

# jSNMP Enterprise™ 3.2 FAQ

## Basic Information

### **Q – What is SNMP?**

A – SNMP is the acronym for Simple Network Management Protocol. SNMP is an Internet Standard Protocol designed to give a user the capability of remotely managing devices from different vendors by polling and setting values and monitoring network events. SNMP network management consists of two major components: a Manager such as a jSNMP application or HP OpenView and an Agent running in each of the devices you wish to manage. For more information see [SNMP & CMIP: An Introduction to Network Management](#).

### **Q – What is the jSNMP Package?**

A – jSNMP is a highly optimized and scalable Java package for communicating with SNMP devices and services. It is written entirely in Java and is completely portable. jSNMP provides complete SNMPv1, SNMPv2c, and SNMPv3 support including notification handling.

### **Q – What differentiates jSNMP from other SNMP packages?**

A – Since jSNMP is 100% pure Java, it has the advantage over non-Java SNMP packages in that it will run on any Java 1.1 or later compatible Virtual Machine. To support SNMPv3 encryption, a [Java Cryptography Extension](#) (JCE) must be installed along with a Java 1.2.1 or later Virtual Machine. Additionally, jSNMP presents a set of simple interfaces to developers without requiring them to know the finer details of SNMP. These same interfaces can be accessed from across the network via a RMI or CORBA 2.0 Client. In addition, jSNMP Enterprise is scalable to allow many clients to have concurrent access to the jSNMP server. In fact, benefits of the optimizations made in jSNMP, such as combining multiple requests into one network packet, are greater when using multiple clients. jSNMP is able to do all of this and still be highly optimized and scalable, much more so than traditional SNMP stacks.

### **Q – What distinguishes jSNMP Enterprise v3.2 and later from earlier versions of jSNMP?**

A – Beginning with version 3.0, jSNMP Enterprise added SNMPv2c and SNMPv3 functionality plus the ability to generate traps and informs. jSNMP Enterprise 3.0 also added remote access to jSNMP services via RMI. jSNMP Enterprise 3.0 also introduced optimizations to receive large *bursts* of traps without loss and to allow queries to be made with object names (*sysContact.0*) instead of OIDs (*1.3.6.1.2.1.1.4.0*). Versions since 3.0 incorporate all of the technology and features in the previous versions. Applications written to the interfaces of previous versions of jSNMP will work with the current version but will be limited to making SNMPv1 queries and receiving SNMPv1 traps.

### **Q – What is jMIBC?**

A – Since SNMP Object Identifiers (OID) are difficult to remember, users may prefer to make a SNMP request in a jSNMP Enterprise application with the name associated with an OID instead of the OIDs dotted decimal notation (e.g., *ifAdminStatus.1* instead of *1.3.6.1.2.1.2.2.1.7.1*). jMIBC takes a series of MIB files and produces a dictionary file that is used by the jSNMP Enterprise `SnmplibService` to translate OIDs to/from common names, to retrieve an OID's status, access, type, *abstract* type, and description, and to translate OID enumerated values. Note that beginning with jSNMP Enterprise 3.2, it is no longer necessary to precompile MIBs to dictionary files with jMIBC, as MIBs can be loaded at runtime with the new `jMIBC.loadMib()` method.

### **Q – What is jSNMP Enterprises' support policy?**

A – TECHNICAL SUPPORT is provided, free of additional charge, via electronic mail. Technical questions along with any error codes, exceptions, source code, system configuration, JDK and JVM version numbers, etc. should be sent directly to: [support@jSNMP.com](mailto:support@jSNMP.com). The more relevant information you can provide, the better we can support you.

**Q – What is jSNMP Enterprise' Update and Upgrade policy?**

A – UPDATES are defined as "Product Maintenance Releases" which include, but are not limited to, bug fixes and incompatibility resolutions. These releases are included with the jSNMP license at no additional charge. UPGRADES are defined as "Major Product Releases" which include, but are not limited to, feature enhancement and or additions. These releases are not included in the original license fees. Upgrade pricing policies will be set at the time of the Upgrade release.

**Q – Is it possible to obtain an evaluation copy of the jSNMP package with an expiration date longer than 30 days for use in my university course work?**

A – We offer free extended licenses under the jSNMP Enterprise Educational License Program for educators and their students using jSNMP Enterprise in organized classes, independent study projects, unfunded research, and other not-for-profit educational activities. For further information, visit our web site at [http://www.jsnmp.com/products.html#educational\\_position](http://www.jsnmp.com/products.html#educational_position).

**Q – Does jSNMP support all platforms without any modification? Our target platforms are Linux, Solaris, BSD, SCO, AIX, HP UX, and Windows NT. If not, what issues should be taken care of?**

A – jSNMP Enterprise will run on any OS that has a Java VM that supports basic TCP/IP networking. Note that jSNMP Enterprise requires a JDK 1.1.x or later compatible development and runtime environment. To support DES encryption in SNMPv3 requests, a Java Cryptography Extension (JCE) must be installed along with a JDK 1.2.1 or later compatible development and runtime environment. If the encryption feature of SNMPv3 is not used, a JCE is not required. The additional RMI interfaces require a JDK 1.2.x or later compatible development and runtime environment or any other Java development and runtime environment with RMI support. An RMI environment is not required for local operation.

**Q – AdventNet sent me a document with a comparison with your package. Do you care to comment on that comparison? Do you have your own comparison that describes YOUR advantages? The following points of comparison are of special interest for us: MIB browser inclusion, storing SNMPv3 parameters in a database, and support through the HTTP protocol.**

A – We are reluctant to get in feature-by-feature comparison debate with AdventNet's marketing blurb. We do not include a MIB browser, store SNMPv3 parameters in a database (this is something you can easily do yourself), nor support SNMP via HTTP (jSNMP Enterprise does support communication via RMI and CORBA, and we would recommend the use of RMI for any remote Java support). Note that it does not mention any lack of SNMP support, but a lack of utilities that you may or may not require. Note that we purposely have not included a myriad of features (MIB browser, database support, etc.) to jSNMP Enterprise in order to keep the price as low as possible. We suggest if any of the features of AdventNet's product are *must haves* and you are willing to pay their price (initial cost plus redistribution costs), that you choose their product. If you require solid SNMPv1/2/3 support in Java at a reasonable price (\$995.00 USD per developer which includes a royalty-free runtime redistribution), we think jSNMP Enterprise will be suitable for your needs.

**Q – Is jSNMP J2ME-compliant? We would like to use it on a PDA, so it should be able to run any profile/configuration you need. We're looking at CDC/Personal and CLDC/PDA.**

A – jSNMP uses the J2SE java.net package. As this is not available in CLDC, the standard release of jSNMP wouldn't be compatible with that configuration nor any profile based on that configuration. I am not familiar with CDC, so I can't advise you on that topic. We do sell a source license so you should be able to port jSNMP to whichever configuration or profile you end up with. However, I can't advise you on how much work this would entail.

**Q – We are interested in using jSNMP in a control/monitoring system via a Java Applet. One of our concerns is Jar size requirements, as the Java Applet needs to be downloaded via HTTP, sometimes over a modem. What are the size requirements for jSNMP, just for basic SNMP communication?**

A – The jSNMPEnterprise.jar (677KB) is the full jSNMP client and server implementation with RMI and Corba support. If your Applet is not a jSNMP RMI or Corba client, use the jSNMP.jar (413KB). If it is a jSNMP RMI client, you can use the jSNMPRmiClient.jar (632KB). If it is a jSNMP Corba client, you can use the jSNMPCorbaClient.jar (458 KB).

**Q – I am evaluating jSNMP for use within our product. I am particularly interested in SNMPv3 support. Does jSNMP support discovery and synchronization. i.e. does it allow you to synchronize with an authoritative engine so that requests/traps sent to that engine are considered timely?**

A – jSNMP Enterprise supports both V3 discovery and synchronization (timeliness). When a session is created, the msgAuthoritativeEngineID is fetched along with the msgAuthoritativeEngineBoots and msgAuthoritativeEngineTime. The SnmpV3WalkerTest.java example illustrates this behavior.

## Using jSNMP

**Q – Would you please help me with the installation of jSNMP?**

A – You should have received a jSNMP.zip file as the main distribution. All that is required to install jSNMP is to unzip it. The user's manual will be in the doc directory and will help you getting started.

**Q – I see from your examples how to create an SNMP Manager. Do you have any examples of how to use your framework from a managed application, i.e., Agent side code using varBinds, initiating traps etc?**

A – jSNMP Enterprise is a manager-only toolkit. There is no support for building agents. However, it does have the ability to send traps and informs.

**Q – I am trying to use the SnmpVarBind.getValue() method. I know it returns an Object, but I don't know how to convert the value received to its equivalent String. Please help.**

A – Use the SnmpVarBind.getStringValue() method. If you wish to use the object returned by the SnmpVarBind.getStringValue() method, use the SnmpVarBind.getType() method to retrieve its type. The user's manual contains a table that maps the returned type to its Java object type.

**Q – I can't extract the values of physical address type from an SnmpVarBind. Please help.**

A – Physical addresses are traditionally represented as OCTET STRING in SNMP. When an SnmpCustomer.deliverSuccessfulOrder() method delivers an SnmpVarBind object, the varBind value can be retrieved via its getValue() method, and its type can be determined by its getType() method. OCTET STRING (SnmpConstants.ASN\_OCTSTR) type objects are represented as byte[]. Cast the object returned by getValue() method with byte[]. If your *physical address* value is represented by a base SNMP object other than an OCTET STRING, you will have to appropriately cast the object returned by the getValue() method.

**Q – I want to make requests with object names rather than OIDs. How do I go about it?**

A – jSNMP 3.x has the ability to use RFC1213-MIB (MIB-II) object names. The example application `SnmpV1GetSysInfo.java` illustrates how to retrieve the RFC1213-MIB objects `sysName`, `sysContact`, `sysLocation`, `ifDescr`, and `ifAdminStatus`, as well as load the HOST-RESOURCES-MIB.my MIB file at runtime and retrieve the HOST-RESOURCES-MIB objects `hrSystemUptime`, `hrSystemDate`, `hrSystemInitialLoadDevice`, `hrSystemInitialLoadParameters`, `hrSystemNumUsers`, `hrSystemProcesses`, `hrSystemMaxProcesses`, and `hrMemorySize`. This application illustrates the use of the `SnmpMIBService` interface, which can be used to translate OIDs to/from common names, to retrieve an OID's status, access, type, *abstract* type, and description, and to translate OID enumerated values. To use the `SnmpMIBService` interface, please refer to the *MIBs* section of the [jSNMP Enterprise User's Guide](#). jSNMP Enterprises also provides `jMIBC`, a Java-based MIB Compiler, which produces `SnmpMIBService` compatible dictionary files (i.e. `.jmib` files) from MIBs.

**Q – Is there any way to generate an SNMP trap with jSNMP?**

A – You can send traps and informs via the `placeTrapOrder()` and `placeInformOrder()` methods of the `SnmpService` interface. The `SnmpV1TrapSenderTest.java` file in the examples illustrates how to send traps.

**Q – Is it possible to use jSNMP with Servlets?**

A – jSNMP is a pure Java implementation. Many customers have created Servlets with jSNMP.

**Q – I am developing a product using jSNMP and would like to send out a broadcast on the local subnet and get responses back from all agents on the subnet. Is this possible?**

A – There is no interface within jSNMP to “broadcast” SNMP requests.

**Q – I want to use SNMPv3 encryption. What do I need to do?**

A – jSNMP supports encryption in SNMPv3 requests. To support encryption, a [Java Cryptography Extension](#) (JCE) must be installed. [JCE providers](#) other than Sun's are available. Note that the JCE requires a JDK/JRE or version 1.2.1 or later. If the encryption feature of SNMPv3 is not used, a JCE is not required.

**Q – I want to run a jSNMP SNMPv3 applet in a browser. What do I need to do?**

A – jSNMP applets that make SNMPv3 requests using authentication and/or encryption may fail by throwing a `java.lang.NoClassDefFoundError` exception, as the `MessageDigest` class (Netscape) is not available when using authentication and the JCE package (Netscape and Internet Explorer) is not available when using authentication and/or encryption. The solution is to install a [Java Plug-in](#) in the Browser using JRE 1.2.2 or later and a JCE provider for authentication and encryption.

**Q – I have a multi-get request with different OIDs. What happens if the some of the get requests are in the cache and some are not? My assumption would be that only the requests that are not in the cache get generated. Is this correct, or will the whole get request be sent out?**

A – The cache is granular as you suspect. Each OID is evaluated separately and if it is available in the local cache, that OID will not be re-requested. OIDs that are missing from the cache will be requested. For a cache hit to occur, the following parameters must match:

- OID
- host
- port
- security info
- order info

That is, in order for a value to be available for a cache hit, it must have been requested with all of the above parameters equivalent to the new request's parameters.

**Q – What is the difference between setting the lCacheThreshold argument to 0 in the SnmpOrderInfo class and making atomic requests by setting the bAtomic argument to true in the SnmpService.placeXXXOrder methods?**

A – Setting lCacheThreshold to 0 in the SnmpOrderInfo class means that the order will not be retrieved from the cache, though its result will be put in the cache. It also means that the order MAY not be atomic, that is, the order MAY be split into multiple PDUs or combined into other PDUs. Atomic requests WILL not be retrieved from the cache, WILL not be put in the cache, and WILL be sent in one PDU.

**Q – I have a question about the responses to placeSetOrder() request. In SNMP a SET request is atomic. Why must I specify it in the placeSetOrder() call?**

A – The bAtomic argument to the placeSetOrder() method (if true) informs the jSNMP engine that you wish to send all of the OID SET requests in the same PDU. You should specify true in most cases, as it matches the SNMP specification for SETs. Setting this argument to false allows jSNMP to split up your SET request into multiple PDUs if it desires. This allows you to specify more OIDs in your placeSetOrder() than could possibly fit in a single PDU.

**Q – I see that you now have an option that disables the cache. Why would I want to disable it?**

A – The cache in jSNMP Enterprise is optimized for servicing large numbers of jSNMP clients. It does improve latency and reduce network traffic when cache hits occur. However, some customers have reported that under continuous heavy load, the memory used by the cache can become quite large (multiple megabytes). It has also been noted that not all SNMP management systems benefit from a cache (i.e. cache hits are specifically avoided). Setting the system property `jsnmp.cacheDisable=true` will conserve both memory and CPU. This option is recommended if the cache is not perceived as providing specific benefit or if resource problems occur under heavy load.

**Q – We are working on internationalizing our product, and a question came up in regards to jSNMP. Is there a way to retrieve the character encoding that the SNMP Agent is talking in? I noticed that jSNMP return Strings represented in ASCII bytes from SnmpVarBind. Am I correct in assuming that the byte values are unaltered before this point? If the SNMP Agent sends bytes in Shift\_JIS, are they delivered in SnmpVarBind as is?**

A – jSNMP does not have the ability to determine what character set the remote agent is using. It does translate different character sets correctly, if the file encoding parameter is set correctly. For more information about encodings, see Sun's tutorial on [internationalization](#).

**Q – We have a problem with an IP address that uses DHCP. I am using the DNS name to access SNMP. The provider switches IP addresses every day. What happens is that the old address is remembered by jSNMP until Tomcat is rebooted. Do you have any suggestions for this problem?**

A – When created, jSNMP's SnmpAuthoritativeSession simply calls `InetAddress.getByName()` on the hostname and stores the result. `InetAddress.getByName()` is only called once during the lifetime of the SnmpAuthoritativeSession in order to improve performance. *The InetAddress class has a cache to store successful as well as unsuccessful host name resolutions. The positive caching is there to guard against DNS spoofing attacks; while the negative caching is used to improve performance. By default, the result of positive host name resolutions are cached forever, because there is no general rule to decide when it is safe to remove cache entries. The result of unsuccessful host name resolution is cached for a very short period of time (10 seconds) to improve performance. Under certain circumstances where it can be determined that DNS spoofing attacks are not possible, a Java security property can be set to a different Time-to-live (TTL) value for positive caching. Likewise, a system admin can configure a different negative caching TTL value when needed. Thus, you will have to create a new SnmpAuthoritativeSession and adjust the TTL value for positive caching.*

**Q – We would like to make jSNMP requests synchronously. How do we implement blocking requests?**

A – While we don't have a specific example for you, you will find it quite easy to implement synchronous requests. First, your `SnmCustomer.deliverSuccessfulOrder()` and `SnmCustomer.deliverFailedOrder()` methods need to call the `notify()` method (see [http://java.sun.com/j2se/1.3/docs/api/java/lang/Object.html#notify\(\)](http://java.sun.com/j2se/1.3/docs/api/java/lang/Object.html#notify())) and you need to implement the `wait()` method (see [http://java.sun.com/j2se/1.3/docs/api/java/lang/Object.html#wait\(\)](http://java.sun.com/j2se/1.3/docs/api/java/lang/Object.html#wait())) after making each request.

**Q – If I create a `SnmOrderInfo` object with a timeout of 5 seconds, 2 retries and a cache threshold of 1000 milliseconds, how many times is the deliver failed order called before the manager stops sending requests? As an example, my `placeGetBulkOrder()` method is placed with 4 OIDs and 10 maximum repetitions. My `deliverFailedOrder()` method simply prints a message indicating a failure has occurred. What I am currently observing is that the `deliverFailedOrder()` is called indefinitely. My expectation is that the calls would be stopped after 2 retries. Is my assumption incorrect?**

A – You will receive either 1 `deliveredSuccessfulOrder()` or 1 `deliverFailedOrder()` call for each OID or potential OID (i.e. `placeGetBulkOrder()`) in a request. A `deliverFailedOrder()` will be called once the number of retries has been exhausted. It will not happen indefinitely. Note that with a `placeGetBulkOrder()` placed with 4 OIDs and 10 maximum repetitions, you could receive 40 `deliveredSuccessfulOrder()/deliverFailedOrder()` calls, as you have 40 potential OIDs.

**Q – I am sending a `placeGetBulkOrder()` request with 4 repeater OIDs in the request. In my `deliverSuccessfulOrder()` implementation I am able to detect when I reach the end of the table. However, I continue to receive orders for what seems to be an indefinite amount of time. The MIB on our device has the potential of being very large. Once that I have detected the end of my table, is there a way I can tell the `SnmService` to stop sending these orders?**

A – There is no way to stop a `placeGetBulkOrder()` request (or any other request) from delivering successful and/or failed orders. The number of successful/failed orders delivered will be determined by the number of non-repeater OIDs and repeater OIDs times the number of max repetitions. jSNMP is designed to guarantee a response for each order requested. As you have noticed this can be quite large. We suggest that when you walk a table, you limit the number of repetitions to a smaller number and make multiple requests until the table walk is finished.

**Q – How can I tell when a `getBulk` request has ended?**

A – The `placeGetBulkOrder()` will return a value that is the one more than the last order number that will be passed as the `iOrderNum` argument to your `deliverSuccessfulOrder()` and `deliverFailedOrder()` methods. All you have to do is wait for order confirmations for all of these orders through your `deliverSuccessfulOrder()` and `deliverFailedOrder()` methods. Look at the `SnmV2GetBulkApplet.java` file in the examples directory of the distribution.

**Q – The agent that we are using has a problem accepting more than 24 `varBinds` in a single set PDU. When I use the `placeSetOrder()` with more than 24 OIDs, jSNMP seems to send them all at once, and the agent cannot handle them. What should I do?**

A – The maximum number of `SnmClerks` defaults to 25, which represents the number of OIDs that will be packed into a single non-atomic request PDU. Use the command line option `-Djsnmp.maxClerk=24`.

**Q – If I place a `getBulk` request with `maxRepetitions > 25`, does the maximum number of `SnmClerks` get changed to the larger value? If I place a `getBulk` request with `maxRepetitions=25` against a table containing 75 entries, then are the `SnmClerks` from the original `SnmInvoice` reused for subsequent `getBulk` requests, or are separate `SnmInvoices` and/or `SnmClerks` created?**

A – With `placeGetBulkOrder()` requests, the number of clerks is ignored. The `SnmInvoice` will create as many clerks as necessary.

**Q – An SNMP agent that I used to be able to access is no longer accessible. I think you have a trace capability within jSNMP. Would you tell me how I can trace using this?**

A – From the command line run `java -Doutback.trace=trace -Djsnmp.dumpPackets=true your_application`.

**Q – I've just run the SnmpV2Walker.java example against one of our devices and it worked pretty darn well, until the following:**

```
...
10819: 1.3.6.1.4.1.8845.1.2.1.2.1.1.1.17825796 (channel-4)
10820: 1.3.6.1.4.1.8845.1.2.1.2.1.1.1.17825797 (channel-5)
10821: 1.3.6.1.4.1.8845.1.2.1.2.1.1.1.17825798 (channel-6)
10822: order failed (General Error)
Ending MIB Walk
```

**Can you help me understand what the error means and what could cause it?**

A – This appears to be a normal termination and I suspect that `1.3.6.1.4.1.8845.1.2.1.2.1.1.1.17825798` is the last OID in your MIB or the last OID you have permission to access in your MIB. An order fails when it can't do the GETNEXT. The text "General Error" can be agent specific. Sometimes it will return "No Such Name" in the same situation.

**Q – Is jSNMP thread safe? My problem is the iOrderNumber. Is one iOrderNumber shared by every SnmpCustomer (big problem) or do I have one iOrderNumber per SnmpCustomer? Is the iOrderNumber sent (by callback) to all SnmpCustomers, or is it sent to only one SnmpCustomer?**

A – jSNMP Enterprise is thread safe. Orders and their corresponding `iOrderNumbers` are not shared between `SnmpCustomers`. All orders (`placeXXXOrder`) placed by a `SnmpCustomer` are returned to its `deliverSuccessfulOrder()` or `deliverFailedOrder()` methods only.

**Q – I currently create a SnmpAuthoritativeSession for each combination of host, port, context, user, authentication password, privacy password, authentication scheme, privacy scheme, and security level. Is that the way to do it? Does that mean that on every change for one of those parameters I should create a new session? Isn't the creation of a session a heavy operation; does it create a new thread; does it do a large memory allocation?**

A – If you require SNMPv3 authentication and privacy, you are creating the session correctly. It does not create a new thread nor does it allocate large amounts of memory. Since you are creating a SNMPv3 session, it is necessary to do discovery by communicating with the remote agent to establish the remote MTU, engine ID, boot time, and engine time. This is the reason it takes some time, as it requires 1 or 2 communication passes between your client and agent. This will not happen with SNMPv1/v2 sessions. It is recommended that sessions be created once and reused as necessary.

**Q – Please correct me if I am wrong. After making a getBulkOrder() request, jSNMP will report SnmpConstants.SNMP\_ERR\_RESPONSENOTGENERATED for failed deliveries due to the maximum PDU size being reached in the agent.**

A – You are correct. jSNMP guarantees a response (success or not) to all outstanding orders. With a `getBulk` request you place *non-repeaters* + (*repeaters* \* *repetitions*) orders. Since the agent responds with less than the number of `varBinds` expected, jSNMP reports the `SnmpConstants.SNMP_ERR_RESPONSENOTGENERATED` error for each of the orders (`varBinds`) not received. Remember the number of repetitions will vary from agent to agent and at what point the repeater starts in the MIB due to the PDU size limit of each agent. Reaching the end of a MIB with a `getBulk` request has a similar but slightly different behavior. When reaching the end of a MIB with a `getBulk` request, your `deliverFailedOrder()` method will receive a `SnmpConstants.SNMP_ERR_ENDOFMIBVIEW` error for the first `varBind` past the end of the MIB and `SnmpConstants.SNMP_ERR_RESPONSENOTGENERATED` for the rest of the `varBinds` past the end of the MIB.

**Q – Is it possible to make a `getBulkOrder()` request without non-repeaters?**

A – Yes, by setting the `szNonRepeaterOIDs` argument of your `placeGetBulkOrder()` call to null.

**Q – In what order does jSNMP deliver table rows and columns when making a `getBulk` request?**

A – Most SNMP agents deliver `getBulk` requests on tables by rows ... c1.1 c1.2 c1.3 c2.1 c2.2 c2.3 etc. When jSNMP receives the packet to the `getBulk` request, it separates out each `varBind` from the `varBindList` and delivers them on separate threads to your `deliverSuccessfulOrder()` method. Since they are delivered on separate threads, there is no guarantee that they will be delivered in order, though they usually are.

**Q – I am using the `SnmpConstants.SNMP_VERSION_xxx` constants from jSNMP in my code. The behavior was not as I anticipated, so I printed the values of the three constants. I got the following result:**

```
SnmpConstants.SNMP_VERSION_1 = 0
SnmpConstants.SNMP_VERSION_2 = 1
SnmpConstants.SNMP_VERSION_3 = 3
```

**Can you explain why the values are not sequential?**

A – The versions are defined with these values in the SNMP specifications. We decided to use the same values instead of doing extra conversions within our code.

**Q – I am facing a problem while retrieving values for the `ifDescr` object in interfaces table in MIB-II. This variable has a type of type `OctetString`, but when I use the `getStringValue()` method on the `varBind`, it gives me a value in hexadecimal numbers. For other strings in the system group, I am able to see the proper text. What is causing jSNMP to show the value in hex instead of in printable text? Here is what I am getting:**

```
Value for OID: 1.3.6.1.2.1.2.2.1.2.16777219 is:
0x33,0x43,0x6f,0x6d,0x20,0x45,0x74,0x68,0x65,0x72,0x4c,0x69,0x6e,0x6b,0x20,0x50,0x43,0x49,0x00
Type: 4
```

A – What you are seeing is the result of your agent sending null-terminated `DisplayStrings`. Note that `DisplayStrings` are just `OctetStrings` and can contain any character (0x00 to 0xff). The `getStringValue()` method creates a `String` from the internal representation of the object value (here `byte[]`). If any character is non-printable, it creates a hex string as you have noticed. To get around this, first determine the object type with the `getType()` method, and if it is `SnmpConstants.ASN_OCTSTR`, get the object value with the `getValue()` method, which will be of type `byte[]`, and create your own `String` from the `byte[]` object.



## jSNMP Traps

**Q – I can run the SnmpV1V2V3TrapListenerTest example just fine on a Windows 95/NT machine, but on a UNIX platform I get the following message:**

```
java.net.BindException: Permission denied
```

**What's up?**

A – One possibility is that there is another SNMP application already listening on the trap port. Also, the `BindException` that you are getting may be a Unix-specific issue. Only the "root" user on a Unix platform is allowed to open up a socket with a port number less than 1024. `SnmpV1V2V3TrapListenerTest` uses the default notification port 162. Either run `SnmpV1V2V3TrapListenerTest` as "root" or modify the `SnmpV1V2V3TrapListenerTest.java` file to create the `SnmpTrapListener` with a port greater than 1023 (don't forget to reconfigure your SNMP agent to send traps and informs to the new port).

**Q – My jSNMP application seems to be losing traps. What's up?**

A – Previous versions of jSNMP were very susceptible to losing traps and informs during trap or inform *bursts*. jSNMP Enterprise 3.x is able to handle sustained *bursts* of 100-400 or more incoming traps/informs per second without loss depending on the operating environment. This was calculated on a 400MHZ Intel P3 Windows NT machine. In fact, *bursts* of over 1000 traps per second were received without loss, but not at a sustained rate. Note that the trap/inform queue capacity and expansion size can be manipulated through `SnmpServiceConfiguration` class methods. This should not be necessary as the defaults, though large, are optimal.

**Q – My jSNMP application receives traps okay, but the address of the agent is different from the agent that sent the trap. What's up?**

A – SNMP traps can be delivered by a different agent than the one that originated the trap. The `SnmpTrapEvent.getAgentIPAddress()` method returns the address of the originating agent and the `SnmpTrapEvent.getSendersIPAddress()` method returns the address of the sending agent.

**Q – Would you please tell me why, when sending a large number of traps, we are seeing a large number of threads being created?**

A – In order to handle traps as quickly as possible, jSNMP uses a single thread to pull the trap packet off of the stack as the trap comes in and places it in a queue; a second thread pulls them off of the queue spawning another thread for decoding and delivery. Note that your callback method should return as soon as possible in order for the spawned thread to die.

**Q – My jSNMP application is failing to send traps, and we get the following message:**

```
Wed Jul 10 16:46:16 EDT 2002, 79, Thread-3, SNMP service unable to send PDU to 68.38.118.247  
(request too big)
```

**I guess the PDU is becoming bigger than it can handle. We must have too many varBinds (18). Please let me know the maximum size of a PDU or how can I provide a workaround for this problem.**

A – SNMPv1/2c defines a max PDU size of 484 bytes. SNMPv3 determines the PDU size through discovery with the SNMP agent. jSNMP uses these as defaults. The SNMPv1/2c defaults can be over-ridden with the command line configuration parameter `jsnmp.ignoreV1V2PduSizeLimit`, which will set the PDU size limit to the socket send/receive buffer size (usually 8192 depending on the OS). Note that the socket send/receive buffer size can also be over-ridden with the command line configuration parameter `jsnmp.socketBufferSize` or through the `SnmpServiceConfiguration.setSocketBufferSize()` method.

**Q – I use jSNMP to generate a coldStart alarm on one machine and send it to another machine where the details of the alarm are printed out. However, when I generate the alarm I am specifying its trap type as 1 but on the receivers end it says the trap type is 0. I was wondering if you could explain what I am doing wrong?**

A – You are doing nothing wrong. What you are seeing is one of the vagaries in the translation of SNMPv1 and SNMPv2 traps. In jSNMP the `SnmpTrapEvent` class is based upon the SNMPv2 trap specification (<http://www.ietf.org/rfc/rfc1905.txt> - RFC 1905 - Protocol Operations for Version 2 of the Simple Network Management Protocol). That is, all traps are specified by an enterprise OID and trap code. The generic SNMPv1 traps, `coldStart`, `warmStart`, `linkDown`, `linkUp`, `authenticationFailure`, and `egpNeighborLoss`, are converted to traps with an enterprise OID of 1.3.6.1.6.3.1.1.5 and corresponding trap codes of 1, 2, 3, 4, 5, and 6 (see <http://www.ietf.org/rfc/rfc2576.txt> - RFC 2576 - Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework). When you send your `coldStart` trap with `java SnmpV1TrapSenderTest 10.24.39.237 public 1.3.6.1.6.3.1.1.5 1`, you are correctly specifying the OID and trap number. When you receive traps with an enterprise OID of 1.3.6.1.6.3.1.1.5, the trap type will be zero based. Other enterprise OIDs' trap types will be one based.

## jSNMP Errors, Exceptions, and Other Issues

**Q – I'm using jSNMP within an applet. Why does the browser throw a SecurityException when I try to communicate with an SNMP device?**

A – As of version 2.01, jSNMP Enterprise is capable of running in either Netscape Communicator 4.5 or Microsoft IE 4.01 or later versions. Security issues are still possible though, depending on how your applet is packaged for the web. Java security restrictions inside browsers prevent untrusted applets from calling various Java APIs. These APIs include many networking calls required to do SNMP. Enabling an applet to run in a browser environment generally involves the following two steps:

1. allow the applet to be trusted, either by packaging it in a signed archive (JAR or CAB), or placing it on the local CLASSPATH
2. call vendor-specific APIs to grant the required rights to the applet.

For more information on applet security, see Sun's [Frequently Asked Questions - Java Security](#) or Netscape's [Security Documentation](#). Example applets are included in jSNMP Enterprise, which demonstrate jSNMP-based applications within a browser.

**Q – When I tried to compile the SnmpV1GetSysInfo.java application I got the following error:**

```
Exception in thread "main" java.lang.NoClassDefFoundError:
SnmpV1GetSysInfo
```

**What did I do wrong?**

A – You are probably having a CLASSPATH problem. Make sure that "." is in your CLASSPATH.

**Q – Why do I get the exception NoClassDefFoundError/java.math.BigInteger when using jSNMP?**

A – jSNMP requires a JDK1.1.x or newer compliant runtime environment. The class `java.math.BigInteger` is specific to JDK1.1 and later versions. If you are using jSNMP within a browser it must likewise support JDK1.1.x or a later version. If you are using Internet Explorer 3.x or Netscape Navigator 3.x you must upgrade to Internet Explorer 4.0 or later or Netscape Communicator 4.x or later.

**Q – I received a NULL SnmpVarBind in my SnmpCustomer deliverSuccessfulOrder() method. What's up?**

A – It is possible (though rare) that an `SnmpCustomer.deliverSuccessfulOrder()` method will receive a NULL `SnmpVarBind`. This will happen when a remote agent's response does not contain any varBinds. While unlikely, jSNMP programmers should test for NULL on the `SnmpVarBind` argument.

**Q – I am running the applet I developed in my JDKs appletviewer. It gives me the following**

```
java.lang.ExceptionInInitializerError:  
  java.security.AccessControlException: access denied  
    <java.util.PropertyPermission browser read>
```

**The above happens when I call SnmpLocalInterfaces.getService(). What's up?**

A – You must specifically grant the applet the permissions needed to run. You can either add the following permissions to your java.policy file:

```
permission java.util.PropertyPermission "outback.trace", "read";  
permission java.util.PropertyPermission "jsnmp.maxClerks", "read";  
permission java.util.PropertyPermission "jsnmp.loadRFC1213MIB", "read";  
permission java.util.PropertyPermission "jsnmp.socketBufferSize", "read";  
permission java.util.PropertyPermission "jsnmp.clerkThreadPool", "read";  
permission java.util.PropertyPermission "jsnmp.bufferDelay", "read";  
permission java.util.PropertyPermission "jsnmp.cachePrunePeriod", "read";  
permission java.util.PropertyPermission "jsnmp.cacheExpireFloor", "read";  
permission java.util.PropertyPermission "jsnmp.cacheExpireFactor", "read";  
permission java.util.PropertyPermission "jsnmp.cacheDisable", "read";  
permission java.util.PropertyPermission "jsnmp.trapQueueCapacity", "read";  
permission java.util.PropertyPermission "jsnmp.trapQueueExpansion", "read";  
permission java.util.PropertyPermission "jsnmp.ignoreV1V2PduSizeLimit", "read";  
permission java.util.PropertyPermission "jsnmp.forceGC", "read";  
permission java.util.PropertyPermission "jsnmp.dumpPackets", "read";  
permission java.util.PropertyPermission "jsnmp.retrieveTimeStamps", "read";  
permission java.util.PropertyPermission "jsnmp.retryPackedTimeouts", "read";  
permission java.util.PropertyPermission "jsnmp.properties", "read";  
permission java.util.PropertyPermission "nocmufixup", "read";  
permission java.util.PropertyPermission "browser", "read";
```

or add the following single permission:

```
permission java.security.AllPermission;
```

Alternately, you can use a policy file supplied on the command line. For instance:

```
appletviewer -J-Djava.security.policy=policy.all yourSnmpApplet
```

**Q – I am using Microsoft Visual J++ 1.1 and am having trouble compiling the sample applications. Does your product work with the Microsoft tool?**

A – Either upgrade to Microsoft Visual J++ Version 6.0 or see the Microsoft Knowledge Base Article [INFO: Use New Java 1.1 Language Features with Visual J++ 1.1.](#)

## Using jMIBC

**Q – Would you please help me with the installation of jMIBC?**

A – You should have received a jMIBC.jar file in the distribution. The jMIBC.jar file can be installed anywhere on a system. The user's manual will be in the doc directory and will help you getting started.

**Q – I am using JDK 1.1, how do I compile a MIB?**

A – If you are using JDK 1.1, you will need to set the CLASSPATH environment variable to run jMIBC. For example, this could be accomplished with the following command:

```
set CLASSPATH=C:\jsnmp\jMIBC.jar;%CLASSPATH%
```

for Win32 systems, or

```
export CLASSPATH=/jsnmp/jMIBC.jar:$CLASSPATH
```

for Unix systems. Running jMIBC can be accomplished with the following command:

```
java jMIBC [options] <mibfile>
```

**Q – I am using JDK 1.2, how do I compile a MIB?**

A – For JDK1.2 and JDK1.3, you can follow the instructions for JDK1.1, or you can run jMIBC without setting the CLASSPATH environment variable. For example, this can be accomplished with the following command:

```
java -jar jMIBC.jar [options] <mibfile>
```

The `-jar` flag tells the Java interpreter that the application is packaged in the JAR file format

**Q – Can we use the parsing capabilities of the jMIBC in order to use it for showing a MIB tree in an SNMP MIB browser that we write? Does the SnmpMIBService export the MIB tree that is parsed?**

A – The `SnmpMIBService` is used to resolve OIDs to object names and enumerated names to enumerated values and vice versa. With a simple walk of an agent you can build a tree of OIDs and names. You can export that to the database of your choice.

**Q – Do jMIBC and the jMIBC.loadMib() method support SMI version1 and version2 MIBS?**

A – Yes.

**Q – Do jMIBC and the jMIBC.loadMib() method parse the IMPORT section of the MIB module and include all the required MIBs?**

A – Yes, if the required MIBs are available.

## jMIBC Errors, Exceptions, and Other Issues

**Q – Why did I get the following error:**

**Error: Unknown syntax of <type> for <name> in <file>.**

A – This message will be displayed if an object has a syntax type that has not been declared. The recommended way to handle this error is to include all dependent MIB files.

**Q – Why did I get the following error:**

**Error: <name> from <file> is undefined.**

**Please define using the `-b` option or include all dependent MIB files.**

A – This message will be displayed if a reference to an OID name is encountered that has not been found in the file. The recommended way to correct his problem is to include all the dependent MIB files using the `-i` option in the correct order. The dependent MIB files can be determined by looking at the `IMPORT` definitions and including all the MIB files referenced in the `FROM` clause.

**Q – Why did I get the following warning:**

**Warning: Redefinition of <name> in <file>  
Previously found in <file>.**

A – This warning indicates that a type, identifier, or OID has been redefined. The last definition found will be used. This warning will only occur if the definitions differ. The recommended way to handle this is to either remove the redefinition or make the definitions identical.

**Q – Why did I get the following error:**

**Error: Parsing file <file>**

A – This message indicates that there was an error parsing the input file. The file name where the error is detected is printed on the error line. Additional details of the error will follow and typically include the line number and column where the error was encountered. The following error message indicates that an enumeration begins with a digit:

*Error: Parsing file <file>  
Encountered "1" at line 8036, column 25.  
Was expecting one of:  
<LCASEFIRST\_IDENT\_TKN> ...  
<UCASEFIRST\_IDENT\_TKN> ...*

The following example illustrates an error where the RFC file was not edited to remove the extraneous text:

*Error: Parsing file <file>  
Encountered "Working" at line 1, column 9.  
Was expecting one of:  
"(" ...  
"DEFINITIONS" ...  
"FORCE-INCLUDE" ...  
"EXCLUDE" ...*

**Q – How does the [CERT Advisory CA-2002-03](#) apply to jSNMP Enterprise?**

A – jSNMP Enterprises acknowledges the potential of SNMP vulnerabilities as identified in the following CERT advisories:

1. [VU#854306](#) – Multiple vulnerabilities in SNMPv1 request handling
2. [VU#107186](#) – Multiple vulnerabilities in SNMPv1 trap handling

jSNMP Enterprises has investigated how these vulnerabilities may impact the jSNMP Enterprise product and has determined the following:

1. VU#854306 – This advisory is not applicable to jSNMP, as jSNMP does not accept or process SNMP Get, Set, or GetNext PDUs; rather, jSNMP sends those requests to SNMP agents and processes subsequent responses.
2. VU#107186 – jSNMP v3.2 passed the 24,098 applicable tests in the PROTOS c06-snmpv1 test suite. jSNMP v3.1 failed only one test with undesirable behavior. No consequences, other than potential denial-of-service, are known and there have been no reported instances of this vulnerability being exploited in the jSNMP product.

We recommend that our customers upgrade to the latest available version of jSNMP.

## jSNMP and RMI

### Q – What is RMI?

A – RMI (Remote Method Invocation) is a distributed object architecture that allows programmers to develop distributed Java programs with the same syntax and semantics used for non-distributed programs. For more information see Sun's [Java Remote Method Invocation](#).

### Q – How do I enable my application to access the remote RMI jSNMP services?

A – First create an `RMI_SnmpClient` object. Next replace all method calls that use the previously created `SnmpLocalInterfaces` object with the equivalent `RMI_SnmpClient` method calls. That's all there is too it.

### Q – What is the best way to provide classes under RMI?

A – You should use a web server as the source of all classes. The following instructions illustrate running a jSNMP RMI Server under Red Hat Linux 6.2 using the Apache web server and the Java 2 SDK 1.3:

1. create the `jSNMPEnterprises/jSNMP/examples/classes` directory under the document root (`/home/httpd/html`)
2. place all jSNMP classes and jars into the `jSNMPEnterprises/jSNMP/examples/classes` directory
3. place all of your HTML files (i.e., `RmiV1WalkerApplet.html`) referencing the jSNMP classes in the `jSNMPEnterprises/jSNMP/examples` directory
4. in a console window, set the CLASSPATH to the `jSNMPEnterprises/jSNMP/examples/classes` directory and start the RMI registry and jSNMP RMI server, specifying the RMI codebase

```
cd /home/httpd/html/jSNMPEnterprises/jSNMP/examples/classes
export JAVA_HOME=/usr/java/jdk1.3.0_02/jre
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=../jSNMPEnterprise.jar: \
$JAVA_HOME/lib/ext/jce1_2_1.jar:$JAVA_HOME/lib/ext/sunjce_provider.jar: \
$JAVA_HOME/lib/ext/local_policy.jar:$JAVA_HOME/lib/ext/US_export_policy.jar
rmiregistry&
java -Djava.rmi.server.codebase=http://mywebserver/jSNMPEnterprises/jSNMP/examples/classes \
-Djava.security.policy=policy RMIServer
```

5. ensure that the web server can resolve the hostnames of any clients that attach to it, either through DNS or a local host file
6. use a browser or the JDK appletviewer to view the applet

```
appletviewer http://mywebserver/jSNMPEnterprises/jSNMP/examples/RmiV1WalkerApplet.html
```

Note that Sun also provides a [small pure-Java HTTP server](#) that can be used in place of the Apache web server. Also, more discussion on the RMI registry, servers, and applets can be found at <http://java.sun.com/j2se/1.3/docs/guide/rmi/getstart.doc.html#7445>.

### Q – When I run your `RmiV1WalkerApplet.html` under Internet Explorer, I got the following error:

**Couldn't connect to jSNMP RMI Server foo.foo.com  
(java.rmi.RemoteException: No registry handler present)**

#### What's up?

A – You need to download the RMI classes for Internet Explorer. Download the [RMI classes](#) from Microsoft and unzip the `rmi.zip` file into your Windows java directory. Any unzip utility capable of handling long filenames can be used, such as [WinZip](#). Alternately, you can get a nice bundled package from [IBM](#) that installs the RMI classes in the right location.

**Q – When I run your RmiV2TrapListenerApplet under Netscape and add a host, it gives me an exception.**

**What's up?**

A – You are most likely loading the applet from your local file system. The problem here is related to security privileges and principals in Netscape. In order to successfully request a set of privileges to perform normally forbidden actions in Netscape (like opening a listening socket, contacting arbitrary hosts, etc.), the applet must be loaded from a trusted principal. Normally this is accomplished using signed applets. You can also accomplish the same thing using a Netscape workaround called [Activating Codebase Principals](#). What this does is tell the browser to treat a codebase named in an HTML file as a trusted principal, just as if it were signed. You must also use a web server as the source of all classes, as explained in the answer to the previous question. Note that Netscape recommends against using this approach in a production setting.

**Q – When I run my RMI application, it gives me the following java.net.ConnectException:**

**Connection refused: no further information**

**What's up?**

A – You probably aren't running a jSNMP RMI server. A jSNMP RMI server must be run with a security policy under the RMI registry server on the same machine. Please refer to the User's Guide for more information.

## jSNMP and CORBA

**Q – What is CORBA?**

A – CORBA (Common Object Request Broker Architecture) is a distributed object architecture that allows software objects to interact across networks. CORBA objects can interact regardless of implementation language, location, or host platform. CORBA was first introduced in 1991 by the [Object Management Group \(OMG\)](#), a consortium of over 800 software vendors, software developers and end users. CORBA addresses the need for interoperability among the rapidly proliferating number of hardware and software products available today.

**Q – placeGetOrder() doesn't seem to be supported with the CORBA jSNMP services?**

A – That is correct. Only those interfaces present in jSNMP 2.0 are supported in the CORBA interfaces. None of the new interfaces or SNMPv2c and SNMPv3 are supported with the CORBA interfaces. If you require these new interfaces, SNMPv2c, or SNMPv3 in a distributed environment, the RMI interfaces in jSNMP 3.x are available. Alternatively, the RMI interfaces in jSNMP 3.x may be used with [RMI over IIOP](#).

**Q – How do I enable my application to access the remote CORBA jSNMP services?**

A – Compile the IDL to generate the stubs particular to your language and environment. The generated stubs then provide a local interface to the remote CORBA jSNMP service. Because the jSNMP service is exposed through CORBA IDL, a developer can access distributed jSNMP services in any language that supports a CORBA 2.0 IDL mapping.

**Q – I just generated the CORBA stubs; why would I use the included Java Client to communicate with the remote jSNMP service instead of talking directly through the stubs?**

A – To make accessing the remote jSNMP services easier for Java developers, we've included a Java wrapper to the CORBA Client. This provides Java developers an easy way of incorporating the distributed jSNMP CORBA services in their applications without learning all the intricacies of CORBA development. One of the great benefits of the Java CORBA Client is that it exposes the same interfaces as the local call jSNMP service. The Enterprise developer can therefore use the same code to access local or remote instances of the jSNMP service.

**Q – Does the Java Client use CORBA?**

A – Yes, but you will still need to compile the jSNMP IDL to generate stubs compatible with your ORB.

**Q – What ORBs does jSNMP Enterprise Support?**

A – jSNMP Enterprise uses the standard IDL to Java mapping, which should allow it to work correctly with any ORB that supports this standard. jSNMP Enterprise 3.x has been tested successfully with Sun's JDK 1.3 ORB. While jSNMP Enterprise 2.x was tested with Inprise's VisiBroker 3.x and Iona's OrbixWeb 3.x, Sun's JDK 1.3 ORB is the only ORB currently supported by jSNMP Enterprises.

**Q – Do I need to be a CORBA developer to use jSNMP Enterprise?**

A – We provide the remote Java interfaces so that you don't need to be a CORBA expert to use jSNMP Enterprise. You should be familiar with CORBA though and be able to compile the IDL that comes with jSNMP Enterprise to generate the CORBA stubs. However, unless you are implementing a client in a language other than Java, you don't need to communicate through these CORBA interfaces; jSNMP Enterprise does this for you. The only other step required of the developer is to communicate the object references of the server components to the client. This may be done through a naming service or through a file system. The task is discussed in more detail in the user's manual.

**Q – What else do I need to use CORBA with jSNMP Enterprise?**

A – You will need a CORBA 2.0 ORB development environment compatible with the standard CORBA to Java mapping. ORB vendors providing environments include Iona, Visigenic, and Javasoft. Sun's JDK 1.3 includes a CORBA ORB that is compatible with jSNMP.

## Miscellaneous

**Q – How does jSNMP Enterprise relate to Sun's JMX and JDMK?**

A – jSNMP Enterprise is a toolkit for building SNMP manager applications. Sun's [Java Management Extensions \(JMX\)](#) and [Java Dynamic Management Kit \(JDMK\)](#) are focused on building Java-based management agents that can communicate with managers using a variety of protocols (RMI, HTTP, etc.). The JMX and JDMK do include an SNMP Manager API with limited functionality, but it is not the primary focus of the package.

**Q – Are jSNMP and jMIBC Year 2000 compliant?**

A – Yes, jSNMP has been examined and determined to be Year 2000 compliant. jSNMP depends on a third-party environment (e.g. operating systems, Java virtual machines, SNMP agents, etc.) in order to operate. Although jSNMP Enterprises is not aware of any Year 2000 issues with these third-party components, it cannot guarantee their Year 2000 compliance and does not assume any responsibility for their non-compliance.