SNMP Context Mapping MIB

draft-kkoushik-snmp-context-map-mib-02.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

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.

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

Abstract

This document defines a MIB module to manage the usage of SNMP contexts. Specifically, within for an SNMP agent which implements multiple copies/instances of some other MIB module, this MIB module provides a mapping between SNMP Contexts and the individual instances of that other MIB module.

Table of Contents

1. Introduction .................................................... 2
2. Terminology ..................................................... 3
3. SNMP Context mapping feature - High-Level Picture ........... 3
4. Protocol Operations ............................................ 3
5. Object Definitions ............................................. 4
6. Security Considerations ....................................... 11
7. Example of usage .............................................. 11
8. References ........................................................ 12
1. Introduction

With the advent of newer technologies, an increasing number of technologies which used to be defined such that only one instance could exist within a device, are now being implemented at a different granularity such that multiple copies/instances can exist in one device at the same time.

An excellent example for this behavior is with the OSPF-MIB (RFC 1850). When it was originally designed, there was no concept of multiple OSPF instances running on the same system and there was no built-in mechanism to handle such circumstances.

However according to Section 4.1.1 of RFC 4577, a PE router that attaches to more than one OSPF domain must run an independent instance of OSPF for each domain. Each OSPF instance is associated to a VRF (see section 3 of RFC4364). This means that OSPF-MIB must now support multiple VRF contexts and within each context the objects in the OSPF-MIB can be indexed by the same OID but represent different data.

One way to overcome this issue would be to update the OSPF-MIB to have an additional variable in the INDEX clause. This would require deprecating and re-defining just about all objects in the MIB; in effect, a modified copy of the original MIB. This change can be severely disadvantageous to existing deployments.

MPLS-LSR-STD-MIB[RFC3813], [BFD-MIB], BGP-MIB[RFC4273], [ISIS-MIB] and IP-FORWARD-MIB[RFC4292] are other MIBs which are also affected in the same way.

Another example is with the BRIDGE-MIB[RFC4188] where each Bridge entity in a system can contain different subsets of data indexed by the same OID.

A better way to overcome the issue is to use multiple SNMP contexts (see section 3.3.1 of RFC 3411). However, as and when significant usage is made of SNMP contexts (e.g., for multiple MIBs and/or multiple domains), then there is a need for the contexts themselves to be manageable. This document defines management information for this purpose.
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 BCP 14, RFC 2119 [RFC2119].

2. Terminology

This document uses terminology from the document describing the MPLS architecture [RFC3031] and from the document describing MPLS Layer-3 VPNs (L3VPN) [RFC4364], as well as the SNMP architecture [RFC3411].

Throughout this document, the use of the terms "Provider Edge (PE) and Customer Edge (CE)" or "PE/CE" will be replaced by "PE" in all cases except when a network device is a CE when used in the carrier’s carrier model.

3. SNMP Context mapping feature - High-Level Picture:

The use of SNMP contexts as a solution to the problem of accessing SNMP MIBs on a per-context basis requires no modifications to existing MIBs. Instead, this solution requires that the Network Management Station (NMS) or any other manager desiring to manage an agent, be made aware that a VPN Identifier or a Bridge Identifier or some other data distinguishing identifier can be mapped to a SNMP Context field for every request.

The MIB presented in this draft can be used to map a vacmContextName into any data distinguishing identifier which can retrieve data from the appropriate instance.

Every SNMPv3 operation acts within the context of an SNMP Context. The name of that SNMP Context is carried in the message header and serves as part of the naming structure of MIB objects (see section 3.3.1 of RFC 3411). When all of an agent’s management information is in the same SNMP Context, then the name of that Context need only be used implicitly. In contrast, when multiple SNMP Contexts are in use, then the name of a particular Context must be used explicitly to determine which items of management information are referenced by which OIDs.

For older versions of SNMP, a mapping between the SNMP community and the SNMP context can be maintained using the SNMP-COMMUNITY-MIB.

4. Protocol Operations
5. Object Definitions

SNMP-CONTEXT-MAPPING-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY,
   OBJECT-TYPE
   FROM SNMPv2-SMI

SnmpAdminString
   FROM SNMP-FRAMEWORK-MIB

MODULE-COMPLIANCE,
OBJECT-GROUP
   FROM SNMPv2-CONF

RowStatus,
StorageType
   FROM SNMPv2-TC

PwIndexType
   FROM PW-TC-STD-MIB

snmpContextMappingMIB MODULE-IDENTITY
   LAST-UPDATED        "200802140000Z"
   ORGANIZATION        "Cisco Systems Inc."
   CONTACT-INFO        "Kiran Koushik A S
                       Email: kkoushik@cisco.com
                       Thomas Nadeau
                       Email: thomas.nadeau@bt.com
                       Chinna Pellacuru
                       Email: pcn@cisco.com"

   DESCRIPTION
   "Copyright (C) The IETF Trust (2008). The initial
   version of this MIB module was published in RFC XXXX.
   -- RFC Editor: Please replace XXXX with RFC number & remove
   -- this note.

   For full legal notices see the RFC itself or see:
   http://www.ietf.org/copyrights/ianamib.html

   A single SNMP agent sometimes needs to support multiple
instances of the same MIB module, and does so through the use of multiple SNMP contexts. This typically occurs because the technology has evolved to have extra dimension(s), i.e., one or more extra data and/or identifier values which are different in the different contexts, but were not defined in INDEX clause(s) of the original MIB module. In such cases, network management applications need to know the specific data/identifier values in each context, and this MIB module provides mapping tables which contain that information.

Within a network there can be multiple Virtual Private Networks (VPNs) configured using Virtual Routing and Forwarding Instances (VRFs). Within a VPN there can be multiple topologies when Multi-topology Routing (MTR) is used. Also, Interior Gateway Protocols (IGPs) can have multiple protocol instances running on the device.

With MTR routing and VRFs, a router now needs to support multiple instances of several existing MIB modules, and this can be achieved if the router’s SNMP agent provides access to each instance of the same MIB module via a different SNMP context (see Section 3.1.1 of RFC 3411). For MTR routing and VRFs, a different SNMP context is needed depending on one or more of the following: the VRF, the topology-identifier, and the routing protocol instance. In other words, the router’s management information can be accessed through multiple SNMP contexts where each such context represents a specific VRF, a specific topology-identifier, and/or a specific routing protocol instance. This MIB module provides a mapping of each such SNMP context to the corresponding VRF, the corresponding topology, and the corresponding routing protocol instance. Some SNMP contexts are independent of VRFs, independent of a topology, or independent of a routing protocol instance, and in such a case, the mapping is to the zero length string.

We have also added the mapping to dot1dBasePort from BRIDGE-MIB and the vplsConfigIndex from VPLS-MIB as we feel that these data distinguishing identifiers will be applicable to a larger subset of the MIBs that are already defined. Similarly a mapping from the PwIndex from PWE3-PW-TC-MIB to the dot1dBasePort from BRIDGE-MIB has been defined here.

As technology evolves more we may need additional identifiers to identify the context. Then we would need to add those additional identifiers into the mapping. We must caution that since there are huge number MIB modules defined, if even a small fraction of them needs to have multiple instances in SNMP contexts, then what this
The paragraph proposes will NOT scale. We request the MIB users to judiciously choose the data distinguishing identifiers which map to a SNMP context.

Copyright (C) The IETF Trust (2008). This version of this MIB module is part of RFC XXX; see the RFC itself for full legal notices.

-- RFC Ed.: replace XXX with actual RFC number & remove this

"note

REVISION "2008021400000Z"
DESCRIPTION
"Initial version of the MIB module."
::= { experiment xxx }

snmpContextMappingMIBObjects OBJECT IDENTIFIER
::= { snmpContextMappingMIB 1 }

snmpContextMappingMIBConformance OBJECT IDENTIFIER
::= { snmpContextMappingMIB 2 }

snmpContextMappingTable OBJECT-TYPE
SYNTAX  SEQUENCE OF SNMPContextMappingEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
"This table contains information on which
snmpContextMappingVacmContextName is mapped to
which VRF, topology, and routing protocol instance.

This table is indexed by SNMP VACM context.

Configuring a row in this table for an SNMP context does not require that the context be already defined, i.e., a row can be created in this table for a context before the corresponding row is created in RFC 3415’s vacmContextTable.

To create a row in this table, a manager must set
snmpContextMappingRowStatus to either ‘createAndGo’ or
‘createAndWait’.

To delete a row in this table, a manager must set
snmpContextMappingRowStatus to ‘destroy’.
::= { snmpContextMappingMIBObjects 1 }
SYNTAX SNMPContextMappingEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Information relating to a single mapping of
snmpContextMappingVacmContextName to the corresponding VRF,
the corresponding topology, and the corresponding routing
protocol instance."
INDEX { snmpContextMappingVacmContextName }
::= { snmpContextMappingTable 1 }

SNMPContextMappingEntry ::= 
SEQUENCE {
  snmpContextMappingVacmContextName  SnmpAdminString,
  snmpContextMappingVrfName          SnmpAdminString,
  snmpContextMappingTopologyName     SnmpAdminString,
  snmpContextMappingProtoInstName    SnmpAdminString,
  snmpContextMappingBridgePort       Unsigned32,
  snmpContextMappingVplsIndex        Unsigned 32,
  snmpContextMappingStorageType      StorageType,
  snmpContextMappingRowStatus        RowStatus,
  snmpContextMappingACType           INTEGER,
  snmpContextMappingPwIndex          pwIndexType
}

snmpContextMappingVacmContextName  OBJECT-TYPE
SYNTAX     SnmpAdminString (SIZE(0..32))
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
"The vacmContextName given to the SNMP context.

This is a human readable name identifying a particular
SNMP VACM context at a particular SNMP entity.
The empty contextName (zero length) represents the
default context."
::= { snmpContextMappingEntry 1 }

snmpContextMappingVrfName  OBJECT-TYPE
SYNTAX     SnmpAdminString (SIZE(0..32))
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"The value of an instance of this object identifies
the name given to the VRF to which the SNMP context
is mapped to.

This is typically a human-readable string. This is
the same ASCII string used in the router’s console
interface to refer to this VRF."
When the value of this object is the zero length string it indicates that the SNMP context is independent of any VRF.

DEFVAL { ''H }   -- the zero length string
::= { snmpContextMappingEntry 2 }

snmpContextMappingTopoName  OBJECT-TYPE
SYNTAX     SnmpAdminString (SIZE(0..32))
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"The value of an instance of this object identifies the name given to the topology to which the SNMP context is mapped to.

This is typically a human-readable string. This is the same ASCII string used in the router’s console interface to refer to this topology.

When the value of this object is the zero length string it indicates that the SNMP context is independent of any topology."

DEFVAL { ''H }   -- the zero length string
::= { snmpContextMappingEntry 3 }

snmpContextMappingProtoInstName  OBJECT-TYPE
SYNTAX     SnmpAdminString (SIZE(0..32))
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"The value of an instance of this object identifies the name given to the protocol instance to which the SNMP context is mapped to.

This is typically a human-readable string. This is the same ASCII string used in the router’s console interface to refer to this protocol instance.

When the value of this object is the zero length string it indicates that the SNMP context is independent of any protocol instance."

DEFVAL { ''H }   -- the zero length string
::= { snmpContextMappingEntry 4 }

snmpContextMappingBridgePort  OBJECT-TYPE
SYNTAX     Unsigned32
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"When snmpContextMappingACType is set to interface(1),
the value of an instance of this object identifies
the dot1dBasePort to which the
SNMP context is mapped to.

When the value of this object is zero
it indicates that the SNMP context is independent
of dot1dBasePort."
DEFVAL { 0 }
REFERENCE "RFC 4188 - Section 4."
::= { snmpContextMappingEntry 5 }

snmpContextMappingVplsIndex OBJECT-TYPE
SYNTAX   Unsigned32
MAX-ACCESS read-create
STATUS   current
DESCRIPTION
"The value of an instance of this object identifies
vplsConfigIndex to which the SNMP context is mapped to.

When the value of this object is zero
it indicates that the SNMP context is independent
of vplsConfigIndex."
DEFVAL { 0 }
REFERENCE "draft-ietf-l2vpn-vpls-mib-00.txt"
::= { snmpContextMappingEntry 6 }

snmpContextMappingStorageType OBJECT-TYPE
SYNTAX   StorageType
MAX-ACCESS read-create
STATUS   current
DESCRIPTION
"The storage type for this conceptual row.

Conceptual rows having the value ‘permanent’ need not
allow write-access to any columnar objects in the row."
DEFVAL { nonVolatile }
::= { snmpContextMappingEntry 7 }

snmpContextMappingRowStatus OBJECT-TYPE
SYNTAX   RowStatus
MAX-ACCESS read-create
STATUS   current
DESCRIPTION
"This object facilitates the creation, modification, or
deletion of a conceptual row in this table.

::= { snmpContextMappingEntry 8 }

snmpContextMappingACType OBJECT-TYPE
SYNTAX INTEGER {
    interface (1),
    pseudoWire (2)
} 
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The value of an instance of this object identifies whether this context is applicable to an attachmentCircuit which is an interface or a pseudoWire.

If the value of this object is interface(1), then the snmpContextMappingBridgePort identifies the dot1dBasePort to which this context maps to.
If the value of this object is pseudoWire(2), then the snmpContextMappingPwIndex identifies the dot1dBasePort to which this context maps to.
When the value of snmpContextMappingVplsIndex is 0, then this object is not relevant."

::= { snmpContextMappingEntry 9 }

snmpContextMappingPwIndex OBJECT-TYPE
SYNTAX PwIndexType
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"When snmpContextMappingACType is set to pseudoWire(2), the value of an instance of this object identifies the PwIndex to which the SNMP context is mapped to. The dot1dBasePort to which this PwIndex maps to is defined in snmpContextMappingBridgePort.

When the value of snmpContextMappingVplsIndex is zero it indicates that the SNMP context is independent of snmpContextMappingPwIndex."

::= { snmpContextMappingEntry 10 }

-- Conformance

snmpContextMappingMIBCompliances
OBJECT IDENTIFIER ::= { snmpContextMappingMIBConformance 1 }

snmpContextMappingMIBGroups
OBJECT IDENTIFIER ::= { snmpContextMappingMIBConformance 2 }
-- Compliance

snmpContextMappingMIBCompliance MODULE-COMPLIANCE
 STATUS current
 DESCRIPTION
 "The compliance statement for entities which implement
 the SNMP-CONTEXT-MAPPING-MIB."

MODULE
 MANDATORY-GROUPS {
  snmpContextMappingDataGroup
 }

OBJECT snmpContextMappingVrfName
 MIN-ACCESS read-only
 DESCRIPTION "Write access is not required."

OBJECT snmpContextMappingTopologyName
 MIN-ACCESS read-only
 DESCRIPTION "Write access is not required."

OBJECT snmpContextMappingProtoInstName
 MIN-ACCESS read-only
 DESCRIPTION "Write access is not required."

OBJECT snmpContextMappingBridgePort
 MIN-ACCESS read-only
 DESCRIPTION "Write access is not required."

OBJECT snmpContextMappingVplsIndex
 MIN-ACCESS read-only
 DESCRIPTION "Write access is not required."

OBJECT snmpContextMappingStorageType
 MIN-ACCESS read-only
 DESCRIPTION "Write access is not required."

OBJECT snmpContextMappingRowStatus
 MIN-ACCESS read-only
 DESCRIPTION "Create/delete/modify access to the
 snmpContextMappingTable is not required."

::= { snmpContextMappingMIBCompliances 1 }
6. Security Considerations

The MIB module described in this document in association with SNMP-COMMUNITY-MIB and SNMPv3 framework is useful for accessing subsets of data based on various data distinguishing identifiers.

There are objects in this MIB which are configurable via SNMP. If these are configured incorrectly, there can be potential data access violations.

There are a number of management objects defined in these MIB modules with a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability.
7. Example of usage:

In `snmpContextMappingTable`:

<table>
<thead>
<tr>
<th>snmpContextMappingContextName</th>
<th>&quot;contextA&quot;{Index}</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmpContextMappingVrfName</td>
<td>&quot;customerA&quot;</td>
</tr>
<tr>
<td>snmpContextMappingTopologyName</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>snmpContextMappingProtoInstName</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>snmpContextMappingBridgePort</td>
<td>0</td>
</tr>
<tr>
<td>snmpContextMappingVplsIndex</td>
<td>0</td>
</tr>
<tr>
<td>snmpContextMappingStorageType</td>
<td>(2)volatile</td>
</tr>
<tr>
<td>snmpContextMappingRowStatus</td>
<td>1(active)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>snmpContextMappingContextName</th>
<th>&quot;contextB&quot;{Index}</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmpContextMappingVrfName</td>
<td>&quot;customerB&quot;</td>
</tr>
<tr>
<td>snmpContextMappingTopologyName</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>snmpContextMappingProtoInstName</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>snmpContextMappingBridgePort</td>
<td>0</td>
</tr>
<tr>
<td>snmpContextMappingVplsIndex</td>
<td>0</td>
</tr>
<tr>
<td>snmpContextMappingStorageType</td>
<td>(2)volatile</td>
</tr>
<tr>
<td>snmpContextMappingRowStatus</td>
<td>1(active)</td>
</tr>
</tbody>
</table>

In `ospfHostTable` from RFC 1850:

[For OSPF Instance on VRF "customerA"]

<table>
<thead>
<tr>
<th>ospfHostIpAddress</th>
<th>&quot;1.2.3.4&quot;{Index}</th>
</tr>
</thead>
<tbody>
<tr>
<td>ospfHostTOS</td>
<td>8</td>
</tr>
<tr>
<td>ospfHostMetric</td>
<td>&quot;ab&quot;</td>
</tr>
<tr>
<td>ospfHostStatus</td>
<td>&quot;1(enabled)&quot;</td>
</tr>
<tr>
<td>ospfHostAreaID</td>
<td>&quot;1.1.1.1&quot;</td>
</tr>
</tbody>
</table>

[For OSPF Instance on VRF "customerB"]

<table>
<thead>
<tr>
<th>ospfHostIpAddress</th>
<th>&quot;1.2.3.4&quot;{Index}</th>
</tr>
</thead>
<tbody>
<tr>
<td>ospfHostTOS</td>
<td>8</td>
</tr>
<tr>
<td>ospfHostMetric</td>
<td>&quot;ab&quot;</td>
</tr>
<tr>
<td>ospfHostStatus</td>
<td>&quot;1(enabled)&quot;</td>
</tr>
<tr>
<td>ospfHostAreaID</td>
<td>&quot;1.1.1.2&quot;</td>
</tr>
</tbody>
</table>

In the above case, we can use the context mapping to distinguish data between different OSPF instances even though the OID indexes are the same.

8. References

8.1. Normative References
8.2. Informative References

9. Acknowledgments

We would like to thank Keith McCloghrie for his insightful comments and expert suggestions.

We would also like to thank Chinna Narasimha Reddy and Madhavi.

Rohit Mediratta has contributed to the VPLS updates to this draft and we would like to thank him for that.

10. IANA Considerations

-- (Note to RFC-Editor:)
-- We request that you assign contiguous RFC numbers to the
-- IANA is requested to root MIB objects in the MIB module
-- contained in this document under the transmission subtree.
--

11. Authors’ Addresses

Thomas D. Nadeau
BT
BT Centre
81 Newgate Street
EC1A 7AJ
London
Email: tom.nadeau@bt.com

A S Kiran Koushik
Cisco Systems Inc
12515 Research Blvd, Bldg 4
Austin TX 78759
Email: kkoushik@cisco.com

Full Copyright Statement

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents
(http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.