

SellaNMS 2.0

Administrator's Manual

SellaNMS 2.0 Administrator's Manual v1.0
Copyright © Digital Genesis Software 2005, 2006.
All rights reserved.
Last Update 01/16/06

Table of Contents

1	Introduction	7
1.1	Overview	7
1.2	About The Manual	7
1.3	Conventions	7
1.4	Features	8
1.5	How To Help	9
1.5.1	Feedback	9
1.5.2	Donate Hardware	9
1.5.3	Become A Developer	9
2	Support	11
2.1	Mailing List	11
2.2	Forums	11
2.3	Professional Services	11
3	Installation	12
3.1	Requirements	12
3.1.1	Operating System	12
3.1.2	Compiler	12
3.1.3	Net-SNMP	12
3.1.4	MySQL	13
3.1.5	libdbi	13
3.2	Extracting	14
3.3	Building	14
3.4	Installing	15
3.5	Configuring	15
3.5.1	Network Element	15
3.5.2	Configure	15
3.6	Starting	16
3.6.1	Startup	16
3.6.2	Processes	16
3.6.3	Discovery	17
3.7	Shutdown	18
4	Configuration	20
4.1	Network Element	20
4.1.1	ICMP	20
4.1.2	SNMP	20
4.1.2.1	Juniper	21
4.1.2.2	Cisco	21
4.1.2.3	Extreme	21
4.1.2.4	Net-SNMP (Linux/BSD/Solaris)	22
4.1.3	SNMP Trap	22
4.1.3.1	Juniper	22
4.1.3.2	Cisco	23
4.1.3.3	Extreme	23
4.1.3.4	Net-SNMP (Linux/BSD/Solaris)	24

4.1.4	Syslog	24
4.1.4.1	Juniper	24
4.1.4.2	Cisco	24
4.1.4.3	Extreme	24
4.1.4.4	syslogd (Linux/BSD/Solaris)	25
4.2	Syntax	25
4.2.1	Configuration Style	25
4.2.2	Comments	26
4.2.3	Includes	27
4.2.4	Merging	27
4.3	Files/Directories	28
5	Modules	30
5.1.1	Overview	30
5.1.2	sella_nms Configuration	30
5.1.3	Common Configuration	31
5.1.4	System	31
5.1.4.1	Syntax Overview	31
5.1.4.2	system	33
5.1.4.2.1	Syntax	33
5.1.4.2.2	Hierarchy	33
5.1.4.2.3	Description	33
5.1.4.2.4	Options	33
5.1.4.3	storage	34
5.1.4.3.1	Syntax	34
5.1.4.3.2	Hierarchy	34
5.1.4.3.3	Description	34
5.1.4.3.4	Options	34
5.1.4.4	snmp-mib	36
5.1.4.4.1	Syntax	36
5.1.4.4.2	Hierarchy	36
5.1.4.4.3	Description	36
5.1.4.4.4	Options	36
5.1.4.5	default	37
5.1.4.5.1	Syntax	37
5.1.4.5.2	Hierarchy	37
5.1.4.5.3	Description	37
5.1.4.5.4	Options	37
5.1.4.6	Schedule	39
5.1.4.6.1	Syntax	39
5.1.4.6.2	Hierarchy	39
5.1.4.6.3	Description	39
5.1.4.6.4	Options	39
5.1.5	discovery	40
5.1.5.1	Syntax Overview	40
5.1.5.2	discovery	41
5.1.5.2.1	Syntax	41

5.1.5.2.2	Hierarchy	41
5.1.5.2.3	Description	41
5.1.5.2.4	Options	41
5.1.5.3	icmp	42
5.1.5.3.1	Syntax	43
5.1.5.3.2	Hierarchy	44
5.1.5.3.3	Description	44
5.1.5.3.4	Options	44
5.1.5.4	snmp	44
5.1.5.4.1	Syntax	45
5.1.5.4.2	Hierarchy	46
5.1.5.4.3	Description	46
5.1.5.4.4	Options	46
5.1.5.5	topology	47
5.1.5.5.1	Syntax	47
5.1.5.5.2	Hierarchy	47
5.1.5.5.3	Description	47
5.1.5.5.4	Options	48
5.1.6	monitor	49
5.1.6.1	Syntax Overview	49
5.1.6.2	monitor	49
5.1.6.2.1	Syntax	49
5.1.6.2.2	Hierarchy	49
5.1.6.2.3	Description	50
5.1.6.2.4	Options	50
5.1.6.3	icmp	51
5.1.6.3.1	Syntax	51
5.1.6.3.2	Hierarchy	52
5.1.6.3.3	Description	52
5.1.6.3.4	Options	52
5.1.6.3.5	Policies	54
5.1.6.4	snmptrap	56
5.1.6.4.1	Syntax	56
5.1.6.4.2	Hierarchy	57
5.1.6.4.3	Description	57
5.1.6.4.4	Options	57
5.1.6.4.5	Policies	57
5.1.6.5	syslog	59
5.1.6.5.1	Syntax	59
5.1.6.5.2	Hierarchy	59
5.1.6.5.3	Description	60
5.1.6.5.4	Options	60
5.1.6.5.5	Policies	60
5.1.6.6	state	62
5.1.6.6.1	Syntax	62
5.1.6.6.2	Hierarchy	62

5.1.6.6.3	Description	62
5.1.6.6.4	Options	63
5.1.6.6.5	Policies	63
5.1.7	output	63
5.1.7.1	Syntax Overview	63
5.1.7.2	output	64
5.1.7.2.1	Syntax	64
5.1.7.2.2	Hierarchy	64
5.1.7.2.3	Description	64
5.1.7.2.4	Options	64
5.1.7.3	gui	66
5.1.7.3.1	Syntax	66
5.1.7.3.2	Hierarchy	66
5.1.7.3.3	Description	66
5.1.7.3.4	Options	67
5.1.7.3.5	Policies	67
5.1.7.4	email	68
5.1.7.4.1	Syntax	68
5.1.7.4.2	Hierarchy	69
5.1.7.4.3	Description	69
5.1.7.4.4	Options	69
5.1.7.4.5	Policies	70
5.1.7.5	syslog	72
5.1.7.5.1	Syntax	72
5.1.7.5.2	Hierarchy	72
5.1.7.5.3	Description	72
5.1.7.5.4	Options	73
5.1.7.5.5	Policies	73
5.1.8	internal	73
5.1.8.1	Syntax Overview	73
5.1.8.2	internal	74
5.1.8.2.1	Syntax	74
5.1.8.2.2	Hierarchy	74
5.1.8.2.3	Description	74
5.1.8.2.4	Options	74
5.1.8.3	tidy	75
5.1.8.3.1	Syntax	75
5.1.8.3.2	Hierarchy	75
5.1.8.3.3	Description	75
5.1.8.3.4	Options	76
5.1.8.3.5	Policies	76
6	GUI	77
7	Policy Framework	78
7.1	Overview	78
7.2	Syntax Overview	78
7.3	Variable Substitution	78

7.4	Policy Statements	80
7.4.1	term	80
7.4.1.1	from clause	80
7.4.1.2	then clause	81
7.4.1.2.1	set	81
7.4.1.2.2	output	82
7.4.1.2.3	evaluate	83
7.4.1.2.4	apply-map	83
7.4.1.2.5	action	84
7.4.1.2.6	apply-format	84
7.5	Format Statements	85
7.5.1	term	85
7.5.1.1	from clause	85
7.5.1.2	then clause	86
7.6	Map Statements	87
7.6.1	term	87
7.6.1.1	from clause	87
7.6.1.2	then clause	88
7.6.1.2.1	query	88
7.6.1.2.2	set	89
8	Troubleshooting	90
8.1	Logging	90
8.2	trace-options	90
9	Advanced Topic	92
9.1	Distributed Installs	92
10	Appendix	94
10.1	Limitations	94
10.2	Licenses	94
10.2.1	GNU General Public License	94

1 Introduction

This section of the manual covers an overview of SellaNMS, details the conventions used within this manual, touches on the features of SellaNMS and explains how you can help with this project.

1.1 Overview

SellaNMS is an open source Network Management System designed to be an extendable carrier grade solution for networks of all sizes.

It will discovery and monitor nearly any SNMP aware device (router, switch, server, etc.) and will accept input from undiscovered elements. It is known to work with [Juniper*](#), [Cisco*](#), [Extreme*](#), [Lucent/Livingston*](#) routers/switches, [Net-SNMP](#) configured servers and Linux based servers. Additionally, it has been used with [Arbor Networks*](#) Peakflow products.

The newest versions of SellaNMS are available at:

http://www.digitalgenesis.com/software/sella_nms/

**Juniper Networks, JUNOS, JUNOSe, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. Cisco and Cisco IOS are registered trademark of Cisco Systems, Inc. Extreme Networks is a registered trademark of Extreme Networks, Inc. Arbor Networks and Peakflow are registered trademarks of Arbor Networks, Inc. Lucent and Livingston are registered trademarks of Lucent, Inc.*

All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

1.2 About The Manual

This manual covers the the SellaNMS back-end daemons responsible for interacting with the network. SellaNMS GUIs are separate projects that primarily interact output from the SellaNMS back-end. Their administration is covered in a separate manual.

1.3 Conventions

Throughout the manual, the SellaNMS application is also referred to as “the system”, “the NMS” or “SellaNMS”. Each of these refer to the SellaNMS application and should be interpreted as such.

To aid the reader in interpreting the manual, the following visual cues are used:

- **blue text** – Used to denote configuration samples, commands and hyperlinks.

- **bold** – Used to highlight sections of a configuration example related the previous description. Also used to highlight items in a list, if a description follows.
- **italics** – Used to highlight significant SellaNMS terms such as module names, subsystem names and configuration sections. In syntax blocks it is used to denote replaceable values. Also used for references to sections of the manual and notes.
- **single quotes** - Used to highlight SellaNMS configuration keywords. Also used for executable scripts and programs within a description.
- **double quotes** - Used to highlight items that may be confusing to the reader without some form of separation such as filename, directories or suggested substitutions such as “.tar.*”.

NOTE: These visual cues with the exception of bold are not used within example configuration snippets.

Commands are listed after the typical command prompt symbol of the mode the user should be in. The following is a list of prompts used within the manual.

- **\$** - Server shell prompt for a normal (non-root) user.
- **#** - Server shell prompt for the root user, Juniper configuration edit mode, Extreme CLI prompt and Cisco CLI prompt.
- **>** - Juniper router prompt for the CLI (not edit mode).
- **(config)#** - Cisco configuration edit mode.
- **mysql>** - Interactive MySQL CLI prompt.

1.4 Features

SellaNMS has a wide variety of features making it an ideal solution for many networks.

- **Flexibility** - Modules have a wide variety of configuration options that adjust how they operate and perform. Additionally, the policy framework provides a great deal of flexibility on how the system processes data and correlates external data sources.
- **Extendable Design** - Modules can be written to extend the functionality of the system. The core application and libraries have a C++ API and work is being done to provide a Perl API, making it easier to write modules for the system.
- **Reliability** - Modules are separated into different processes and run independently of each other. This isolates the impact of bugs and errors to the affected module. Should a module fail, the remainder of the system continues and the watchdog process recovers the failed module.

- **Scalability** - SellaNMS is a multi-threaded and multi-process system written with highly optimized low memory algorithms. This allows the system to handle large workloads on average servers and take full advantage of servers with multiple CPUs. Additionally, the system can be configured to run it's modules distributed across multiple servers. This provides scalability to handle the largest of networks.
- **Portability** - The core application, libraries and modules are written in C and C++ and compile with GCC. This allows the system to be ported to a wide variety of UNIX based systems.
- **Open System** - The full source code for the system is available under the GPL license. The core application, libraries and modules are all available for examination and modification. Data is stored within a MySQL database and can be accessed via third-party applications.

1.5 How To Help

There are many ways to help the SellaNMS project. Below we cover a few of them.

1.5.1 Feedback

Providing feedback, both positive and negative, is a huge help to the project. Without sufficient feedback, it will take us longer to move the project towards what end-users need.

If you run into problems compiling or running SellaNMS please report those issues to the Forums or Mailing List (see *Support*). Additionally, if you successfully install and run the system, we'd like to know.

1.5.2 Donate Hardware

If you have hardware you would like to donate to the project, we will gladly accept it. We can use any additional manageable networking gear (Juniper, Cisco, Foundry, Extreme, Force10, Riverstone, etc.) to improve our test network. AC powered units are preferred. We can also use additional server platforms to aid us in porting.

Contact us via the Mailing List (see *Support*) if you are interested in donating.

1.5.3 Become A Developer

Become a SellaNMS developer and contribute directly to the project. This can mean writing new modules, developing policies for new network elements, providing MIBs, improving documentation or even writing a SellaNMS GUI.

Contact us via the Mailing List (see *Support*) if you're interested in becoming a developer.

2 Support

This section of the manual covers the available sources for SellaNMS support.

2.1 Mailing List

The SellaNMS User Mailing List is available at:

http://mailman.genesismuds.com/mailman/listinfo/sella_nms-user

The SellaNMS Developer Mailing List is available at:

http://mailman.genesismuds.com/mailman/listinfo/sella_nms-developer

2.2 Forums

The Support and Development Forums are available at:

<http://www.digitalgenesis.com/support/forum/>

2.3 Professional Services

Professional services are available for installation support, policy development and configuration support and development of new features and modules.

If you are interested in our professional services, please contact us by emailing sellanms@digitalgenesis.com.

3 Installation

This section of the manual covers requirements, configuration, building and starting SellaNMS.

3.1 Requirements

3.1.1 Operating System

SellaNMS was designed and implemented on [RedHat Linux](#) and [Fedora Core](#) distributions. It was implemented to be compatible with UNIX based system, however has not been tested or ported to any other platforms to date.

The system is know to run on:

- Fedora Core 4
- Fedora Core 2
- RedHat ES4
- RedHat ES3
- RedHat 9
- RedHat 8
- Debian Linux

3.1.2 Compiler

SellaNMS currently requires the GCC C/C++ compiler to build. The system is known to build with versions 2.95.3, 3.x and 4.0.1. Any version of GCC past 3.0 should build properly.

Work has been put into making the system compile with Intel's C/C++ compiler, however the current release does not build with it.

3.1.3 Net-SNMP

SellaNMS requires the Net-SNMP libraries and headers to build. It is known to build with versions 4.2.6, 5.0.6, 5.0.8 and 5.2.1. Nearly any version of Net-SNMP should be compatible.

Net-SNMP can be downloaded from:

<http://www.net-snmp.org>

On Fedora Core systems, you can install the required packages by running (as root):

```
# yum install net-snmp net-snmp-devel net-snmp-libs net-snmp-utils
```

On RedHat ES4 systems, you can install the required packages by running (as root):

```
# up2date -i net-snmp net-snmp-devel net-snmp-libs net-snmp-utils
```

Please consult the MySQL manual to find details on how to configure the MySQL administrator login and start it up.

3.1.4 MySQL

SellaNMS requires the MySQL client libraries and a working MySQL database. The system interacts with MySQL via the the libdbi abstraction layer. It is known to be compatible with versions 3.23.48, 4.0.13, 4.0.18 and 4.1.13a.

Any version of MySQL beyond 3.23.32 MySQL should be compatible, however we suggest at least the newest 4.0.x release, since the query cache feature provides a performance enhancement.

MySQL can be downloaded from:

<http://www.mysql.com>

On Fedora Core systems, you can install the required packages by running (as root):

```
# yum install mysql mysql-server mysqlclient10
```

On RedHat ES4 systems, you can install the required packages by running (as root):

```
# up2date -i mysql mysql-server mysqlclient10
```

Please consult the MySQL manual to find details on how to configure the MySQL administrator login and start it up.

3.1.5 libdbi

SellaNMS requires the libdbi and libdbi-drivers libraries and headers to build. It is known to work with versions 6.7, 0.7.1, 0.7.2 and 0.8.1. We recommend version 0.8.0 or better since, the library contained a memory leak in earlier versions.

If you build libdbi and libdbi-drivers from scratch, be sure to build with MySQL driver support.

The libdbi package can be downloaded from:

<http://libdbi.sourceforge.net/>

The libdbi-drivers package can be downloaded from:

<http://libdbi-drivers.sourceforge.net/>

On Fedora Core systems, you can install the required packages by running (as root):

```
# yum install libdbi libdbi-devel libdbi-dbd-mysql libdbi-dbd-mysql libdbi-drivers
```

On RedHat ES4 systems, you can install the required packages by running (as root):

```
# up2date -i libdbi libdbi-devel libdbi-dbd-mysql libdbi-dbd-mysql libdbi-drivers
```

3.2 Extracting

Before you can extract the distribution archive, make sure you have the newest copy from:

ftp://ftp.digitalgenesis.com/pub/sella_nms/

With the newest archive, you can extract the archive by running (replacing “2.x.x” with the proper filename):

```
$ tar xvzf sella_nms-2.x.x.tar.gz
```

Or if you downloaded the BZ2 archive, by running:

```
$ tar xvjf sella_nms-2.x.x.tar.gz2
```

The archive will extract the distribution into a directory that matches the filename, minus the “.tar.*” extension.

3.3 Building

To build SellaNMS, change to the distribution directory and run the configure script by running:

```
$ cd sella_nms-2.x.x
$ ./configure
```

The configure script may take a few minutes to complete. Without any arguments specified it will use the default settings to generate the Makefiles to compile the system.

If the configure script encounters any errors or finds that the required software is not installed, it will abort with a message reporting what it encountered. If the message isn’t descriptive enough, check the end of the configure log file which is found in “src/config.log” for hints.

Next, compile the system by running:

```
$ make all
```

Compiling will take between 5 and 30 minutes depending on the speed of your server. If you encounter any errors during the compiling process please report them (see *Support*).

3.4 Installing

Once compiling has completed, installing is the next step. You can install SellaNMS by running (as root):

```
$ su -  
# make install
```

Once the system is installed, it will startup a configuration script to walk you through the configuration of the system itself. This script can be run at any time by running 'config-sella_nms.sh' (installed in "/usr/local/sella_nms/sbin/" by default).

Make sure to configure your PHP enabled web server (Apache) to serve up the Web GUI (installed in "/usr/local/sella_nms/html" by default). The Web GUI will be deprecated in an upcoming release in favor of a stand alone GUI. Visit <http://www.digitalgenesis.com> for details.

3.5 Configuring

3.5.1 Network Element Configuration

Before starting up SellaNMS, you'll want to verify that your routers, switches, and servers have been properly configured to allow incoming ICMP and SNMP requests from your SellaNMS server. See the *Configuration* section of the manual for more information on this.

3.5.2 System Configuration

If you elected to not run the configuration script (during 'make install'), you will need to load the SellaNMS database schema into your MySQL server and finish configuring your sella_nms.conf configuration file.

You can run the configuration script again to take care of the configuration details for you by running (as root):

```
# /usr/local/sella_nms/sbin/config-sella_nms.sh
```

If you would prefer to manually prepare and configure SellaNMS, follow the next steps.

For MySQL 4.x run (from the top of the SellaNMS distribution):

```
$ mysql -u root -p sella_nms < support/sella_nms-mysql.sql
$ mysql -u root -p mysql
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,LOCK TABLES,CREATE TEMPORARY
      TABLES,CREATE,DROP ON sella_nms.* TO sella_nms@localhost
      IDENTIFIED BY 'password';
```

For MySQL 3.23.x run (from the top of the SellaNMS distribution):

```
$ mysql -u root -p sella_nms < support/sella_nms-mysql.sql
$ mysql -u root -p mysql
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON sella_nms.* TO
      sella_nms@localhost IDENTIFIED BY 'password';
```

Edit your configuration file by running (as root):

```
# cd /usr/local/sella_nms/etc
# vi sella_nms.conf
```

3.6 Starting

3.6.1 Startup

Run the following commands (as root) to startup SellaNMS:

```
# cd /usr/local/sella_nms/sbin
# ./sella_nms
```

SellaNMS will fork into the background and begin starting up its configured modules. See the *Troubleshooting* section if SellaNMS fails to startup or work as expected.

3.6.2 Processes

Starting up SellaNMS launches several modules each of which run as separate daemons.

You can view these processes by utilizing the 'admin-sella_nms.sh' script (as root). Below is an example:

```
# cd /usr/local/sella_nms/sbin/
# ./admin-sella_nms.sh -p
nobody 10665 0.0 0.5 10620 5748 ? SN Nov30 0:00 sella_nms [running]
nobody 10666 0.0 0.5 11472 6120 ? SN Nov30 0:07 int.tidyd [running]
nobody 10669 0.1 1.1 26068 11568 ? SN Nov30 19:53 out.guid [running]
nobody 10670 0.0 0.8 22436 8848 ? SN Nov30 6:58 mon.stated [running]
nobody 10672 0.7 1.3 35992 14360 ? SN Nov30 97:18 mon.syslogd [running]
nobody 10676 0.8 5.1 85004 17434 ? SN Nov30 96:30 mon.SNMP trapd [running]
nobody 20866 1.2 3.3 89988 18341 ? SN Nov30 99:19 mon.icmpd [running]
nobody 14229 0.0 0.7 13208 17812 ? SN Nov30 1:08 out.emaild [running]
nobody 15558 2.5 0.6 11968 16556 ? SN Nov30 0:12 out.syslogd [running]
```

Optionally, you can use the ‘ps’ command to view these processes. Below is an example:

```
$ ps auxw | grep -E 'sella_nms|dsc\.|mon\.|out\.|int\.'
nobody 10665 0.0 0.5 10620 5748 ? SN Nov30 0:00 sella_nms [running]
nobody 10666 0.0 0.5 11472 6120 ? SN Nov30 0:07 int.tidyd [running]
nobody 10669 0.1 1.1 26068 11568 ? SN Nov30 19:53 out.guid [running]
nobody 10670 0.0 0.8 22436 8848 ? SN Nov30 6:58 mon.stated [running]
nobody 10672 0.7 1.3 35992 14360 ? SN Nov30 97:18 mon.syslogd [running]
nobody 10676 0.8 5.1 85004 17434 ? SN Nov30 96:30 mon.SNMP trapd [running]
nobody 20866 1.2 3.3 89988 18341 ? SN Nov30 99:19 mon.icmpd [running]
nobody 14229 0.0 0.7 13208 17812 ? SN Nov30 1:08 out.emaild [running]
nobody 15558 2.5 0.6 11968 16556 ? SN Nov30 0:12 out.syslogd [running]
```

If you need SellaNMS or an individual module to reread its configuration, you can send the HUP signal. If the ‘sella_nms’ daemon is signaled, every module will reread its configuration and restart. However, if you only need one module to reread its configuration, you can signal the individual module, resulting in less disruption to the system.

3.6.3 Discovery

Part of the startup process is to begin the *discovery* modules. This behavior can be stopped by removing the ‘startup’ keyword from the *discovery schedule* section in the “sella_nms.conf” configuration file.

The discovery process will first scan the prefixes provided in the “sella_nms.conf” configuration file with ICMP. It will then use SNMP to poll each device that responded to ICMP. Finally, the data polled via SNMP is processed by the *topology* module to build the network topology and populate the database. This process will take a different amount of time on each network. For smaller networks, discovery should complete in just a few minutes. Larger networks (300+ elements) may take over 30 minutes.

The progress of the *discovery* modules can be monitored by using the 'admin-sella_nms.sh' script. The "-d" option, will display a discovery summary and the last 5 entries (by default) in the ICMP and SNMP discovery tables. Below is an example.

```
# cd /usr/local/sella_nms/sbin/
# ./admin-sella_nms.sh -d 1
[Discovery Information]
Discovery started; 360 ICMP hosts located, 9882 SNMP OIDs collected.
Previously completed discovery contains 329 elements, 31999 interfaces and 32107 IPs
Displaying the last 1 records (out of 360) from table discovery_icmp (descending order).
***** 1. row *****
      id: 14233
timestamp: 2005-12-09 04:00:09
      ip: 12.34.56.1

Displaying the last 1 records (out of 9882) from table discovery_snmp (descending order).
***** 1. row *****
      id: 9794980
timestamp: 2005-12-09 04:06:41
      ip: 12.34.56.40
variable: .1.3.6.1.2.1.15.3.1.4.66.192.246.226
value: 4
```

3.7 Shutdown

Since SellaNMS modules each run as separate daemons, shutting down the system needs to be performed in a specific way.

To shutdown SellaNMS and all of its modules, send the TERM (-15) signal (the default signal) to the 'sella_nms' daemon using the 'kill' command (as root). This will cause all of the modules to shutdown cleanly. Below is an example:

```
$ su -
# ps auxw | grep sella_nms
nobody 10665 0.0 0.5 10620 5748 ? SN Nov30 0:00 sella_nms [running]
# kill 10665
```

NOTE: Do not use the KILL (-9) signal to shutdown the 'sella_nms' daemon. This will cause the SellaNMS modules to continue running and will require you to manually shutdown each of them.

After issuing the 'kill' command to the 'sella_nms' daemon, SellaNMS will signal each of its modules asking them to shutdown cleanly. Any modules that do not fully shutdown within 10 seconds will be forcefully terminated. Finally, the 'sella_nms' daemon will finish shutting down.

Should the 'sella_nms' daemon be terminated uncleanly, you may need to shutdown its modules manually. If this happens, send a TERM (-15) signal to each of the SellaNMS modules. If any modules fail to shutdown after 10 seconds, use the KILL (-9) signal to forcefully terminate them. The *Processes* section has an example of how to get the process IDs for all of the SellaNMS modules.

4 Configuration

This section of the manual walks through the various aspects of configuring SellaNMS and configuring your network elements (routers, switches, etc.) to work with the system. You may need to consult your equipment vendors documentation to properly configure them.

4.1 Network Element

Your network elements may need to be configured to get the full benefit of SellaNMS. This section will walk you through some of the common configuration changes.

4.1.1 ICMP

Most manageable network elements respond to ICMP or ping. If you filter incoming traffic to your network elements, you'll need to verify that your SellaNMS server is allowed to send ICMP echo requests to it.

You can test if ICMP echo requests are being accepted by your network element by using the 'ping' command. In the example below, replace 12.34.56.1 with the IP of the network element you're testing:

```
$ ping 12.34.56.1
PING 12.34.56.1 (12.34.56.1) 56(84) bytes of data.
64 bytes from 12.34.56.1: icmp_seq=0 ttl=254 time=0.392 ms
64 bytes from 12.34.56.1: icmp_seq=1 ttl=254 time=0.395 ms
64 bytes from 12.34.56.1: icmp_seq=2 ttl=254 time=0.387 ms
```

4.1.2 SNMP

Most manageable network elements support SNMP queries. These network elements usually need to be configured to accept incoming queries. If you filter incoming SNMP traffic to your network elements, you'll need to verify that your SellaNMS server is allowed to communicate.

If you have the Net-SNMP utilities installed on your SellaNMS server you can use the 'snmpwalk' command to test this. In the example below, replace "my_comm" with your community and 12.34.56.1 with the IP of the network element you're testing.

```
$ snmpwalk -c my_comm 12.34.56.1 system
SNMPv2-MIB::sysDescr.0 = STRING: Juniper Networks, Inc. t320 internet router,
kernel JUNOS 7.2-20050831.0 #0: 2005-0 Build date: 2005-08-31 08:00:21 UTC
Copyright (c) 1996-2005 Juniper Networks, Inc.
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.2636.1.1.1.2.7
SNMPv2-MIB::sysUpTime.0 = Timeticks: (266827794) 30 days, 21:11:17.94
```

```
SNMPv2-MIB::sysContact.0 = STRING: support@domain.com
SNMPv2-MIB::sysName.0 = STRING: KYTO-CORE-01
SNMPv2-MIB::sysLocation.0 = STRING: Kyoto, Japan
SNMPv2-MIB::sysServices.0 = INTEGER: 4
```

4.1.2.1 Juniper

SNMP is disabled by default on Juniper routers. Below is a configuration snippet showing a valid SNMP configuration.

```
> show configuration snmp
snmp {
  location "Kyoto, Japan";
  contact "support@domain.com";
  community my_comm {
    clients {
      /* Your SellaNMS server. */
      192.168.0.55/32;
    }
  }
}
```

Below are the set commands needed to build this configuration:

```
> edit
# set snmp location "Kyoto, Japan"
# set snmp contact "support@domain.com"
# set snmp community my_comm clients 192.168.0.55/32
# commit
```

4.1.2.2 Cisco

SNMP is disabled by default on Cisco routers and switches. Below are example commands to enable SNMP for a specific host. You should verify that "access-list 10" is not already in use.

```
# conf t
(config)# access-list 10 permit 192.168.0.55
(config)# snmp-server location Kyoto, Japan
(config)# snmp-server contact support@domain.com
(config)# snmp-server chassis-id KYTO-CORE-01
(config)# snmp-server community my_comm RO 10
(config)# exit
# write
```

4.1.2.3 Extreme

Below are example commands to enable SNMP for a specific host.

```
# configure snmp sysName "KYTO-CORE-01"  
# configure snmp sysLocation "Kyoto, Japan"  
# configure snmp sysContact "support@domain.com"  
# configure snmp add community readonly my_comm  
# configure access-profile "PERMIT_SNMP" mode permit  
# configure access-profile "PERMIT_SNMP" add 5 permit ipaddress 192.168.0.55/32  
# configure snmp access-profile readonly PERMIT_SNMP  
# save configuration
```

4.1.2.4 Net-SNMP (Linux/BSD/Solaris)

If you build Net-SNMP from the source code, it will walk you through the configuration. If you need to manually configure Net-SNMP, edit the "snmpd.conf" configuration file (default "/etc/snmp/snmpd.conf"). You will need to set the following options:

```
rocommunity my_comm 127.0.0.1  
rocommunity my_comm 192.168.0.55/32  
syslocation "Kyoto, Japan"  
syscontact support@domain.com
```

After making these configuration changes to "snmpd.conf", send a HUP (-1) signal to 'snmpd' to have the changes take effect.

4.1.3 SNMP Trap

Most manageable network elements are able to generate SNMP Traps. These network elements need to be configured with the host name or IP address of a SNMP trap collector.

4.1.3.1 Juniper

Below is a configuration snippet showing a valid SNMP trap configuration.

```

> show configuration snmp
snmp {
  location "Kyoto, Japan";
  contact "support@domain.com";
  trap-group my_comm {
    version v2;
    categories {
      chassis;
      link;
      remote-operations;
      routing;
      startup;
      rmon-alarm;
    }
  }
  targets {
    /* Your SellaNMS server. */
    192.168.0.55;
  }
}

```

Below are the set commands needed to build this configuration:

```

> edit
# set snmp location "Kyoto, Japan"
# set snmp contact "support@domain.com"
# set snmp trap-group my_comm version v2
# set snmp trap-group my_comm categories chassis
# set snmp trap-group my_comm categories link
# set snmp trap-group my_comm categories remote-operations
# set snmp trap-group my_comm categories routing
# set snmp trap-group my_comm categories startup
# set snmp trap-group my_comm categories rmon-alarm
# set snmp trap-group my_comm targets 192.168.0.55
# commit

```

4.1.3.2 Cisco

Below is a configuration snippet showing a valid SNMP trap configuration.

```

# conf t
(config)# snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
(config)# snmp-server enable traps config
(config)# snmp-server enable traps envmon fan shutdown supply temperature status
(config)# snmp-server enable traps rf
(config)# snmp-server enable traps bgp
(config)# snmp-server host 192.168.0.55 my_comm
(config)# exit
# write

```

4.1.3.3 Extreme

Below is a configuration snippet showing a valid SNMP trap configuration.

```
# configure snmp add trapreceiver 192.168.0.55 port 162 community "my_comm" mode
enhanced
# save configuration
```

4.1.3.4 Net-SNMP (Linux/BSD/Solaris)

Net-SNMP's 'snmpd' daemon is capable of generating SNMP traps for authentication failures, cold start traps and user defined traps. See the man page for "snmpd.conf" to see how to configure Net-SNMP for SNMP traps.

4.1.4 Syslog

Most manageable network elements are able to log to remote syslog servers. These network elements need to be configured with the host name or IP address of a remote syslog server.

4.1.4.1 Juniper

Below is a configuration snippet showing a valid syslog configuration.

```
> show configuration system syslog
system {
  syslog {
    /* Your SellaNMS server. */
    host 192.168.0.55 {
      any any
    }
  }
}
```

Below are the set commands needed to build this configuration:

```
> edit
# set system syslog host 192.168.0.55 any any
# commit
```

4.1.4.2 Cisco

Below is a configuration snippet showing a valid syslog configuration.

```
# conf t
(config)# logging trap debugging
(config)# logging 192.168.0.55
(config)# exit
# write
```

4.1.4.3 Extreme

Below is a configuration snippet showing a valid syslog configuration.

```
# conf syslog add 192.168.0.55 local0
# enable syslog
# save configuration
```

4.1.4.4 syslogd (Linux/BSD/Solaris)

The syslog daemon on most Linux/BSD/Solaris servers can be configured to send logs to remote syslog servers. The configuration file for syslogd is usually “/etc/syslog.conf”. If your server uses a different syslog daemon such as syslog-ng, your configuration file may be elsewhere and will require different configuration changes.

For RedHat Linux and Fedora Core, you can direct syslog messages to a remote server by adding a line with the facility and priority followed by a host name with the “@” symbol prepended to it. Example below:

```
*.info;mail.none;authpriv.none;cron.none    @192.168.0.55
authpriv.*                                  @192.168.0.55
```

After adding these lines to your configuration, restart the syslog daemon by running (as root):

```
# /etc/rc.d/init.d/syslog restart
```

4.2 Syntax

4.2.1 Configuration Style

The SellaNMS configuration files are based on a hierarchy or tree structure. The basic style follows a 'keyword value' pair, or a 'keyword block' pair. All 'keyword value' pairs are terminated by a semicolon as shown below:

```
order 10;
```

All “key block” pairs are a keyword followed by a “{ }” block or a “[]” block set. Either style of blocks may be used in place of the other. The two are purely available for cosmetic reasons.

Each block main contain a series of “keyword value” pairs, “keyword block” pairs, values or a mix.

Below is an example from the “sella_nms.conf” configuration file:

```

discovery {
  icmp {
    description "Polls and records ICMP reachability for discovery";
    executable "dsc.icmpd";
    order 10;
    disable;

    prefix {
      192.168.0.0/24;
      192.168.10.0/24;
    }
    policy {
      input [ EXAMPLE_IN1 EXAMPLE_IN2 ];
      output [ EXAMPLE_OUT1 EXAMPLE_OUT2 ];
    }
  }
}

```

4.2.2 Comments

Bash, C, C++/Java and SQL comment styles are supported within the configuration files.

Below is an example of a bash style comment (pound character):

```

discovery {
  icmp {
    description "Polls and records ICMP reachability for discovery";
    executable "dsc.icmpd";
    order 10;
    # disable;
  }
}

```

Below is an example of a multi-line C style comment (/* ... */ block):

```

discovery {
  icmp {
    description "Polls and records ICMP reachability for discovery";
    executable "dsc.icmpd";
    /* order 10;
    disable;
    */
  }
}

```

Below is an example of a C++/Java comment (double slashes):

```

discovery {
  icmp {
    description "Polls and records ICMP reachability for discovery";
    executable "dsc.icmpd";
    order 10;
  }
  // disable;
}

```

Below is an example of an SQL comment (double dashes):

```

discovery {
  icmp {
    description "Polls and records ICMP reachability for discovery";
    executable "dsc.icmpd";
    order 10;
  }
  -- disable;
}

```

4.2.3 Includes

The configuration files support file and directory includes and nested inclusions. Nesting is supported up to a depth of 255. If needed, the limit can be altered by editing FLATFILE_MAXDEPTH within "flatfile.h" and recompiling.

This feature is useful for creating user specific policies outside of the policies included with SellaNMS. This makes it easier to identify which are specific to your installation.

Below is an example of a file include statement:

```
#include "/usr/local/sella_nms/etc/policy-statements.conf"
```

Below is an example of a directory include statement:

```
#include "/usr/local/sella_nms/etc/policy.d"
```

All files within the directory "/usr/local/sella_nms/etc/policy.d" will be included as if they were each specified separately.

4.2.4 Merging

Configuration merging is a feature of the SellaNMS configuration system. If a configuration hierarchy overlaps at the same level with another configuration hierarchy, they will be automatically (and silently) merged.

This feature is primarily available to allow sections of the policy-options hierarchy to be broken into separate more manageable files. This allows for the easy addition of policies for new vendor equipment.

We don't suggest writing configurations like this, but for example, view the following configuration snippet:

```
discovery {
  icmp {
    description "Polls and records ICMP reachability for discovery";
    executable "dsc.icmpd";

    prefix {
      192.168.0/24;
    }

    prefix {
      192.168.10.0/24;
    }

    policy {
      input [ EXAMPLE_IN1 ];
      output [ EXAMPLE_OUT1 ];
    }

    policy {
      input [ EXAMPLE_IN2 ];
      output [ EXAMPLE_OUT2 ];
    }
  }
}
```

The previous configuration snippet will be merged into:

```
discovery {
  icmp {
    description "Polls and records ICMP reachability for discovery";
    executable "dsc.icmpd";

    prefix {
      192.168.0/24;
      192.168.10.0/24;
    }

    policy {
      input [ EXAMPLE_IN1 EXAMPLE_IN2 ];
      output [ EXAMPLE_OUT1 EXAMPLE_OUT2 ];
    }
  }
}
```

4.3 Files/Directories

The SellaNMS configuration files are broken into separate files and directories to make them easier to maintain. Below, we detail out the various files and directories used.

- **sella_nms.conf** – This is the top level configuration file. It contains system and module specific configuration options used to adjust how the modules run and which policies are applied to them. Most general user configuration changes take place within this file.
- **policy-statements.conf** – The primary policy statement configuration file. It contains the generic non-vendor specific policies.
- **policy.d** – Directory containing vendor specific policy, format and map statement configuration files. Any file within this directory will be included into the configuration.
- **format-statements.conf** – The primary format configuration file. It contains generic non-vendor specific format statements which are applied to policy statements to define how variables function and enforce proper input and output for modules.
- **map-statements.conf** – The primary map configuration file. It contains generic non-vendor specific map statements which are applied to policies and formats to map external data to the data provided by a module, as a policy is evaluated.

5 Modules

This section of the manual covers the syntax and options for the configuration of the SellaNMS subsystems and modules.

5.1.1 Overview

Modules make up the majority of SellaNMS. They are the workers of the system, each responsible for performing a different set of tasks. Each module is grouped into one of SellaNMS's subsystems; discovery, monitor, output and internal.

Each module has a set of 'sella_nms' (watchdog process) configuration options, a set of common configuration options, and a set of unique configuration options.

5.1.2 sella_nms Configuration

The 'sella_nms' daemon is responsible for starting and stopping the system's modules based on a schedule. It is also responsible for monitoring the health of the modules and restarting them should they fail or behave beyond the configured thresholds. While it shares some source code in common with the systems modules, it is technically a watchdog process and not a module.

Every module's configuration block may contain any of the following options, which are read by the 'sella_nms' daemon:

- **config-file** – The configuration file to be used by the module (default: use compiled in value).
- **description** – A description field used to describe the module (default: none).
- **disable** – Disable (don't run) the module (default: not set).
- **environment** – The environment variables to set before running the module (default: none).
- **executable** – The executable for this module (default: none).
- **facility** – The syslog facility to log messages from this module. (default: local0).
- **group** – The UNIX group to run the module as. (default: use primary group associated with the user).
- **nice** – The niceness at which to run this module (default: 5).
- **order** – The order in which to start this module, relative to the order of other modules. Lower numbered modules are started before higher numbered modules. Modules at the same order are started at the same time. All modules at the current order must be started and complete (shutdown) before moving to the next order with the exception of order 0. Order 0 is treated as a non-order and will not block the startup of other orders.

- **pid-file** – The file used to store the process ID (PID) of the module (default: use value compiled into module).
- **restart** – Restart the module if it stops running (default: active).
- **no-restart** – Do not restart the module if it stops running (default: not active).
- **restart-limit** – The number of times to restart the module before automatically disabling it (default: ULONG_MAX)
- **schedule** – The schedule at which to run the module (default: none). See the ‘system default schedule’ portion of the *Modules* section of the manual for more information.
- **stat-file** – The file to store the module stats in (default: use value compiled into module).
- **user** – The user to run the module as (default: current user).

5.1.3 Common Configuration

The majority of SellaNMS modules use the ‘system’ section of the configuration to specify how to connect to the database and which SNMP MIBs to read in.

Most SellaNMS modules support the following configuration options within the module specific section of their configurations:

- **trace-options** – Enables a set of trace (debug) options used to troubleshoot a module. See the Troubleshooting section of the manual for more information.
- **policy input** – Specify a set of input policies for the module. Input policies are primarily used to reject unwanted incoming data.
- **policy output** – Specify a set of output policies for the module. Output policies are primarily used to manipulate data and specify what data should be passed to the output subsystem.

5.1.4 System

The ‘system’ section of the configuration file contains general options that apply to SellaNMS and its modules.

5.1.4.1 Syntax Overview

The following block details the top level configuration syntax for the ‘system’ section of the configuration:

```

system {
  trace-options [ ... ];
  storage {
    database-server-name {
      debug;
      driver (mysql | pgsql);
      driver-directory path-to-driver-directory;
      server (localhost | host-name | ip-address);
      port port-number;
      socket path-to-unix-socket;
      compression;
      tty tty;
      options options;
      database database;
      username username;
      password password;
      pool {
        debug;
        timeout milliseconds;
        trim-delay seconds;
        min-size size;
        max-size size;
      }
    }
  }
  snmp-mib {
    path-to-mib-directory;
    ...
  }
  default {
    config-file path-to-configuration-file;
    description description;
    directory path-to-binary-directory;
    disable;
    environment environment-variables;
    executable path-to-executable-file;
    facility facility;
    group unix-group;
    launch-delay seconds;
    nice niceness;
    order run-order;
    pid-file path-to-process-id-file;
    (restart | no-restart);
    restart-limit limit;
    schedule {
      startup;
      delay seconds;
      cron enhanced-vixie-cron-format;
      second second;
      minute minute;
      hour hour;
      day day;
      day-of-week day-of-week;
      month month;
      day-of-month day-of-month;
    }
    stat-file path-to-stats-file;
    user unix-user;
  }
}

```

```
}  
}
```

5.1.4.2 system

5.1.4.2.1 Syntax

```
system {  
    trace-options [ ... ];  
    storage { ... }  
    snmp-mib { ... }  
    default { ... }  
}
```

5.1.4.2.2 Hierarchy

top

5.1.4.2.3 Description

The 'system' section of the configuration contains general options that apply to SellaNMS and its modules.

5.1.4.2.4 Options

- **trace-options** – The trace-options to apply to the 'sella_nms' daemon (default: none). See the *Troubleshooting* section for more information on how to use this option.
- **storage** – The storage related configuration options (default: none).
- **snmp-mib** – The SNMP MIB related configuration options (default: none).
- **default** – The default options for modules started by the 'sella_nms' daemon (default: none).

5.1.4.3 storage

5.1.4.3.1 Syntax

```
storage {  
    database-server-name {  
        debug;  
        driver (mysql | pgsql | driver-name);  
        driver-directory path-to-driver-directory;  
        server (localhost | host-name | ip-address);  
        port port-number;  
        socket path-to-unix-socket;  
        compression;  
        tty tty;  
        options options;  
        database database;  
        username username;  
        password password;  
        pool {  
            debug;  
            timeout seconds;  
            trim-delay seconds;  
            min-size size;  
            max-size size;  
        }  
    }  
}
```

5.1.4.3.2 Hierarchy

```
system { ... }
```

5.1.4.3.3 Description

Contains one or more database configuration sets.

5.1.4.3.4 Options

- **server-database-name** – A descriptive name for the configuration set. Multiple configuration sets may be specified using different server-database-names (default: none).
- **debug** – Enable debug output for the storage library, excluding the connection pool code. (default: not active).
- **driver** – The libdbi driver to use to connect to the database (default: mysql).
 - **mysql** – The mysql libdbi driver.
 - **pgsql** – The PostgreSQL libdbi driver.
 - **driver-name** – The libdbi driver name.
NOTE: Only the MySQL driver code has been tested and some MySQL specific SQL statements are currently used within SellaNMS.
- **driver-directory** – The libdbi driver directory to search for the specified driver (default: use directory compiled into libdbi).
- **server** – The database server to connect to (default: localhost).

- **localhost** – Connect to the local server via a UNIX socket.
- **host-name** – Connect to host-name via a TCP socket.
- **ip-address** – Connect to IP addresss via a TCP socket.
- **port** – TCP port to use to connect to the database sever (default: use driver default port).
- **socket** – The UNIX socket to use to connect to the database (default: use driver default socket). This option is only available when using the MySQL driver.
- **compression** – Enable compression of the database between SellaNMS and the database server (default: not active). This option is only available when using the MySQL driver.
- **tty** – The tty to use to connect to the database (default: use driver default tty). This option is only available when using the PostgreSQL driver.
- **options** – The options to use when connecting to the database (default: none). This option is only available when using the PostgreSQL driver.
- **database** – The database name to use (default: sella_nms).
- **username** – The username to use when connecting to the database (default: none).
- **password** – The password to use when connecting to the database (default: none).
- **pool** – The storage library's connection pool provides a dynamic set of active connections to the database. It improves performance for modules that need to frequently open and close connections to the database and prevents file descriptor depletion (default: enabled).
- **pool debug** - Enable debug output for the storage libraries connection pool code. (default: not active).
- **pool timeout** – The number of seconds a database connection must be idle before the connection pool will reconnect it (default: 300 seconds).
- **pool trim-delay** – The number of seconds a database connection must be idle before the connection pool will consider it for removal from the pool (default: 30 seconds).
- **pool min-size** – The minimum number of connections the connection pool will maintain with the database (default: 2).
- **pool max-size** – The maximum number of connections the connection pool will maintain with the database (default: 32).

5.1.4.4 snmp-mib

5.1.4.4.1 Syntax

```
snmp-mib {  
    path-to-mib-directory;  
    ...  
}
```

5.1.4.4.2 Hierarchy

```
system { ... }
```

5.1.4.4.3 Description

Contains a set of SNMP MIB directories.

5.1.4.4.4 Options

- **path-to-mib-directory** – The full path to a SNMP MIB directory. All MIBs within this directory will be loaded (default: none). Child directories will not be loaded, unless specified on a separate line.

5.1.4.5 default

5.1.4.5.1 Syntax

```
default {  
    config-file path-to-configuration-file;  
    description description;  
    directory path-to-binary-directory;  
    disable;  
    environment environment-variables;  
    executable path-to-executable-file;  
    facility facility;  
    group unix-group;  
    nice niceness;  
    order run-order;  
    pid-file path-to-process-id-file;  
    (restart | no-restart);  
    restart-limit limit;  
    schedule {  
        startup;  
        delay seconds;  
        launch-delay seconds;  
        cron enhanced-vixie-cron-format;  
        second second;  
        minute minute;  
        hour hour;  
        day day;  
        day-of-week day-of-week;  
        month month;  
        day-of-month day-of-month;  
    }  
    stat-file path-to-stats-file;  
    user unix-user;  
}
```

5.1.4.5.2 Hierarchy

```
system { ... }
```

5.1.4.5.3 Description

Contains the default settings for all modules started by the 'sella_nms' daemon. If any of these options are specified within a module's section of the configuration, it will override the value specified here.

5.1.4.5.4 Options

- **config-file** – The configuration file to be used by the module (default: use compiled in value).
- **description** – A description field used to describe the module (default: none).
- **disable** – Disable (don't run) the module (default: not set).
- **environment** – The environment variables to set before running the module (default: none).
- **executable** – The executable for this module (default: none).

- **facility** – The syslog facility to log messages from this module. (default: local0).
- **group** – The UNIX group to run the module as. (default: use primary group associated with the user).
- **nice** – The niceness at which to run this module (default: 5).
- **order** – The order in which to start this module, relative to the order of other modules. Lower numbered modules are started before higher numbered modules. Modules at the same order are started at the same time. All modules at the current order must be started and complete (shutdown) before moving to the next order with the exception of order 0. Order 0 is treated as a non-order and will not block the startup of other orders.
- **pid-file** – The file used to store the process ID (PID) of the module (default: use value compiled into module).
- **restart** – Restart the module if it stops running (default: active).
- **no-restart** – Do not restart the module if it stops running (default: not active).
- **restart-limit** – The number of times to restart the module before automatically disabling it (default: ULONG_MAX)
- **schedule** – The schedule at which to run the module (default: none).
- **stat-file** – The file to store the module stats in (default: use value compiled into module).
- **user** – The user to run the module as (default: current user).

5.1.4.6 Schedule

5.1.4.6.1 Syntax

```
schedule {  
    startup;  
    delay seconds;  
    launch-delay seconds;  
    cron enhanced-vixie-cron-format;  
    second second;  
    minute minute;  
    hour hour;  
    day day;  
    day-of-week day-of-week;  
    month month;  
    day-of-month day-of-month;  
}
```

5.1.4.6.2 Hierarchy

```
system default { ... }  
subsystem-name { ... }  
subsystem-name module-name { ... }
```

5.1.4.6.3 Description

Contains options used to control the schedule used to run a module.

5.1.4.6.4 Options

- **startup** – Run the module on startup even if there is a schedule specified. The schedule will be followed after the first startup of the module (default: not active).
- **delay** – The number of seconds to delay or wait between starting the module. If a cron based schedule is also used, the delay seconds will be added the specified start time for the first run only (default: 0 seconds).
- **launch-delay** – The number of seconds to delay after starting this module. This feature provides a way to prevent more than one module from starting at exactly the same time (default: 2 seconds).
- **cron** – The schedule at which to startup the module (default: “* * * * *”). The format is exactly the same as Vixie CRON notation, except a seconds field is added to the left of the minutes field. See the crontab(5) man page for more information. This option will override the use of the second, minute, hour, day, dayofweek, month and dayofmonth options.
- **schedule second** – The seconds that the module should be started (default: all).
- **minute** – The minutes that the module should started (default: all).

- **hour** – The hours that the module should be started (default: all).
- **day** – The days that the module should be started (default: all).
- **day-of-week** – The days of the week that the module should be started (default: all).
- **month** – The months that the module should be started (default: all).
- **day-of-month** – The days of the month that the module should be started (default: all).

5.1.5 discovery

The 'discovery' section of the configuration file contains options for the discovery subsystem and all discovery modules.

The discovery modules are responsible for collecting information from the network's elements, which can be used by the NMS to produce more meaningful alarms and searchable information.

5.1.5.1 Syntax Overview

The following block details the top level configuration syntax for the 'discovery' section of the configuration:

```

discovery {
  disable;
  nice niceness;
  (restart | no-restart);
  schedule {
    startup;
    delay seconds;
    launch-delay seconds;
    cron enhanced-vixie-cron-format;
    second second;
    minute minute;
    hour hour;
    day day;
    day-of-week day-of-week;
    month month;
    day-of-month day-of-month;
  }
  icmp { ... }
  snmp { ... }
  topology { ... }
  module-name { ... }
}

```

5.1.5.2 discovery

5.1.5.2.1 Syntax

```

discovery {
  disable;
  schedule { ... }
  icmp { ... }
  snmp { ... }
  topology { ... }
  module-name { ... }
}

```

5.1.5.2.2 Hierarchy

top

5.1.5.2.3 Description

The ‘discovery’ section of the configuration contains options for the discovery modules.

5.1.5.2.4 Options

- **disable** – Disable (don’t run) the module (default: not set).
- **nice** – The niceness at which to run this module (default: 5).
- **restart** – Restart the module if it stops running (default: active).
- **no-restart** – Do not restart the module if it stops running (default: not active).
- **schedule** – The schedule at which to run the module (default: none). See the ‘system default schedule’ portion of the *Modules* section of the manual for more information.

- **icmp** – The configuration options for the ICMP discovery module (default: use module specific defaults).
- **snmp** – The configuration options for the SNMP discovery module (default: use module specific defaults).
- **topology** – The configuration options for the topology discovery module (default: use module specific defaults).
- **module-name** – The configuration options for the 'module-name' module (default: use module specific defaults). Module names are not hard coded and any unique name maybe used to group the configuration options for the module.

5.1.5.3 icmp

5.1.5.3.1 Syntax

```
icmp {
    /* 'sella_nms' configuration options */
    config-file path-to-configuration-file;
    description description;
    directory path-to-binary-directory;
    disable;
    environment environment-variables;
    executable path-to-executable-file;
    facility facility;
    group unix-group;
    nice niceness;
    order run-order;
    pid-file path-to-process-id-file;
    (restart | no-restart);
    restart-limit limit;
    schedule {
        startup;
        delay seconds;
        launch-delay seconds;
        cron enhanced-vixie-cron-format;
        second second;
        minute minute;
        hour hour;
        day day;
        day-of-week day-of-week;
        month month;
        day-of-month day-of-month;
    }
    stat-file path-to-stats-file;
    user unix-user;

    /* Common configuration options */
    trace-options [ ... ];
    policy {
        input [ ... ];
        output [ ... ];
    }

    /* Unique configuration options */
    source ip-address;
    interface interface-name;
    transmit count;
    required count;
    timeout milliseconds;
    prefix {
        cidr-prefix;
        ...
    }
}
```

5.1.5.3.2 Hierarchy

discovery { ... }

5.1.5.3.3 Description

The 'discovery icmp' section of the configuration contains options for the ICMP discovery module.

This module polls network elements using ICMP ECHO_REQUEST packets and expects active elements to respond with ICMP ECHO_REPLY packets. Elements that are determined to be active are stored within the database and used to seed the SNMP discovery module's polling process.

5.1.5.3.4 Options

- **source** – The IP address to source packets from this module from (default: chosen by kernel). This option overrides the 'interface' option.
- **interface** – The interface to take the IP address to source packets from this module (default: chosen by kernel). This option is overridden by the 'source' option.
- **transmit** – The number of packets to send to each destination polled by this module (default: 3 packets).
- **required** – The number of packets that must be received back from a polled destination to be considered active (default: 1 packet).
- **timeout** – The number of milliseconds to wait before assuming a polled destination will not respond.
- **prefix** – The prefix(es) to use to generate the destinations to poll (default: none). This must be specified as a prefix in CIDR notation (eg: 192.168.55.0/24).

NOTE: See the `sella_nms` Configuration and Common Configuration section of the manual for detailed information on their respective options.

5.1.5.3.5 Policies

5.1.5.4 snmp

5.1.5.4.1 Syntax

```
snmp {  
    /* 'sella_nms' configuration options */  
    config-file path-to-configuration-file;  
    description description;  
    directory path-to-binary-directory;  
    disable;  
    environment environment-variables;  
    executable path-to-executable-file;  
    facility facility;  
    group unix-group;  
    nice niceness;  
    order run-order;  
    pid-file path-to-process-id-file;  
    (restart | no-restart);  
    restart-limit limit;  
    schedule {  
        startup;  
        delay seconds;  
        launch-delay seconds;  
        cron enhanced-vixie-cron-format;  
        second second;  
        minute minute;  
        hour hour;  
        day day;  
        day-of-week day-of-week;  
        month month;  
        day-of-month day-of-month;  
    }  
    stat-file path-to-stats-file;  
    user unix-user;  
  
    /* Common configuration options */  
    trace-options [ ... ];  
    policy {  
        input [ ... ];  
        output [ ... ];  
    }  
  
    /* Unique configuration options */  
    source ip-address;  
    interface interface-name;  
    threads count;  
    commit-count count;  
    community [ ... ];  
    timeout milliseconds;  
    retry count;  
    prefix {  
        cidr-prefix;  
        ...  
    }  
}
```

```
oid {
    snmp-oid;
    ...
}
}
```

5.1.5.4.2 Hierarchy

```
discovery { ... }
```

5.1.5.4.3 Description

The 'discovery snmp' section of the configuration contains options for the SNMP discovery module.

5.1.5.4.4 Options

- **source** – The IP address to source packets from this module from (default: chosen by kernel). This option overrides the 'interface' option.
- **interface** – The interface to take the IP address to source packets from this module (default: chosen by kernel). This option is overridden by the 'source' option.
- **threads** – The number of threads to use to poll the network (default: 32 threads). Each thread polls one network element at a time.
- **commit-count** – The number SNMP OIDs to commit to the database a time (default: 300 OIDs).
- **community** – The community or communities to use to poll the network elements (default: public).
- **timeout** – The number of milliseconds to wait before assuming that the transmitted SNMP request was lost and retransmitting (default: 1000).
- **retry** – The number of retry attempts before assuming the network element will no longer respond (default: 3).
- **prefix** – The prefix(es) to use to generate the destinations to poll (default: none). This must be specified as a prefix in CIDR notation (eg: 192.168.55.0/24).

NOTE: See the sella_nms Configuration and Common Configuration section of the manual for detailed information on their respective options.

5.1.5.4.5 Policies

5.1.5.5 topology

5.1.5.5.1 Syntax

```
topology {  
    /* 'sella_nms' configuration options */  
    config-file path-to-configuration-file;  
    description description;  
    directory path-to-binary-directory;  
    disable;  
    environment environment-variables;  
    executable path-to-executable-file;  
    facility facility;  
    group unix-group;  
    nice niceness;  
    order run-order;  
    pid-file path-to-process-id-file;  
    (restart | no-restart);  
    restart-limit limit;  
    schedule {  
        startup;  
        delay seconds;  
        launch-delay seconds;  
        cron enhanced-vixie-cron-format;  
        second second;  
        minute minute;  
        hour hour;  
        day day;  
        day-of-week day-of-week;  
        month month;  
        day-of-month day-of-month;  
    }  
    stat-file path-to-stats-file;  
    user unix-user;  
  
    /* Common configuration options */  
    trace-options [ ... ];  
    policy {  
        input [ ... ];  
        output [ ... ];  
    }  
  
    /* Unique configuration options */  
    duplicate-algo algorithm;  
    merge-algo algorithm;  
    duplicate-percent percentage;  
    merge-percent percentage;  
    maximum-entry-delete count;  
}
```

5.1.5.5.2 Hierarchy

```
discovery { ... }
```

5.1.5.5.3 Description

The 'discovery topology' section of the configuration contains options for the topology discovery module.

5.1.5.5.4 Options

- **duplicate-algo** – The algorithm used to determine if two newly polled network elements are the same element or two different elements (default: a-and-b). If the elements are determined to be duplicates, the two elements are merged. Valid options are listed below:
 - **a-and-b** – A (first element) and B (second element) will be considered the same element if and only if both A and B have greater than or equal to the 'match-percent' of interfaces in common.
 - **a-or-b** – A (first element) and B (second element) will be considered the same element if and only if A or B have greater than or equal to the 'match-percent' of interfaces in common.
- **merge-algo** – The algorithm used to determine if a newly polled network element is the same as an element within the existing network discovery (default: hostname-or-merge-percentage). Valid options are listed below:
 - **hostname-or-merge-percentage** – If the hostname of two elements match or the percentage of interfaces matching is greater than or equal to the 'merge-percent', merge the elements.
 - **hostname-and-merge-percentage** – If the hostname of two elements match and the percentage of interfaces matching is greater than or equal to the 'merge-percent', merge the elements.
 - **hostname-only** – If the hostname of two elements match, merge the elements.
 - **merge-percentage-only** – If the percentage of interfaces matching is greater than or equal to the 'merge-percent', merge the elements.
- **duplicate-percent** – The percentage used for the 'duplicate-algo' option (default: 90 percent).
- **merge-percent** – The percentage used for the 'merge-algo' option (default: 95 percent).
- **maximum-entry-delete** – The maximum number of entries to delete from the SNMP discovery information in one command (default: 500 entries).
-

NOTE: See the `sella_nms` Configuration and Common Configuration section of the manual for detailed information on their respective options.

5.1.5.5.5 Policies

5.1.6 monitor

The 'monitor' section of the configuration file contains options for the monitor subsystem and all monitoring modules.

The monitor modules are responsible for polling and collecting information from the network's elements, processing that data and generating alarms or actions based on that input.

5.1.6.1 Syntax Overview

The following block details the top level configuration syntax for the 'monitor' section of the configuration:

```

monitor {
  disable;
  nice niceness;
  (restart | no-restart);
  schedule {
    startup;
    delay seconds;
    launch-delay seconds;
    cron enhanced-vixie-cron-format;
    second second;
    minute minute;
    hour hour;
    day day;
    day-of-week day-of-week;
    month month;
    day-of-month day-of-month;
  }
  icmp { ... }
  snmptrap { ... }
  syslog { ... }
  state { ... }
  module-name { ... }
}

```

5.1.6.2 monitor

5.1.6.2.1 Syntax

```

monitor {
  disable;
  schedule { ... }
  icmp { ... }
  snmptrap { ... }
  syslog { ... }
  state { ... }
  module-name { ... }
}

```

5.1.6.2.2 Hierarchy

[top](#)

5.1.6.2.3 Description

The ‘monitor’ section of the configuration contains options for the monitor modules.

5.1.6.2.4 Options

- **disable** – Disable (don’t run) the module (default: not set).
- **nice** – The niceness at which to run this module (default: 5).
- **restart** – Restart the module if it stops running (default: active).
- **no-restart** – Do not restart the module if it stops running (default: not active).

- **schedule** – The schedule at which to run the module (default: none). See the ‘system default schedule’ portion of the *Modules* section of the manual for more information.
- **icmp** – The configuration options for the ICMP monitoring module (default: use module specific defaults).
- **snmptrap** – The configuration options for the SNMP Trap monitoring module (default: use module specific defaults).
- **syslog** – The configuration options for the syslog monitoring module (default: use module specific defaults).
- **state** – The configuration options for the state monitoring module (default: use module specific defaults).
- **module-name** – The configuration options for the ‘module-name’ module (default: use module specific defaults). Module names are not hard coded and any unique name maybe used to group the configuration options for the module.

5.1.6.3 icmp

5.1.6.3.1 Syntax

```
icmp {  
    /* 'sella_nms' configuration options */  
    config-file path-to-configuration-file;  
    description description;  
    directory path-to-binary-directory;  
    disable;  
    environment environment-variables;  
    executable path-to-executable-file;  
    facility facility;  
    group unix-group;  
    nice niceness;  
    order run-order;  
    pid-file path-to-process-id-file;  
    (restart | no-restart);  
    restart-limit limit;  
    schedule {  
        startup;  
        delay seconds;  
        launch-delay seconds;  
        cron enhanced-vixie-cron-format;  
        second second;  
        minute minute;  
        hour hour;  
        day day;  
        day-of-week day-of-week;  
        month month;  
        day-of-month day-of-month;  
    }  
    stat-file path-to-stats-file;  
    user unix-user;  
  
    /* Common configuration options */  
    trace-options [ ... ];  
    policy {  
        input [ ... ];  
        output [ ... ];  
    }  
  
    /* Unique configuration options */  
    source ip-address;  
    interface interface-name;  
    interval seconds;  
    interval-adjust seconds;  
    payload string;  
    timeout milliseconds;  
    transmit count;  
    ttl hops;  
    tos value;  
    dscp value;  
    precedence value;  
    queue-depth count;  
    calc-interval seconds;  
    calc-interval-adjustment seconds;
```

```

calc-queue-depth count;
calc-queue-lower-threshold percentage;
calc-queue-upper-threshold percentage;
policy-queue-depth count;
policy-queue-limit count;
policy-queue-lower-threshold percentage;
policy-queue-upper-threshold percentage;
suppression {
    packet-loss percentage;
    rtt milliseconds;
    jitter milliseconds;
    normalized-rtt milliseconds;
}
}

```

5.1.6.3.2 Hierarchy

```
monitor { ... }
```

5.1.6.3.3 Description

The 'monitor icmp' section of the configuration contains options for the ICMP monitor module.

This module polls IPv4 addresses configured on network elements using ICMP ECHO_REQUEST packets and expects the elements to respond with ICMP ECHO_REPLY packets. The module will track the packet loss, RTT (round trip time), jitter and variation in RTT (normalized RTT) from each polled IPv4 destination.

5.1.6.3.4 Options

- **source** – The IP address to source packets from this module from (default: chosen by kernel). This option overrides the 'interface' option.
- **interface** – The interface to take the IP address to source packets from this module (default: chosen by kernel). This option is overridden by the 'source' option.
- **interval** – The interval in seconds in which to poll all destinations (default: 60 seconds). Packets are transmitted evenly across the interval period and not flooded all at once towards a destination.
- **interval-adjustment** – The number of seconds the module may adjust the 'interval' either up or down by (per interval period), when the server is unable to keep up (default: 5 seconds).
- **payload** – The payload of the transmitted ICMP packets (default: "SellaNMS – ICMP Monitor Module").
- **timeout** – The maximum amount of time in milliseconds to allow a destination to respond before assuming it will not (default: 5000 milliseconds).

- **transmit** – The number of packets to transmit to each destination during an interval period (default: 3 packets). Packets to the same destination are not transmitted until the previous set of destinations is complete. They are transmitted evenly across the interval period.
- **ttl** – The time to live (TTL) to set within the transmitted packets header (default: use OS default). Each layer 3 element along the path towards the destination will decrement the TTL. Should the TTL reach 0, then packet will be discarded and not reach its destination.
- **tos** – The value to set the ToS bits within the transmitted packets header to (default: 0).
- **dscp** – The value to set the DSCP bits within the transmitted packets header to (default: 0)
- **precedence** – The value to set the IP Precedence bits within the transmitted packets header to (default: 0)
- **queue-depth** – The maximum depth for the modules queues. (default: 4000).
- **calc-interval** – The interval in seconds at which to perform calculations against the modules collected data (default: 30 seconds). This option is effects how quickly the module will be able to detect changes in the network.
- **calc-interval-adjustment** – The number of seconds the 'calc-interval' may be adjusted by (per 'calc-interval') if the server is unable to keep up with the current 'calc-interval' (default: 5 seconds).
- **calc-queue-depth** – The depth to allow the module's calc-queue (data pending insertion into the calculation threads data structures) to reach before dropping new incoming data for the queue (default: use 'queue-depth'). This option will override the 'queue-depth' option.
- **calc-queue-lower-threshold** – The lower threshold of the percentage of the calc-queue's maximum depth at which to decrease the 'interval' by 'interval-adjustment' seconds (default: 15 percent). The 'interval' will not be adjusted below the originally configured value.
- **calc-queue-upper-threshold** – The upper threshold of the percentage of the calc-queue's maximum depth at which to increase 'interval' by 'interval-adjustment' seconds (default: 70 percent).
- **policy-queue-depth** – The depth to allow the module's policy-queue (data pending processing by a policy) before dropping new incoming data for the queue (default: 4 times the calc-queue-depth).
- **policy-queue-limit** – The number of policy queue entries to be processed at one time (default: 4000).

- **policy-queue-lower-threshold** – The lower threshold of the percentage of the policy-queue’s maximum depth at which to decrease the ‘calc-interval’ by ‘calc-interval-adjustment’ seconds (default: 15 percent). The ‘calc-interval’ will not be adjusted below the originally configured value.
- **policy-queue-upper-threshold** – The upper threshold of the percentage of the policy-queue’s maximum depth at which to increase ‘calc-interval’ by ‘calc-interval-adjustment’ seconds (default: 70 percent).
- **suppression** – Settings used to suppress or ignore minor changes in calculated data verses the previous data. Data that is suppressed is not passed to the policy-queue and is not processed by the policy framework. This results in a significant savings in CPU cycles, since a policy will not have to be evaluated every interval period for every destination.
- **suppression disable** – Disable all suppression features (default: not set). This is not recommended, but is useful if you suspect the suppression code is not functioning properly.
- **suppression packet-loss** – The amount of variation in packet loss between the current value and the previous value to suppress (default: 5 percent).
- **suppression rtt** – The amount of variation in RTT between the current value and the previous value (default: 30 milliseconds).
- **suppression jitter** – The amount of variation in jitter between the current value and the previous value (default: 30 milliseconds).
- **suppression normalized-rtt** – The amount of variation in normalized RTT between the current value and the previous value (default: 30 milliseconds).

NOTE: See the `sella_nms` Configuration and Common Configuration section of the manual for detailed information on their respective options.

5.1.6.3.5 Policies

Input policies can be used to limit which IPv4 addresses from your network elements are polled. The following standard input policies are included with the distribution:

- **ACCEPT_ALL** – Accepts all addresses.
- **REJECT_ALL** – Rejects all addresses.
- **REJECT_LOOPBACK** – Rejects loopback (127.0.0/8) address space.
- **REJECT_PRIVATE** – Rejects private (RFC1918) address space.
- **REJECT_EDGE_INTERFACE** – Rejects address space that is on the edge of the network (addresses that are not used to connect to other elements).

- **REJECT_BACKBONE_INTERFACE** – Rejects address space that is on the backbone of the network (addresses used to connect to other network elements).

Output policies may be used to specify which conditions generate alarms. The following standard input policies are included with the distribution:

- **PACKET_LOSS** – Alarms on heavy and light packet loss. Heavy packet loss is 90+% over 60 seconds. Light packet loss is defined at 30+% over 60 seconds.
- **TOS_CHANGE** – Alarms if the transmitted ToS, DSCP or IP Precedence return value is different than the transmitted value. This detects elements that have mis-configured CoS settings.

5.1.6.4 snmptrap

5.1.6.4.1 Syntax

```
snmp {  
    /* 'sella_nms' configuration options */  
    config-file path-to-configuration-file;  
    description description;  
    directory path-to-binary-directory;  
    disable;  
    environment environment-variables;  
    executable path-to-executable-file;  
    facility facility;  
    group unix-group;  
    nice niceness;  
    order run-order;  
    pid-file path-to-process-id-file;  
    (restart | no-restart);  
    restart-limit limit;  
    schedule {  
        startup;  
        delay seconds;  
        launch-delay seconds;  
        cron enhanced-vixie-cron-format;  
        second second;  
        minute minute;  
        hour hour;  
        day day;  
        day-of-week day-of-week;  
        month month;  
        day-of-month day-of-month;  
    }  
    stat-file path-to-stats-file;  
    user unix-user;  
  
    /* Common configuration options */  
    trace-options [ ... ];  
    policy {  
        input [ ... ];  
        output [ ... ];  
    }  
  
    /* Unique configuration options */  
    source ip-address;  
    interface interface-name;  
    port ip-port;  
    proxy {  
        convert (v1|v2|none);  
        ip-address;  
        ...  
    }  
    queue-depth count;  
    rate-interval seconds;  
    rate-clean count;  
    exit-after count;  
}
```

5.1.6.4.2 Hierarchy

`monitor { ... }`

5.1.6.4.3 Description

The 'monitor snmptrap' section of the configuration contains options for the SNMP trap monitor module.

5.1.6.4.4 Options

- **source** – The IP address used to determine which interface to listen for incoming SNMP traps on (default: all interfaces). This option overrides the 'interface' option.
- **interface** – The interface to listen for incoming SNMP traps on (default: all interfaces). This option is overridden by the 'source' option.
- **port** – The UDP port on which to listen for incoming SNMP traps on (default: 162).
- **proxy** – Proxy incoming SNMP traps to each IP address specified within the proxy configuration block (default: none).
- **proxy convert** – Convert to the specified version of SNMP trap before the SNMP trap is proxied (default: none). Valid options are v1, v2 and none;
- **queue-depth** – The maximum depth the module's queue may reach before incoming data is dropped (default: 1000).
- **rate-interval** – The interval in seconds to use for calculating the average rate of incoming SNMP traps per second (default: 60 seconds).
- **rate-clean** – The number of increment operations against the rate library before cleaning stale entries (default: 2500).
- **exit-after** – The number of SNMP traps to receive before shutting down the module (default: 0).

NOTE: See the sella_nms Configuration and Common Configuration section of the manual for detailed information on their respective options.

5.1.6.4.5 Policies

Input policies can be used to limit which incoming data is processed by the module. The following standard input policies are included with the distribution:

- **ACCEPT_ALL** – Accepts all addresses.
- **REJECT_ALL** – Rejects all addresses.
- **REJECT_RATE_15** – Rejects source that are sending more than 15 inputs per second. Prevents a single source from overloading the module.
- **REJECT_RATE_30** – Rejects source that are sending more than 30 inputs per second. Prevents a single source from overloading the module.

- **REJECT_AUTH_FAILURE_TRAP** – Rejects authentication failure SNMP traps.
- **AUTHORIZE_TRAP** – Rejects SNMP traps that do not originate from the specified source and do not have the correct community. This policy must be modified to use your SNMP trap sources and community.

Output policies may be used to specify which conditions generate alarms. The following standard input policies are included with the distribution:

- **LINK_TRAP** – Alarms on link up and down SNMP traps.
- **RMON_TRAP** – Alarms on RMON SNMP traps.
- **OSPF_TRAP** – Alarms on common OSPF SNMP traps.
- **BGP_TRAP** – Alarms on common BGP SNMP traps.
- **NETSNMP_TRAP** – Alarms on common Net-SNMP SNMP traps.
- **ARBOR_SP_TRAP** – Alarms on Arbor Service Provider SNMP traps.
- **ARBOR_DOS_TRAP** – Alarms on Arbor DoS SNMP traps.
- **NETSCREEN_TRAP** – Alarms on common Juniper Netscreen related SNMP traps.
- **JUNIPER_CHASSIS_TRAP** – Alarms on Juniper M, T and J series chassis related SNMP traps.
- **JUNIPER_MPLS_TRAP** – Alarms on Juniper M, T and J series MPLS related SNMP traps.
- **EXTREME_HARDWARE_TRAP** – Alarms on Extreme Networks hardware related SNMP traps.
- **EXTREME_SOFTWARE_TRAP** – Alarms on Extreme Networks software related SNMP traps.
- **CATCH_ALL_TRAP** – Sends all SNMP traps which were not caught by a policy to the syslog output module for logging.

5.1.6.5 syslog

5.1.6.5.1 Syntax

```
syslog {  
    /* 'sella_nms' configuration options */  
    config-file path-to-configuration-file;  
    description description;  
    directory path-to-binary-directory;  
    disable;  
    environment environment-variables;  
    executable path-to-executable-file;  
    facility facility;  
    group unix-group;  
    nice niceness;  
    order run-order;  
    pid-file path-to-process-id-file;  
    (restart | no-restart);  
    restart-limit limit;  
    schedule {  
        startup;  
        delay seconds;  
        launch-delay seconds;  
        cron enhanced-vixie-cron-format;  
        second second;  
        minute minute;  
        hour hour;  
        day day;  
        day-of-week day-of-week;  
        month month;  
        day-of-month day-of-month;  
    }  
    stat-file path-to-stats-file;  
    user unix-user;  
  
    /* Common configuration options */  
    trace-options [ ... ];  
    policy {  
        input [ ... ];  
        output [ ... ];  
    }  
  
    /* Unique configuration options */  
    source ip-address;  
    interface interface-name;  
    port ip-port;  
    duplicate-history-max count;  
    duplicate-history-age seconds;  
    duplicate-fuzz-factor value;  
    queue-depth count;  
    rate-interval seconds;  
    rate-clean count;  
    exit-after count;  
}
```

5.1.6.5.2 Hierarchy

```
monitor { ... }
```

5.1.6.5.3 Description

The 'monitor syslog' section of the configuration contains options for the syslog monitor module.

5.1.6.5.4 Options

- **source** – The IP address used to determine which interface to listen for incoming syslog messages on (default: all interfaces). This option overrides the 'interface' option.
- **interface** – The interface to listen for incoming syslog messages on (default: all interfaces). This option is overridden by the 'source' option.
- **port** – The UDP port on which to listen for incoming syslog messages on (default: 514).
- **duplicate-history-max** – The maximum number of previously received syslog messages to keep in the history (default: 25).
- **duplicate-history-age** – The maximum age a previously received syslog message will be kept in the history for comparison against new messages (default: 5 seconds).
- **duplicate-fuzz-factor** – The fuzzy value by which a syslog messages may vary from a previously received message to be considered a duplicate (default: 25). The larger the value, the less similar a syslog message may be and still be considered a duplicate.
- **queue-depth** – The maximum depth the module's queue may reach before incoming data is dropped (default: 1000).
- **rate-interval** – The interval in seconds to use for calculating the average rate of incoming syslog messages per second (default: 60 seconds).
- **rate-clean** – The number of increment operations against the rate library before cleaning stale entries (default: 2500).
- **exit-after** – The number of syslog messages to receive before shutting down the module (default: 0).
- **maximum-entry-delete** – The maximum number of entries to delete from the SNMP discovery information in one command (default: 500 entries).

NOTE: See the `sella_nms` Configuration and Common Configuration section of the manual for detailed information on their respective options.

5.1.6.5.5 Policies

Input policies can be used to limit which incoming data is processed by the module. The following standard input policies are included with the distribution:

- **ACCEPT_ALL** – Accepts all addresses.

- **REJECT_ALL** – Rejects all addresses.
- **REJECT_RATE_15** – Rejects source that are sending more than 15 inputs per second. Prevents a single source from overloading the module.
- **REJECT_RATE_30** – Rejects source that are sending more than 30 inputs per second. Prevents a single source from overloading the module.
- **REJECT_GENERIC_SYSLOG** – Rejects generic low value syslog messages.
- **REJECT_GENERIC_JUNIPER_SYSLOG** – Rejects low value Juniper syslog messages.
- **REJECT_GENERIC_NETSCREEN_SYSLOG** – Rejects generic low value Juniper Netscreen syslog messages.
- **REJECT_GENERIC_CISCO_SYSLOG** – Rejects generic low value Cisco syslog messages.
- **REJECT_GENERIC_EXTREME_SYSLOG** – Rejects generic low value Extreme syslog messages.
- **REJECT_GNERIC_NETSCREEN_SYSLOG** – Rejects generic low value Juniper Netscreen syslog messages.

Output policies may be used to specify which conditions generate alarms. The following standard input policies are included with the distribution:

- **LINUX_SYSLOG** – Alarms on Linux syslog messages.
- **JUNIPER_HEALTH_SYSLOG** – Alarms on Juniper health related syslog messages.
- **JUNIPER_BGP_SYSLOG** – Alarms on Juniper BGP related syslog messages.
- **JUNIPER_VPN_SYSLOG** – Alarms on Juniper VPN and Pseudo-wire related syslog messages.
- **CISCO_GENERIC_SYSLOG** – Alarms on generic Cisco syslog messages.
- **CISCO_BGP_SYSLOG** – Alarms on Cisco BGP related syslog messages.
- **CISCO_7500_HEALTH_SYSLOG** – Alarms on Cisco 7500 series health related syslog messages.
- **CISCO_7500_NOTIFICATION_SYSLOG** – Alarms on Cisco 7500 series notification related syslog messages.
- **CISCO_7200_HEALTH_SYSLOG** – Alarms on Cisco 7200 series health related syslog messages.
- **CISCO_7200_NOTIFICATION_SYSLOG** – Alarms on Cisco 7200 series notification related syslog messages.
- **CATCH_ALL_SYSLOG** – Sends all syslog messages which were not caught by a policy to the syslog output module for logging.

5.1.6.6 state

5.1.6.6.1 Syntax

```
state {  
    /* 'sella_nms' configuration options */  
    config-file path-to-configuration-file;  
    description description;  
    directory path-to-binary-directory;  
    disable;  
    environment environment-variables;  
    executable path-to-executable-file;  
    facility facility;  
    group unix-group;  
    nice niceness;  
    order run-order;  
    pid-file path-to-process-id-file;  
    (restart | no-restart);  
    restart-limit limit;  
    schedule {  
        startup;  
        delay seconds;  
        launch-delay seconds;  
        cron enhanced-vixie-cron-format;  
        second second;  
        minute minute;  
        hour hour;  
        day day;  
        day-of-week day-of-week;  
        month month;  
        day-of-month day-of-month;  
    }  
    stat-file path-to-stats-file;  
    user unix-user;  
  
    /* Common configuration options */  
    trace-options [ ... ];  
    policy {  
        input [ ... ];  
        output [ ... ];  
    }  
  
    /* Unique configuration options */  
    interval seconds;  
    expire-interval seconds;  
    max-delete count;  
    process-element;  
}
```

5.1.6.6.2 Hierarchy

```
monitor { ... }
```

5.1.6.6.3 Description

The 'monitor state' section of the configuration contains options for the state monitor module.

5.1.6.6.4 Options

- **interval** – The interval in seconds at which to process state changes (default: 30 seconds).
- **expire-interval** – The interval in seconds at which to evaluate state entries for removal (default: 3600 seconds).
- **max-delete** – The maximum number of entries to delete from the database in one command (default: 250 entries).
- **process-element** – Enable processing of element state changes (default: not set). This is not recommended, since it will significantly increase the amount of processing this module must perform and offers little to no benefit.

NOTE: See the `sella_nms` Configuration and Common Configuration section of the manual for detailed information on their respective options.

5.1.6.6.5 Policies

Input policies can be used to limit which incoming data is processed by the module. The following standard input policies are included with the distribution:

- **ACCEPT_ALL** – Accepts all addresses.
- **REJECT_ALL** – Rejects all addresses.

Output policies may be used to specify which conditions generate alarms. The following standard input policies are included with the distribution:

- **UPDATE_STATE** – Updates state changes for all included modules.
- **EXPIRE_STATE** – Expire and removes stale and non-referenced state entries.

5.1.7 output

The 'output' section of the configuration file contains options for the output subsystem and all output modules.

The output modules are responsible for generating output for various purposes such as making alarms available for display on a GUI, logging to the servers syslog, or emailing an administrator.

5.1.7.1 Syntax Overview

The following block details the top level configuration syntax for the 'output' section of the configuration:

```

output {
  disable;
  nice niceness;
  (restart | no-restart);
  schedule {
    startup;
    delay seconds;
    launch-delay seconds;
    cron enhanced-vixie-cron-format;
    second second;
    minute minute;
    hour hour;
    day day;
    day-of-week day-of-week;
    month month;
    day-of-month day-of-month;
  }
  gui { ... }
  email { ... }
  syslog { ... }
  module-name { ... }
}

```

5.1.7.2 output

5.1.7.2.1 Syntax

```

output {
  disable;
  schedule { ... }
  gui { ... }
  email { ... }
  syslog { ... }
  module-name { ... }
}

```

5.1.7.2.2 Hierarchy

top

5.1.7.2.3 Description

The ‘output’ section of the configuration contains options for the monitor modules.

5.1.7.2.4 Options

- **disable** – Disable (don’t run) the module (default: not set).
- **nice** – The niceness at which to run this module (default: 5).
- **restart** – Restart the module if it stops running (default: active).
- **no-restart** – Do not restart the module if it stops running (default: not active).
- **schedule** – The schedule at which to run the module (default: none). See the ‘system default schedule’ portion of the *Modules* section of the manual for more information.

- **gui** – The configuration options for the GUI output module (default: use module specific defaults).
- **email** – The configuration options for the email output module (default: use module specific defaults).
- **syslog** – The configuration options for the syslog output module (default: use module specific defaults).
- **module-name** – The configuration options for the 'module-name' module (default: use module specific defaults). Module names are not hard coded and any unique name maybe used to group the configuration options for the module.

5.1.7.3 gui

5.1.7.3.1 Syntax

```
gui {
    /* 'sella_nms' configuration options */
    config-file path-to-configuration-file;
    description description;
    directory path-to-binary-directory;
    disable;
    environment environment-variables;
    executable path-to-executable-file;
    facility facility;
    group unix-group;
    nice niceness;
    order run-order;
    pid-file path-to-process-id-file;
    (restart | no-restart);
    restart-limit limit;
    schedule {
        startup;
        delay seconds;
        launch-delay seconds;
        cron enhanced-vixie-cron-format;
        second second;
        minute minute;
        hour hour;
        day day;
        day-of-week day-of-week;
        month month;
        day-of-month day-of-month;
    }
    stat-file path-to-stats-file;
    user unix-user;

    /* Common configuration options */
    trace-options [ ... ];
    policy {
        input [ ... ];
        output [ ... ];
    }

    /* Unique configuration options */
    interval seconds;
    expire-interval seconds;
    queue-depth count;
    max-delete count;
}
```

5.1.7.3.2 Hierarchy

```
output { ... }
```

5.1.7.3.3 Description

The 'output gui' section of the configuration contains options for the GUI output module.

This module accepts alerts from other module's and organizes the alerts for presentation in a GUI.

5.1.7.3.4 Options

- **interval** – The interval in seconds at which new alerts are read from the database (default: 15 seconds).
- **expire-interval** – The interval in seconds at which to evaluate entries for expiration and removal (default: 120 seconds).
- **queue-depth** – The maximum depth the module's queue may reach before new data will be accepted (default: 250).
- **max-delete** – The maximum number of entries to delete from the database in one command (default: 250 entries).

NOTE: See the `sella_nms` Configuration and Common Configuration section of the manual for detailed information on their respective options.

5.1.7.3.5 Policies

Input policies can be used to limit which alerts are processed. The following standard input policies are included with the distribution:

- **ACCEPT_ALL** – Accepts all alerts.
- **REJECT_ALL** – Rejects all alerts.

Output policies may be used to manipulate alerts. The following standard output policies are included with the distribution:

- **UPDATE_GUI_EVENT** – Updates GUI alert events. Updates include handling duplicate alerts, recovery alerts clearing a failure and a failure alert clearing a recovery.
- **EXPIRE_GUI_STORAGE** – Expires and removes GUI alerts.

5.1.7.4 email

5.1.7.4.1 Syntax

```
email {
    /* 'sella_nms' configuration options */
    config-file path-to-configuration-file;
    description description;
    directory path-to-binary-directory;
    disable;
    environment environment-variables;
    executable path-to-executable-file;
    facility facility;
    group unix-group;
    nice niceness;
    order run-order;
    pid-file path-to-process-id-file;
    (restart | no-restart);
    restart-limit limit;
    schedule {
        startup;
        delay seconds;
        launch-delay seconds;
        cron enhanced-vixie-cron-format;
        second second;
        minute minute;
        hour hour;
        day day;
        day-of-week day-of-week;
        month month;
        day-of-month day-of-month;
    }
    stat-file path-to-stats-file;
    user unix-user;

    /* Common configuration options */
    trace-options [ ... ];
    policy {
        input [ ... ];
        output [ ... ];
    }

    /* Unique configuration options */
    interval seconds;
    queue-depth count;
    default {
        server [ (hostname | ip-address) ... ];
        port tcp-port;
        timeout seconds;
        from [email-address ... ];
        reply-to [ email-address ... ];
        to [ email-address ... ];
        cc [ email-address ... ];
        bcc [ email-address ... ];
        subject string;
        body string;
        priority value;
    }
}
```

```

    length value;
}
alias {
    alias-name {
        server [ (hostname | ip-address) ];
        port tcp-port;
        timeout seconds;
        from [ email-address ... ];
        reply-to [ email-address ... ];
        to [ email-address ... ];
        cc [ email-address ... ];
        bcc [ email-address ... ];
        subject string;
        body string;
        priority value;
        length value;
    }
    ...
}
}

```

5.1.7.4.2 Hierarchy

```
output { ... }
```

5.1.7.4.3 Description

The 'output email' section of the configuration contains options for the email output module.

This module accepts alerts from other module's and delivers email via SMTP. This module is a SMTP client and does not rely on a local mail server to deliver email for it.

5.1.7.4.4 Options

- **interval** – The interval in seconds at which new alerts are read from the database (default: 15 seconds).
- **queue-depth** – The maximum depth the module's queue may reach before new data will be accepted (default: 250).
- **default** – The default settings to use for delivering email (default: none). This settings will also be used as the default settings for all specified aliases.
- **default server** – The set of servers to deliver email to (default: localhost). The module will attempt to deliver email via SMTP to each server listed in the order they are listed until one succeeds or the end of the list is reached.
- **default port** – The TCP port to use when connecting to the server (default: 25).
- **default timeout** – The number of seconds to give a SMTP server to accept an incoming connection before moving onto the next (default: 2 seconds).
- **default from** – The set of email address to source email from (default: none).

- **default reply-to** – The set of email address to have the receiver reply to (default: none).
- **default to** – The set of email address to deliver email to (default: none).
- **default cc** – The set of email address to carbon copy (CC) email to (default: none).
- **default bcc** – The set of email address to blind carbon copy (BCC) email to (default: none).
- **default subject** – The subject of the email (default: value of 'brief' from the policy output).
- **default body** – The body of the email (default: value of 'detail' from the policy output).
- **default priority** – The priority of the email (default: 0).
- **default length** – The maximum length the email may use (default: 0). If an email is longer than the specified length, it will be broken into parts and emailed separately. This feature is designed to accommodate pagers that can accept only limited amounts of text per message.
- **alias** – Contains a set of available aliases that may be referenced within a policy to deliver email that doesn't use the settings from the 'output email default' configuration block. To use an alias within a policy, set a variable named 'alias' to the name of the desired alias within the policy's 'then set' block.
- **alias alias-name** – The name of an alias which may be referenced within a policy. Each alias may contain any of the options found within the 'output email default' configuration block. Any setting not specified within the alias will default to the setting from the 'output email default' configuration block.

NOTE: See the `sella_nms` Configuration and Common Configuration section of the manual for detailed information on their respective options.

5.1.7.4.5 Policies

Input policies can be used to limit which alerts are processed. The following standard input policies are included with the distribution:

- **ACCEPT_ALL** – Accepts all alerts.
- **REJECT_ALL** – Rejects all alerts.

Output policies may be used to manipulate alerts. The following standard input policies are included with the distribution:

- **EMAIL_NOTIFICATION** – Uses the default settings specified in the 'output email default' section of the configuration to email alerts.

- **EMAIL_NOTIFICATION_OFFHOURS** – Policy that emails alerts only during off hours using the default settings specified in the 'output email default' section of the configuration.
- **EMAIL_NOTIFICATION_ROTATION** – Policy that emails different email-addresses each day of the week. This policy must be edited prior to use.
- **EMAIL_HOLIDAY_SCHEDULE** – Policy that emails different users based on calendar dates. This policy must be edited prior to use.
- **EMAIL_WEEKEND_SCHEDULE** – Policy that emails different users based on calendar dates. This policy must be edited prior to use.

5.1.7.5 syslog

5.1.7.5.1 Syntax

```
syslog {  
    /* 'sella_nms' configuration options */  
    config-file path-to-configuration-file;  
    description description;  
    directory path-to-binary-directory;  
    disable;  
    environment environment-variables;  
    executable path-to-executable-file;  
    facility facility;  
    group unix-group;  
    nice niceness;  
    order run-order;  
    pid-file path-to-process-id-file;  
    (restart | no-restart);  
    restart-limit limit;  
    schedule {  
        startup;  
        delay seconds;  
        launch-delay seconds;  
        cron enhanced-vixie-cron-format;  
        second second;  
        minute minute;  
        hour hour;  
        day day;  
        day-of-week day-of-week;  
        month month;  
        day-of-month day-of-month;  
    }  
    stat-file path-to-stats-file;  
    user unix-user;  
  
    /* Common configuration options */  
    trace-options [ ... ];  
    policy {  
        input [ ... ];  
        output [ ... ];  
    }  
  
    /* Unique configuration options */  
    interval seconds;  
    queue-depth count;  
}
```

5.1.7.5.2 Hierarchy

```
output { ... }
```

5.1.7.5.3 Description

The 'output syslog' section of the configuration contains options for the syslog output module.

This module accepts alerts from other module's and logs them to the server's syslog daemon. Messages from SellaNMS itself are not logged via this module and is handled individually by each module.

5.1.7.5.4 Options

- **interval** – The interval in seconds at which new alerts are read from the database (default: 15 seconds).
- **queue-depth** – The maximum depth the module's queue may reach before new data will be accepted (default: 250).

NOTE: See the `sella_nms` Configuration and Common Configuration section of the manual for detailed information on their respective options.

5.1.7.5.5 Policies

Input policies can be used to limit which alerts are processed. The following standard input policies are included with the distribution:

- **ACCEPT_ALL** – Accepts all alerts.
- **REJECT_ALL** – Rejects all alerts.

Output policies may be used to manipulate alerts. There are no output policies included with the distribution.

5.1.8 internal

The 'internal' section of the configuration contains options for the internal subsystem and internal modules.

5.1.8.1 Syntax Overview

The following block details the complete configuration syntax for the 'internal' section of the configuration:

```

output {
  disable;
  nice niceness;
  (restart | no-restart);
  schedule {
    startup;
    delay seconds;
    launch-delay seconds;
    cron enhanced-vixie-cron-format;
    second second;
    minute minute;
    hour hour;
    day day;
    day-of-week day-of-week;
    month month;
    day-of-month day-of-month;
  }
  tidy { ... }
  module-name { ... }
}

```

5.1.8.2 internal

5.1.8.2.1 Syntax

```

internal {
  disable;
  schedule { ... }
  tidy { ... }
  module-name { ... }
}

```

5.1.8.2.2 Hierarchy

[top](#)

5.1.8.2.3 Description

The ‘output’ section of the configuration contains options for the monitor modules.

5.1.8.2.4 Options

- **disable** – Disable (don’t run) the module (default: not set).
- **nice** – The niceness at which to run this module (default: 5).
- **restart** – Restart the module if it stops running (default: active).
- **no-restart** – Do not restart the module if it stops running (default: not active).
- **schedule** – The schedule at which to run the module (default: none). See the ‘system default schedule’ portion of the *Modules* section of the manual for more information.
- **tidy** – The configuration options for the tidy internal module (default: use module specific defaults).

5.1.8.3 tidy

5.1.8.3.1 Syntax

```
tidy {  
    /* 'sella_nms' configuration options */  
    config-file path-to-configuration-file;  
    description description;  
    directory path-to-binary-directory;  
    disable;  
    environment environment-variables;  
    executable path-to-executable-file;  
    facility facility;  
    group unix-group;  
    nice niceness;  
    order run-order;  
    pid-file path-to-process-id-file;  
    (restart | no-restart);  
    restart-limit limit;  
    schedule {  
        startup;  
        delay seconds;  
        launch-delay seconds;  
        cron enhanced-vixie-cron-format;  
        second second;  
        minute minute;  
        hour hour;  
        day day;  
        day-of-week day-of-week;  
        month month;  
        day-of-month day-of-month;  
    }  
    stat-file path-to-stats-file;  
    user unix-user;  
  
    /* Common configuration options */  
    trace-options [ ... ];  
    policy {  
        input [ ... ];  
        output [ ... ];  
    }  
  
    /* Unique configuration options */  
    interval seconds;  
    output-age seconds;  
}
```

5.1.8.3.2 Hierarchy

```
internal { ... }
```

5.1.8.3.3 Description

The 'internal tidy' section of the configuration contains options for the tidy internal module.

This module is responsible for removing stale data which is not specific to another module from the database.

5.1.8.3.4 Options

- **interval** – The interval in seconds at which the database is checked for data that should be removed (default: 45 seconds).
- **output-age** – The maximum age in seconds that an entry may remain within the databases output table before it is removed (default: 300 seconds).

NOTE: See the sella_nms Configuration and Common Configuration section of the manual for detailed information on their respective options.

5.1.8.3.5 Policies

Input polices can be used to limit which data is processed. There are no input policies included with the distribution.

Output policies may be used to limit which data is processed. There are no output policies included with the distribution.

6 GUI

This section of the manual covers SellaNMS Web GUI.

7 Policy Framework

7.1 Overview

The policy framework is one of the major components of SellaNMS. It is responsible for providing a flexible high performance method of processing data in and out of the systems modules.

The policy framework is broken into three major components; policy-statements, format-statements and map-statements. Each of these components are contained with the 'policy-options' section of the configuration.

7.2 Syntax Overview

The following block details the top level configuration syntax for the 'policy-options' section of the configuration:

```
policy-options {  
  policy-statement policy-statement-name { ... }  
  ...  
  format-statement format-statement-name { ... }  
  ....  
  map-statement map-statement-name { ... }  
  ...  
}
```

7.3 Variable Substitution

The policy framework has the ability to dynamically substitute variables and values in a *term's from* or *then* clause using the data set being processed by the policy. This feature works for both *policy statements* and *map statements*.

To perform a substitution, wrap a variable name included within the policy data set in the "@" symbol. Before the clause is evaluated, the variable will be replaced with the value matching the variable within the policy data set.

Below is an example assuming that the policy data set contains a variable of 'prefix':

```

policy-options {
  policy-statement EXAMPLE_POLICY {
    term EXAMPLE_TERM {
      from {
        source network;
        prefix [ 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 ];
      }
      then {
        output {
          gui {
            brief "This is a brief information about prefix @prefix@";
            detail "This is the detailed information about prefix @prefix@";
            channel default;
          }
        }
        accept;
      }
    }
    apply-format EXAMPLE_FORMAT;
  }
}

```

In addition to substitution, an operator can be applied to a variable or value by prepending a keyword followed by a colon to the front of it. For example, if the policy data set contained the SNMP OID for “risingAlarm”, you would specify “snmp:risingAlarm” to resolve the SNMP OID from MIBs loaded into SellaNMS. This feature can be combined with variable substitution.

Below is an example of using this feature within a policy:

```

policy-options {
  policy-statement EXAMPLE_POLICY {
    term EXAMPLE_TERM {
      from {
        protocol SNMP trap;
        event snmp:risingAlarm;
      }
      then {
        output {
          gui {
            brief "Received RMON alarm @snmp:eventDescription@";
            channel rmon;
          }
        }
      }
      accept;
    }
  }
  apply-format EXAMPLE_FORMAT;
}

```

7.4 Policy Statements

Policy statements are a flexible method of configuring how a module processes data. In its basic form, a policy is a simple program that is run from top to bottom evaluating variables and possibly taking action(s). Policies can be chained together to produce more sophisticated programs.

A *policy statement* is built from one or more *terms* and an *apply-format statement*.

7.4.1 term

Each *policy statement term* may have a *from* and/or *then* clause, which define what the *term* does.

7.4.1.1 from clause

A *term's from* clause is a set of conditions that are evaluated by the policy framework. If the *from* clause evaluates true, the *term's then* clause will be executed, otherwise the policy framework will move to the next *term*.

The *from* clause conditions are a list of variables paired with a value or set of values. The method of comparison between the variable and the value(s) is defined within the applied format statement. See the *format statement* section of the manual for more details.

Below is an example of a *from* clause:

```

policy-options {
  policy-statement EXAMPLE_POLICY {
    term EXAMPLE_TERM {
      from {
        source network;
        prefix [ 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 ];
      }
      then {
        accept;
      }
    }
    apply-format EXAMPLE_FORMAT;
  }
}

```

7.4.1.2 then clause

A *term's then* clause defines what work will be done if the *term's from* clause evaluates true. The *then* clause may contain the following constructs:

- set
- output
- evaluate
- apply-map
- action

7.4.1.2.1 set

A *set* block, allows the *term* to set (or overwrite) variable(s) within the policy's data set. Below is an example of a *set* block within a *term's then* clause:

```

policy-options {
  policy-statement EXAMPLE_POLICY {
    term EXAMPLE_TERM {
      from {
        source    network;
        prefix    [ 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 ];
      }
      then {
        /* Set example-var1 to 100 and example-var2 to "nihon no tsurugi". */
        set {
          example-var1 100;
          example-var2 "nihon no tsurugi";
        }
        accept;
      }
    }
    apply-format EXAMPLE_FORMAT;
  }
}

```

7.4.1.2.2 output

An *output* block, allows the *term* to send a message to the output subsystem. Each block listed within the *output* block is a separate message to a different module within the output subsystem. The following variables can be specified within a block. Most of these variables are already defined within the policies data set and will be automatically passed onto the output subsystem.

- **timestamp** – Timestamp for when the message was received.
- **element** – The element database id for this set of data.
- **interface** – The interface database id for this set of data.
- **ip** – The ip database id for this set of data.
- **source** – The source of this set of data.
- **module** – The output module to send this message to.
- **protocol** – The protocol that originated this set of data.
- **channel** – The channel to deliver this message to.
- **state** – The state of the specified element, interface or ip.
- **brief** – A brief description of the message for display.
- **detail** – A detailed description of the message for display.

Below is an example of an *output* block within a *term's then* clause:

```

policy-options {
  policy-statement EXAMPLE_POLICY {
    term EXAMPLE_TERM {
      then {
        output {
          gui {
            brief "This is brief information about prefix @prefix@";
            detail "This is detailed information about prefix @prefix@";
            channel default;
          }
        }
      }
      accept;
    }
  }
  apply-format EXAMPLE_FORMAT;
}

```

7.4.1.2.3 evaluate

An *evaluate* block, allows the *then* clause to evaluate a sub-policy. This is generally used only in special cases such as resolving BGP error codes to an error message.

Below is an example of a policy with an *evaluate* block:

```

policy-options {
  policy-statement EXAMPLE_POLICY {
    term EXAMPLE_TERM {
      then {
        evaluate {
          policy-statement EXAMPLE_SUB-POLICY {
            sub-var 100;
          }
        }
      }
      accept;
    }
  }
  apply-format EXAMPLE_FORMAT;
}

```

7.4.1.2.4 apply-map

An *apply-map statement*, allows the *then* clause to apply one or more maps which are specific to the *term*. This is useful when a *term* needs some additional information which is not available within the policy data set.

Below is an example of a *apply-map statement* a *term's then* clause:

```

policy-options {
  policy-statement EXAMPLE_POLICY {
    term EXAMPLE_TERM {
      then {
        apply-map EXAMPLE_MAP;
        accept;
      }
    }
    apply-format EXAMPLE_FORMAT;
  }
}

```

7.4.1.2.5 action

An *action*, allows the *term* to adjust how the policy framework continues to process the policy. The following can be specified for an action:

- **accept** – Stop evaluating and accept the policy data set. The module will receive a return value of accept.
- **reject** – Stop evaluating and reject the policy data set. The module will receive a return value of reject.
- **next-input** – Stop evaluating and request the next input (data set) from the module.
- **next-term** – Stop evaluating the current term and move to next term. If no terms remain, move to the next policy. If no policy remains, return the default action.
- **next-policy** – Stop evaluating the current policy and move to the next policy. If no policy remains, return the default action.

Below is an example of a *then* clause using an action:

```

policy-options {
  policy-statement EXAMPLE_POLICY {
    term EXAMPLE_TERM {
      then {
        reject;
      }
    }
    apply-format EXAMPLE_FORMAT;
  }
}

```

7.4.1.2.6 apply-format

The *apply-format* statements sets the format to use for the policy. The format specifies the rules that the policy adheres to. See the *format statements* section of the manual for more information.

Below is an example of the *apply-format* statement within a policy:

```

policy-options {
  policy-statement EXAMPLE_POLICY {
    term EXAMPLE_TERM {
      then {
        accept;
      }
    }
  }
  apply-format EXAMPLE_FORMAT;
}

```

7.5 Format Statements

Format statements are a flexible method of configuring a policy statement. Format statements specify which variables within the policy data set may be used, which variables are required, how the corresponding policy data set value will be compared to the policy value, and what policy value may be specified. Additionally, a format statement can specify a map statement, which will be executed before the policy is evaluated.

A format statement is built from one of more *terms*.

7.5.1 term

Each *format statement term* may have a *from* and/or *then* clause, which details what the *term* defines.

7.5.1.1 from clause

A *format statement term's from* clause specified which variables may be used within a *policy statement term's from* clause.

A variable can be specified using a 'variable' block, or a 'group' block. Each block type can specify the following:

- **required** – If the variable or group of variables are required to be specified in the *policy statement term's from* clause. Set to either 'yes; or 'no'. Defaults to 'no'.
- **type** – The type of data stored within this variable. Set to 'string', 'integer' or 'prefix'. Defaults to 'string'.
- **function** – The function used to compare the policy data set value to the policy value. Set to 'equal-to', 'not-equal', 'at-most', 'at-least', 'more-than', 'less-than' or 'ignore'. Defaults to 'equal-to'.
- **value** – The set of acceptable values that may specified for the policy value. Default allows any policy value.
- **variable** – The set of acceptable variables to allow within the *policy statement term's from* clause. Only applies to 'group' blocks.

Below is an example of a format-statement with both a 'variable' block and a 'group' block.

```
policy-options {
  format-statement EXAMPLE_FORMAT {
    term EXAMPLE_TERM {
      from {
        variable source { /* 'variable' block */
          required no;
          type string;
          function equal-to;
          value [ network storage ];
        }
        variable prefix { /* 'variable' block */
          required yes;
          type prefix;
          function equal-to;
        }
        group EXAMPLE_GROUP { /* 'group' block */
          required no;
          type integer;
          function equal-to;

          variable element;
          variable interface;
          variable ip;
        }
      }
      then {
        action {
          default accept;
          value [ accept reject ];
        }
      }
    }
  }
}
```

7.5.1.2 then clause

A *format statement term's* then clause specifies the default *action* and which *actions* may be specified.

Below is an example of a *format statement term's* then clause:

```

policy-options {
  format-statement EXAMPLE_FORMAT {
    term EXAMPLE_TERM {
      from {
        variable source {
          required no;
          type string;
          function equal-to;
          value [ network storage ];
        }
      }
      then {
        action {
          default accept;
          value [ accept reject ];
        }
      }
    }
  }
}

```

7.6 Map Statements

A *map statement* defines how to map data from an external source into a policy's data set.

A *map statement* is built from one or more *terms*.

7.6.1 term

Each *map statement term* may have a *from* and/or *then* clause, which details what the *term* defines.

7.6.1.1 from clause

A *map statements term's from* clause is a set of conditions that are evaluated by the policy framework. If the *from* clause evaluates true, the *term's then* clause will be executed, otherwise the policy framework will move to the next *map statement term*.

The *from* clause's conditions are evaluated by verifying that certain variables either exist or do not exist within the policy's data set. These conditions can be evaluated using the following:

- **defined** – Variables specified must exist within the policy data set.
- **defined-except** – Variables specified must not exist within the policy data set.

Below is an example of a *map statement term's from* clause:

```

policy-options {
  map-statement EXAMPLE_MAP {
    term EXAMPLE_TERM1 {
      from {
        defined source;
        defined-except element-id;
      }
      then {
        query {
          database {
            sql "SELECT element.id AS element_id FROM element, interface, ip,
ip_data WHERE element.id = interface.element AND interface.id = ip.interface
AND ip_data.ip = ip.id AND ip_data.value = '@source@' LIMIT 1";
          }
        }
        set {
          element-id @element_id@;
        }
      }
    }
  }
}

```

7.6.1.2 then clause

A *map statement term's then* clause defines what work will be done if the *term's from* clause evaluates true. The *then* clause may contain the following constructs:

- query
- set

7.6.1.2.1 query

A *query* block allows the *term* to request information from a data source and apply that new data into the policy's data set. Currently, the only supported data source is the *database*, but this will be expanded in a future release.

Below is an example of a *query* against the *database* within a *map statement term's then* clause:

```

policy-options {
  map-statement EXAMPLE_MAP {
    term EXAMPLE_TERM1 {
      from {
        defined source;
        defined-except element-id;
      }
      then {
        query {
          database {
            sql "SELECT element.id AS element_id FROM element,
interface, ip, ip_data WHERE element.id = interface.element AND
interface.id = ip.interface AND ip_data.ip = ip.id AND ip_data.value =
'@source@' LIMIT 1";
          }
        }
        set {
          element-id @element_id@;
        }
      }
    }
  }
}

```

7.6.1.2.2 set

A *set* block allows the *term* to set (or overwrite) variable(s) within the policy's data set.

Below is an example of a *set* block within a *map statement term's then* clause:

```

policy-options {
  map-statement EXAMPLE_MAP {
    term EXAMPLE_TERM1 {
      from {
        defined source;
        defined-except element-id;
      }
      then {
        query {
          database {
            sql "SELECT element.id AS element_id FROM element, interface, ip,
ip_data WHERE element.id = interface.element AND interface.id = ip.interface
AND ip_data.ip = ip.id AND ip_data.value = '@source@' LIMIT 1";
          }
        }
        set {
          element-id @element_id@;
        }
      }
    }
  }
}

```

8 Troubleshooting

This section of the manual covers several methods of troubleshooting the configuration and behavior of SellaNMS.

8.1 Logging

By default, all debug output from SellaNMS and its modules are logged to your servers syslog daemon. The default syslog configuration on most servers sends this output to the file `/var/log/messages`. When you're initially setting up the SellaNMS, it can be helpful to monitor the log file via the 'tail' command. As new input to the file arrives, it will be displayed on your terminal. You can run 'tail', as seen below (ctrl-c to exit):

```
$ tail -f /var/log/messages
```

You can also display this output on your terminal by running SellaNMS or a module in the foreground. This can be done by specifying the '-f' argument on the command line, as seen below:

```
$ ./sella_nms -f
```

8.2 trace-options

SellaNMS has configurable debugging output via a feature called 'trace-options'. When this feature is not active, only significant events are logged.

There are three ways of triggering output from the 'trace-options' feature. The first way is by specifying the 'trace-options' keyword within the "sella_nms.conf" configuration file as a keyword for a module or under the system section. The second way is by specifying a debug level on the command line via the "-d" argument. Additionally, the debug level can be increased while a module is running by sending the signal USR1. If the debug level hits 9, it will wrap back to level 0.

A set of options can be specified for the 'trace-options' keyword to provide different output. Any combination of options can be used to provide the output desired. Below is a list of the available trace-options and a description of the output they produce.

- **debug** – Logs additional debug output. This typically includes information useful for understanding the operation of a module and provides notifications of conditions that would not normally be logged.
- **config** – Logs the module's configuration options with the

values being used for each. This is useful to see the default options and values, or to troubleshoot when a module has chosen to use a different value than you specified (ie: invalid or out of range value was provided).

- **policy** – Enables the ‘policy-in’ and ‘policy-out’ options.
- **policy-in** – Logs the input provided to a policy evaluated by the module. This is useful for troubleshooting policies.
- **policy-out** – Logs the output from an evaluated policy. This is useful for troubleshooting policies.
- **policy-sql** – Logs any SQL query performed while evaluating a policy and the number of records returned from it.
- **policy-map** – Logs the output from any map-statement used while evaluating a policy.
- **sql** – Logs any SQL queries performed by the module. Usually used only by developers.
- **thread** – Logs the creation and destruction of threads within the module. Usually used only by developers.
- **timer** – Logs any timed algorithms used with the module. Usually used only by developers.
- **internal** – Logs low to medium volume output for data structure and variables. Usually used only by developers.
- **dump** – Logs high volume output for data structure and variables. Usually used only by developers.
- **all** – Enables all trace-options except ‘internal’ and ‘dump’.
- **developer** – Enables all trace-options including ‘internal’ and ‘dump’.

Below is an example of enabling the ‘debug’, ‘config’ and ‘thread’ ‘trace-options’ for the *icmp montor* module within the “sella_nms.conf” configuration file.

```
monitor {  
  icmp {  
    trace-options [ debug config thread ];  
    description "Polls and processes ICMP reachability events";  
    executable "mon.icmpd";  
  
    policy {  
      input [ REJECT_LOOPBACK ];  
      output [ PACKET_LOSS ];  
    }  
  }  
}
```

9 Advanced Topic

This section of the manual covers topics which fall outside the scope of normal users.

9.1 Distributed Installs

SellaNMS was designed to support distributed installs. It can be configured to run in a distributed setup across multiple servers to provide additional scalability. Its design also automatically takes advantage of servers with multiple CPUs.

There are many combinations of distributed setups that can be implemented. These setups can be configured to run on hundreds of servers, although most typical distributed setups are only with a few servers. See the *Limitations* section of the manual in the *Appendix* before implementing any of these setups.

The simplest and most common distributed setup is running SellaNMS on a separate server from the MySQL database. To configure this setup, simply run MySQL on a separate server and configure SellaNMS to connect to that remote server rather than localhost. This can be configured within the 'system storage' section of the configuration.

```
system {  
  storage {  
    primary {  
      driver mysql;  
      server 192.168.0.56;  
      port 3306;  
      database sella_nms;  
      username sella_nms;  
      password secret;  
    }  
  }  
}
```

The second most common distributed setup includes separating the SellaNMS *monitor* modules from the *discovery*, *output* and *internal* modules. To implement this setup, install SellaNMS on two servers and then *disable* the *monitor* modules on one server and *disable* the *discovery*, *output* and *internal* modules on the other. Optionally, the MySQL database could be run on a third server.

A distributed setup which networks with large volumes of incoming SNMP traps and syslog messages may find useful is running multiple servers for *syslog* and *SNMP trap monitor* modules. To implement this setup, install SellaNMS on as many servers as needed and run the *syslog* and *SNMP trap monitor* modules on all but one of the servers. The last server will run the remaining modules. Your network elements would then need to be configured to deliver SNMP traps and syslog messages to one of the available *syslog* and/or *SNMP trap* servers. Optionally, the MySQL database could be run on a separate server.

NOTE: With any distributed setup, take care not to run more than one set of discovery, output or internal modules. If you chose to run more than one icmp monitor module, use an input policy to prevent overlap of the prefixes monitored. Finally, when running multiple syslog and snmptrap monitor modules, sending the same SNMP trap or syslog message to different servers will result in duplicate alarms. Violating any of these rules, may lead to unexpected behavior from SellaNMS.

10 Appendix

This section of the manual contains additional information related to SellaNMS.

10.1 Limitations

Current limitations of SellaNMS include:

- **Portability** – Additional work needs to be done to improve the portability of the POSIX thread code.
- **SQL Syntax** – The SQL syntax in portion of SellaNMS may use MySQL specific SQL syntax. Additional work may be needed to support other database back-ends via libdbi. The system has only been tested with MySQL.
- **Database Fail-over** – The use of the database fail-over features of SellaNMS has not been tested and may not function properly.
- **Distributed Installs** – The use of duplicate modules against the same database is not tracked or account for by SellaNMS. Because of this, running duplicate discovery, output or internal modules may result in unexpected behavior. Additionally, running duplicate monitor modules to monitor the same devices may result in unexpected behavior (this can be avoided with proper configuration).

10.2 Licenses

10.2.1 GNU General Public License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, 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 Library 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