

GetRange Operation for the Simple Network Management Protocol (SNMP)
draft-irtf-nmrg-snmpp-getrange-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](#).

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 May 28, 2004.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

This memo defines the GetRangeRequest-PDU for the Simple Network Management Protocol (SNMP). The GetRangeRequest-PDU enables a command generator to read a large amount of management information with a minimum number of protocol operations and without having to guess a suitable repetition count and without reading data beyond the information the command generator is interested in (overshoot effect).

Table of Contents

1.	Introduction	3
2.	Definitions	4
3.	Elements of Procedure	4
4.	Examples	7
4.1	Retrieving Columns from a Single Table	7
4.2	Retrieving Columns from Multiple Tables	7
4.3	Retrieving Tables with Holes	8
5.	Discussion	9
6.	Acknowledgements	9
	Normative References	10
	Informative References	10
	Author's Address	11
	Intellectual Property and Copyright Statements	12

1. Introduction

The second version of the protocol operations for the Simple Network Management Protocol (SNMP) defined in [RFC 3416](#) [[RFC3416](#)] includes the GetBulkRequest-PDU for reading large amounts of management information. The GetBulkRequest-PDU can be seen as a generalization of the GetNextRequest-PDU since it basically asks the command generator to perform repeated get-next operations internally. The max-repetitions parameter in the GetBulkRequest-PDU defines the maximum number of repetitions. The command responder might perform fewer repetitions taking into account constraints for the response message or other processing constraints.

The usage of the GetBulkRequest-PDU has shown some problems. First, command generators often do not know a suitable value for the max-repetitions parameter. If the parameter is chosen too small, more requests are needed which reduces the overall benefit. If the max-repetitions parameter is set too large, variables might be retrieved that are not interesting for the command generator and subsequently discarded (overshoot effect). This can have severe effects if the retrieval of the unwanted data internally within the command generator is expensive [[STBULK](#)].

This memo introduces the GetRangeRequest-PDU which addresses these issues with the GetBulkRequest-PDU while otherwise following the same design principles:

1. The GetRangeRequest-PDU operates like all other SNMP PDUs on a lexicographically ordered list of variables and does not assume knowledge about conceptual tables.
2. The GetRangeRequest-PDU uses the same PDU format as all other SNMP PDUs.
3. The GetRangeRequest-PDU keeps the non-repetitions parameter of the GetBulkRequest-PDU since it is often required to retrieve the value of sysUpTime in addition to other variables (especially for counter variables).
4. The GetRangeRequest-PDU requires no knowledge of the MIBs supported by an application and it can be used with existing instrumentations and extensible agent protocols such as AgentX .
5. The GetRangeRequest-PDU deals with holes or columns from conceptual tables with different number of instances in such a way that no uninteresting data or fill data is transmitted.

The basic improvement over the GetBulkRequest-PDU is to replace the

max-repetitions parameter with a number of bumper object identifiers (OIDs) which define where the processing should stop, thus avoiding the need to estimate a suitable value for the max-repetitions parameter. The idea to use bumper OIDs (or short bumpers) was first introduced in [\[STBUMP\]](#).

The GetRangeRequest-PDU belongs to the Read class and the Confirmed class as defined in [RFC 3411](#) [\[RFC3411\]](#).

2. Definitions

```
NMRG-SNMP-GETRANGE-PDU DEFINITIONS ::= BEGIN

IMPORTS PDU, VarBindList, max-bindings
    FROM SNMPv2-PDU;

GetRangeRequest-PDU ::=
    [9] IMPLICIT                      -- xxx IANA, can we use [9] ??
        GetRangePDU

GetRangePDU ::=                      -- identical in structure to PDU
    SEQUENCE {
        request-id
            INTEGER (-2147483648..2147483647),

        non-repeaters
            INTEGER (0..max-bindings),

        bumpers                      -- number of bumper varbinds
            INTEGER (0..max-bindings),

        variable-bindings            -- values are ignored
            VarBindList
    }

END
```

3. Elements of Procedure

A GetRangeRequest-PDU is generated and transmitted at the request of an application. The purpose of the GetRangeRequest-PDU is to request the transfer of a potentially large amount of data, including, but not limited to, the efficient and rapid retrieval of large tables.

Upon receipt of a GetRangeRequest-PDU, the receiving SNMP entity processes each variable binding in the variable-binding list to produce a Response-PDU with its request-id field having the same

value as in the request.

The values of the non-repeaters (N) and bumpers (B) field together with the first elements in the variable bindings specify the processing requested. One variable binding in the Response-PDU is requested for the first N variable bindings in the request. The following B elements in the variable bindings are so called bumpers which split the list of lexicographically ordered variables into sublists. Multiple variable bindings are requested for each of the remaining (R) variable bindings in the request.

If N is greater than zero, the first through the (N)-th variable bindings of the Response-PDU are each produced as follows:

1. The variable is located which is in the lexicographically ordered list of the names of all variables which are accessible by this request and whose name is the first lexicographic successor of the variable binding's name in the incoming GetRangeRequest-PDU. The corresponding variable binding's name and value fields in the Response-PDU are set to the name and value of the located variable
2. If the requested variable binding's name does not lexicographically precede the name of any variable accessible by this request, i.e., there is no lexicographic successor, then the corresponding variable binding produced in the Response-PDU has its value field set to "endOfMibView", and its name field set to the variable binding's name in the request.

If B is greater than zero, the next B variable bindings are conceptually copied into the bumpers list (BL). If R is non-zero, the remaining variable bindings in the request are conceptually copied into the repeater list (RL). The (N + 1)-th and subsequent variable bindings of the Response-PDU are each produced in a similar manner:

1. Select the first element of the repeater list RL and the first element of the bumpers list BL.
2. The variable is located which is in the lexicographically ordered list of the names of all variables which are accessible by this request and whose name is the first lexicographic successor of the selected variable binding's name in the repeater list RL.
3. If the name of the located variable is lexicographically smaller than the name of the selected variable binding's name in the bumpers list BL, then the variable binding's name and value fields in the Response-PDU are set to the name and value of the located variable. The selected element in the repeater list RL is

updated with the name of the located variable.

4. Otherwise, the variable binding's name field in the Response-PDU is set to the name of the selected bumpers variable binding and the variable binding's value field is set to "endOfMibView". The selected elements are marked as "done" in the repeater list RL and the bumpers list BL.
5. The next element in the repeater list RL and the next element in the bumpers list BL which is not yet marked "done" is selected. If the end of the lists is reached, the first or a subsequent element not yet marked "done" is selected. Processing stops if no such element exists. Goto step number 2 if new elements were selected.

While the GetRangeRequest-PDU retrieves all the requested lists of variables, the response may be generated with a lesser number of variable bindings (possibly zero) for either of two reasons.

1. If the size of the message encapsulating the Response-PDU containing the requested number of variable bindings would be greater than either a local constraint or the maximum message size of the originator, then the response is generated with a lesser number of variable bindings. This lesser number is the ordered set of variable bindings with some of the variable bindings at the end of the set removed, such that the size of the message encapsulating the Response-PDU is approximately equal to but no greater than either a local constraint or the maximum message size of the originator. Note that the number of variable bindings removed has no relationship to the values of N, B, or R.
2. In the event that the processing of a request with many repetitions requires a significantly greater amount of processing time than a normal request, then a command responder application may terminate the request with less than the full number of repetitions, providing at least one repetition is completed.

If the processing of any variable binding fails for a reason other than listed above, then the Response-PDU is re-formatted with the same values in its request-id and variable-bindings fields as the received GetRangeRequest-PDU, with the value of its error-status field set to "genErr", and the value of its error-index field is set to the index of the variable binding in the original request which corresponds to the failed variable binding.

Otherwise, the value of the Response-PDU's error-status field is set to "noError", and the value of its error-index field to zero.

The generated Response-PDU (possibly with an empty variable-bindings field) is then encapsulated into a message. If the size of the resultant message is less than or equal to both a local constraint and the maximum message size of the originator, it is transmitted to the originator of the GetRangeRequest-PDU. Otherwise, the snmpSilentDrops counter defined in [RFC 3418](#) [[RFC3418](#)] is incremented and the resultant message is discarded.

4. Examples

The following examples assume that an agent implements the ifTable and ifXTable of the IF-MIB [[RFC2863](#)] with five rows (identified by the ifIndex values 1 to 5) and the ipNetToMediaTable of the IP-MIB [[RFC2011](#)] with two rows (identified by 192.0.2.1, 192.0.2.25).

4.1 Retrieving Columns from a Single Table

Retrieve the values for ifAdminStatus and ifOperStatus for all rows in the ifTable (assuming that only seven variable bindings fit into a Response-PDU).

```
GetRangeRequest [ non-repeaters = 1, bumpers = 2 ]
  ( sysUpTime,
    ifOperStatus, ifLastChange,
    ifAdminStatus, ifOperStatus )
```

```
Response ( sysUpTime.0 = "12",
  ifAdminStatus.1 = "up", ifOperStatus.1 = "up",
  ifAdminStatus.2 = "up", ifOperStatus.2 = "up",
  ifAdminStatus.3 = "up", ifOperStatus.3 = "down" )
```

```
GetRangeRequest [ non-repeaters = 1, bumpers = 2 ]
  ( sysUpTime,
    ifOperStatus, ifLastChange,
    ifAdminStatus.3, ifOperStatus.3 )
```

```
Response ( sysUpTime.0 = "13",
  ifAdminStatus.4 = "up", ifOperStatus.4 = "down",
  ifAdminStatus.5 = "up", ifOperStatus.5 = "down",
  ifOperStatus = [endOfMibView], ifLastChange = [endOfMibView] )
```

4.2 Retrieving Columns from Multiple Tables

Retrieve the values of ifDescr, ifName, ipAdEntAddr, ipAdEntIfIndex, ipAdEntNetMask for all rows in the ifTable and the ipAdEntTable (assuming that only nine variable bindings fit into a Response-PDU).


```
GetRangeRequest [ non-repeaters = 1, bumpers = 4 ]
  ( sysUpTime,
    ifType, ifInMulticastPkts, ipAdEntNetMask, ipAdEntBcastAddr,
    ifDescr, ifName, ipAdEntIfIndex, ipAdEntNetMask )

Response ( sysUpTime = "32",
  ifDescr.1 = "lo", ifName.1 = "lo",
  ipAdEntIfIndex.127.0.0.1 = "1",
  ipAdEntNetMask.127.0.0.1 = "255.0.0.0",
  ifDescr.2 = "eth0", ifName.2 = "eth0",
  ipAdEntIfIndex.192.0.2.1 = "2",
  ipAdEntNetMask.192.0.2.1 = "255.255.255.0" )

GetRangeRequest [non-repeaters = 1, bumpers = 4 ]
  ( sysUpTime,
    ifType, ifInMulticastPkts, ipAdEntNetMask, ipAdEntBcastAddr,
    ifDescr.2, ifName.2,
    ipAdEntIfIndex.192.0.2.1, ipAdEntNetMask.192.0.2.1 )

Response ( sysUpTime = "33",
  ifDescr.3 = "eth1", ifName.3 = "eth1",
  ipAdEntNetMask = [endOfMibView], ipAdEntNetMask = [endOfMibView],
  ifDescr.4 = "eth2", ifName.4 = "eth2",
  ifDescr.5 = "eth3", ifName.5 = "eth3" )

GetRangeRequest [non-repeaters = 1, bumpers = 2 ]
  ( sysUpTime,
    ifType, ifInMulticastPkts,
    ifDescr.5, ifName.5 )

Response ( sysUpTime = "34",
  ifType = [endOfMibView], ifInMulticastPkts = [endOfMibView] )
```

4.3 Retrieving Tables with Holes

Retrieve the values of ifDescr and ifAlias. Assume that the instance ifAlias.2 does not exist in the MIB view (assuming that twelve variable bindings fit into a Response-PDU).

```
GetRangeRequest [ non-repeaters = 1, bumpers = 2 ]
  ( sysUpTime,
    ifType, ifCounterDiscontinuityTime,
    ifDescr, ifAlias )

Response ( sysUpTime = "42",
  ifDescr.1 = "lo", ifAlias.1 = "loopback interface",
  ifDescr.2 = "eth0", ifAlias.3 = "",
```



```
ifDescr.3 = "eth1", ifAlias.4 = "",  
ifDescr.4 = "eth2", ifAlias.5 = "",  
ifDescr.5 = "eth3", ifCounterDiscontinuityTime = [endOfMibView],  
ifType = [endOfMibView] )
```

5. Discussion

Many proposals have been developed so far to improve the bulk retrieval performance characteristics of SNMP. The proposed solutions range from rather simple to rather drastic changes and additions to the existing protocol. The new PDU proposed in this memo certainly falls into the first category. By using this new PDU in combination with OID compression techniques and transports supporting larger message sizes (such as SNMP over TCP [[RFC3430](#)]), significant performance improvements can be achieved since (a) the manager does not have to guess a suitable value for the max-repetitions parameter of the GetBulkRequest-PDU and holes are handled in a way which avoids to return variables which are of no value for the command generator.

Dave Perkins proposed a GetColsRequest-PDU which is more powerful than the GetRangeRequest-PDU described in this memo. The GetColsRequest-PDU operates on conceptual rows and supports a filter expression to select the desired rows. The response is returned in a compact encoding which suppresses redundant OID components. However, the GetColsRequest-PDU does not allow to retrieve data from multiple arbitrary tables with a single request and it requires that the SNMP engine has knowledge about table indexing to be able to extract values of auxiliary objects by unpacking instance identifiers.

Wes Hardacker proposed a set of object-oriented PDUs which provide many new features. The processing of these PDUs requires MIB knowledge in the SNMP engine. The EOS working group of the IETF was not able to achieve consensus on these new object-oriented PDUs.

6. Acknowledgements

The Network Management Research Group (NMRG) has discussed bulk data retrieval improvements for SNMP in several meetings. This document was inspired by many ideas that came up during these discussions. The submissions to the EOS IETF working group further helped to shape this document.

The idea to introduce bumper objects to mark the end of lists was first described by M. Malowidzki [[STBUMP](#)].

Some paragraphs and phrases are taken from the second version of the protocol operations for the Simple Network Management Protocol (SNMP)

[RFC3416] written by R. Presuhn, J. Case, K. McCloghrie, M. Rose, and S. Waldbusser.

Normative References

- [RFC3411] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, [RFC 3411](#), December 2002.
- [RFC3416] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, [RFC 3416](#), December 2002.
- [RFC3418] Presuhn, R., Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, [RFC 3418](#), December 2002.

Informative References

- [STBULK] Sprenkels, R. and J. Martin-Flatin, "Bulk Transfers of MIB Data", Simple Times 7(1), March 1999.
- [STBUMP] Malowidzki, M., "GetBulk Worth Fixing", Simple Times 10(1), December 2002.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", [RFC 2863](#), June 2000.
- [RFC2011] McCloghrie, K., "SNMPv2 Management Information Base for the Internet Protocol using SMIV2", [RFC 2011](#), November 1996.
- [RFC3430] Schoenwaelder, J., "Simple Network Management Protocol (SNMP) over Transmission Control Protocol (TCP) Transport Mapping", [RFC 3430](#), December 2002.

Author's Address

Juergen Schoenwaelder
International University Bremen
Campus Ring 1
28725 Bremen
Germany

Phone: +49 421 200-3587

EMail: j.schoenwaelder@iu-bremen.de

Intellectual Property Statement

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.

Full Copyright Statement

Copyright (C) The Internet Society (2003). 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 assignees.

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.

Acknowledgment

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