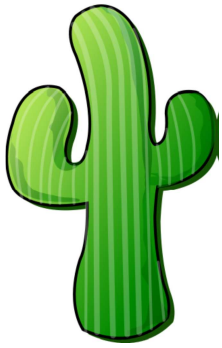
# Cacti

SNMPAgent Plugin Version 0.2

Usage Guide



# License

# Table of Contents

# 1 Preface

During one of my projects I had to deal with the formal process being necessary to generate SNMP notifications for foreign SNMP notification receivers. I spent some time to read up about the creation of MIBs, the different rules, syntax, et cetera, et cetera and then I thought: “Hey, we already have a private enterprise number registered by IANA.”

But the creation of a MIB for Cacti is only one part of the story. The challenge is to parse the MIB file and to make all related data reachable for SNMP. And this was the birth of the SNMPAgent plugin. Well, the SNMPAgent is *only* a system plugin. And the initial version did not offer any kind of a GUI.

This document will focus on its successor, providing an interface to review the content of the SNMP cache and offering the ability to register individual SNMP notification receivers. SNMPAgent works internally with a new PHP class which will allow the plugin itself as well as other plugins to update and maintain the content of the SNMP cache in a fast and efficient way. Yes, you have read correctly, Cacti plugins can hook into the SNMPAgent, register their own MIB (SMIv2) and send out SNMP notifications (informs/traps) automatically to all receivers listing to these messages. And therefore developers will never have to think about the different OIDs and required versions of SNMP. All they have to know is the notification name and the list of defined varbinds.

Like the initial version already did, this version includes some Cacti templates to demonstrate how you can use the SNMPAgent to monitor your Cacti server by Cacti itself using SNMP.

Andreas Braun

(browniebraun)

# 2 Cacti MIB Tree

## 2.1 Structure of Cacti's Management Information Base

```
           |
           +- cactiAppl
           +- cactiStats
           +- cactiEvents
           +- cactiPlugins
           +- cactiMibGroups
```

## 2.2 Expanded Cacti MIB Tree

```
REVISION        "201402030000Z"
           +- cactiAppl
           |  |
           |  +- cactiApplLastUpdate
           |  +- cactiApplVersion
           |  +- cactiApplSnmpVersion
           |  +- cactiApplRrdtoolVersion
           |  +- cactiApplPollerEnabled
           |  +- cactiApplPollerType
           |  +- cactiApplPollerInterval
           |  +- cactiApplLoadBalance
           |  +- cactiApplSpineMaxThreads
           |  +- cactiApplSpineScriptServers
           |  +- cactiApplSpineScriptTimeout
           |  +- cactiApplSpineMaxOids
           |  |
           |  +- cactiApplDeviceTable
           |  |  |
           |  |  +- cactiApplDevicEntry
           |  |     |
           |  |     +- cactiApplDeviceIndex
           |  |     +- cactiApplDeviceDescription
           |  |     +- cactiApplDeviceHostname
           |  |     +- cactiApplDeviceStatus
           |  |     +- cactiApplDeviceEventCount
           |  |     +- cactiApplDeviceFailDate
           |  |     +- cactiApplDeviceRecoveryDate
           |  |     +- cactiApplDeviceLastError
           |  |
           |  +- cactiApplPollerTable
           |  |  |
           |  |  +- cactiApplPollerEntry
           |  |     |
           |  |     +- cactiApplPollerIndex
           |  |     +- cactiApplPollerHostname
           |  |     +- cactiApplPollerIpAddress
           |  |     +- cactiApplPollerLastUpdate
           |  |
           |  +- cactiApplPluginTable
           |     |
           |     +- cactiApplPluginEntry
           |        |
           |        +- cactiApplPluginIndex
           |        +- cactiApplPluginType
           |        +- cactiApplPluginName
           |        +- cactiApplPluginStatus
           |        +- cactiApplPluginVersion
           |
           +- cactiStats
           |  |
           |  +- cactiStatsLastUpdate
           |  +- cactiStatsRecacheTime
```

```
|  +- cactiStatsRecachedHosts
|  +- cactiStatsLocalPollerRuntime
|  +- cactiStatsTotalsDevices
|  +- cactiStatsTotalsDataSources
|  +- cactiStatsTotalsGraphs
|  |
|  +- cactiStatsTotalsDeviceStatusTable
|  |  |
|  |  +- cactiStatsTotalsDeviceStatusEntry
|  |     |
|  |     +- cactiStatsTotalsDeviceStatusIndex
|  |     +- cactiStatsTotalsDeviceStatusCounter
|  |
|  +- cactiStatsDeviceTable
|  |  |
|  |  +- cactiStatsDeviceEntry
|  |     |
|  |     +- cactiStatsDeviceIndex
|  |     +- cactiStatsDeviceHostname
|  |     +- cactiStatsDeviceMinTime
|  |     +- cactiStatsDeviceMaxTime
|  |     +- cactiStatsdeviceCurTime
|  |     +- cactiStatsDeviceAvgTime
|  |     +- cactiStatsDeviceTotalPolls
|  |     +- cactiStatsDeviceFailedPolls
|  |     +- cactiStatsDeviceAvailability
|  |
|  +- cactiStatsPollerTable
|  |  |
|  |  +- cactiStatsPollerEntry
|  |     |
|  |     +- cactiStatsPollerIndex
|  |     +- cactiStatsPollerHostname
|  |     +- cactiStatsPollerRunTime
|  |     +- cactiStatsPollerMethod
|  |     +- cactiStatsPollerConcurrentProcesses
|  |     +- cactiStatsPollerThreads
|  |     +- cactiStatsPollerHosts
|  |     +- cactiStatsPollerHostsPerProcess
|  |     +- cactiStatsPollerItems
|  |     +- cactiStatsPollerRrrdsProcessed
|  |     +- cactiStatsPollerUtilization
|  |
|  +- cactiStatsTotalsDeviceStatusUnknown
|  +- cactiStatsTotalsDeviceStatusDown
|  +- cactiStatsTotalsDeviceStatusRecovering
|  +- cactiStatsTotalsDeviceStatusUp
|  +- cactiStatsTotalsDeviceStatusDisabled
|
+- cactiEvents
|  |
|  +- cactiEventAttributes
|  |  |
|  |  + - cactiEventDescription
|  |
|  +- cactiEventNotifications
|     |
|     + - cactiNotify
|     + - cactiNotifyDeviceDown
|     + - cactiNotifyDeviceRecovering
|     + - cactiNotifyPollerRuntimeExceeding
|     + - cactiNotifyDeviceFailedPoll
|
+- cactiPlugins
|  |
|  +- thold(1)
|  +- boost(2)
|  +- dsstats(3)
|
+- cactiMibGroups
   |
   + - cactiApplPollerGroup
   + - cactiApplSpineGroup
   + - cactiStatsTotalsDeviceGroup
   + - cactiNotifyGroup
```

# 3 System Requirements

- Cacti 0.8.8a (or above)
- PIA 3.1 (or above)
- Net-SNMP

# 4 Base Installation

The current manual is based on Cacti 0.8.8a and Plugin Architecture (PIA) 3.1. This plugin requires a hand full of hooks which are currently not part of Cacti. Most probably all required code changes will be part of Cacti 0.8.9

Please download this plugin to the <cacti_path>/plugin directory. And unpack the archive (.tgz or zip), so that all files will reside in <cacti_path>/plugin/SNMPAgent.

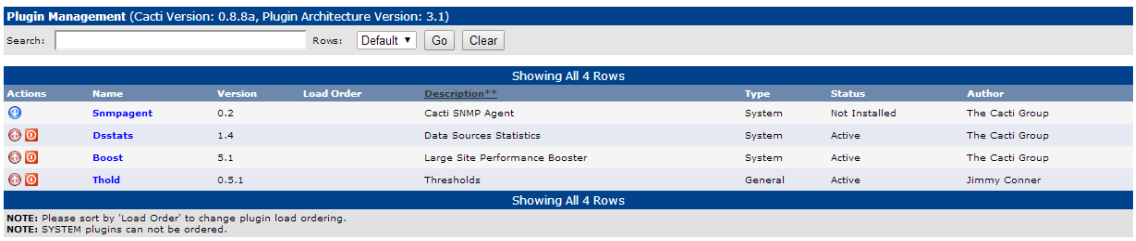
Within a sub-directory named “patches” you will find a folder containing all pre-patched files for Cacti 0.8.8a as well as 0.8.8b. Replace the original files of your local Cacti installation by these prepared ones. Alternatively you can execute the included patch file directly from the cacti main directory to update your system:

```
pi@raspi ~/patch/cacti_088a_test $ patch --dry-run -p1 -i plugins/snmpagent/patches/cacti-
0.8.8a/cacti088a.patch
patching file data_sources.php
patching file graphs.php
patching file host.php
patching file include/plugins.php
patching file poller.php
patching file settings.php
```

If successful:

```
patch -p1 -i plugins/snmpagent/patches/cacti-0.8.8a/cacti088a.patch
```

Now open the plugin management console and hit the install icon to start the installation procedure.




SNMPAgent automatically starts to parse all registered MIB files and to build up the SNMP cache. This step will take a while for midsized/large environments relating to the number of monitored hosts and registered MIB files. The default setup running on a system monitoring 2000 hosts takes around 17-20 seconds.

By hitting the “enable” button the SNMPAgent becomes active.

# 5 Extended Installation

On top of the traditional plugin installation described in chapter 4, the Cacti SNMPAgent requires a special configuration of your Net-SNMP instance if you would like to monitor Cacti by Cacti itself or any other NMS using SNMP.

## 5.1 Background

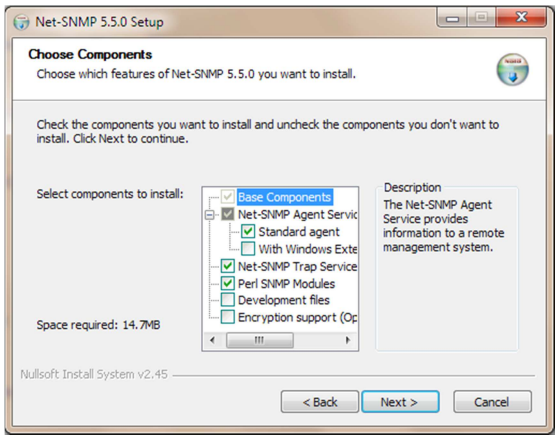
Net-SNMP provides several methods to extend the SNMP agent for custom applications. The Cacti SNMPAgent plugin focus on a method called "pass_persist", because this way allows having a memory persistent PHP script running. In detail every SNMP request against the Cacti Enterprise OID will be automatically forwarded to the Cacti SNMPAgent, which will respond in a standardized format.

Due to performance reasons the SNMP cache will be refreshed only every 60 seconds. Following steps will explain how you can setup your system to support this method.

## 5.2 Windows

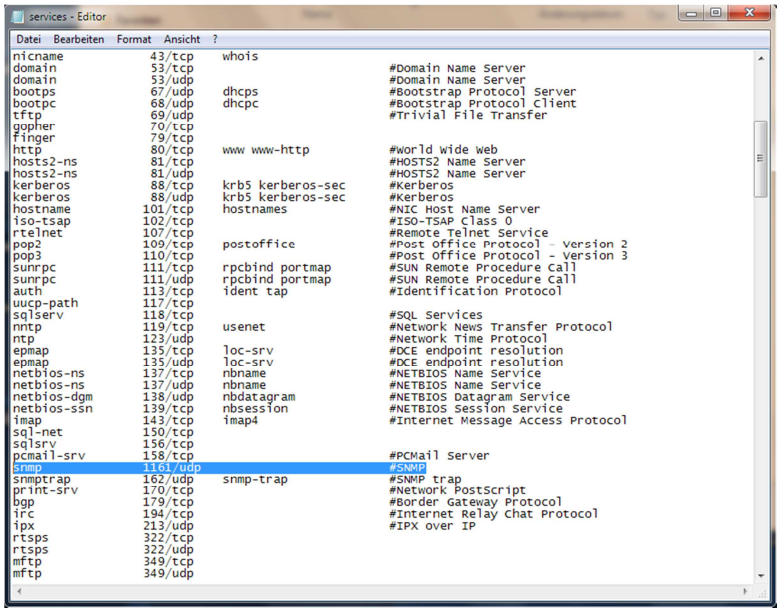
The setup of SNMP for Windows is a little bit tricky, because the common installation of Net-SNMP as a replacement for the Microsoft Windows Service does not support the extension of the SNMP agent using the pass_persist method.

An alternative to the use of the winExtDLL is to proxy requests from Net-SNMP to the Microsoft SNMP service. This means both services will run in co-existence to each other. In that case please install the Net-SNMP Agent Service as "Standard Agent" only.

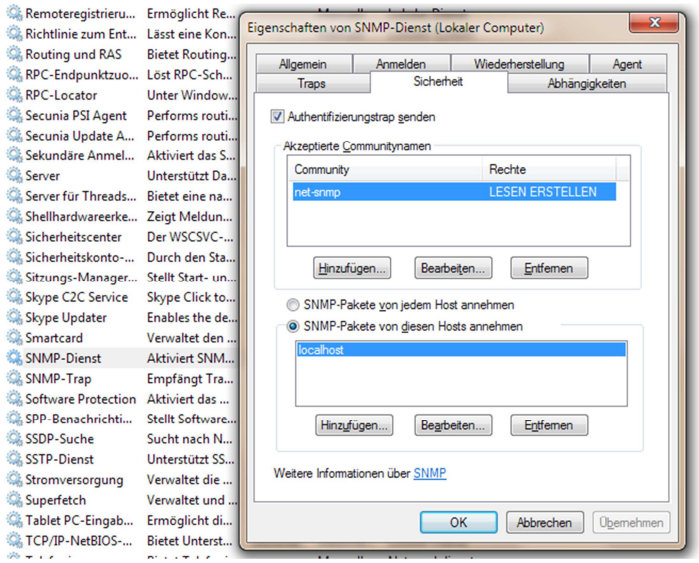


To allow both SNMP agents to run in parallel, it is necessary to change the default TCP/IP port either of the Microsoft or of the Net-SNMP service. It depends on your environment which agent you want to modify, but in the following example we will modify the Microsoft SNMP agent to listen on another port than 161 and the Net-SNMP to work as a SNMP proxy:

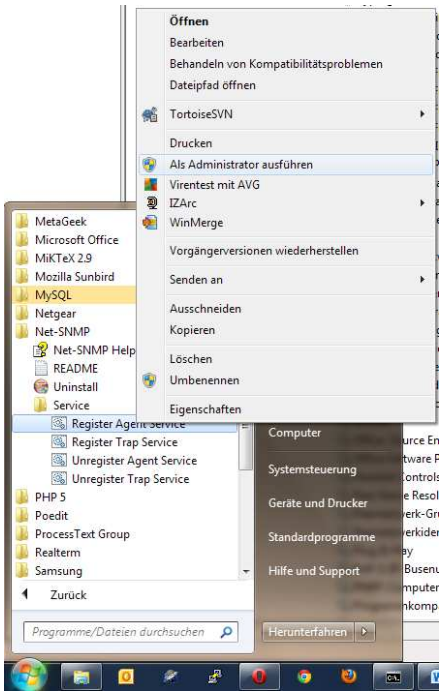
1. Edit with administrator permissions the SERVICES file located in %systemroot%\system32\drivers\etc and change the default port for SNMP from 161 to 1161.

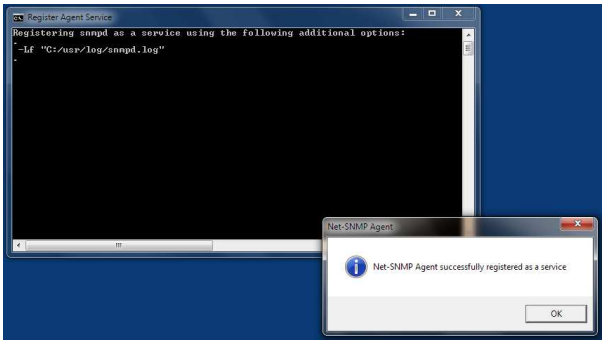



2. After saving that modification configure the Microsoft agent to only accept requests from localhost and add a read+create community string, which will later allow Net-SNMP to get full SNMP access:



3. Restart the Microsoft SNMP service. Note, that external services cannot query the MS agent directly anymore after that setup.
4. Now we can register the Net-SNMP service which will listen to the default port 161. Therefore execute as administrator the “Register Agent Service” script which is part of your Net-SNMP installation package.




5. Per default Net-SNMP does not include a configuration file for its SNMP daemon. With regard to the default installation paths of Net-SNMP go to C:\usr\etc\snmp and create a new file named “snmpd.conf”.
   As long as you do not have to monitor special applications like MS Exchange running on the same server you only need a few lines to configure the Net-SNMP service:

```
proxy -v 1 -c net-snmp localhost:1161 .1.3
pass_persist .1.3.6.1.4.1.23925 php <cacti_path>\plugins\snmpagent\persist.php
rocommunity <your_community_string>
```

   The first line ensures that all requests against branches of .1.3 will be forwarded automatically to the Microsoft SNMP agent. This is necessary to keep the monitoring of your server resources up and running within Cacti, due to the fact that the HOST-RESOURCE-MIB has not implemented in Net-SNMP yet.
   The second configuration line instructs the Net-SNMP agent to forward all requests against the Cacti MIB tree to the memory persistent script the Cacti SNMP-agent offers.

Define a community string that allows Cacti or another application to run SNMP queries against this server.

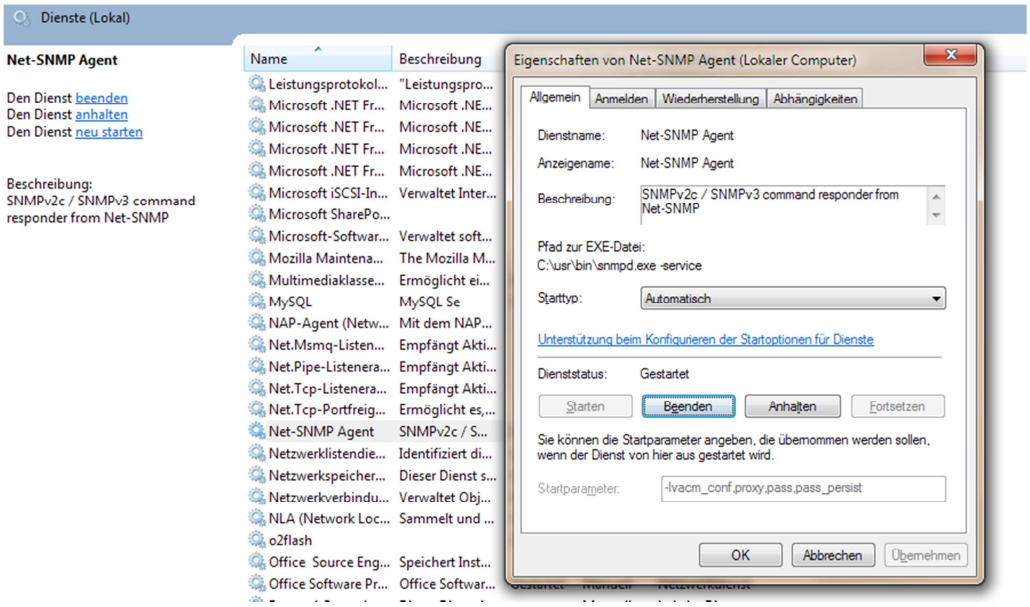
If you would like to see all numeric object identifies (OIDs) being translated into textual object identifiers automatically, then put a copy of the CACTI-MIB and all other plugin related MIBs into the default mib folder of Net-SNMP (e.g. c:/usr/share/snmp/mibs).

Update the second configuration file named “snmp.conf”:

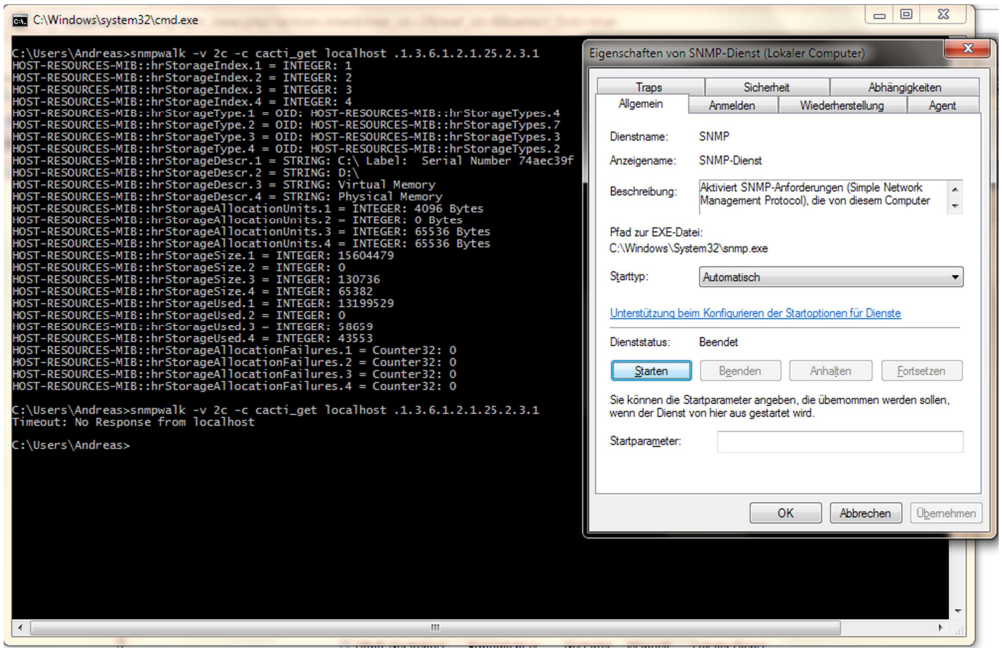
The last two lines of this configuration example define that the CACTI-MIB as well as the CACTI-SNMPAGENT-MIB have to be included.

```
mibdirs C:/usr/share/snmp/mibs
persistentDir C:/usr/snmp/persist
tempFilePattern C:/usr/temp/snmpdXXXXXX
mibs +CACTI-MIB
mibs +CACTI-SNMPAGENT-MIB
```

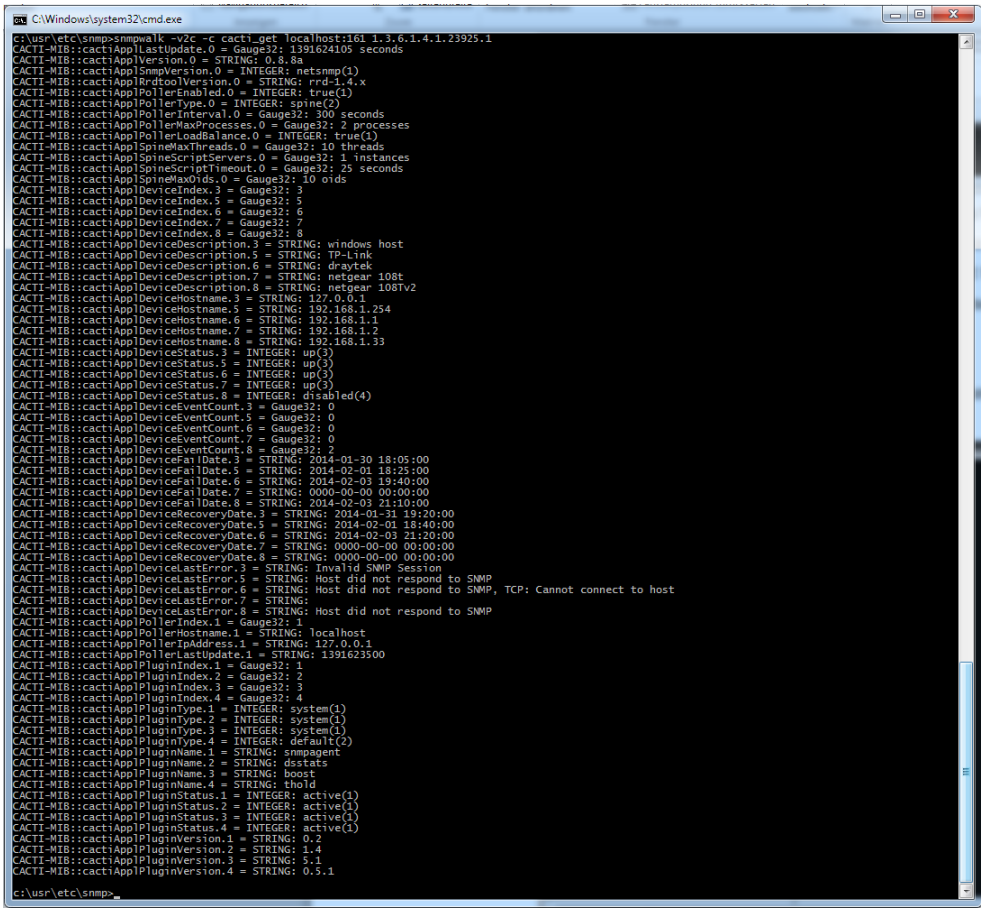
6. Start the Net-SNMP service by using start parameter “`-Ivacm_conf,proxy,pass,pass_persist`”. This specifies the modules that should be initialized when the agents starts.




7. Test the Net-SNMP agent to verify that requests against the host MIB will be forwarded to the MS agent. As shown below you can check that easily by running a snmpwalk against the Net-SNMP service querying the host MIB.




8. Now let's start a first snmpwalk against the branch "cactiAppl" of the Cacti MIB. The output will be generated by the SNMPAgent's persist script and should look similar to the following screenshot.



With a short look to the Linux configuration part, I'm pretty sure some of you Windows guys will be a little bit surprised about its length. Anyways, don't be disappointed and go forward to chapter 7.

## 5.3 Linux

Fortunately Linux admins don't have to run multiple SNMP agents in parallel. In most cases it should be enough to configure the SNMP daemon, so let's start by editing its configuration file (e.g. /etc/snmp/snmpd.conf).

At first we have to ensure that access to the Cacti MIB tree will be granted.

```
###############################################################################
#
#  ACCESS CONTROL
#
                                                #  system + hrSystem groups only
#view   systemonly  included   .1.3.6.1.2.1.1
#view   systemonly  included   .1.3.6.1.2.1.25.1
view   all         included   .1.3.6.1.4.1.23925
                                                #  Full access from the local host
#rocommunity public  localhost
                                                #  Default access to basic system info
rocommunity <your_community_string>
```

Additionally add a read community string if necessary before we have to register the pass_persist script of the SNMPAgent.

```
###############################################################################
#
#  EXTENDING THE AGENT
#

#
#  Arbitrary extension commands
#
# extend    test1   /bin/echo  Hello, world!
# extend-sh test2   echo Hello, world! ; echo Hi there ; exit 35
#extend-sh test3   /bin/sh /tmp/shtest

#  Note that this last entry requires the script '/tmp/shtest' to be created first,
#    containing the same three shell commands, before the line is uncommented

#  Walk the NET-SNMP-EXTEND-MIB tables (nsExtendConfigTable, nsExtendOutput1Table
#     and nsExtendOutput2Table) to see the resulting output

#  Note that the "extend" directive supercedes the previous "exec" and "sh" directives
#  However, walking the UCD-SNMP-MIB::extTable should still returns the same output,
#     as well as the fuller results in the above tables.


#
#  "Pass-through" MIB extension command
#
#pass .1.3.6.1.4.1.8072.2.255  /bin/sh       PREFIX/local/passtest
#pass .1.3.6.1.4.1.8072.2.255  /usr/bin/perl PREFIX/local/passtest.pl
pass_persist .1.3.6.1.4.1.23925 <cacti_path>/plugins/snmpagent/persist.php
```

Make sure that the pass_persist script is executable and update the the path to PHP at the beginnig of the persist script if necessary (or put the right path to PHP in front of the path to the persist script).

```
chmod a+x <cacti_path>/plugins/snmpagent/persist.php
```

After that you can restart the snmp daemon.

```
e.g.:  sudo /etc/init.d/snmpd restart
```

Now the script will automatically respond to a SNMP query running against the Cacti Enterprise OID.

# 6 Updates

If you have already used a previous version of SNMPAgent then you should proceed following steps to upgrade your system:

1. Copy the archive of the new version into your Cacti plugin folder
2. Disable the SNMPAgent using Cacti's Plugin Management Console
3. Stop the Net-SNMP daemon
4. Delete the "snmpagent" folder
5. Extract the archive
6. If necessary replace all Cacti system files by the prepared ones the new snmpagent folder includes or use the included patch file instead.

| Update to | Patch files required | |
|---|---|---|
| | 0.8.8a | 0.8.8b |
| 0.2.x | yes | yes |

7. Enable the SNMPAgent using Cacti's Plugin Management Console
8. Start the Net-SNMP daemon

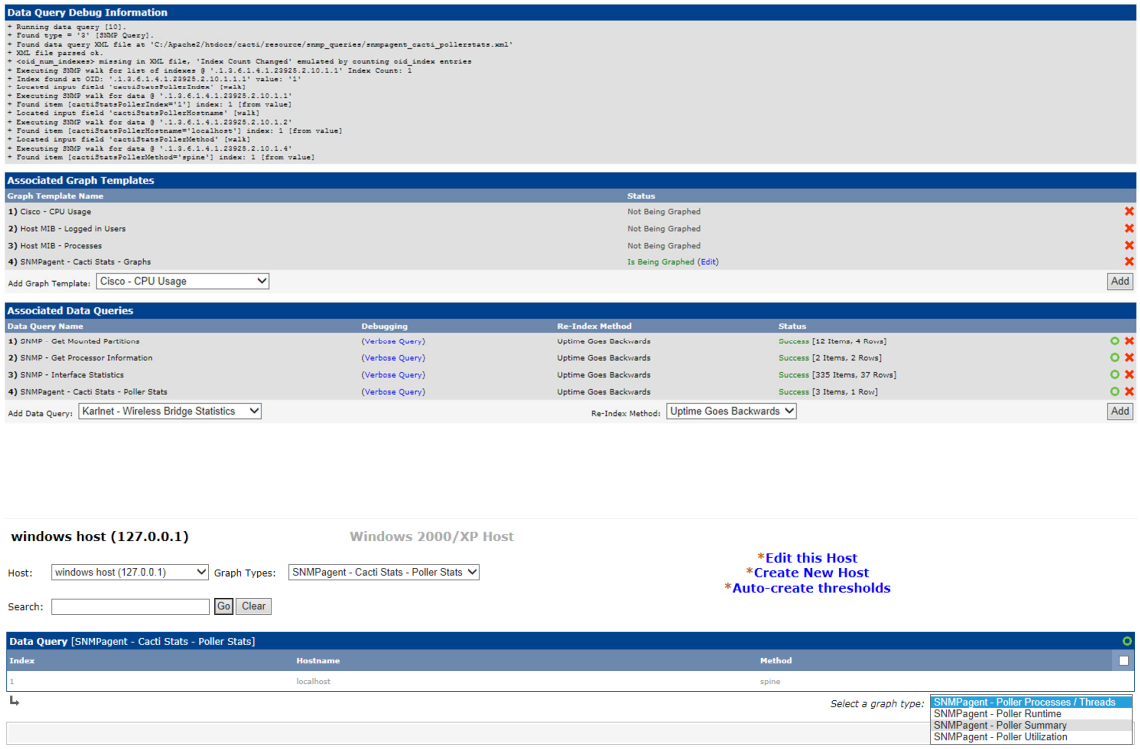
# 7 Monitoring Cacti with Cacti

## 7.1 Templates

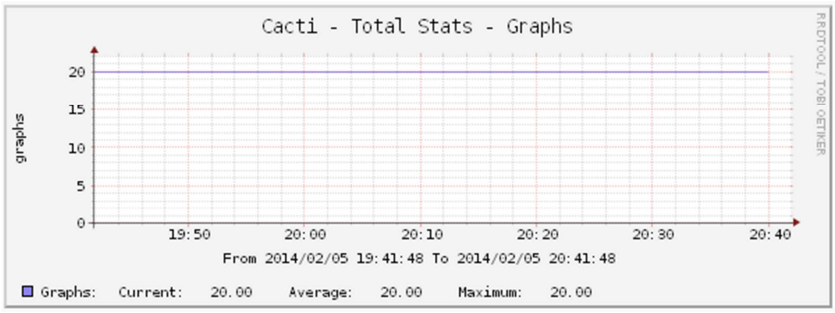
Well, I'm wound up to a high pitch what will be built by you, the Cacti community, in the future, but to demonstrate how you can use the SNMPAgent plugin to monitor Cacti by itself the SNMPAgent already offers a data query that allows monitoring some specific KPIs of Cacti. The snmp query and templates are located under:

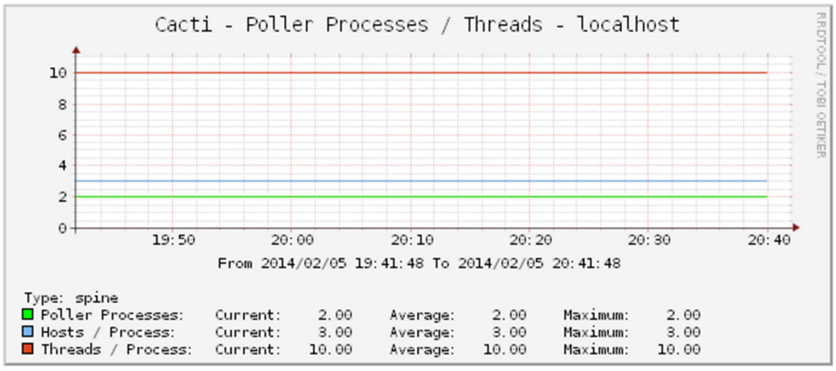
```
<cacti_path>/plugins/snmpagent/templates/cacti/poller_statistics
<cacti_path>/plugins/snmpagent/templates/cacti/totals
```

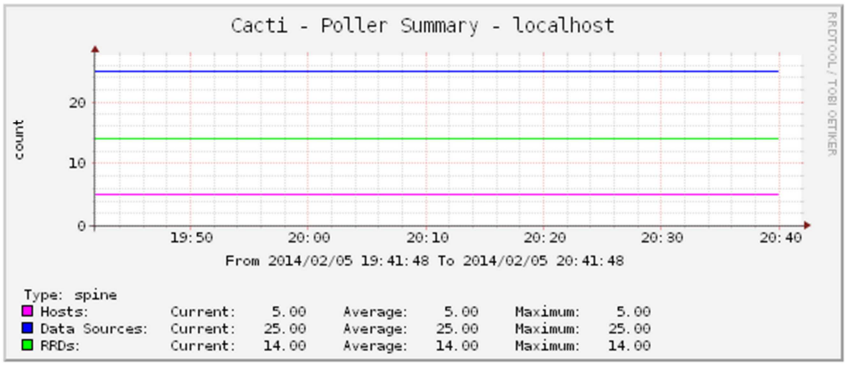
Copy the snmp query to <cacti_path>/resource/snmp_queries and use Cacti's import functionality to add the new templates to your system. After that you can add the new data query / graph template to your Cacti host and create new graphs:

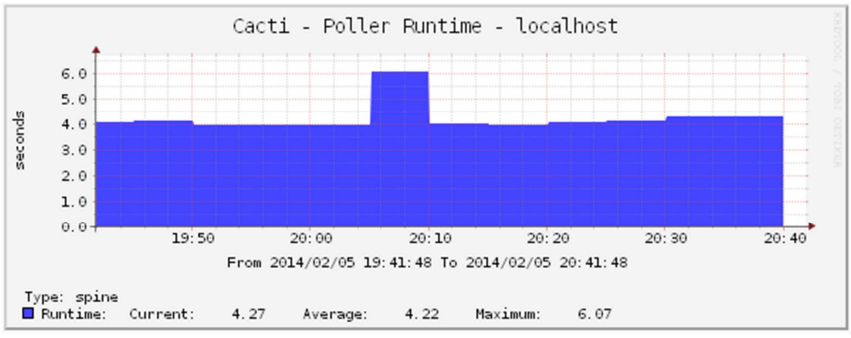


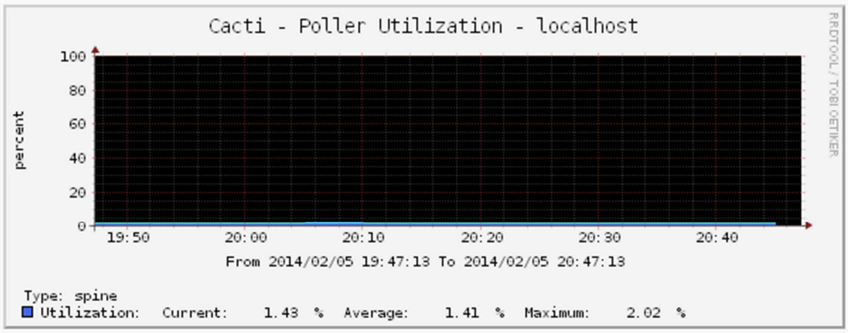
## 7.2 Graphs


Cacti - Total Stats - Graphs
graphs
20
15
10
5
0
19:50
20:00
20:10
20:20
20:30
20:40
From 2014/02/05 19:41:48 To 2014/02/05 20:41:48
Graphs: Current: 20.00 Average: 20.00 Maximum: 20.00
RRDTOOL / TOBI OETIKER


Cacti - Poller Processes / Threads - localhost
10
8
6
4
2
0
19:50
20:00
20:10
20:20
20:30
20:40
From 2014/02/05 19:41:48 To 2014/02/05 20:41:48
Type: spine
Poller Processes: Current: 2.00 Average: 2.00 Maximum: 2.00
Hosts / Process: Current: 3.00 Average: 3.00 Maximum: 3.00
Threads / Process: Current: 10.00 Average: 10.00 Maximum: 10.00
RRDTOOL / TOBI OETIKER


Cacti - Poller Summary - localhost
count
20
10
0
19:50
20:00
20:10
20:20
20:30
20:40
From 2014/02/05 19:41:48 To 2014/02/05 20:41:48
Type: spine
Hosts: Current: 5.00 Average: 5.00 Maximum: 5.00
Data Sources: Current: 25.00 Average: 25.00 Maximum: 25.00
RRDs: Current: 14.00 Average: 14.00 Maximum: 14.00
RRDTOOL / TOBI OETIKER


Cacti - Poller Runtime - localhost
seconds
6.0
5.0
4.0
3.0
2.0
1.0
0.0
19:50
20:00
20:10
20:20
20:30
20:40
From 2014/02/05 19:41:48 To 2014/02/05 20:41:48
Type: spine
Runtime: Current: 4.27 Average: 4.22 Maximum: 6.07

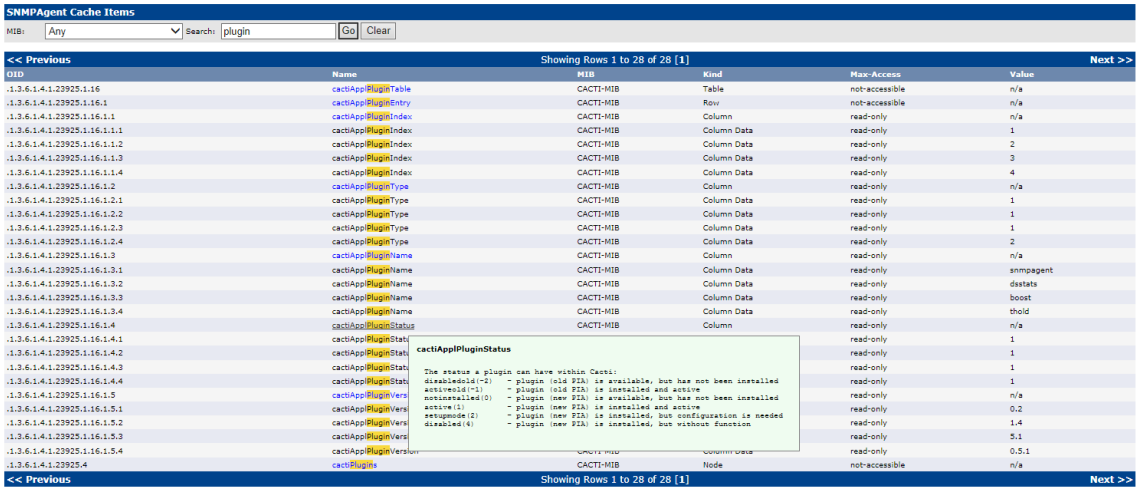

Cacti - Poller Utilization - localhost
percent
100
80
60
40
20
0
19:50
20:00
20:10
20:20
20:30
20:40
From 2014/02/05 19:47:13 To 2014/02/05 20:47:13
Type: spine
Utilization: Current: 1.43 % Average: 1.41 % Maximum: 2.02 %
RRDTOOL / TOBI OETIKER

# 8 Monitoring Cacti with a custom NMS

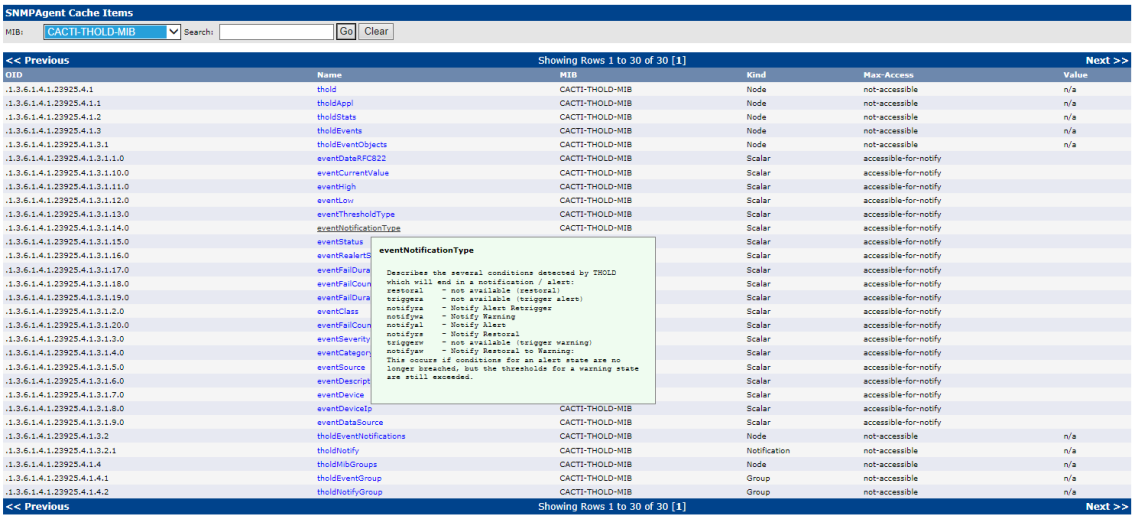
Feel free to contact me directly if you would like to show your examples here.

# 9 SNMPAgent Utilities - Inside the SNMP Cache

With version 0.2 administrators are able to directly take a look at the content of the SNMP Cache using the Cacti graphical user interface. Go to Console->System Utilities and click on “View SNMPAgent Cache”.




This interface allows searching for specific OIDs and object names of all registered MIB files.




No, that is not a fake - Thold already supports the SNMPAgent plugin since SVN revision 2197.

# 10 SNMP Notifications

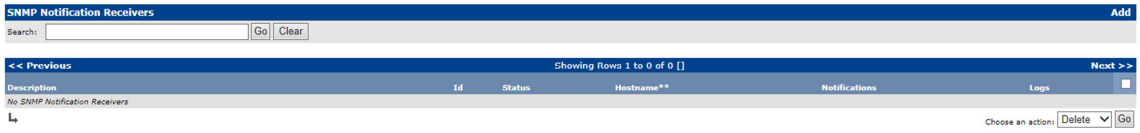
If you are using SNMP notification receivers, that will automatically generate tickets, emails, short messages and so on, and you are still searching for a way how to report events detected by Cacti, then you should take a look at the following section.

## 10.1 Presets

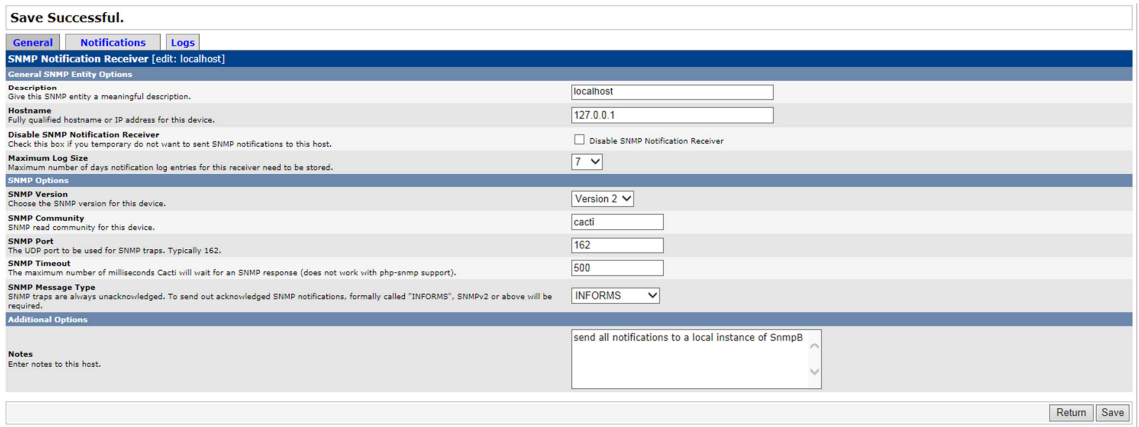
SNMPAgent internally uses the snmptrap command of NET-SNMP to generate notifications. Therefore the path to snmptrap has to be updated under “Configuration”->”Settings”->Tab: “Misc”.

## 10.2 Registration

This plugin adds a new item called “SNMP Notification Receivers” to the Cacti console. If this entry is not visible under “Configuration” then update your user realms first.

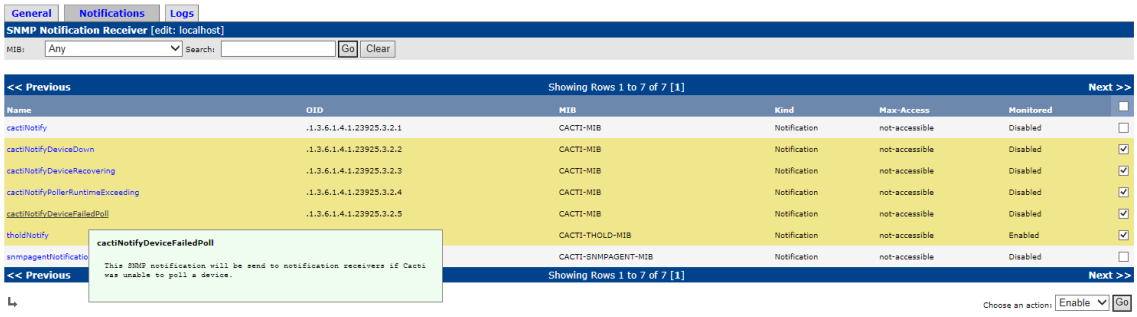



Once opened click on “Add” in the right upper corner register a new receiver and fill out all input field before you click on “create”. Please note that I’ve locally installed the great and free SnmpB tool (GPLv2) before to demonstrate this functionality. SnmpB is available for Windows, Linux and MacOS.

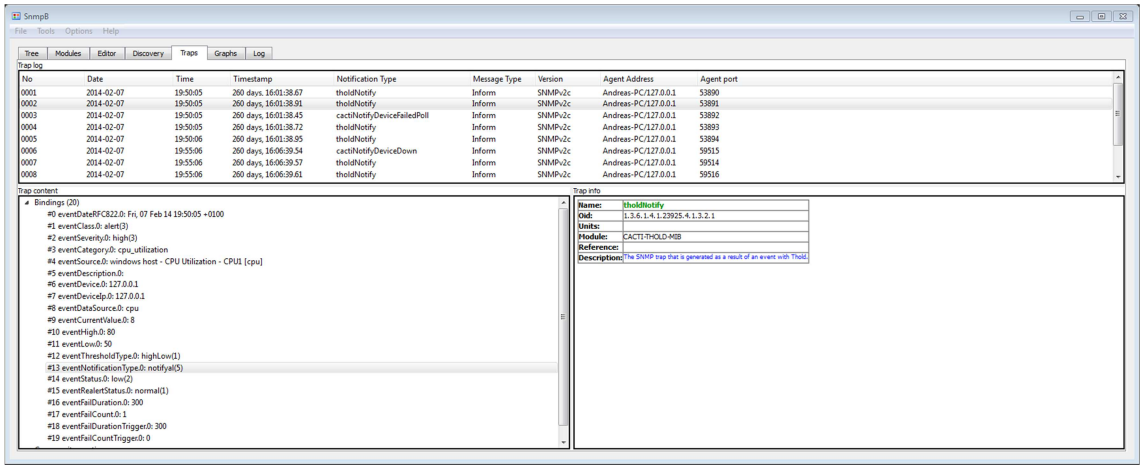



With a click on the second tab we will a get a dynamic list of registered notifications. “Dynamic” means that the list will automatically change if plugins will be installed or uninstalled. But please keep in mind that the SNMPAgent never

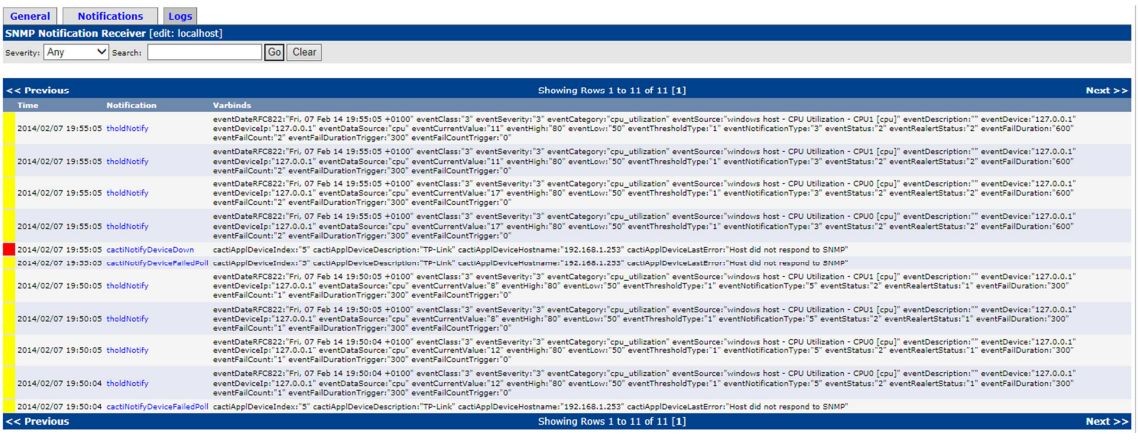
drops monitored notifications on its own. They only become invisible / inactive as long as the related plugin has not been reinstalled again. This means that you can upgrade a plugin without being worried about losing the configuration of related notifications objects.




Once we have enabled some of those notifications the SNMPAgent will automatically start to generate SNMP traps/informs forwarded to that notification receiver.




Do not forget that you have to add all Cacti MIBs to your SNMP receiver. Only having all MIBs being installed your receiver will be able to parse received messages correctly. All notifications being generated can also be reviewed for the predefined period of time under tab “log” or in the global notifications log (see SNMPAgent Utilities).



## 10.3 Debugging

Currently the bad implementation of Net-SNMP for Windows avoids that the output values can be returned correctly to Cacti. At the moment the only way to debug the notifications being generated by SNMPAgent is to increase the Cacti's logging level up to "high":