

NET-SNMP 安装配置手册

2008.3.11

张兵

Aaron Cheung

移动互联网事业部第四开发部

东软集团（成都）有限公司

Address: 成都都江堰市青城山东软大道 1 号东软软件园 B5 楼

Postcode: 611844

QQ: 375627443

Email: zhang.bing@neusoft.com

Gmail: zhangcqu@gmail.com

目 录

第 1 章 net-snmp安装及配置	1-1
1.1 安装.....	1-1
1.2 设置net-snmp自启动.....	1-3
1.3 设置环境变量	1-3
第 2 章 net-snmp测试.....	2-3
2.1 本地通过localhost测试.....	2-3
2.2 本地通过IP测试.....	2-4
2.3 远程通过IP测试.....	2-5
第 3 章 snmpd.conf详解	3-6
3.1 通道控制配置	3-7
3.1.1 定义安全体名称	3-7
3.1.2 定义安全组	3-7
3.1.3 定义视图	3-8
3.1.4 向安全组授权相应的视图.....	3-8
3.2 系统联系人信息.....	3-9
3.3 进程检查.....	3-10
3.4 可执行脚本	3-11
3.5 磁盘检查.....	3-12
3.6 负载均衡检查	3-13
3.7 可扩展部分	3-14
3.8 通过控制.....	3-15
3.9 其它	3-16
第 4 章 net-snmp指令.....	4-16
4.1 snmpget	4-16
4.2 snmpwalk	4-16
4.3 snmpconf	4-16
4.4 snmpd	4-17
4.5 snmpgetnext	4-17
4.6 snmpset	4-17
4.7 snmpbulkget.....	4-17
4.8 snmptrap	4-17
4.9 snmptrapd	4-17
4.10 snmpinform	4-17
4.11 snmptable	4-17

4.12 snmpstatus.....	4-18
4.13 snmpbulkwalk	4-18
4.14 snmpdelta	4-18
4.15 snmptest	4-18
4.16 snmptranslate	4-18
4.17 snmpusm	4-18
4.18 snmpvacm.....	4-18
4.19 snmpdf	4-18
第 5 章 JAVA开发.....	5-18
5.1 NET-SNMP采集示例程序（Java）	5-18
5.2 NET-SNMP主动发送示例程序（Java）	5-20
5.3 运行结果.....	5-21
第 6 章 Linux常用OID.....	6-22
6.1 CPU	6-22
6.2 内存.....	6-22
6.3 磁盘.....	6-23
6.4 示例.....	6-23
第 7 章 snmpd.conf示例配置	7-23

图目录

图 1-1 解压net-snmp-5.3.2.tar.gz	1-1
图 1-2 进入解压后的目录	1-1
图 1-3 configure命令及参数	1-1
图 1-4 configure摘要	1-2
图 1-5 编译及安装命令	1-2
图 1-6 添加snmpd.conf文件	1-2
图 1-7 snmpd.conf原文件内容	1-2
图 1-8 snmpd.conf修改后文件内容	1-3
图 1-9 net-snmp自启动代码	1-3
图 1-10 net-snmp环境变量	1-3
图 2-1 本地通过localhost测试	2-3
图 2-2 本地通过localhost测试后的部分显示结果	2-4
图 2-3 本地通过IP测试	2-4
图 2-4 本地通过IP测试	2-5
图 2-5 远程通过IP测试	2-6
图 3-1 snmpd.conf定义安全体名称	3-7
图 3-2 snmpd.conf定义安全组	3-7
图 3-3 snmpd.conf定义安全组属性表	3-7
图 3-4 snmpd.conf定义视图	3-8
图 3-5 snmpd.conf向安全组授权的相应视图	3-8
图 3-6 系统联系人信息	3-9
图 3-7 进程检查	3-10
图 3-8 可执行脚本	3-11
图 3-9 磁盘检查	3-12
图 3-10 负载均衡检查	3-13
图 3-11 可扩展部分	3-14
图 3-12 通过控制	3-15
图 3-13 其它	3-16
图 5-1 程序执行结果	5-21

表目录

表 3-1 snmpd.conf定义安全体名称属性表.....3-7

表 3-2 snmpd.conf定义视图的属性表3-8

表 3-3 snmpd.conf向安全组授权的相应视图3-8

表 6-1 CPU常用OID6-22

表 6-2 内存常用OID6-22

表 6-3 磁盘常用OID6-23

第1章 net-snmp 安装及配置

1.1 安装

步骤1: 解压

```
[root@localhost poc]# tar -xvf net-snmp-5.3.2.tar.gz
```

图1-1 解压 net-snmp-5.3.2.tar.gz

步骤2: configure

1) 进入源文件目录

```
[root@localhost poc]# cd net-snmp-5.3.2
```

图1-2 进入解压后的目录

2) configure

```
[root@localhost net-snmp-5.3.2]# ./configure --prefix=/usr/local/net-snmp --enable-mfd-rewrites --with-default-snmp-version="2" --with-sys-contact="Aaron,E_mail;zhang.bing@neusoft.com" --with-sys-location="China" --with-logfile="/var/log/snmpd.log" --with-persistent-directory="/var/net-snmp"
```

图1-3 configure 命令及参数

prefix: net-snmp 将要安装的路径

enable-mfd-rewrites: 允许用新的 MFD 重写可用的 mid 模块

with-default-snmp-version: 默认的 SNMP 版本

with-sys-contact: 可以配置该设备的联系人

with-sys-location: 该设备的位置

with-logfile: 日志文件路径

with-persistent-directory: 不变数据存储目录

3)configure 摘要

```

-----
Net-SNMP configuration summary:
-----

SNMP Versions Supported:    1 2c 3
Net-SNMP Version:           5.3.2
Building for:                linux
Network transport support:   Callback Unix TCP UDP
SNMPv3 Security Modules:    usm
Agent MIB code:              mibII ucd_snmp snmpv3mibs notification notific
ation-log-mib target agent_mibs agentx disman/event-mib disman/schedule util
ities host
SNMP Perl modules:          disabled
Embedded perl support:      disabled
Authentication support:     MD5 SHA1
Encryption support:         DES AES
WARNING: New version of the Event MIB which may be subtly different from
the original implementation - configure with 'disman/old-event-mib' for the
previous version
-----

```

图1-4 configure 摘要

步骤3: 编译并安装

```
[root@localhost net-snmp-5.3.2]# make && make install
```

图1-5 编译及安装命令

步骤4: 配置 snmpd.conf

- 1) 将 EXAMPLE.conf 文件复制到/usr/local/net-snmp/share/snmp, 并重命名为 snmpd.conf

```
[root@localhost net-snmp-5.3.2]# cp EXAMPLE.conf /usr/local/net-snmp/share/s
nmp/snmpd.conf
```

图1-6 添加 snmpd.conf 文件

- 2) 将snmpd.conf中如图 1-7的内容修改为如图 1-8所示

```

#          sec.name  source          community
com2sec local      localhost      COMMUNITY
com2sec mynetwork NETWORK/24      COMMUNITY

```

图1-7 snmpd.conf 原文件内容

```

#          sec.name  source          community
com2sec local      localhost      public
com2sec mynetwork 192.168.228.254      public
com2sec mynetwork 192.168.228.155      public

```

图1-8 snmpd.conf 修改后文件内容

【注意】

在编辑 snmpd.conf 可使用空格，但不能使用 TAB 键，否则会出现错误

1.2 设置 net-snmp 自启动

在/etc/rc.local文件的末尾加上如图 1-9所示代码

```
/usr/local/net-snmp/sbin/snmpd -c /usr/local/net-snmp/share/snmp/snmpd.conf &
```

图1-9 net-snmp 自启动代码

1.3 设置环境变量

在/etc/profile文件的export命令前加上如图 1-10所示代码

```
PATH=/usr/local/net-snmp/bin:/usr/local/net-snmp/sbin:$PATH
```

图1-10 net-snmp 环境变量

第2章 net-snmp 测试

在配置 net-snmp 的时候，配置了三种情况：本地通过 localhost 访问、本地通过 IP 访问、远程通过 IP 测试。因此，测试的时候也分三种情况。

2.1 本地通过 localhost 测试

步骤1： 运行如图 2-1所示命令

```
[root@localhost ~]# snmpwalk -v 2c -c public localhost if
```

图2-1 本地通过 localhost 测试

步骤2： 测试后的显示结果如图 2-2所示，表示该种情况正常。


```
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifIndex.5 = INTEGER: 5
IF-MIB::ifIndex.6 = INTEGER: 6
IF-MIB::ifIndex.7 = INTEGER: 7
IF-MIB::ifIndex.8 = INTEGER: 8
IF-MIB::ifIndex.9 = INTEGER: 9
IF-MIB::ifIndex.10 = INTEGER: 10
IF-MIB::ifIndex.11 = INTEGER: 11
IF-MIB::ifIndex.12 = INTEGER: 12
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: peth0
IF-MIB::ifDescr.3 = STRING: sit0
IF-MIB::ifDescr.4 = STRING: vif0.0
IF-MIB::ifDescr.5 = STRING: eth0
IF-MIB::ifDescr.6 = STRING: vif0.1
IF-MIB::ifDescr.7 = STRING: veth1
IF-MIB::ifDescr.8 = STRING: vif0.2
```

图2-2 本地通过 localhost 测试后的部分显示结果

2.2 本地通过 IP 测试

步骤1: 运行如图 2-3所示

```
[root@localhost ~]# snmpwalk -v 2c -c public 192.168.228.254 if
```

图2-3 本地通过 IP 测试

步骤2: 测试结果如图 2-4所示，表示该种情况正常。

```
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifIndex.5 = INTEGER: 5
IF-MIB::ifIndex.6 = INTEGER: 6
IF-MIB::ifIndex.7 = INTEGER: 7
IF-MIB::ifIndex.8 = INTEGER: 8
IF-MIB::ifIndex.9 = INTEGER: 9
IF-MIB::ifIndex.10 = INTEGER: 10
IF-MIB::ifIndex.11 = INTEGER: 11
IF-MIB::ifIndex.12 = INTEGER: 12
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: peth0
IF-MIB::ifDescr.3 = STRING: sit0
IF-MIB::ifDescr.4 = STRING: vif0.0
IF-MIB::ifDescr.5 = STRING: eth0
IF-MIB::ifDescr.6 = STRING: vif0.1
IF-MIB::ifDescr.7 = STRING: veth1
IF-MIB::ifDescr.8 = STRING: vif0.2
```

图2-4 本地通过 IP 测试

2.3 远程通过 IP 测试

因为在客户机上的时候，可能没安装net-snmp，因此也就不能运行net-snmp的命令，所以需要第三方软件进行测试。这儿使用的是AdventNet MibBrowser。如图 2-5所示，输入IP地址、端口及community，选定左边菜单的OID。然后通过菜单【Operations】→【Get】菜单获取值。

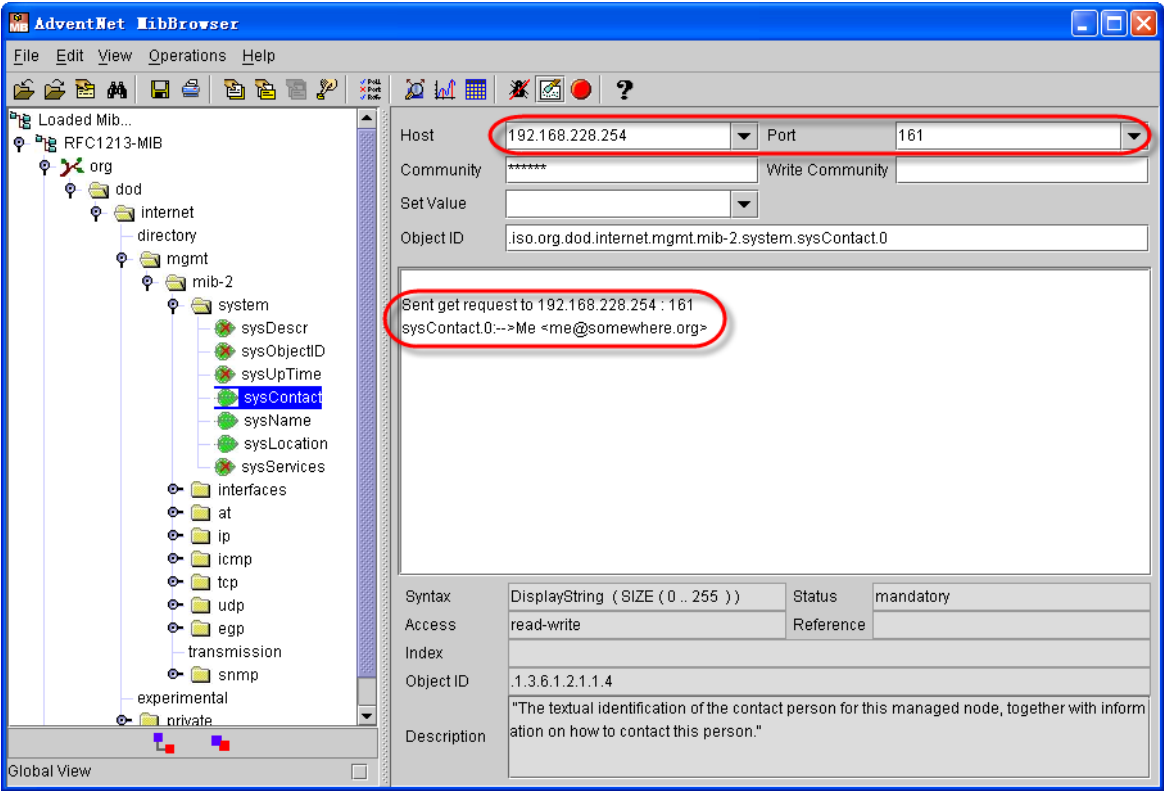


图2-5 远程通过 IP 测试

第3章 snmpd.conf 详解

snmpd.conf 的配置包括通道控制（Access Control）、系统联系人信息（System contact information）、进程检查（process check）、可执行脚本（executables/scripts）、磁盘检查（disk checks）、负载均衡检查（load average checks）、可扩展部分（extensible sections）、通过控制（Pass through control）、其它等部分。

一般情况只需要修改3.1 的3.1.1 就可以满足常规需求。

【注意】

在编辑 snmpd.conf 可使用空格，但不能使用 TAB 键，否则会出现错误

3.1 通道控制配置

3.1.1 定义安全体名称

```
#####
# First, map the community name (COMMUNITY) into a security name
# (local and mynetwork, depending on where the request is coming
# from):

#          sec.name  source          community
com2sec local      localhost        public
com2sec mynetwork  192.168.228.254    public
com2sec mynetwork  192.168.228.155    public
```

图3-1 snmpd.conf 定义安全体名称

表3-1 snmpd.conf 定义安全体名称属性表

字段	注释
sec.name	安全体名称。
source	定义请求的来源，在 IP 协议中，这个数据是 IP 地址。在 net-snmp 中可以对来源 IP 加以控制，但这个特性不是 SNMP 规定的，是 net-snmp 扩展的。
community	共同体名称

3.1.2 定义安全组

```
#####
# Second, map the security names into group names:

#          sec.model  sec.name
group MyRWGroup v1    local
group MyRWGroup v2c    local
group MyRWGroup usm    local
group MyROGroup v1     mynetwork
group MyROGroup v2c    mynetwork
group MyROGroup usm    mynetwork
```

图3-2 snmpd.conf 定义安全组

图3-3 snmpd.conf 定义安全组属性表

字段	注释
	安全组名称，如 “MyRWGroup”
sec.model	安全模式，可选值为 v1、v2c、usm
sec.name	安全体名称

3.1.3 定义视图

```
####
# Third, create a view for us to let the groups have rights to:

#           incl/excl subtree                                mask
view all    included .1                                     80
```

图3-4 snmpd.conf 定义视图

表3-2 snmpd.conf 定义视图的属性表

字段	注释
	视图名，如 “all”
incl/excl	对下面的 MIB 子树是包括还是排除
subtree	视图中涉及的 MIB 子树
mask	掩码

3.1.4 向安全组授权相应的视图

```
####
# Finally, grant the 2 groups access to the 1 view with different
# write permissions:

#           context sec.model sec.level match read  write notif
access MyROGroup ""      any      noauth  exact  all   none  none
access MyRWGroup ""      any      noauth  exact  all   all   none
```

图3-5 snmpd.conf 向安全组授权的相应视图

表3-3 snmpd.conf 向安全组授权的相应视图

字段	属性
	安全组，如 “MyROGroup”

context	上下文，v1、v2c 中始终为空
sec.model	安全模式，可选值为 v1、v2c、usm
sec.level	安全级别，可选值为 auth、noauth、priv，v1、v2c 中只能为 noauth
match	前缀，指定 context 如何与 PDU 中的 context 匹配，V3 使用
read	授权的读视图
write	授权的写视图
notif	授权的 trap 视图

3.2 系统联系人信息

```
#####
# System contact information
#
# It is also possible to set the sysContact and sysLocation system
# variables through the snmpd.conf file.  **PLEASE NOTE** that setting
# the value of these objects here makes these objects READ-ONLY
# (regardless of any access control settings).  Any attempt to set the
# value of an object whose value is given here will fail with an error
# status of notWritable.
#
syslocation Right here, right now.
syscontact Aaron <zhang.bing@neusoft.org>
#
# Example output of snmpwalk:
# % snmpwalk -v 1 -c public localhost system
# system.sysDescr.0 = "SunOS name sun4c"
# system.sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.sunos4
# system.sysUpTime.0 = Timeticks: (595637548) 68 days, 22:32:55
# system.sysContact.0 = "Me <me@somewhere.org>"
# system.sysName.0 = "name"
# system.sysLocation.0 = "Right here, right now."
# system.sysServices.0 = 72
```

图3-6 系统联系人信息

可以通过如下命令获得联系人信息：

```
snmpwalk -v 1 -c public localhost system
```

3.3 进程检查

```
#####
# Process checks.
#
# The following are examples of how to use the agent to check for
# processes running on the host. The syntax looks something like:
#
# proc NAME [MAX=0] [MIN=0]
#
# NAME: the name of the process to check for. It must match
# exactly (ie, http will not find httpd processes).
# MAX: the maximum number allowed to be running. Defaults to 0.
# MIN: the minimum number to be running. Defaults to 0.
#
# Examples:
#
# Make sure mountd is running
proc mountd
#
# Make sure there are no more than 4 ntalkds running, but 0 is ok too.
proc ntalkd 4
#
# Make sure at least one sendmail, but less than or equal to 10 are running.
proc sendmail 10 1
# A snmpwalk of the prTable would look something like this:
#
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.2
# enterprises.ucdavis.procTable.prEntry.prIndex.1 = 1
# enterprises.ucdavis.procTable.prEntry.prIndex.2 = 2
# enterprises.ucdavis.procTable.prEntry.prIndex.3 = 3
# enterprises.ucdavis.procTable.prEntry.prNames.1 = "mountd"
# enterprises.ucdavis.procTable.prEntry.prNames.2 = "ntalkd"
# enterprises.ucdavis.procTable.prEntry.prNames.3 = "sendmail"
# enterprises.ucdavis.procTable.prEntry.prMin.1 = 0
# enterprises.ucdavis.procTable.prEntry.prMin.2 = 0
# enterprises.ucdavis.procTable.prEntry.prMin.3 = 1
# enterprises.ucdavis.procTable.prEntry.prMax.1 = 0
```

图3-7 进程检查

可以通过如下命令获得检查进程后的结果:

```
snmpwalk -v 1 -c public localhsot .1.3.6.1.4.1.2021.2
```

3.4 可执行脚本

```
#####
# Executables/scripts
#
#
# You can also have programs run by the agent that return a single
# line of output and an exit code. Here are two examples.
#
# exec NAME PROGRAM [ARGS ...]
#
# NAME:      A generic name.
# PROGRAM:   The program to run. Include the path!
# ARGS:      optional arguments to be passed to the program
#
# a simple hello world
exec echotest /bin/echo hello world
#
# Run a shell script containing:
#
# #!/bin/sh
# echo hello world
# echo hi there
# exit 35
#
# Note: this has been specifically commented out to prevent
# accidental security holes due to someone else on your system writing
# a /tmp/shtest before you do. Uncomment to use it.
#
#exec shelltest /bin/sh /tmp/shtest
#
# Then,
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.8
# enterprises.ucdavis.extTable.extEntry.extIndex.1 = 1
# enterprises.ucdavis.extTable.extEntry.extIndex.2 = 2
# enterprises.ucdavis.extTable.extEntry.extNames.1 = "echotest"
# enterprises.ucdavis.extTable.extEntry.extNames.2 = "shelltest"
# enterprises.ucdavis.extTable.extEntry.extCommand.1 = "/bin/echo hello world"
# enterprises.ucdavis.extTable.extEntry.extCommand.2 = "/bin/sh /tmp/shtest"
# enterprises.ucdavis.extTable.extEntry.extResult.1 = 0
# enterprises.ucdavis.extTable.extEntry.extResult.2 = 35
# enterprises.ucdavis.extTable.extEntry.extOutput.1 = "hello world."
# enterprises.ucdavis.extTable.extEntry.extOutput.2 = "hello world."
# enterprises.ucdavis.extTable.extEntry.extErrFix.1 = 0
# enterprises.ucdavis.extTable.extEntry.extErrFix.2 = 0
#
# Note that the second line of the /tmp/shtest shell script is cut
# off. Also note that the exit status of 35 was returned.
```

图3-8 可执行脚本

可以通过如下命令获得结果：

```
snmp -v 1 -c public localhost .1.3.6.1.4.1.2021.8
```


3.5 磁盘检查

```
#####
# disk checks
#
# The agent can check the amount of available disk space, and make
# sure it is above a set limit.
#
# disk PATH [MIN=DEFDISKMINIMUMSPACE]
#
# PATH:  mount path to the disk in question.
# MIN:   Disks with space below this value will have the Mib's errorFlag set.
#        Default value = DEFDISKMINIMUMSPACE.
#
# Check the / partition and make sure it contains at least 10 megs.
#
disk / 10000
#
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.9
# enterprises.ucdavis.diskTable.dskEntry.diskIndex.1 = 0
# enterprises.ucdavis.diskTable.dskEntry.diskPath.1 = "/" Hex: 2F
# enterprises.ucdavis.diskTable.dskEntry.diskDevice.1 = "/dev/dsk/c201d6s0"
# enterprises.ucdavis.diskTable.dskEntry.diskMinimum.1 = 10000
# enterprises.ucdavis.diskTable.dskEntry.diskTotal.1 = 837130
# enterprises.ucdavis.diskTable.dskEntry.diskAvail.1 = 316325
# enterprises.ucdavis.diskTable.dskEntry.diskUsed.1 = 437092
# enterprises.ucdavis.diskTable.dskEntry.diskPercent.1 = 58
```

图3-9 磁盘检查

可以通过如下命令获得结果：

```
snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.9
```

3.6 负载均衡检查

```
#####
# load average checks
#
# load [1MAX=DEFMAXLOADAVE] [5MAX=DEFMAXLOADAVE] [15MAX=DEFMAXLOADAVE]
#
# 1MAX:   If the 1 minute load average is above this limit at query
#         time, the errorFlag will be set.
# 5MAX:   Similar, but for 5 min average.
# 15MAX:  Similar, but for 15 min average.
#
# Check for loads:
load 12 14 14
#
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.10
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.1 = 1
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.2 = 2
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.3 = 3
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.1 = "Load-1"
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.2 = "Load-5"
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.3 = "Load-15"
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.1 = "0.49" Hex: 30 2E 34 39
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.2 = "0.31" Hex: 30 2E 33 31
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.3 = "0.26" Hex: 30 2E 32 36
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.1 = "12.00"
```

图3-10 负载均衡检查

可以通过如下命令获得结果：

```
snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.10
```

3.7 可扩展部分

```
#####  
# Extensible sections.  
#  
  
# This alleviates the multiple line output problem found in the  
# previous executable mib by placing each mib in its own mib table:  
  
# Run a shell script containing:  
#  
# #!/bin/sh  
# echo hello world  
# echo hi there  
# exit 35  
#  
# Note: this has been specifically commented out to prevent  
# accidental security holes due to someone else on your system writing  
# a /tmp/shtest before you do. Uncomment to use it.  
#  
# exec .1.3.6.1.4.1.2021.50 shelltest /bin/sh /tmp/shtest  
  
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.50  
# enterprises.ucdavis.50.1.1 = 1  
# enterprises.ucdavis.50.2.1 = "shelltest"  
# enterprises.ucdavis.50.3.1 = "/bin/sh /tmp/shtest"  
# enterprises.ucdavis.50.100.1 = 35  
# enterprises.ucdavis.50.101.1 = "hello world."
```

图3-11 可扩展部分

可以通过如下命令获得结果：

```
snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.50
```

3.8 通过控制

```
#####
# Pass through control.
#
# Usage:
#   pass MIBOID EXEC-COMMAND
#
# This will pass total control of the mib underneath the MIBOID
# portion of the mib to the EXEC-COMMAND.
#
# Note:  You'll have to change the path of the passtest script to your
# source directory or install it in the given location.
#
# Example:  (see the script for details)
#           (commented out here since it requires that you place the
#           script in the right location. (its not installed by default))
#
# pass .1.3.6.1.4.1.2021.255 /bin/sh PREFIX/local/passtest
#
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.255
# enterprises.ucdavis.255.1 = "life the universe and everything"
# enterprises.ucdavis.255.2.1 = 42
# enterprises.ucdavis.255.2.2 = OID: 42.42.42
# enterprises.ucdavis.255.3 = Timeticks: (363136200) 42 days, 0:42:42
# enterprises.ucdavis.255.4 = IPAddress: 127.0.0.1
# enterprises.ucdavis.255.5 = 42
```

图3-12 通过控制

可以通过如下命令获得结果：

```
snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.255
```

3.9 其它

```
#####
# Subagent control
#
# The agent can support subagents using a number of extension mechanisms.
# From the 4.2.1 release, AgentX support is being compiled in by default.
# To use this mechanism, simply uncomment the following directive.
#
# master agentx
#
# Please see the file README.agentx for more details.
#
#####
# Further Information
#
# See the snmpd.conf manual page, and the output of "snmpd -H".
# MUCH more can be done with the snmpd.conf than is shown as an
# example here.
```

图3-13 其它

第4章 net-snmp 指令

net-snmp 提供了许多工具使用和调试 SNMP,但从我个人的使用来看常用的指令有 snmpd、snmpwalk、snmpget、snmpconf 四条指令。下面将所有指令的功能。

4.1 snmpget

模拟 SNMP 的 GetRequest 操作的工具。用来获取一个或几个管理信息。用来读取管理信息的内容。

4.2 snmpwalk

利用 GetNextRequest 对给定的管理树进行遍历的工具。一般用来对表格类型管理信息进行遍历。

4.3 snmpconf

生成 snmpd 配置文件的工具。用于生成 snmpd 的各种配置文件,用作模板,以生成用户级配置文件。

4.4 snmpd

net-snmp 开发的主代理程序，包括众多标准 MIB 的实现，还可以使用子代理进行扩展，是一个功能强大的 SNMP 代理。运行 snmpd 后，操作系统直接具备了 SNMP 协议支持，可以被管理站管理。

4.5 snmpgetnext

模拟 SNMP 的 GetNextRequest 操作的工具。用来获取一个管理信息实例的下一个可用实例数据。

4.6 snmpset

模拟 SNMP 的 SetRequest 操作的工具。用来设置可以写的管理信息。一般用来配置设备或对设备执行操作。

4.7 snmpbulkget

模拟 SNMP 的 GetBulkRequest 操作的工具。用来读取大块的数据。一般在大量读取大块数据时使用以提高带宽利用率，并且比使用 snmpget、snmpgetnext、及 snmpwalk 有更强的容错能力，代理会返回尽可能多的数据，比其他命令更有保证。

4.8 snmptrap

模拟发送 trap 的工具。用来发送模拟 Trap。一般用来测试管理站安装和配置是否正确，或者用来验证开发的 Trap 接收程序是否可以正常工作。

4.9 snmptrapd

接收并显示 Trap 的工具。一般用在代理的开发过程中，接收代理发来的 Trap，并将 PDU 细节打印出来，用来测试 Trap 发送功能是否正确。

4.10 snmpinform

模拟发送 InformRequest 的工具。跟 snmptrap 类似，用来发送模拟的带应答的 Trap，以测试管理站或自己开发的接收程序。

4.11 snmptable

使用 GetNextRequest 和 GetBulkRequest 操作读取表信息，以列表形式显示的工具。

4.12 snmpstatus

使用 **SNMP** 实体中读取几个重要的管理信息以确定设备状态的工具。用来简单测试设备状态。

4.13 snmpbulkwalk

利用 **GetBulkRequest** 实现对给定管理树进行遍历的工具。对表格类形管理信息进行遍历读取。

4.14 snmpdelta

用来监视 **Integer** 类型的管理对象，会及时报告值改变情况的工具。用来监测一个设备或开发中的代理。

4.15 snmptest

一个复杂的工具，可以监测和管理一个网络实体的信息，通过 **SNMP** 请求操作与管理实体通信。

4.16 snmptranslate

将对象名字和标识符相互转换的工具。用于数据格式的对象标识符和可读式字符串的数据名称的转换。类似于域名和 **IP** 地址的关系。

4.17 snmpusm

SNMPv3 **USM** 配置工具。用于 **SNMPv3** 的用户管理。

4.18 snmpvacm

为一个网络实体或维护 **SNMPv3** 的基于视图访问控制参数的工具。用于维护 **SNMPv3** 的视图访问控制。

4.19 snmpdf

通过 **SNMP** 访问并显示网络实体磁盘利用情况的工具。用来监测网络实体的磁盘。

第5章 JAVA 开发

5.1 NET-SNMP 采集示例程序（Java）

```
package com.aaron.snmp4j;
```

```
import java.io.IOException;
import org.snmp4j.CommunityTarget;
import org.snmp4j.PDU;
import org.snmp4j.Snmp;
import org.snmp4j.event.ResponseEvent;
import org.snmp4j.mp.SnmpConstants;
import org.snmp4j.smi.OID;
import org.snmp4j.smi.OctetString;
import org.snmp4j.smi.UdpAddress;
import org.snmp4j.smi.VariableBinding;
import org.snmp4j.transport.DefaultUdpTransportMapping;
import java.util.logging.Logger;

/**
 * SNMP管理类
 *
 * @author <a href="mailto:zhang.bing@neusoft.com">张兵</a>
 * @version $Revision 1.1 $ 2008-1-15 下午02:08:25
 */
public class SNMPManage {
    // SNMP的空闲CPU百分比的OID
    private static final String OID_FREE_CPU = "1.3.6.1.4.1.2021.11.11.0";

    /**
     * 获得空闲CPU百分比的OID
     *
     * @return String
     */
    public static String getOID_FREE_CPU() {
        return OID_FREE_CPU;
    }

    /**
     * 根据OID得到对应OID的值
     * @param address
     * @param OID
     * @param community
     * @return
     */
    public String getValueByOID(String address, String OID, String community)
    {
        try {
            // 使用DefaultUdpTransportMapping对象实例化SNMP对象
            Snmp snmp = new Snmp(new DefaultUdpTransportMapping());
            // 实例化CommunityTarget对象并设置其属性
            CommunityTarget target = new CommunityTarget();
            target.setCommunity(new OctetString(community));
            target.setVersion(SnmpConstants.version2c);
            target.setAddress(new UdpAddress(address));
            target.setRetries(1);
            target.setTimeout(5000);
            //
            snmp.listen();
            // 请求
            PDU request = new PDU();
            request.setType(PDU.GET);
        }
    }
}
```



```

        request.add(new VariableBinding(new OID(OID)));
        // 响应
        PDU response = null;
        ResponseEvent responseEvent = snmp.send(request, target);
        response = responseEvent.getResponse();
        // 分析响应结果
        if (response != null) {
            if (response.getErrorIndex() == PDU.noError
                && response.getErrorStatus() == PDU.noError) {
                String pause = responseEvent.getResponse()
                    .getVariableBindings().toString();
                return pause;
            } else {
                return "geting is the fail.";
            }
        } else {
            return "response is null";
        }
    } catch (IOException ioe) {
        ioe.printStackTrace();
        return "geting is the exception.";
    }
}

/**
 * 主程序
 *
 * @param args
 */
public static void main(String args[]) {
    // 日志
    Logger logger = Logger.getLogger("global");
    SNMPManage snmpManage = new SNMPManage();
    // SNMP地址及端口
    String address = "192.168.228.254/161";
    // SNMP的共同体
    String community = "public";
    // 执行getResponse方法并显示在后台
    logger.info(snmpManage.getResponse(address,
        SNMPManage.getOID_FREE_CPU(), community));
}
}

```

5.2 NET-SNMP 主动发送示例程序（Java）

```

/**
 * 主动发送告警信息
 * @param currenttime
 * @param alarmcode
 * @param systemName
 * @param content
 * @param code
 */
public static void SendSnmptTrap(String currenttime, String alarmcode,
    String systemName, String content, int code) {

```

```

// TODO Auto-generated method stub
logger.info("time=" + currenttime);
logger.info("alarmcode=" + alarmcode);
logger.info("systemName=" + systemName);
logger.info("content=" + content);
SnmpVarBindList lgarbage = new SnmpVarBindList();
lgarbage.addVarBind(new SnmpVarBind(new
SnmpOid(SNMPPParam.EnterpriseOid
    + ".1.1"), new SnmpString(currenttime)));
lgarbage.addVarBind(new SnmpVarBind(new
SnmpOid(SNMPPParam.EnterpriseOid
    + ".1.2"), new SnmpString(alarmcode)));
lgarbage.addVarBind(new SnmpVarBind(new
SnmpOid(SNMPPParam.EnterpriseOid
    + ".1.3"), new SnmpString(systemName)));
lgarbage.addVarBind(new SnmpVarBind(new
SnmpOid(SNMPPParam.EnterpriseOid
    + ".1.4"), new SnmpString(content)));
SnmpTrap snmpTrap = null;
if (code == 1) {
    snmpTrap = new SnmpTrap(
        new SnmpOid(SNMPPParam.EnterpriseOid + ".1"), lgarbage);
} else if (code == 2) {
    snmpTrap = new SnmpTrap(
        new SnmpOid(SNMPPParam.EnterpriseOid + ".2"), lgarbage);
} else if (code == 3) {
    snmpTrap = new SnmpTrap(
        new SnmpOid(SNMPPParam.EnterpriseOid + ".3"), lgarbage);
}
try {
    snmpTrap.setCommunityString("public");
    snmpTrap.setDestinationAddress(InetAddress
        .getByName(SNMPPParam.DestinationAddress));

    SnmpV3AdaptorServer snmpV3AdaptorServer = new
SnmpV3AdaptorServer(
    Integer.parseInt(SNMPPParam.DestinationPort) - 1);
    snmpV3AdaptorServer.setTrapPort(new Integer(Integer
        .parseInt(SNMPPParam.DestinationPort)));
    snmpV3AdaptorServer.start();
    snmpTrap.sendV2(snmpV3AdaptorServer);

} catch (Exception e) {
    logger.info(e.getMessage());
}
}

```

5.3 运行结果

```

2008-1-15 14:21:01 com.aaron.snmp4j.SNMPManage main
信息: [1.3.6.1.4.1.2021.11.11.0 = 100]

```

图5-1 程序执行结果

第6章 Linux 常用 OID

Linux 常用的 OID 包括 CPU、内存、磁盘三大部分，下面将列出其一般情况下的 OID。

6.1 CPU

表6-1 CPU 常用 OID

Load	
1 minute Load	.1.3.6.1.4.1.2021.10.1.3.1
5 minute Load	.1.3.6.1.4.1.2021.10.1.3.2
15 minute Load	.1.3.6.1.4.1.2021.10.1.3.3
CPU	
percentage of user CPU time	.1.3.6.1.4.1.2021.11.9.0
raw user cpu time	.1.3.6.1.4.1.2021.11.50.0
percentages of system CPU time	.1.3.6.1.4.1.2021.11.10.0
raw system cpu time	.1.3.6.1.4.1.2021.11.52.0
percentages of idle CPU time	.1.3.6.1.4.1.2021.11.11.0
raw idle cpu time	.1.3.6.1.4.1.2021.11.53.0
raw nice cpu time	.1.3.6.1.4.1.2021.11.51.0

6.2 内存

表6-2 内存常用 OID

Total Swap Size	.1.3.6.1.4.1.2021.4.3.0
Available Swap Space	.1.3.6.1.4.1.2021.4.4.0
Total RAM in machine	.1.3.6.1.4.1.2021.4.5.0
Total RAM used	.1.3.6.1.4.1.2021.4.6.0
Total RAM Free	.1.3.6.1.4.1.2021.4.11.0
Total RAM Shared	.1.3.6.1.4.1.2021.4.13.0
Total RAM Buffered	.1.3.6.1.4.1.2021.4.14.0
Total Cached Memory	.1.3.6.1.4.1.2021.4.15.0

6.3 磁盘

表6-3 磁盘常用 OID

Path where the disk is mounted	.1.3.6.1.4.1.2021.9.1.2.1
Path of the device for the partition	.1.3.6.1.4.1.2021.9.1.3.1
Total size of the disk/partion (kBytes)	.1.3.6.1.4.1.2021.9.1.6.1
Available space on the disk	.1.3.6.1.4.1.2021.9.1.7.1
Used space on the disk	.1.3.6.1.4.1.2021.9.1.8.1
Percentage of space used on disk	.1.3.6.1.4.1.2021.9.1.9.1
Percentage of inodes used on disk	.1.3.6.1.4.1.2021.9.1.10.1
System Uptime	.1.3.6.1.2.1.1.3.0

6.4 示例

表 6-1、表 6-2、表 6-3为常用的OID，为了验证OID的有效性、net-snmp的配置、MIB库是否正确，可以通过如下命令进行验证：

```
snmpget -v 1 -c "community" target_name_or_ip OID
```

例如通过如下指令获得硬盘总大小：

```
[root@localhost snmp]# snmpget -v 1 -c "public" 192.168.228.254 .1.3.6.1.4.1.2021.9.1.6.1
UCD-SNMP-MIB::dskTotal.1 = INTEGER: 75799584
```

第7章 snmpd.conf 示例配置

```
#####
#
# EXAMPLE.conf:
# An example configuration file for configuring the ucd-snmp snmpd agent.
#
#####
#
# This file is intended to only be an example. If, however, you want
# to use it, it should be placed in /usr/local/net-snmp/etc/snmp/snmpd.conf.
# When the snmpd agent starts up, this is where it will look for it.
#
```

```
# You might be interested in generating your own snmpd.conf file using
# the "snmpconf" program (perl script) instead. It's a nice menu
# based interface to writing well commented configuration files. Try it!
#
# Note: This file is automatically generated from EXAMPLE.conf.def.
# Do NOT read the EXAMPLE.conf.def file! Instead, after you have run
# configure & make, and then make sure you read the EXAMPLE.conf file
# instead, as it will tailor itself to your configuration.

# All lines beginning with a '#' are comments and are intended for you
# to read. All other lines are configuration commands for the agent.

#
# PLEASE: read the snmpd.conf(5) manual page as well!
#

#####
# Access Control
#####

# YOU SHOULD CHANGE THE "COMMUNITY" TOKEN BELOW TO A NEW KEYWORD
ONLY
# KNOWN AT YOUR SITE. YOU *MUST* CHANGE THE NETWORK TOKEN BELOW TO
# SOMETHING REFLECTING YOUR LOCAL NETWORK ADDRESS SPACE.

# By far, the most common question I get about the agent is "why won't
# it work?", when really it should be "how do I configure the agent to
# allow me to access it?"
#
# By default, the agent responds to the "public" community for read
# only access, if run out of the box without any configuration file in
# place. The following examples show you other ways of configuring
# the agent so that you can change the community names, and give
# yourself write access as well.
```

```
#
# The following lines change the access permissions of the agent so
# that the COMMUNITY string provides read-only access to your entire
# NETWORK (EG: 10.10.10.0/24), and read/write access to only the
# localhost (127.0.0.1, not its real ipaddress).
#
# For more information, read the FAQ as well as the snmpd.conf(5)
# manual page.
```

```
####
# First, map the community name (COMMUNITY) into a security name
# (local and mynetwork, depending on where the request is coming
# from):
```

```
#   sec.name  source      community
com2sec local  localhost  COMMUNITY
com2sec mynetwork NETWORK/24  COMMUNITY
```

```
####
# Second, map the security names into group names:
```

```
#           sec.model  sec.name
group MyRWGroup  v1      local
group MyRWGroup  v2c     local
group MyRWGroup  usm      local
group MyROGroup  v1       mynetwork
group MyROGroup  v2c      mynetwork
group MyROGroup  usm      mynetwork
```

```
####
# Third, create a view for us to let the groups have rights to:
```

```
#       incl/excl subtree      mask
view all  included .1          80
```

####

Finally, grant the 2 groups access to the 1 view with different

write permissions:

```
#          context sec.model sec.level match  read  write notif
access MyROGroup ""    any    noauth  exact all  none  none
access MyRWGroup ""    any    noauth  exact all  all   none
```

#####

System contact information

#

It is also possible to set the sysContact and sysLocation system
variables through the snmpd.conf file. ****PLEASE NOTE**** that setting
the value of these objects here makes these objects READ-ONLY
(regardless of any access control settings). Any attempt to set the
value of an object whose value is given here will fail with an error
status of notWritable.

syslocation Right here, right now.

syscontact Me <me@somewhere.org>

Example output of snmpwalk:

```
# % snmpwalk -v 1 -c public localhost system
# system.sysDescr.0 = "SunOS name sun4c"
# system.sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.sunos4
# system.sysUpTime.0 = Timeticks: (595637548) 68 days, 22:32:55
# system.sysContact.0 = "Me <me@somewhere.org>"
# system.sysName.0 = "name"
# system.sysLocation.0 = "Right here, right now."
# system.sysServices.0 = 72
```

```
# -----

#####

# Process checks.
#
# The following are examples of how to use the agent to check for
# processes running on the host. The syntax looks something like:
#
# proc NAME [MAX=0] [MIN=0]
#
# NAME: the name of the process to check for. It must match
#       exactly (ie, http will not find httpd processes).
# MAX:  the maximum number allowed to be running. Defaults to 0.
# MIN:  the minimum number to be running. Defaults to 0.

#
# Examples:
#
# Make sure mountd is running
proc mountd

# Make sure there are no more than 4 ntalkds running, but 0 is ok too.
proc ntalkd 4

# Make sure at least one sendmail, but less than or equal to 10 are running.
proc sendmail 10 1

# A snmpwalk of the prTable would look something like this:
#
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.2
# enterprises.ucdavis.procTable.prEntry.prIndex.1 = 1
# enterprises.ucdavis.procTable.prEntry.prIndex.2 = 2
```



```
# enterprises.ucdavis.procTable.prEntry.prIndex.3 = 3
# enterprises.ucdavis.procTable.prEntry.prNames.1 = "mountd"
# enterprises.ucdavis.procTable.prEntry.prNames.2 = "ntalkd"
# enterprises.ucdavis.procTable.prEntry.prNames.3 = "sendmail"
# enterprises.ucdavis.procTable.prEntry.prMin.1 = 0
# enterprises.ucdavis.procTable.prEntry.prMin.2 = 0
# enterprises.ucdavis.procTable.prEntry.prMin.3 = 1
# enterprises.ucdavis.procTable.prEntry.prMax.1 = 0
# enterprises.ucdavis.procTable.prEntry.prMax.2 = 4
# enterprises.ucdavis.procTable.prEntry.prMax.3 = 10
# enterprises.ucdavis.procTable.prEntry.prCount.1 = 0
# enterprises.ucdavis.procTable.prEntry.prCount.2 = 0
# enterprises.ucdavis.procTable.prEntry.prCount.3 = 1
# enterprises.ucdavis.procTable.prEntry.prErrorFlag.1 = 1
# enterprises.ucdavis.procTable.prEntry.prErrorFlag.2 = 0
# enterprises.ucdavis.procTable.prEntry.prErrorFlag.3 = 0
# enterprises.ucdavis.procTable.prEntry.prErrMsg.1 = "No mountd process running."
# enterprises.ucdavis.procTable.prEntry.prErrMsg.2 = ""
# enterprises.ucdavis.procTable.prEntry.prErrMsg.3 = ""
# enterprises.ucdavis.procTable.prEntry.prErrFix.1 = 0
# enterprises.ucdavis.procTable.prEntry.prErrFix.2 = 0
# enterprises.ucdavis.procTable.prEntry.prErrFix.3 = 0
#
# Note that the errorFlag for mountd is set to 1 because one is not
# running (in this case an rpc.mountd is, but thats not good enough),
# and the ErrMessage tells you what's wrong. The configuration
# imposed in the snmpd.conf file is also shown.
#
# Special Case: When the min and max numbers are both 0, it assumes
# you want a max of infinity and a min of 1.
#
# -----
```

```
#####
```

```
# Executables/scripts
```

```
#
```

```
#
```

```
# You can also have programs run by the agent that return a single
```

```
# line of output and an exit code. Here are two examples.
```

```
#
```

```
# exec NAME PROGRAM [ARGS ...]
```

```
#
```

```
# NAME: A generic name.
```

```
# PROGRAM: The program to run. Include the path!
```

```
# ARGS: optional arguments to be passed to the program
```

```
# a simple hello world
```

```
exec echotest /bin/echo hello world
```

```
# Run a shell script containing:
```

```
#
```

```
# #!/bin/sh
```

```
# echo hello world
```

```
# echo hi there
```

```
# exit 35
```

```
#
```

```
# Note: this has been specifically commented out to prevent
```

```
# accidental security holes due to someone else on your system writing
```

```
# a /tmp/shctest before you do. Uncomment to use it.
```

```
#
```

```
#exec shelltest /bin/sh /tmp/shctest
```

```
# Then,
```

```
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.8
```

```
# enterprises.ucdavis.extTable.extEntry.extIndex.1 = 1
```

```
# enterprises.ucdavis.extTable.extEntry.extIndex.2 = 2
```

```
# enterprises.ucdavis.extTable.extEntry.extNames.1 = "echotest"
# enterprises.ucdavis.extTable.extEntry.extNames.2 = "shelltest"
# enterprises.ucdavis.extTable.extEntry.extCommand.1 = "/bin/echo hello world"
# enterprises.ucdavis.extTable.extEntry.extCommand.2 = "/bin/sh /tmp/shtest"
# enterprises.ucdavis.extTable.extEntry.extResult.1 = 0
# enterprises.ucdavis.extTable.extEntry.extResult.2 = 35
# enterprises.ucdavis.extTable.extEntry.extOutput.1 = "hello world."
# enterprises.ucdavis.extTable.extEntry.extOutput.2 = "hello world."
# enterprises.ucdavis.extTable.extEntry.extErrFix.1 = 0
# enterprises.ucdavis.extTable.extEntry.extErrFix.2 = 0

# Note that the second line of the /tmp/shtest shell script is cut
# off. Also note that the exit status of 35 was returned.

# -----

#####

# disk checks
#

# The agent can check the amount of available disk space, and make
# sure it is above a set limit.

# disk PATH [MIN=DEFDISKMINIMUMSPACE]
#
# PATH: mount path to the disk in question.
# MIN: Disks with space below this value will have the Mib's errorFlag set.
#     Default value = DEFDISKMINIMUMSPACE.

# Check the / partition and make sure it contains at least 10 megs.

disk / 10000

# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.9
```

```
# enterprises.ucdavis.diskTable.dskEntry.diskIndex.1 = 0
# enterprises.ucdavis.diskTable.dskEntry.diskPath.1 = "/" Hex: 2F
# enterprises.ucdavis.diskTable.dskEntry.diskDevice.1 = "/dev/dsk/c201d6s0"
# enterprises.ucdavis.diskTable.dskEntry.diskMinimum.1 = 10000
# enterprises.ucdavis.diskTable.dskEntry.diskTotal.1 = 837130
# enterprises.ucdavis.diskTable.dskEntry.diskAvail.1 = 316325
# enterprises.ucdavis.diskTable.dskEntry.diskUsed.1 = 437092
# enterprises.ucdavis.diskTable.dskEntry.diskPercent.1 = 58
# enterprises.ucdavis.diskTable.dskEntry.diskErrorFlag.1 = 0
# enterprises.ucdavis.diskTable.dskEntry.diskErrorMsg.1 = ""

# -----

#####

# load average checks
#

#          load          [1MAX=DEFMAXLOADAVE]          [5MAX=DEFMAXLOADAVE]
# [15MAX=DEFMAXLOADAVE]

#
# 1MAX:  If the 1 minute load average is above this limit at query
#        time, the errorFlag will be set.
# 5MAX:  Similar, but for 5 min average.
# 15MAX: Similar, but for 15 min average.

# Check for loads:
load 12 14 14

# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.10
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.1 = 1
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.2 = 2
# enterprises.ucdavis.loadTable.laEntry.loadaveIndex.3 = 3
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.1 = "Load-1"
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.2 = "Load-5"
```

```
# enterprises.ucdavis.loadTable.laEntry.loadaveNames.3 = "Load-15"
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.1 = "0.49" Hex: 30 2E 34 39
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.2 = "0.31" Hex: 30 2E 33 31
# enterprises.ucdavis.loadTable.laEntry.loadaveLoad.3 = "0.26" Hex: 30 2E 32 36
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.1 = "12.00"
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.2 = "14.00"
# enterprises.ucdavis.loadTable.laEntry.loadaveConfig.3 = "14.00"
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.1 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.2 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrorFlag.3 = 0
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMsg.1 = ""
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMsg.2 = ""
# enterprises.ucdavis.loadTable.laEntry.loadaveErrMsg.3 = ""

# -----

#####

# Extensible sections.
#

# This alleviates the multiple line output problem found in the
# previous executable mib by placing each mib in its own mib table:

# Run a shell script containing:
#
# #!/bin/sh
# echo hello world
# echo hi there
# exit 35
#
# Note: this has been specifically commented out to prevent
# accidental security holes due to someone else on your system writing
# a /tmp/shctest before you do. Uncomment to use it.
#
```

```
# exec .1.3.6.1.4.1.2021.50 shelltest /bin/sh /tmp/shtest
```

```
# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.50
```

```
# enterprises.ucdavis.50.1.1 = 1
```

```
# enterprises.ucdavis.50.2.1 = "shelltest"
```

```
# enterprises.ucdavis.50.3.1 = "/bin/sh /tmp/shtest"
```

```
# enterprises.ucdavis.50.100.1 = 35
```

```
# enterprises.ucdavis.50.101.1 = "hello world."
```

```
# enterprises.ucdavis.50.101.2 = "hi there."
```

```
# enterprises.ucdavis.50.102.1 = 0
```

```
# Now the Output has grown to two lines, and we can see the 'hi
```

```
# there.' output as the second line from our shell script.
```

```
#
```

```
# Note that you must alter the mib.txt file to be correct if you want
```

```
# the .50.* outputs above to change to reasonable text descriptions.
```

```
# Other ideas:
```

```
#
```

```
# exec .1.3.6.1.4.1.2021.51 ps /bin/ps
```

```
# exec .1.3.6.1.4.1.2021.52 top /usr/local/bin/top
```

```
# exec .1.3.6.1.4.1.2021.53 mailq /usr/bin/mailq
```

```
# -----
```

```
#####
```

```
# Pass through control.
```

```
#
```

```
# Usage:
```

```
# pass MIBOID EXEC-COMMAND
```

```
#
```

```
# This will pass total control of the mib underneath the MIBOID
```

```
# portion of the mib to the EXEC-COMMAND.
```

```
#
# Note: You'll have to change the path of the passtest script to your
# source directory or install it in the given location.
#
# Example: (see the script for details)
#      (commented out here since it requires that you place the
#      script in the right location. (its not installed by default))

# pass .1.3.6.1.4.1.2021.255 /bin/sh PREFIX/local/passtest

# % snmpwalk -v 1 -c public localhost .1.3.6.1.4.1.2021.255
# enterprises.ucdavis.255.1 = "life the universe and everything"
# enterprises.ucdavis.255.2.1 = 42
# enterprises.ucdavis.255.2.2 = OID: 42.42.42
# enterprises.ucdavis.255.3 = Timeticks: (363136200) 42 days, 0:42:42
# enterprises.ucdavis.255.4 = IpAddress: 127.0.0.1
# enterprises.ucdavis.255.5 = 42
# enterprises.ucdavis.255.6 = Gauge: 42
#
# % snmpget -v 1 -c public localhost .1.3.6.1.4.1.2021.255.5
# enterprises.ucdavis.255.5 = 42
#
# % snmpset -v 1 -c public localhost .1.3.6.1.4.1.2021.255.1 s "New string"
# enterprises.ucdavis.255.1 = "New string"
#

# For specific usage information, see the man/snmpd.conf.5 manual page
# as well as the local/passtest script used in the above example.

#####

# Subagent control
#

# The agent can support subagents using a number of extension mechanisms.
# From the 4.2.1 release, AgentX support is being compiled in by default.
```

To use this mechanism, simply uncomment the following directive.

#

master agentx

#

Please see the file README.agentx for more details.

#

#####

Further Information

#

See the snmpd.conf manual page, and the output of "snmpd -H".

MUCH more can be done with the snmpd.conf than is shown as an

example here.