.]

8.1. FTP Example

This example defines verb enumeration values for the File Transfer Protocol, as defined in RFC 959 and updated by RFC 2228 and RFC 2640. Note that verb name strings specified in the <verbName> field are not limited to 4 characters in length. In the FTP protocol, all the command names are 4 characters in length, and the verb name string should match the official command name as closely as possible.

ftp VERB-IDENTIFIER
DESCRIPTION
"The set of verbs for FTP is derived from the list of commands defined for the File Transfer Protocol, which are identified by case-insensitive strings. The commands are simply listed in the order found in the FTP documentation."
REFERENCE
"File Transfer Protocol, RFC 959, Section 4.1;
FTP Security Extensions, RFC 2228, Section 3;
Internationalization of the File Transfer Protocol, RFC 2640, Section 4.1."
::= {
    user(1),     -- USER NAME
    pass(2),     -- PASSWORD
    acct(3),     -- ACCOUNT
    cwd(4),      -- CHANGE WORKING DIRECTORY
    cdup(5),     -- CHANGE TO PARENT DIRECTORY
    smnt(6),     -- STRUCTURE MOUNT
    rein(7),     -- REINITIALIZE
    quit(8),     -- LOGOUT
    port(9),     -- DATA PORT
    pasv(10),    -- PASSIVE
    type(11),    -- REPRESENTATION TYPE
    stru(12),    -- FILE STRUCTURE
    mode(13),    -- TRANSFER MODE
    retr(14),    -- RETRIEVE
Internet-Draft            RMON Verb Identifiers                July 2000

stor(15),    -- STORE
stou(16),    -- STORE UNIQUE
appe(17),    -- APPEND (with create)
allo(18),    -- ALLOCATE
rest(19),    -- RESTART
rnfr(20),    -- RENAME FROM
rnto(21),    -- RENAME TO
abor(22),    -- ABORT
dele(23),    -- DELETE
rmd(24),     -- REMOVE DIRECTORY
mkd(25),     -- MAKE DIRECTORY
pwd(26),     -- PRINT WORKING DIRECTORY
list(27),    -- LIST
nlst(28),    -- NAME LIST
site(29),    -- SITE PARAMETERS
syst(30),    -- SYSTEM
stat(31),    -- STATUS
help(32),    -- HELP
noop(33),    -- NOOP
auth(34),    -- AUTHENTICATION/SECURITY MECHANISM
adat(35),    -- AUTHENTICATION/SECURITY DATA
pbsz(36),    -- PROTECTION BUFFER SIZE
prot(37),    -- DATA CHANNEL PROTECTION LEVEL
ccc(38),     -- CLEAR COMMAND CHANNEL
mic(39),     -- INTEGRITY PROTECTED COMMAND
conf(40),    -- CONFIDENTIALITY PROTECTED COMMAND
enc(41),     -- PRIVACY PROTECTED COMMAND
lang(42)     -- LANGUAGE
}

8.2.  POP3 Example

This example defines verb enumeration values for the Post Office Protocol, Version 3, as defined in RFC 1939 and updated by RFC 2449.

pop3 VERB-IDENTIFIER
DESCRIPTION
  "The set of verbs for POP3 is derived from the list of commands defined for the Post Office Protocol, which are identified by case-insensitive strings. The commands are simply listed in the order found in the POP3 command summary."
REFERENCE
  "Post Office Protocol, Version 3, RFC 1939, Section 9;
8.3. SNMP Example

This example defines verb enumeration values for the Simple Network Management Protocol, as defined in RFC 1905.

snmp VERB-IDENTIFIER
DESCRIPTION
"The set of verbs for SNMP is derived from the list of PDU transaction types in the Protocol Operations document for SNMPv2. Note that the Response-PDU is not considered a verb, but is classified as belonging to the transaction type associated with the request PDU."
REFERENCE

::= {
    get(1),
    get-next(2),
    get-bulk(3),
    set(4),
    inform(5),
    trap(6),
    report(7)
}
9. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

10. Acknowledgements

This memo is a product of the RMONMIB WG.

11. References

[PIREF]

[RFC1155]

[RFC1157]
[RFC1212]

[RFC1215]

[RFC1901]

[RFC1905]

[RFC1906]

[RFC2021]

[RFC2026]

[RFC2074]

[RFC2119]
S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels" RFC 2119, Harvard University, March 1997.
[RFC2570]
Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to
Version 3 of the Internet-standard Network Management Framework",
RFC 2570, SNMP Research, Inc., TIS Labs at Network Associates,

[RFC2571]
Describing SNMP Management Frameworks", RFC 2571, Cabletron
Systems, Inc., BMC Software, Inc., IBM T. J. Watson Research, April
1999.

[RFC2572]
Processing and Dispatching for the Simple Network Management
Protocol (SNMP)", RFC 2572, SNMP Research, Inc., Cabletron Systems,

[RFC2573]
2573, SNMP Research, Inc., Secure Computing Corporation, Cisco
Systems, April 1999.

[RFC2574]
Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for
version 3 of the Simple Network Management Protocol (SNMPv3)", RFC
2574, IBM T. J. Watson Research, April 1999.

[RFC2575]
Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access
Control Model (VACM) for the Simple Network Management Protocol
(SNMP)", RFC 2575, IBM T. J. Watson Research, BMC Software, Inc.,

[RFC2578]
McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M.,
and S. Waldbusser, "Structure of Management Information Version 2
(SMIv2)", RFC 2578, STD 58, Cisco Systems, SNMPinfo, TU
Braunschweig, SNMP Research, First Virtual Holdings, International
Network Services, April 1999.

[RFC2579]
McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M.,
and S. Waldbusser, "Textual Conventions for SMIv2", RFC 2579, STD
58, Cisco Systems, SNMPinfo, TU Braunschweig, SNMP Research, First
12. Security Considerations

This memo defines the structure of a portion of the Remote Monitoring MIB framework, but does not define any MIB objects, protocol operations, or other mechanisms which can potentially introduce new security risks into a managed system.
13. Author’s Address

Andy Bierman
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA USA 95134
Phone: +1 408-527-3711
Email: abierman@cisco.com

Chris Bucci
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA USA 95134
Phone: +1 408-527-5337
Email: cbucci@cisco.com

Russell Dietz
Apptitude, Inc.
6330 San Ignacio Avenue
San Jose, CA USA 95119
Phone: +1 408-574-2256
Email: rsdietz@apptitude.com

Albin Warth
NetScout Systems, Inc.
4 Technology Park Drive
Westford, MA 01886
Phone: (978) 614-4306
Email: albin@netscout.com
14. Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."