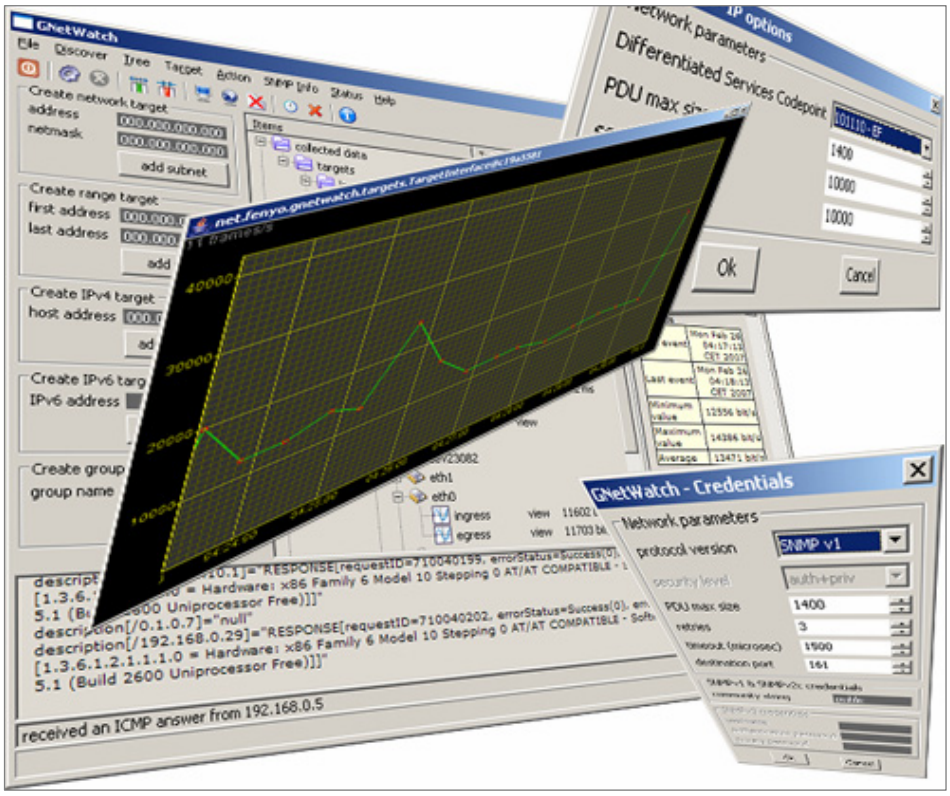
GNetWatch 3.1

USER'S MANUAL

v3.1 – MAY, 26 2008



Copyright 2006, 2007, 2008 Alexandre Fenyo
gnetwatch@fenyo.net - http://fenyo.net
GNU GENERAL PUBLIC LICENSE v2

CONTENTS

FIGURES

TABLES

1. INTRODUCTION

1.1. GNETWATCH AT A GLANCE

GNetWatch is a free open source **Java** application that offers real-time graphical monitoring and analysis of network performance through SNMP and ICMP. To get an instant view of the network state, data are collected, stored, and displayed every few seconds. **Two traffic generation modules** are available. The former can flood UDP packets of any size (**jumbo frames** for instance) and tagged with any DiffServ/ToS flag for QoS and **class of services testing**. The latter can generate a huge quantity of parallel requests to any HTTP(s) server for Web **application load testing**. To automatically discover new hosts, GNetWatch can make use of Ethereal/WireShark and later invoke NMap to get information about the remote systems.

1.2. LICENSE

GNetWatch
Copyright 2006, 2007, 2008 Alexandre Fenyo
gnetwatch@fenyo.net

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users. This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it. (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have. You must make sure that they, too, receive or can get the
source code. And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software. If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary. To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License. The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices
stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in
whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third
parties under the terms of this License.

c) If the modified program normally reads commands interactively
when run, you must cause it, when started running for such
interactive use in the most ordinary way, to print or display an
announcement including an appropriate copyright notice and a
notice that there is no warranty (or else, saying that you provide
a warranty) and that users may redistribute the program under
these conditions, and telling the user how to view a copy of this
License. (Exception: if the Program itself is interactive but
does not normally print such an announcement, your work based on
the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable
source code, which must be distributed under the terms of Sections
1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three
years, to give any third party, for a charge no more than your
cost of physically performing source distribution, a complete
machine-readable copy of the corresponding source code, to be
distributed under the terms of Sections 1 and 2 above on a medium
customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer
to distribute corresponding source code. (This alternative is
allowed only for noncommercial distribution and only if you
received the program in object code or executable form with such
an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it. For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable. However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
form) with the major components (compiler, kernel, and so on) of the
operating system on which the executable runs, unless that component
itself accompanies the executable.

If distribution of executable or object code is made by offering
access to copy from a designated place, then offering equivalent
access to copy the source code from the same place counts as
distribution of the source code, even though third parties are not
compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program
except as expressly provided under this License. Any attempt
otherwise to copy, modify, sublicense or distribute the Program is
void, and will automatically terminate your rights under this License.
However, parties who have received copies, or rights, from you under
this License will not have their licenses terminated so long as such
parties remain in full compliance.

5. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Program or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Program (or any work based on the
Program), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the
original licensor to copy, distribute or modify the Program subject to
these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Program at all. For example, if a patent
license would not permit royalty-free redistribution of the Program by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under
any particular circumstance, the balance of the section is intended to
apply and the section as a whole is intended to apply in other
circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system, which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Program under this License
may add an explicit geographical distribution limitation excluding
those countries, so that distribution is permitted only in or among
countries not thus excluded. In such case, this License incorporates
the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of
this License, you may choose any version ever published by the Free Software
Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than `show w' and `show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.

The directory `3rdParty` in each bundle distribution contains third-party licences.

1.3. ADVANCED PROTOCOLS SUPPORT

Note that **IPv6** and **SNMPv3** are fully supported by GNetWatch.

GNetWatch is being developed in Java with Eclipse, following the Unified Software Development Process.

1.4. GRAPHIC PERFORMANCES

Graphic performances have been particularly optimized: GNetWatch is using both SWT and AWT simultaneously, the main GUI is drawn with SWT in order to get direct access to the underlying windowing system and animated graphs are generated using Java2D over AWT.

1.5. SUPPORTED PLATFORMS

GNetWatch depends on class libraries available in Sun's JRE 5 specification and on the following external packages:

- hibernate-3.1
- commons-collections-3.2
- commons-configuration-1.3
- commons-io-1.2
- commons-jxpath-1.2
- commons-lang-2.3
- commons-logging-1.1
- dom4j-1.6.1
- log4j-1.2.13
- hsqldb
- snmp4j
- swt

These packages are independent from the target platform, except for SWT, the Standard Widget Toolkit (http://www.eclipse.org/swt) that comes with native libraries compiled specifically for your target platform. This is the main reason why different distributions, named bundles, are available.

Moreover, GNetWatch makes use of the browser widget included in SWT. This widget depends on external packages like Mozilla or Internet Explorer, on some operating systems.

At least, GNetWatch should run on the platforms on which the SWT browser widget runs:

- Windows (with Internet Explorer 5 and above)
- Mac (Panther OS X 10.3 and above – Safari-based)
- Linux GTK and Linux Motif: RedHat Enterprise Linux 3 and SuSE 9 should be fully compliant, and some other Linux distributions may require a supported version of Mozilla to be installed.

See the SWT FAQ (http://www.eclipse.org/swt/faq.php) to get more information about the supported platforms for SWT and its browser widget.

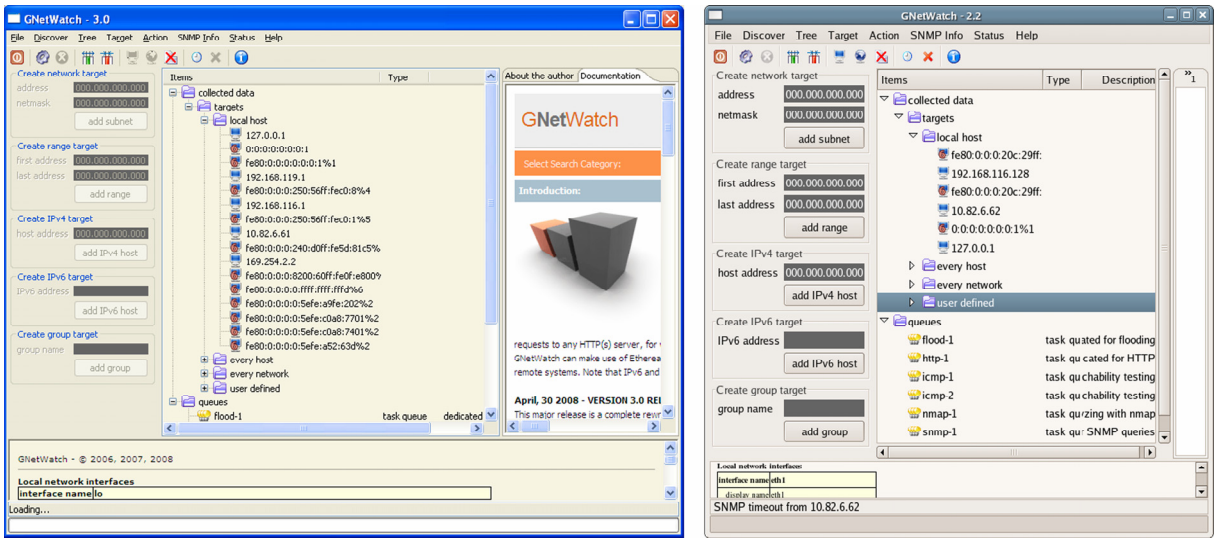


Figure 1 : GNetWatch on MS-Windows and GNetWatch on Linux

2. INSTALLATION

2.1. DIFFERENT WAYS TO INSTALL GNETWATCH

Here are several ways to install GNetWatch:

- download a **bundle** (installation already packaged) for Linux or MS-Windows
- download the GNetWatch JAR file and public domain dependant packages
- download the GNetWatch java sources archive and compile it

2.2. USING A BUNDLE FOR MS-WINDOWS

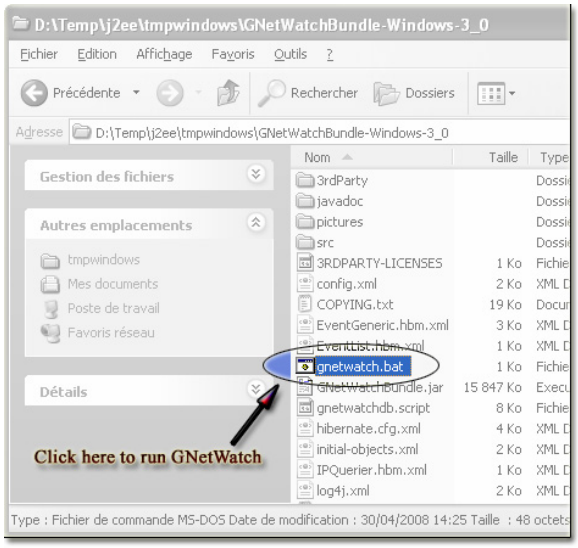


Figure 2 : running GNetWatch under MS-Windows

Just follow these steps:

1. download and install a Java SE Runtime Environment (JRE) compliant with JRE 5 specifications at least (available for instance from http://java.sun.com)
2. download and extract the GNetWatch MS-Windows bundle
3. double-click on `gnetwatch.bat` (see figure "Figure 2 : running GNetWatch under MS-Windows").

Until GNetWatch 2.2, you could run the program by double-clicking on the `GNetWatchBundle.jar` file. Starting with GNetWatch 3.0, **avoid** double-clicking on `GNetWatchBundle.jar` because it may launch GNetWatch without setting a sufficient maximum heap size, so the program may crash later if using an internal database. On the contrary, `gnetwatch.bat` correctly sets the appropriate memory options.

2.3. USING A BUNDLE FOR LINUX

Just follow these steps:

1. download and install a Java SE Runtime Environment (JRE) compliant with JRE 5 specifications at least (available for instance from http://java.sun.com)
2. download and extract the GNetWatch Linux bundle
3. set and export the `MOZILLA_FIVE_HOME` environment variable (see your Mozilla or Firefox documentation)

4. include the GNetWatch installation directory and the `MOZILLA_FIVE_HOME` in the `LD_LIBRARY_PATH` environment variable

5. invoke the Java VM

The figure “Figure 3 : running GNetWatch under Linux” gives an example of running under Linux.

```
user@host% tar zxf GNetWatch-LinuxBundle-version.tar.gz
user@host% cd GNetWatch-LinuxBundle-version
user@host% MOZILLA_FIVE_HOME=/usr/lib/mozilla-1.7.12
user@host% export MOZILLA_FIVE_HOME
user@host% LD_LIBRARY_PATH="$LD_LIBRARY_PATH:$MOZILLA_FIVE_HOME:."
user@host% export LD_LIBRARY_PATH
user@host% java -XX:+AggressiveHeap -jar GNetWatchBundle.jar
```

Figure 3 : running GNetWatch under Linux

2.4. RUNNING EXTERNAL PROGRAMS

2.4.1. STANDARD PROGRAMS

GNetWatch can call NMap and Ethereal/Wireshark:

- Ethereal/Wireshark to track new hosts
- NMap to display information about remote hosts

Before running GNetWatch, you must include `nmap` (or `nmap.exe` on MS-Windows) and `tethereal` (or `tethereal.exe` on MS-Windows) in the PATH. If you installed Wireshark instead of Ethereal, you must rename or copy `tshark` (or `tshark.exe` on MS-Windows) to `tethereal` (or to `tethereal.exe` on MS-Windows).

2.4.2. GENERIC PROGRAMS

GNetWatch can also connect to external probes to monitor targets. This is done by running external programs or tracking changes to file contents. This feature is described later in this document.

3. CONFIGURATION

3.1. MAIN CONFIGURATION FILE: CONFIG.XML

The main configuration file is named `config.xml` and is located in the GNetWatch installation base directory.

This file contains Java configuration properties following this XML format:

```
<entry key="CONFIGURATION_ENTRY">ENTRY_VALUE</entry>
```

Here are the available configuration entries:

- **`net.fenyo.log4j`**

  This entry defines the name of the logging engine configuration file.

- **`net.fenyo.initialobjects`**

  This entry defines the name of a file that contains definitions of user-defined GNetWatch objects that will be built just after GNetWatch start-up.

- **`net.fenyo.genericconffile`**

  This entry defines the name of a file that contains definitions of generic templates used to connect to external probes.

- **`net.fenyo.queues.count.icmp`**

  This entry defines the number of queues that can simultaneously handle actions of type ActionPing.

- **`net.fenyo.queues.count.snmp`**

  This entry defines the number of queues that can simultaneously handle actions of type ActionSNMP.

- **`net.fenyo.queues.count.flood`**

  This entry defines the number of queues that can simultaneously handle actions of type ActionFlood.

- **`net.fenyo.queues.count.http`**

  This entry defines the number of queues that can simultaneously handle actions of type ActionHTTP.

- **`net.fenyo.queues.count.nmap`**

  This entry defines the number of queues that can simultaneously handle actions of type ActionNmap.

- **net.fenyo.queues.count.process**

  This entry defines the number of queues that can simultaneously handle actions of type ActionGenericProcess.

- **net.fenyo.queues.count.source**

  This entry defines the number of queues that can simultaneously handle actions of type ActionGenericSrc.

- **net.fenyo.nmap.timeout**

  This entry defines the timeout in milliseconds used when waiting for Nmap sub-processes.

- **net.fenyo.ipaddresseditor.insertonkeypressed**

  This boolean entry defines the type of Key events that are used by the IPv4 address editor of GNetWatch. When running GNetWatch under SWT+GTK+X11, set it to false. Otherwise, set it to true.

- **net.fenyo.ping.countparameter**

  Since JRE is not efficient with raw sockets, GNetWatch starts external PING processes. This entry defines the option used by GNetWatch to make PING stop after having sent a specific number of ICMP packets.

- **net.fenyo.ping.regex**

  This entry defines a regular expression used by GNetWatch to parse the output of PING processes.

- **net.fenyo.language** and **net.fenyo.country**

  these entries are used to set the desired LOCALE.

  Here is a configuration file example for use with an English MS-Windows system:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>General configuration properties for MS-Windows</comment>
  <entry key="net.fenyo.log4j">log4j.xml</entry>
  <entry key="net.fenyo.initialobjects">initial-objects.xml</entry>
  <entry key="net.fenyo.genericconffile">generic.xml</entry>
  <entry key="net.fenyo.queues.count.icmp">2</entry>
  <entry key="net.fenyo.queues.count.snmp">2</entry>
  <entry key="net.fenyo.queues.count.flood">1</entry>
  <entry key="net.fenyo.queues.count.http">1</entry>
  <entry key="net.fenyo.queues.count.nmap">1</entry>
  <entry key="net.fenyo.queues.count.process">1</entry>
  <entry key="net.fenyo.queues.count.source">1</entry>
  <entry key="net.fenyo.nmap.timeout">120000</entry>
  <entry key="net.fenyo.ipaddresseditor.insertonkeypressed">true</entry>
  <entry key="net.fenyo.ping.countparameter">-n</entry>
  <entry key="net.fenyo.ping.regex">(.|\r|\n)*:.*?([0-9]+)[^0-9]*ms(.|\r|\n)*</entry>
  <entry key="net.fenyo.language"></entry>
  <entry key="net.fenyo.country"></entry>
</properties>
```

  Here is a configuration file example for use with a French Linux system:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>General configuration properties for Linux</comment>
  <entry key="net.fenyo.log4j">log4j.xml</entry>
  <entry key="net.fenyo.initialobjects">initial-objects.xml</entry>
  <entry key="net.fenyo.genericconffile">generic.xml</entry>
  <entry key="net.fenyo.queues.count.icmp">2</entry>
  <entry key="net.fenyo.queues.count.snmp">2</entry>
  <entry key="net.fenyo.queues.count.flood">1</entry>
  <entry key="net.fenyo.queues.count.http">1</entry>
  <entry key="net.fenyo.queues.count.nmap">1</entry>
  <entry key="net.fenyo.queues.count.process">1</entry>
  <entry key="net.fenyo.queues.count.source">1</entry>
  <entry key="net.fenyo.nmap.timeout">120000</entry>
  <entry key="net.fenyo.ipaddresseditor.insertonkeypressed">false</entry>
  <entry key="net.fenyo.ping.countparameter">-c</entry>
  <entry key="net.fenyo.ping.regex">(.|\r|\n)*:.*?([0-9]+)\.[0-9]*[^0-9]*ms(.|\r|\n)*</entry>
  <entry key="net.fenyo.language">fr</entry>
  <entry key="net.fenyo.country">FR</entry>
</properties>
```

The following picture shows the same menu displayed under different localization options.

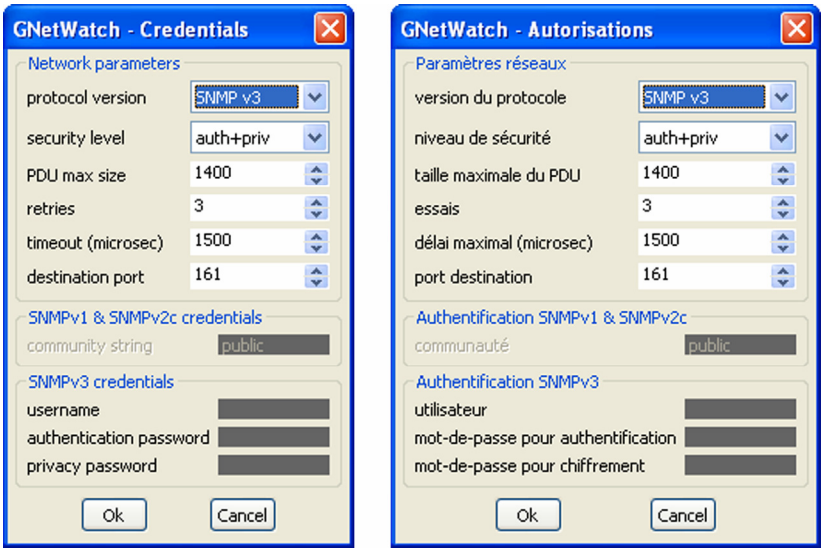


Figure 4 : same menu with different locale

3.2. CREATING OBJECTS AT START-UP

3.2.1. DEFINING OBJECTS

The file defined in the `net.fenyo.initialobjects` configuration entry may be used to make GNetWatch automatically build objects at start-up. Among the many objects GNetWatch lets you build through the GUI, only a few target types with limited features can be built through this file:

- **groups**: trees of groups can be defined, meaning parents of a group are groups.
- **IPv4 targets**: parents of IPv4 targets **must** be groups. IPv4 targets can define specific SNMP properties.
- **IPv6 targets**: parents of IPv6 targets **must** be groups. IPv6 targets can **not** define any specific SNMP properties (you need to use the GUI to define those properties).

- **IPv4 subnets**: parents of IPv4 subnets **must** be groups.
- **IPv4 ranges**: parents of IPv4 ranges **must** be groups.

3.2.2. EXAMPLE

Here is an example of such a configuration file:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<gnetwatch>
  <objects>
    <target targetType="group">
      <name>fenyo.net</name>
    </target>
    <target targetType="group">
      <name>sourceforge.net</name>
    </target>
    <target targetType="group">
      <name>gnetwatch.sourceforge.net</name>
      <parent parentType="group">sourceforge.net</parent>
    </target>
    <target targetType="ipv4">
      <address>66.35.250.209</address>
      <parent parentType="group">gnetwatch.sourceforge.net</parent>
    </target>
    <target targetType="group">
      <name>gw.fenyo.net</name>
      <parent parentType="group">fenyo.net</parent>
    </target>
    <target targetType="ipv4">
      <address>192.168.0.5</address>
      <parent parentType="group">gw.fenyo.net</parent>
    </target>
    <target targetType="ipv4">
      <address>88.170.235.198</address>
      <parent parentType="group">gw.fenyo.net</parent>
      <snmp><version>v2c</version><community>private</community></snmp>
    </target>
    <target targetType="group">
      <name>sandbox</name>
      <parent parentType="group">gw.fenyo.net</parent>
      <parent parentType="group">www.sourceforge.net</parent>
    <target targetType="ipv4">
      <address>127.0.0.1</address>
      <parent parentType="group">sandbox</parent>
    </target>
    </target>
  </objects>
</gnetwatch>
```

The tree structure corresponding to this file looks like:

```
- fenyo.net
  - gw.fenyo.net
    - 192.168.0.5
    - 88.170.235.198
    - sandbox
      - 127.0.0.1
- sourceforge.net
  - gnetwatch.sourceforge.net
    - 66.35.250.209
    - sandbox (same instance as the previous one)
      - 127.0.0.1
```

3.2.3. DEFINING SNMP PARAMETERS

Here is an example of definition of the SNMPv1 properties:

```
<snmp>
  <version>v1</version>
  <community>public</community>
  <pdu-max-size>1400</pdu-max-size>
  <port>161</port>
  <retries>3</retries>
</snmp>
```

Here is an example of definition of the SNMPv2c properties:

```
<snmp>
  <version>v2c</version>
  <community>public</community>
  <pdu-max-size>1400</pdu-max-size>
  <port>161</port>
  <retries>3</retries>
</snmp>
```

Here is an example of definition of the SNMPv3 properties:

```
<snmp>
  <version>v3</version>
  <security>AUTH_PRIV</security>
  <!-- use NOAUTH_NOPRIV to get no authentication nor privacy,
           AUTH_NOPRIV to get authentication but no privacy
           and AUTH_PRIV to get both authentication and privacy -->
  <password-auth>my_secret_for_authentication</password-auth>
  <password-priv>my_secret_for_privacy</password-priv>
  <pdu-max-size>1400</pdu-max-size>
  <port>161</port>
  <retries>3</retries>
</snmp>
```

3.3. USING AN EXTERNAL DATABASE

Since version 3.0, GNetWatch maintains its configuration and the collected data in a JDBC compliant database. It is shipped with an embedded one: HSQLDB, but you can configure an external database.

If you plan to collect a big amount of data or track a lot of targets, you may encounter memory and performance limitations using the internal HSQLB database. In that case, migrate to an external database. You can choose any type of JDBC database supported by Hibernate. The section “Database connection settings” in the file `hibernate.cfg.xml` defines the JDBC driver and connection path to the database.

For instance, to use an external HSQLDB process, apply the following steps:

1. run an external HSQLDB database process with the following command:

   ```
   java -XX:+AggressiveHeap -cp GNetWatchBundle.jar org.hsqldb.Server
   ```

2. change the Database connection settings in the file `hibernate.cfg.xml` like this:

   ```
   <!-- Database connection settings -->
   <property name="connection.driver_class">org.hsqldb.jdbcDriver</property>
   <property name="connection.url">jdbc:hsqldb:hsql://127.0.0.1</property>
   ```

3. Finally, run GNetWatch.

3.4. CONNECTING TO EXTERNAL PROBES

3.4.1. DEFINING TEMPLATES

To connect to external probes, you can choose between two ways:

- *external processes*: GNetWatch can run an external process and parse its numeric output as a probe value;
- *external sources*: GNetWatch can also loop scanning a file for new lines containing numeric output.

The file defined in the `net.fenyo.genericconfigfile` configuration entry is used to easily define standard tasks using external processes of files. The one shipped with GNetWatch contains templates letting you probe remote CPU load and remote free physical memory.

Each template entry contains the following parameters:

- *name*: the template name;
- *title*: the name of the action associated with this template entry;
- *unit*: the unit associated with the numeric value returned by this probe.

Templates of type *process* also contain the following parameters:

- *cmdline*: the command line (process name and arguments);
- *workdir*: the current working directory set by GNetWatch before spawning the process.

Templates of type *source* also contain the following parameter:

- *filename*: path of the file used to collect probed numeric values.

3.4.2. EXAMPLE

Here is an example of such a configuration file:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<gnetwatch>
  <generic>

    <template type="source">
      <name>external file</name>
      <title>external file</title>
      <filename>FILENAME</filename>
      <workdir></workdir>
      <unit></unit>
    </template>

    <template type="process">
      <name>external process</name>
      <title>external process</title>
      <cmdline>COMMAND_LINE</cmdline>
      <workdir></workdir>
      <unit></unit>
```

```
    </template>

    <template type="process">
      <name>MS-Windows target: processor load</name>
      <title>processor load</title>
      <cmdline>generic\procload.bat CPU0 remote_host user@domain password</cmdline>
      <unit>%</unit>
    </template>

    <template type="process">
      <name>MS-Windows target: free physical memory</name>
      <title>free physical memory</title>
      <cmdline>generic\freephysmem.bat remote_host user@domain password</cmdline>
      <unit>kb</unit>
    </template>

  </generic>
</gnetwatch>
```

4. CONCEPTS

4.1. GENERAL CONCEPTS

The major GNetWatch operations are based on the following 5 concepts: **targets**, **actions**, **queues**, **events** and **views**.

4.1.1. TARGETS

A target is a persistent object on which you can perform operations. Some targets can also be a container for other targets. For instance, a TargetIPv4Subnet instance is a target container that holds sub-targets such as TargetIPv4 instances. Note that we name “instance” any target of a specific type.

4.1.2. ACTIONS

An action is an object attached to a target and that can perform an operation on it. Over-loading an HTTP server represented by a TargetIPv4 instance is a typical action. There are currently seven action types:

- ActionPing: used to ping remote hosts;
- ActionSNMP: used to collect throughput of remote network interfaces;
- ActionFlood: used to flood UDP packets to remote targets;
- ActionHTTP: used to load a web or ftp server;
- ActionNmap: used to invoke NMap on a remote target;
- ActionGenericProcess: used to probe a remote value by spawning a sub-process;
- ActionGenericSrc: used to probe a remote value by scanning a file.

4.1.3. QUEUES

To perform its operation, an action must be activated by a circular queue. A queue holds several actions, and activates them one after the other. When every target has been activated, this process loops. Each queue is processed by a dedicated thread so that different queues can perform parallel actions. Each type of action is processed by one or many dedicated queues.

4.1.4. EVENTS

Ann event maintains a time stamped piece of information relative to a specific target. Events are created by actions when they perform their operations. The round-trip-time to contact a host is a typical event.

4.1.5. VIEWS

A view can display many events associated to a single target in a human-readable form. For instance, a view can generate a real-time graph that moves when new events are created by a running action.

4.2. MAIN GUI

The figure “Figure 5 : main window” highlights the nine main areas of the GUI.

1. **Targets area**: this area displays a tree that contains the targets, the actions that perform operations on them and the views used to display events attached to theses targets. The elements inside this tree are named **tree items**.

2. **Queues area**: this area displays a tree containing each queue. Since several queues of a same type can be created for parallel processing, an index number is appended to the queue type. For instance, the figure shows two queues named *icmp-1* and *icmp-2* meaning that two PING can be performed at the same moment.

3. **Target creation area**: on this area, you will find the targets that can be manually created. when pressing the “add” button, the target is inserted as a child of the current (i.e. selected) target in the Targets area.

4. **Menus area**: the eight menus are used to perform most operations, like managing the tree items, adding or removing targets, views and actions, getting specific information or status, etc.

5. **Toolbar area**: this area contains shortcuts for some menu entries.

6. **Panel area**: this area contains two persistent panels (the *About the author* and the *Documentation* panels) and transient panels created when double-clicking on a view. Each transient panel displays statistics about the events associated with a view. To update a view panel, no need to close and re-open it, just double-click on the view in the tree item.

7. **Status bar**: the status bar displays the current operation status. It is updated by actions.

8. **Progress bar**: if a queue is selected in the queues area, the progress bar displays a bar the progresses each time an action in this queue is finished. The bar is fully expanded when every action is done. If the queue is empty, the bar is hidden.

9. **Information panel**: this panel contains textual information that is added by some menu operations (ex.: the “Get system description” submenu entry adds a description dealing with the currently selected target at the end of the information panel).

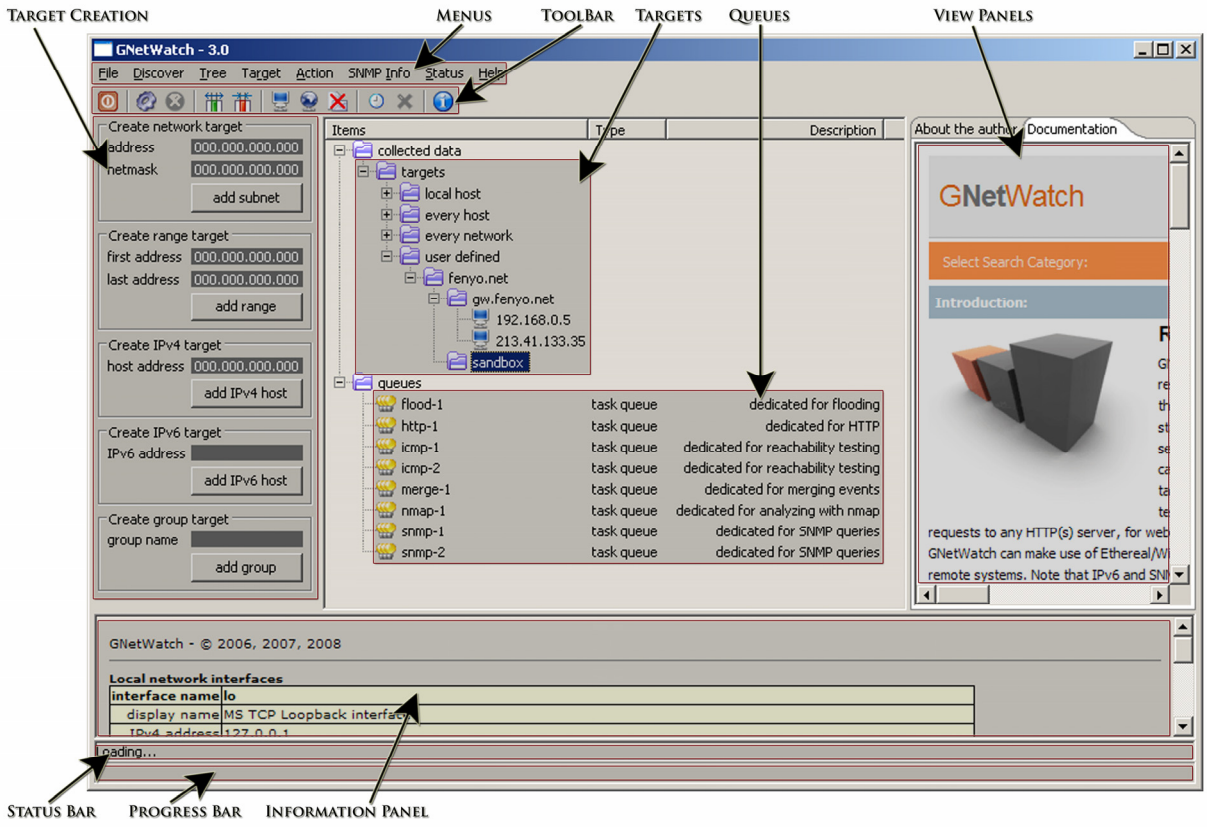


Figure 5 : main window

4.3. GRAPH WINDOW

Each view is associated with an information panel. Views relative to numeric events are also associated with a graph window. These windows can display the numeric values associated with events in two ways:

- **Automatic mode**: the horizontal and vertical scales are automatically adjusted and the window is continuously scrolled.

- **Manual mode**: the user can drag the graph with the mouse and he can also adjust the horizontal and vertical scales manually. This mode appeared in version 3.0.

The figure “Figure 6 : graph window” highlights the main elements of such a window :

1. **Target name** and **view name**: these elements are displayed in the window title.

2. **Scale mode**: the user can switch from the initial automatic scale mode to the manual scale mode by selecting the window and pressing any key or mouse button. To come back in the automatic scale mode, the user must select the window and press the ‘a’ key.

3. **Key bindings**: the top line of the window displays the major key bindings.

4. **Mode**: the current mode is displayed on the left.

5. **Events**: event values are displayed in red and a green line connects them.

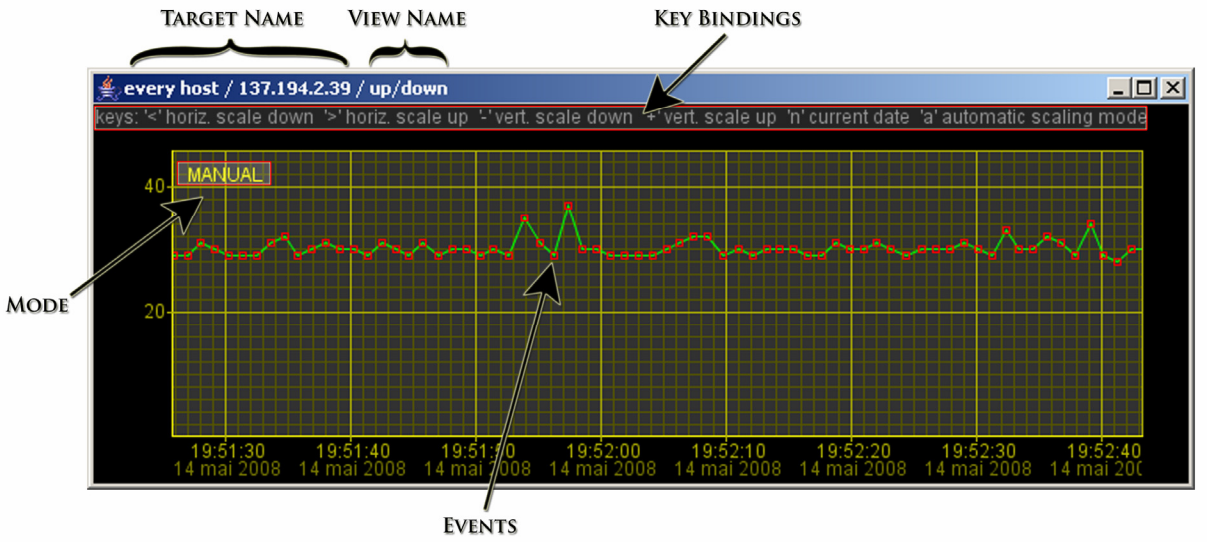


Figure 6 : graph window

5. STEP-BY-STEP OPERATIONS

5.1. INTRODUCTION

This section enumerates the steps you need to complete to achieve the following main operations:

- compute the round-trip-time to reach a host
- explore throughput of remote interfaces using SNMP
- invoke NMap on a remote host
- flood a target with UDP
- load a web server with parallel connections
- discover hosts with Ethereal/WireShark

5.2. COMPUTE THE ROUND-TRIP-TIME TO REACH A HOST

5.2.1. STARTING OPERATIONS

The ICMP network protocol is used to compute the round-trip-time to reach a host. Since the Java TCP/IP API is particularly not suitable to generate ICMP packets, GNetWatch spawns an external program: ping. The program output is parsed to get the delay and an event is generated to store the result in milliseconds.

The figure named “Figure 7 : round-trip-time step-by-step” shows the different steps needed to monitor the round-trip-time to a host have IP address 10.82.0.100 :

- **1st step: create a group**

  Since a node can have multiple addresses (for instance like a router), we first create a group corresponding to our host: the addresses will be children of this group. This type of management will help us organize easily a target tree containing hundreds of items. So we choose a special target type that can contain sub-targets: the group target. To create a new one, just click on the item “user defined” in the targets area, type the name of the group in the left panel for target creation named “create group target” (here we choose `my router`) and press the “add group” button.

- **2nd step: create a target**

  We now want to create the IP target as a child of the group target, so we select the newly-created group, we type the IP address 10.82.0.100 in the left panel for target creation named “create IPv4 target” and finally press the “add IPv4 host” button.

- **3rd step: add an action**

  To regularly compute a ping on this IP target, we need to insert an action in one of the ICMP circular queues and associate it to the target. To do this, you first need to select the IP target in the targets area. Then, in the Action menu, we select the “Ping target” operation. It will automatically create the action, associate it with the IP target and choose an ICMP queue to insert the action in it. You will then see two new children of the IP target:

- *the new action*: the current RTT value is updated on the same line of the action name (in the description column), just after each ICMP packet is received;

- *a new view named "up/down"*: this view appears just after the first event is created by the action and will stay there as long as some ICMP events associated with this target will be stored in the database. Since this view is associated with these events, removing the view would drop the events.

- **4th step: browse the view**

  To browse the events, we double-click on the view. It creates two objects:

  - *a new information pane*l: this panel contains statistics about the events associated with this view. To update those statistics, you need to double-click on the view again. If you removed this panel, just double-click on the view to create it again;

  - *a new graph window*: this graph window displays the events associated with this view. It is automatically updated when new events are created. Moreover, when in automatic mode, it is automatically scaled and scrolled. If you close this window, just double-click on the view to make it appear again.

5.2.2. UNDERSTANDING THE TARGET TREE

We created only three items in the target area: an IP target, an ICMP action and a PING view. But you will see many more items on the figure "Figure 8 : automatic target creation" for two reasons:

1. **Some targets are automatically created.**

   Each IP address of the node on which you run GNetWatch is automatically added under "local host". You will then probably find IP targets associated to your loopback interface: an IPv4 target named "127.0.0.1" and IPv6 target named "::1". You will also find IP targets associated to you network interfaces. With our example, we got an IPv4 target named "10.82.6.61" and a link-local IPv6 target named "fe80:0:0:0:240:d0ff:fe5d:81c5%7".

   Each time an IPv4 target is added, a corresponding network target is added under "every network". For this reason, we got two network targets: "127.0.0.0/255.0.0.0" and "10.0.0.0/255.0.0.0".

2. **Targets can have many parents: a target and its descendant tree of children are displayed under each of its parents.**

   So, when we added the IPv4 target named "10.82.6.61" under the node "my router", the IPv4 target got three parents: the group target named "my router" (because it was selected when we pressed "add IPv4 host"), the group target named "every host" (because every IP target is a child of this group target), the network target named "10.0.0.0/255.0.0.0" because this is the network associated with IP address 10.82.6.61. For the same reason, we will find many instances of the IP targets associated with loopback and other network interfaces.

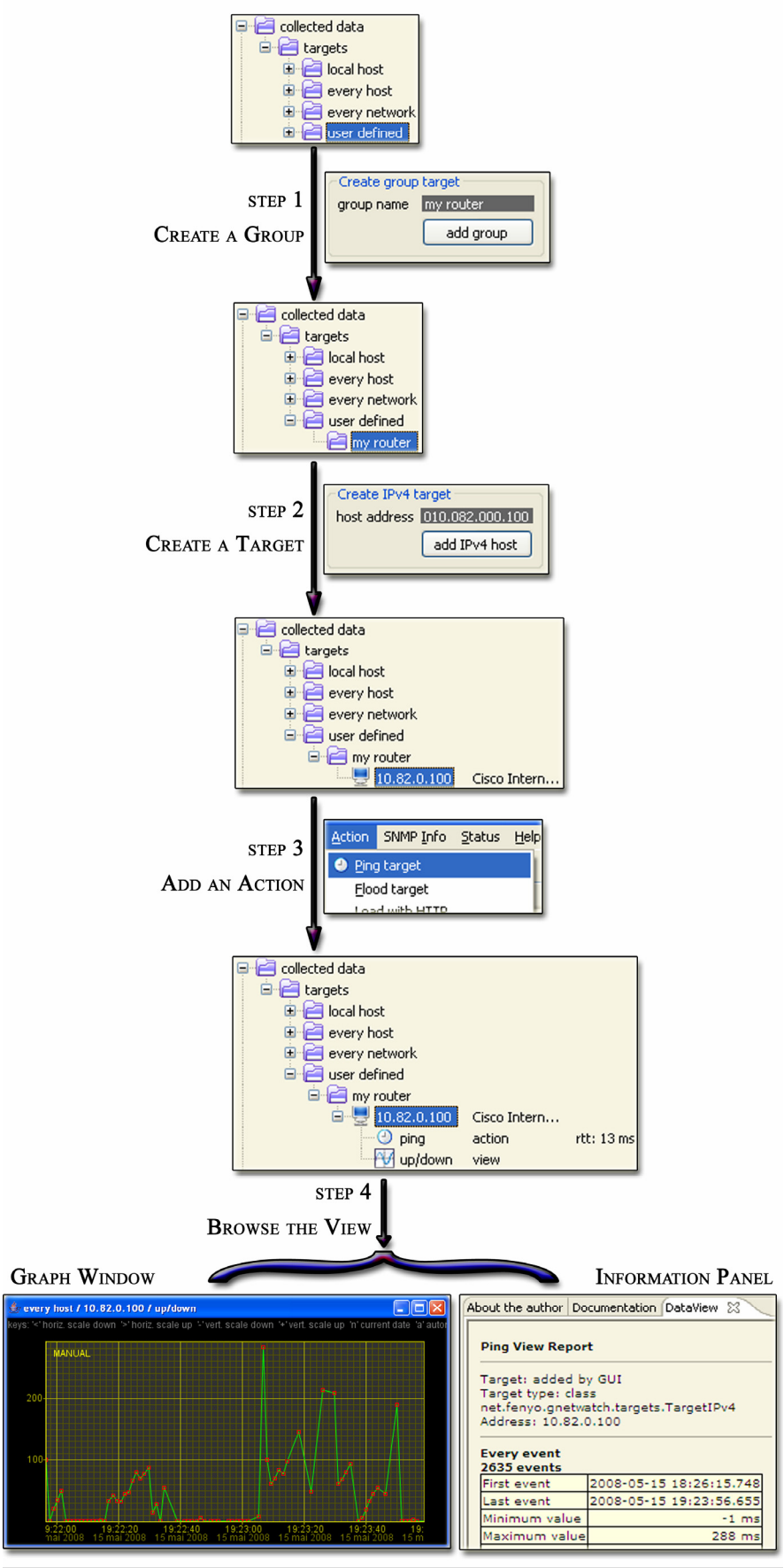


Figure 7 : round-trip-time step-by-step

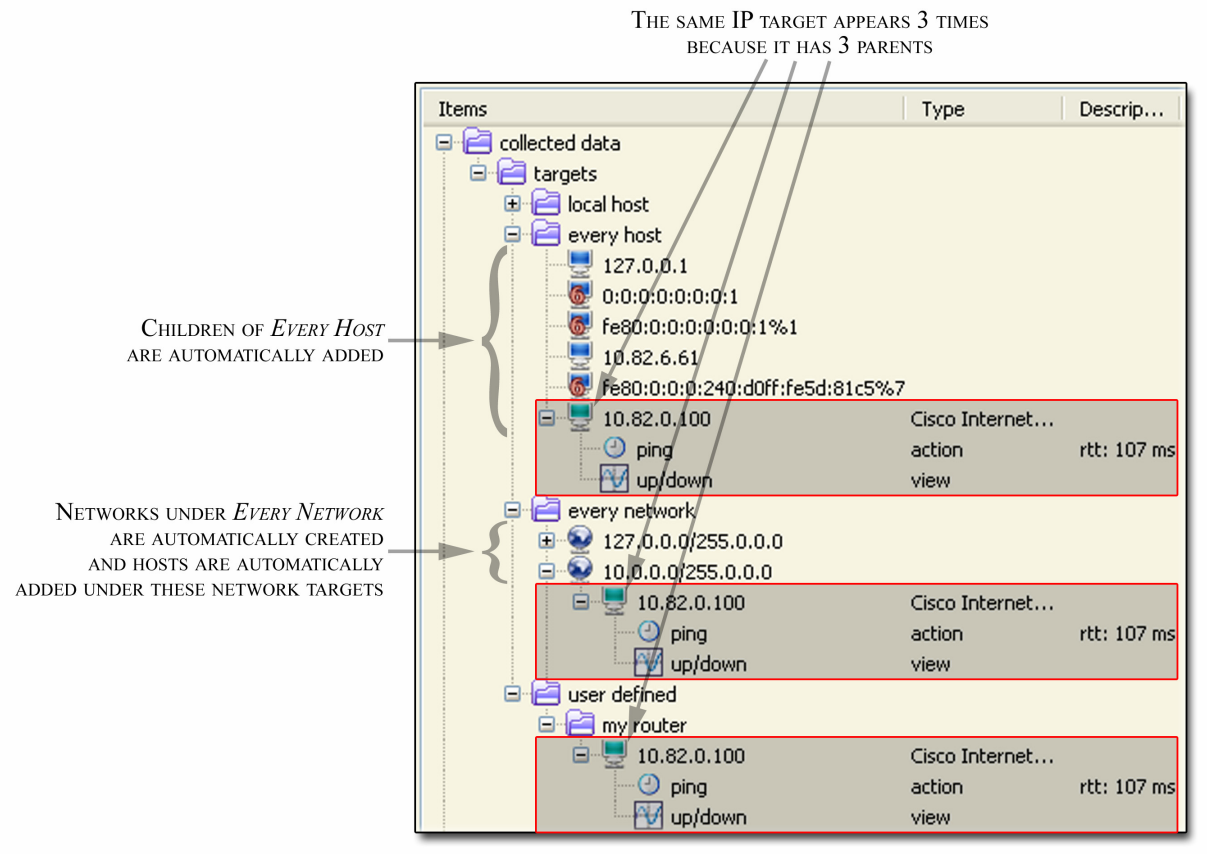


Figure 8 : automatic target creation

5.2.3. CANCELLING OPERATIONS

To cancel the operations we previously did, we need to delete actions, delete views and associated events, and finally remove targets. There are many ways to perform this:

- **first way: one thing at a time**

  We first remove the action: click on the action to select it and use the menu entry “Drop Action” in the “Action” menu.

  We can now remove the view: click on the action to select it and use the menu entry “Drop View” in the “Action” menu. This removes the view and associated events. If we had not previously dropped the action, a new view would rapidly be created again because of the new events instantiated by the action.

  Then we can remove the IP target: click on the IPv4 target item under “my router” to select it and use the menu entry “Detach Element” in the “Target” menu. At this point, the IPv4 target is not removed because you only detached it from one of its three parents. The two remaining parents are the group target named “every host” and the network target named “10.0.0.0/255.0.0.0”. So, you now need to detach the IP target from the network target: click on the IPv4 target item under the network target and repeat the process. The IP target is now definitively removed. It should not be, since it previously had two parents and you detached it from one parent only. The explanation is that just after you detached the IP target from its network parent, there was finally only one parent: the “every host” group. For the same reason a host is automatically added to this group at creation time, it is

automatically deleted when it has not any other parent.

Finally we can remove the group named "my router": click on the group to select it and use the menu entry "Detach Element" in the "Target" menu. Everything is now removed.

- **second way: recursive removal**

  We first remove at once the group named "my router", the "ping" action, the view and the associated events: click on the group named "my router" to select it and use the menu entry "Detach Element" in the "Target" menu. This will recursively detach every target and drop every action, view and associated events.

  We finally need to remove the IP target: for this to be done, we need to detach the IP target from the network target named "10.0.0.0/255.0.0.0": click on the IPv4 target item under the network target and use the menu entry "Detach Element" in the "Target" menu.

5.3. EXPLORE THROUGHPUT OF REMOTE INTERFACES USING SNMP

The figure "Figure 9 : explore via SNMP" describes step-by-step the process of using SNMP to explore a host and get throughput of its remote interfaces.

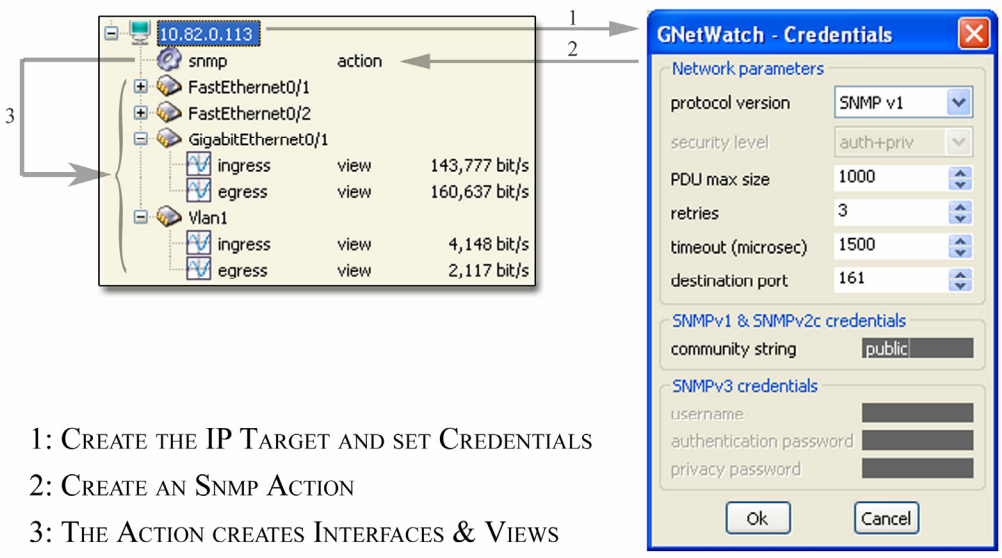


Figure 9 : explore via SNMP

The SNMP credentials are associated with the IP target: you need to set them before adding the SNMP action. Use the menu entry "Set Credentials…" in the "Target" menu to open the credentials dialog. Then choose the appropriate SNMP dialect: SNMPv1, SNMPv2c, SNMPv3 (MD5 + DES).

You will notice a new type of target: interfaces. The first time the SNMP action is invoked by an SNMP queue, it queries the interface list of its parent IP target through SNMP. For each remote interface, it creates an interface target. Then, each time the SNMP action is invoked, it updates the ingress and egress throughput of the interfaces.

With this process, if you remove an interface, it will not be automatically added the next time the action is invoked. You need to drop the action and create it again if you want the interface to appear again.

5.4. Invoke NMap on a remote host

To invoke NMap on a remote host, just create an IP target, select it and choose the menu entry named "Explore via Nmap" in the "Action" menu. It will add an action that will invoke nmap, create a view with the results and finally remove itself from the NMap queue. To display the results, double-click on the view: the information panel will contain the results.

5.5. Flood a target with UDP

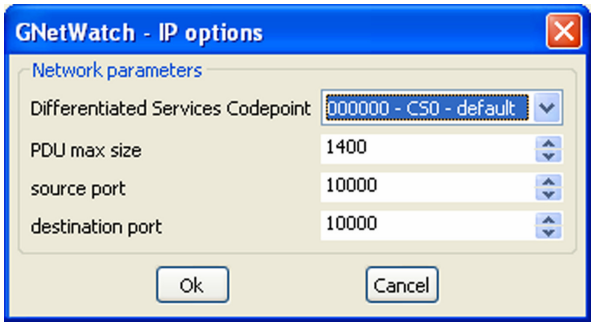


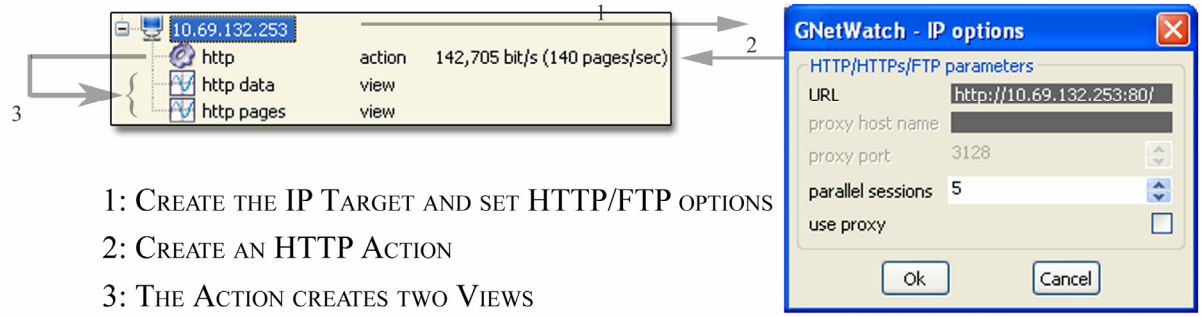
Figure 10 : flooding a target with UDP

GNetWatch can flood packets with any DiffServ codepoint (DSCP/ToS), especially those defined in RFC-2597 (Assured Forwarding Per Hop Behaviour Group) and RFC-2598 (Expedite Forwarding Per Hop Behaviour Group). This is very useful when troubleshooting QoS in IP networks.

Use the menu entry "Set IP options…" in the "Target" menu to open the IP options dialog (see figure "Figure 10 : flooding a target with UDP"). Then select the correct DSCP or ToS, and set the other network parameters (UDP source and destination ports and packet size).

5.6. Load a Web or Ftp server with parallel connections

GNetWatch can load a Web or an FTP server with multiple parallel connections. Use the menu entry "Set HTTP/FTP options…" in the "Target" menu to open the HTTP/FTP dialog and set the URL (http, https and ftp URL schemes are supported), the proxy and the number of parallel sessions. Then add an HTTP action, that will create two views: the former to monitor the generated throughput in bit/s and the latter to monitor the same throughput in pages/sec.



1: Create the IP Target and set HTTP/FTP options

2: Create an HTTP Action

3: The Action creates two Views

Figure 11 : loading a Web/Ftp server with parallel connections

5.7. Discover hosts with Ethereal/WireShark

GNetWatch can automatically discover new hosts by snooping on every network interface at the same time. One Ethereal/WireShark command line process is spawned for each interface

when you use the menu entry named “Start” in the “Discover” menu. Each new IP address discovered is automatically added in the tree under the group named “every host” and the corresponding network is added under “every network”. You may run ping broadcasts to automatically add every host on a local or remote network. To stop every snooping process, use the menu entry named “Stop” in the “Discover” menu.

6. GENERIC PROBES

6.1. INTRODUCTION

Generic probes refer to collecting external numeric values to create events. GNetWatch offers two ways to collect external numeric values:

- *spawning sub-processes*: this is achieved with a generic process action;
- *scanning files*: this is achieved with a generic source action.

6.2. CONFIGURE A TARGET FOR GENERIC PROBE

The only targets that can handle generic probes are group targets. A single group target can only be associated with one generic process action and one generic source action. So, you must create several group targets to collect several probes, even if they scan the same remote host.

To select the parameters of the generic probe, you must select a group target and use the menu entry named “Set generic options…” in the “Target” menu. Then you will see a dialog box named “Generic options”. You can now directly fill the fields as you like, or select a template. If you select a template, the fields will be automatically filled with default values. Then you must customize the fields (hostname, user name, password, etc.).

The figure named “Figure 12 : using a generic probe template” shows the default values that appeared just after having selected a template.

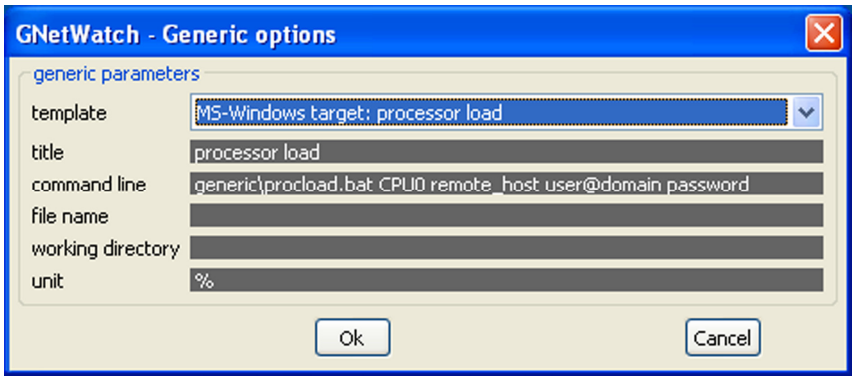


Figure 12 : using a generic probe template

If you want to spawn a sub-process, you do not need to fill the “file name” field. On the contrary, if you want to scan a file, you do not need to fill the “command line” and “working directory” fields.

6.3. SPAWNING SUB-PROCESSES

In order to spawn a sub-process, you need to select the target group and add a “Generic Process” action with the corresponding entry in the “Action” menu. The program name at the beginning of the command line field must be an absolute path or a path relative to the GNetWatch installation base directory. **The program must only return a positive integer.**

6.4. SCANNING FILES

In order to scan a file, you need to select the target group and add a "Generic Source" action with the corresponding entry in the "Action" menu. The file name field must contain an absolute path or a path relative to the GNetWatch installation base directory. **New values can only be positive integers and must be appended to the end of the file.**

6.5. AVAILABLE TEMPLATES

GNetWatch is shipped with 4 generic probe templates:

- *remote MS-Windows processor load*: this template is based on a Visual Basic script that uses WMI (Windows Management Instrumentation) to collect a remote processor load. On the command line, change CPU0 to CPUx to collect the load of CPU number x. You also need to specify the domain/login/password of an account that has sufficient rights to make WMI calls to the remote server.

- *remote MS-Windows free physical memory*: this template is based on a Visual Basic script that uses WMI (Windows Management Instrumentation) to collect the amount of remote free physical memory available. You also need to specify the domain/login/password of an account that has sufficient rights to make WMI calls to the remote server.

- *remote Unix/Linux processor load*: this template is based on a script that uses ssh and uptime to get the remote processor load. Note that you must have configured ssh to accept remote commands without any password (for instance, create a couple of DSA keys on the local host and add the public one to the authorized keys list on the remote host).

- *remote Unix/Linux free physical memory*: this template is based on a script that uses ssh and /proc/meminfo to get the amount of remote physical memory available. Note that you must have configured ssh to accept remote commands without any password (for instance, create a couple of DSA keys on the local host and add the public one to the authorized keys list on the remote host).

Note that the remote MS-Windows templates are only available with the MS-Windows GNetWatch bundle, and that the remote Unix/Linux templates are only available with the Linux GNetWatch bundle.

6.6. EXAMPLE

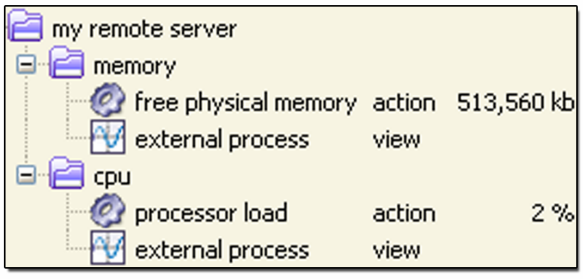


Figure 13 : multiple generic probes for the same target

In this example, we want to collect the free physical memory and the processor load of a single remote system. Since a group target can only be associated with one generic process action, we create the three following group targets:

1. *my remote server*: this group represents the remote system;

2. *memory*: this group is a child of my

remote server, and will be used to get the free physical memory. So, we select this group and use the menu “Set generic options…” to set the free physical memory template. Then we add a generic process action on this group.

3. *cpu*: this group is a child of my remote server, and will be used to get the processor load. So, we select this group and use the menu “Set generic options…” to set the cpu load template. Then we add a generic process action on this group.

The figure named “Figure 13 : multiple generic probes for the same target” shows the resulting tree item.

7. INTERNALS

7.1. UML CLASS DIAGRAM

The UML classes diagram of GNetWatch show the relationships between the different types of Targets, Actions, Queues, Events and Views. The figure “Figure 14 : UML classes diagram” shows the classes diagram of GNetWatch 2.2.

7.2. SOURCES

The sources are available on http://www.gnetwatch.com

7.3. LOCKS

Managing locks is a complex work inside GNetWatch, since many actions may run at the same time in different threads, and since all of them may modify internal data and external DB content.

When possible, GNetWatch wraps thread-unsafe collections with synchronized wrappers, like `Collection.synchronizedMap()` and `Collection.synchronizedList()`.

Otherwise, objects are used to synchronize threads. Here is a list of synchronized objects and classes :

- `GUI.tab_folder (global lock)`
- `GUI.GUI_created (global lock)`
- `GUI.sync_tree (global lock)`
- `synchro (global lock)`
- `ExternalCommand (instances)`
- `CaptureManager (global lock)`
- `CaptureManager.capture_list (global lock)`
- `CaptureManager.listeners (global lock)`
- `Queue (instances)`
- `Queue.actions (instances)`
- `AwtGUI.frame_list (instances)`
- `BasicComponent.sync_value_per_vinterval (instances)`
- `BasicComponent.sync_update (instances)`
- `BasicComponent.events (instances)`
- `VisualElement.initialized (instances)`
- `SNMPQuerier.getSysDescr()::invoked (instance)`
- `registered_components (instance)`

7.4. THREADS

Many thread groups run inside the GNetWatch process, as you can see in table “Table 1 : threads”.

thread	*Origin*
main	`CommandLine.main()`
interrupt	`Background.run()`
GUI	`GUI.run()`

capture-*	`Capture.run()`
repaint	`AwtGUI.run()`
icmp-*	`PingQueue.run()`
snmp-*	`SNMPQueue.run()`
flood-*	`FloodQueue.run()`
http-*	`HTTPQueue.run()`
generic-process-*	`GenericProcessQueue.run()`
generic-source-*	`GenericSrcQueue.run()`
nmap-*	`NmapQueue.run()`
hsqldb	`library thread (HSQLDB)`
DefaultUDPTransportMapping	`library thread (SNMP4J)`
Timer-0	`system thread`

Table 1 : threads

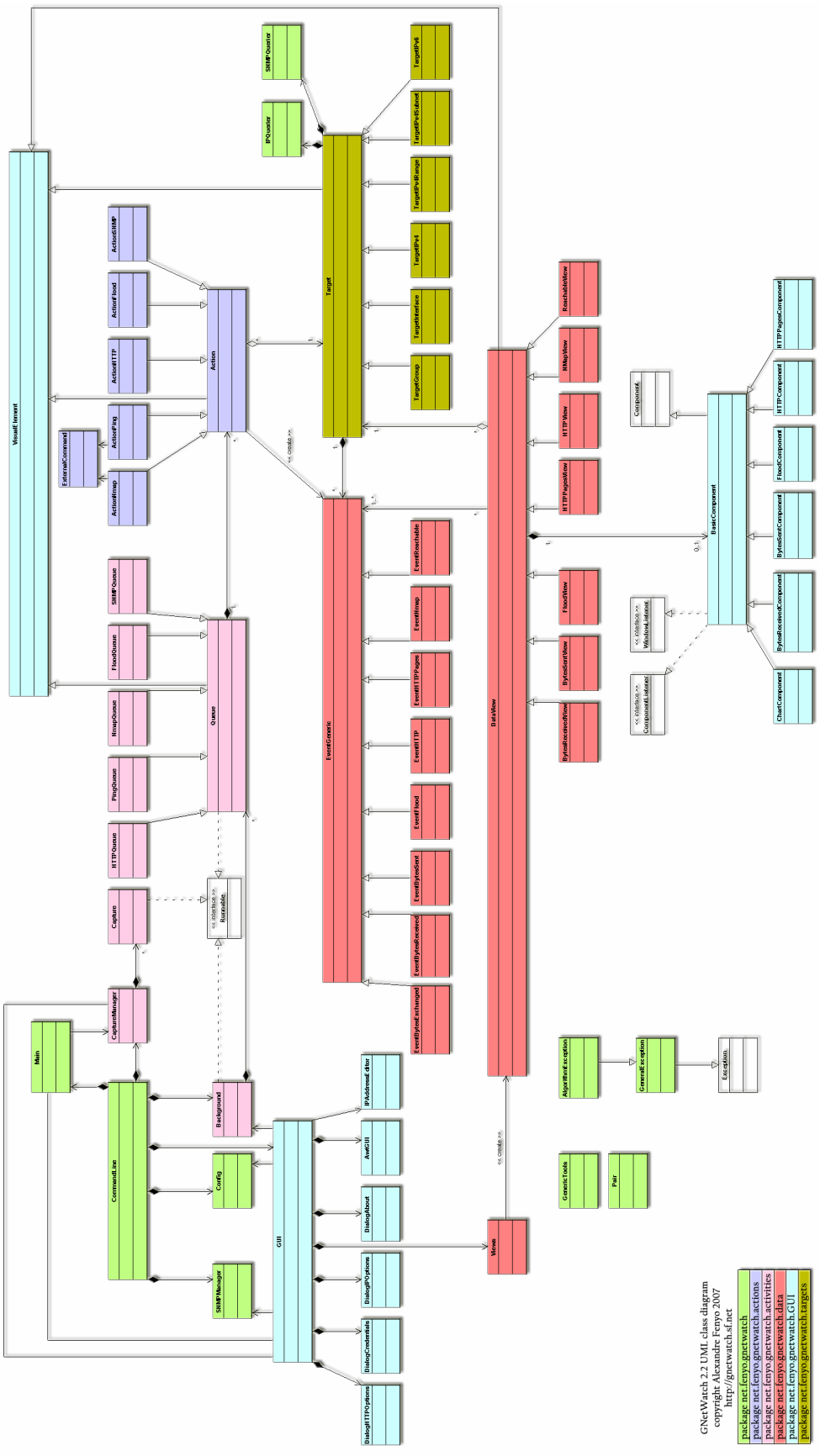


Figure 14 : UML classes diagram