Network Working Group Internet-Draft Expires: January 18, 2002 F. Strauss J. Schoenwaelder TU Braunschweig July 20, 2001

SMIng Internet Protocol Core Modules draft-ietf-sming-inet-modules-02

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of <u>Section 10 of RFC2026</u>.

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/lid-abstracts.txt.

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

This Internet-Draft will expire on January 18, 2002.

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

Abstract

This memo presents SMIng modules that introduce commonly used Internet Protocol specific data definitions. They are provided so that other SMIng modules that would otherwise define their own representations can import them from a common place.

The IETF-INET module defines types and classes for representing Internet Protocol specific information. It builds on <u>RFC 2851</u> and extends it in several ways.

The IETF-INET-FILTER module extends the IETF-INET module by providing generic definitions for typical IP packet filters.

Table of Contents

<u>1</u> .	$\underline{1}$. Introduction	•			<u>3</u>
<u>2</u> .	<u>2</u> . IETF-INET				<u>3</u>
<u>3</u> .	<u>3</u> . IETF-INET-FILTER				<u>9</u>
<u>4</u> .	<u>4</u> . Usage Examples				<u>11</u>
<u>5</u> .	5. Security Considerations				<u>13</u>
<u>6</u> .	<u>6</u> . Acknowledgments				<u>13</u>
	References				<u>13</u>
	Authors' Addresses				<u>13</u>
<u>A</u> .	A. OPEN ISSUES				<u>14</u>
	Full Copyright Statement				<u>15</u>

1. Introduction

SMIng [1] modules frequently need to represent Internet Protocol specific information such as IP addresses. This memo contains SMIng modules which define a core set of SMIng types and classes to be imported by other SMIng modules.

The IETF-INET module provides core SMIng data definitions for the Internet Protocol suite. This module is derived from $[\underline{3}]$.

The IETF-INET-FILTER module provides SMIng data definitions that model Internet Protocol filters and components thereof.

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 [2].

2. IETF-INET

module IETF-INET {

organization	"IETF Next Generation Structure of Management Information Working Group (SMING)";
contact	"Juergen Schoenwaelder
	TU Braunschweig Bueltenweg 74/75 38106 Braunschweig Germany
	Phone: +49 531 391-3289 EMail: schoenw@ibr.cs.tu-bs.de";
description	"This module defines core types and classes for the Internet Protocol suite. This document builds upon <u>RFC 2851</u> and extends it in various ways.";
<pre>revision { date description };</pre>	"2001-03-02"; "Initial revision, published as RFC &rfc.number.";
// // Core type def //	initions for the Internet Protocol suite.
typedef InetPort	Number {

```
type
                    Unsigned32 (0..65535);
    description
       "Represents a 16 bit port number of an Internet
        transport layer protocol. Port numbers are assigned by
        IANA. A list of all assignments is available from
     <http://www.iana.org/>.
     The value zero is object-specific and must be defined as
        part of the description of any object which uses this
        syntax. Examples of the usage of zero might include
        situations where a port number is unknown, or when the
        value zero is used as a wildcard in a filter.";
    reference
       "STD 6 (RFC 768), STD 7 (RFC 793) and RFC 2960";
};
typedef InetProtocolNumber {
    type
                    Unsigned32 (0..255);
    description
       "Represents an 8 bit protocol number which indicates the
        next upper-layer protocol used in the data portion of an
        Internet datagram. Protocol numbers are assigned by
        IANA. A list of all assignments is available at the
        from <http://www.iana.org/>".;
    reference
       "STD 5 (RFC 791) Section 3.1 and RFC 2460 Section 3.";
};
typedef InetDiffServCodePoint {
    type
                    Unsigned32 (-1, 0..63);
    description
       "Represents a 6 bit Differentiated Services Code Point
        (DSCP). The special value -1 may be used to indicate
        e.g. a wildnot in a filter.";
    reference
       "RFC 2474";
};
typedef InetAddress {
    type
                    OctetString;
    description
       "Represents a generic IP version neutral Internet address.";
};
typedef InetAddressPrefixLength {
    type
                    Unsigned32;
    description
       "Denotes the length of a generic Internet network address
```

Strauss & Schoenwaelder Expires January 18, 2002 [Page 4]

prefix. A value of n corresponds to an IP address mask which has n contiguous 1-bits from the most significant bit (MSB) and all other bits set to 0.

InetAddressPrefixLength values that are larger than the maximum length of an IP address for a specific InetAddressType are treated as the maximum significant value applicable for the InetAddressType. The maximum significant value is 32 for the InetAddressType 'ipv4(1)' and 128 for the InetAddressType 'ipv6(2)'. The maximum significant value for the InetAddressType 'dns(16)' is 0.

The value zero is object-specific and must be defined as part of the description of any object which uses this syntax. Examples of the usage of zero might include situations where the Internet network address prefix is unknown or does not apply.";

};

typedef InetAutonomousSystemNumber {
 type Unsigned32;

description

"Represents an autonomous system number which identifies an Autonomous System (AS). An AS is a set of routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS, and using an exterior gateway protocol to route packets to other ASs'. IANA maintains the AS number space and has delegated large parts to the regional registries.

Autonomous system numbers are currently limited to 16 bits (0..65535). There is however work in progress to enlarge the autonomous system number space to 32 bits. This textual convention therefore uses an Unsigned32 value without a range restriction in order to support a larger autonomous system number space.";

```
reference
   "RFC 1771, RFC 1930";
};
//
// Internet Protocol address types for specific IP versions.
//
typedef InetAddressType {
   type Enumeration (unknown(0), ipv4(1), ipv6(2),
```

```
dns(16));
```

Strauss & Schoenwaelder Expires January 18, 2002 [Page 5]

description "A value that represents a type of Internet address.

- unknown(0) An unknown address type. This value MUST be used if the value of the corresponding address attribute is a zero-length string. It may also be used to indicate an IP address which is not in one of the formats defined below.
- ipv4(1) An IPv4 address as defined by the InetAddressIPv4 type.
- ipv6(2) An IPv6 address as defined by the InetAddressIPv6 type.
- dns(16) A DNS domain name as defined by the InetAddressDNS type.

The type SHOULD NOT be subtyped to support future extensions. It MAY be subtyped in compliance statements in order to require only a subset of these address types for a compliant implementation.";

};

```
typedef InetAddressIPv4 {
    type
                    InetAddress (4);
                   "1d.1d.1d";
    format
    description
       "Represents a 32 bit IP version 4 (IPv4) network address:
        octets
                 contents
                                  encoding
        1-4
                 IPv4 address
                                  network-byte order
        If there is a corresponding InetAddressType attribute,
        its value MUST be ipv4(1).";
    reference
       "STD 5 (RFC 791)";
};
typedef InetAddressIPv6 {
    type
                    InetAddress (16 | 20);
    format
                    "2x:2x:2x:2x:2x:2x:2x:2x%4d";
    description
       "Represents a 128 bit IPv6 network address plus an
        optional 32 bit scope identifier:
        octets
                 contents
                                  encoding
```

1-16 IPv6 address network-byte order 17-20 scope identifier network-byte order

If there is a corresponding InetAddressType attribute, its value MUST be ipv6(2).

The scope identifier (bytes 17-20) MUST NOT be present for global IPv6 addresses. For non-global IPv6 addresses (e.g. link-local or site-local addresses), the scope identifier MUST always be present. It contains a link identifier for link-local and a site identifier for site-local IPv6 addresses.

The scope identifier MUST disambiguate identical address values. For link-local addresses, the scope identifier will typically be the interface index (ifIndex as defined in the IF-MIB, <u>RFC 2233</u>) of the interface on which the address is configured.

The scope identifier may contain the special value 0 which refers to the default scope. The default scope may be used in cases where the valid scope identifier is not known (e.g., a management application needs to write a site-local InetAddressIPv6 address without knowing the site identifier value). The default scope SHOULD NOT be used as an easy way out in cases where the scope identifier for a non-global IPv6 address is known.";

reference

"<u>RFC 2373</u>";

};

```
typedef InetAddressDNS {
   type InetAddress (1..255);
   format "255a";
   description
      "Represents a DNS domain name. The name SHOULD be
      fully qualified whenever possible.
      If there is a corresponding InetAddressType attribute,
      its value MUST be dns(16).
      The descriptions of attributes of this type must fully
      describe how (and when) such names are to be resolved
      to IP addresses.";
}
```

};

//

```
SMIng Internet Protocol Core Modules
                                                                July 2001
Internet-Draft
       // Core class definitions for the Internet Protocol suite.
       11
       class InetNetworkEndpoint {
           attribute InetAddressType type {
               access
                           readwrite;
               description
                  "The type of this Internet Protocol endpoint.";
           };
           attribute InetAddress address {
               typemap type {
                   map ipv4 InetAddressIPv4;
                   map ipv6 InetAddressIPv6;
                   map dns InetAddressDNS;
               };
               access
                           readwrite;
               description
                  "The address of this Internet Protocol endpoint.
                   An address value is always interpreted within the
                   context of the type value. The type attribute defines
                   the context.";
           };
           description
               "This class defines a generic Internet Protocol endpoint
                at the network layer.";
       };
       class InetTransportEndpoint {
           attribute InetNetworkEndpoint address {
               description
                  "The network layer endpoint.";
           };
           attribute InetPortNumber port {
               description
                  "The transport layer port number.";
           };
           description
              "This class defines a generic transport layer endpoint for
               the Internet Protocol suite.";
       };
       class InetSubnet {
           attribute InetNetworkEndpoint endpoint {
               description
                  "A network layer endpoint within the Internet
                   Protocol subnet.";
           }
```

Strauss & Schoenwaelder Expires January 18, 2002 [Page 8]

```
attribute InetAddressPrefixLength prefix {
               description
                  "The address prefix which identifies the subnet
                   portion of the address of the network layer
                   endpoint.";
           };
           description
              "This class defines a generic Internet Protocol subnet.";
       };
       class InetPortNumberRange {
           attribute InetPortNumber start {
                           readwrite;
               access
               description
                  "The first port number in the port range.";
           };
           attribute InetPortNumber end {
               access
                           readwrite;
               description
                  "The last port number in the port range.";
           };
           description
              "This class represents a range of consecutive Internet
               transport layer port numbers. The start and end port
               numbers are included in the range of consecutive port
               numbers.";
       };
   };
3. IETF-INET-FILTER
   module IETF-INET-FILTER {
       import IETF-INET (InetProtocolNumber,
                          InetDiffServCodePoint,
                          InetPortNumberRange,
                          InetSubnet);
       import IETF-SMING (Counter64,
                          DisplayString255);
```

- organization "IETF Next Generation Structure of Management Information Working Group (SMING)";
- contact "Juergen Schoenwaelder
 - TU Braunschweig

```
Bueltenweg 74/75
                 38106 Braunschweig
                 Germany
                 Phone: +49 531 391-3289
                 EMail: schoenw@ibr.cs.tu-bs.de";
                "This module defines core filter classes for
description
                 the Internet protocol suite.";
revision {
                "2001-03-02";
    date
    description "Initial revision, published as RFC &rfc.number;.";
};
class Filter {
    attribute DisplayString255 name {
                    readwrite;
        access
        description
           "The administratively assigned name of the filter.";
    };
    attribute Counter64 packets {
        access
                    readonly;
        units
                    "packets";
           "The number of packets matching this filter.";
    };
    attribute Counter64 bytes {
                    readonly;
        access
                    "bytes";
        units
        description
           "The number of bytes contained in packets matching
            this filter.";
    };
    description
       "A generic filter. Classes derived from this generic filter
        must introduce additional attributes which define the
        filter parameters.";
};
class InetFilter : Filter {
    attribute InetSubnet srcSubNet {
        description
           "The subnet to match against the packet's
            source address.";
    };
    attribute InetSubnet dstSubNet {
        description
           "The subnet to match against the packet's
```

Strauss & Schoenwaelder Expires January 18, 2002 [Page 10]

```
destination address.";
    };
    attribute InetPortRange srcPortRange {
        description
           "The port range to match against the packet's
            source port."
    };
    attribute InetPortRange dstPortRange {
        description
           "The port range to match against the packet's
            destination port.";
    };
    attribute InetProtocolNumber protocol {
                    readwrite;
        access
        description
           "The protocol number to match against the packet's
            Internet Protocol version. A value of zero
            matches any IP version.";
    };
    attribute InetDiffServCodePoint dscp {
        access
                    readwrite;
        description
           "The value that the DSCP in the packet must have to
            match this entry. A value of -1 indicates that a
            specific DSCP value has not been defined and thus all
            DSCP values are considered a match.";
    }
    description
       "This class represents a generic Internet Protocol filter.
        Classes derived from this class may add attributes which
        define further filter criteria.";
};
```

<u>4</u>. Usage Examples

};

The following example shows how a TCP connection might be modelled. The class definition below could be used to derive an IP version independent MIB for representing open TCP connections.

```
module IETF-TCP {
    import IETF-INET (InetTransportEndpoint);
    // ...
```

```
typedef TcpConnectionState {
    type
            Enumeration (closed(1), listen(2), synSent(3),
                         synReceived(4), established(5),
                         finWait1(6), finWait2(7), closeWait(8),
                         lastAck(9), closing(10), timeWait(11));
    description
       "The state of a TCP connection.";
    reference
       "STD 7 (RFC 793)";
};
typedef TcpConnectionCtrl {
            Enumeration (none(0), delete(1));
    type
    description
       "The enumerated numbers defined by this type allow to control
        TCP connections. The value none(0) will be reported whenever
        an attribute of this type is read.
        The only value which may be written is delete(1). If a
        management station writes the value delete(1), then this
        has the effect of deleting the TCB (as defined in RFC 793)
        of the corresponding connection on the managed node,
        resulting in immediate termination of the connection.
        As an implementation-specific option, a RST segment may be
        sent from the managed node to the other TCP endpoint (note
        however that RST segments are not sent reliably).";
};
class TcpConnection {
    attribute InetTransportEndpoint local {
        description
           "The local endpoint of the TCP connection.";
    };
    attribute InetTransportEndpoint remote {
        description
           "The remote endpoint of the TCP connection.";
    };
    attribute TcpConnectionState state {
        access
                    readonly;
        description
           "The current state of the TCP connection.";
    };
    attribute TcpConnectionCtrl ctrl {
        access
                    readwrite;
        description
           "A control which allows to change the state of the
            TCP connection.";
```

Strauss & Schoenwaelder Expires January 18, 2002 [Page 12]

```
};
    description
        "This class contains information about a particular current
        TCP connection.";
    };
};
```

5. Security Considerations

This module does not define any management objects. Instead, it defines a set of SMIng types and classes which may be used by other SMIng modules to define management objects. These data definitions have no security impact on the Internet.

<u>6</u>. Acknowledgments

Some definitions in this document are derived from <u>RFC 2851</u> [3], which was written by M. Daniele, B. Haberman, S. Routhier and J. Schoenwaelder.

References

- [1] Strauss, F. and J. Schoenwaelder, "SMIng Next Generation Structure of Management Information", <u>draft-ietf-sming-02.txt</u>, July 2001.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", <u>RFC 2119</u>, <u>BCP 14</u>, March 1997.
- [3] Daniele, M., Haberman, B., Routhier, S. and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", <u>RFC 2851</u>, June 2000.

Authors' Addresses

Frank Strauss TU Braunschweig Bueltenweg 74/75 38106 Braunschweig Germany

Phone: +49 531 391-3266 EMail: strauss@ibr.cs.tu-bs.de URI: <u>http://www.ibr.cs.tu-bs.de/</u>

Juergen Schoenwaelder TU Braunschweig Bueltenweg 74/75 38106 Braunschweig Germany Phone: +49 531 391-3289 EMail: schoenw@ibr.cs.tu-bs.de

URI: http://www.ibr.cs.tu-bs.de/

Appendix A. OPEN ISSUES

- What else is missing? There might be more core type or class definitions that should go into the IETF-SMING-INET module.
- Are the filters sufficiently flexible? The filters probably need more work to cover more cases. Should the IETF-INET-FILTER module become a separate document?
- More examples needed? Is it useful to include more examples, e.g. on the usage of filters or subnets?
- Dscp Definition Does the InetDiffServCodePoint type definition really belong into this module?

InetAddressDNS Format - 255a or 255t? Length restriction?

- Usage of the terms Endpoint and Address Check the attribute identifiers and descriptions of InetTransportEndpoint and InetSubnet: when should the term endpoint be used, and when address?
- InetProtocolNumber and InetTransportEndpoint Should an InetProtocolNumber attribute be added to the InetTransportEndpoint?
- Undocumented typemap keyword This feature needs more work. We either define such a type casting mechanism or we add a real discriminated union to the SMIng type system.

Strauss & Schoenwaelder Expires January 18, 2002 [Page 14]

Full Copyright Statement

Copyright (C) The Internet Society (2001). 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.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

Strauss & Schoenwaelder Expires January 18, 2002 [Page 15]