Pandora v1.2

Installation Manual

***Pandora* v1.2Installation Manual**

First Edition(v1.2) Edition

Published November 19th, 2006

Revision History

Revision 1.0 11 Sept 2006
Submitted.
Revision 1.1 20 Sept 2006
First draft for review.
Revision 1.2 21 Nov 2006
Last review.

Table of Contents

Chapter 1. Introduction to Pandora FMS

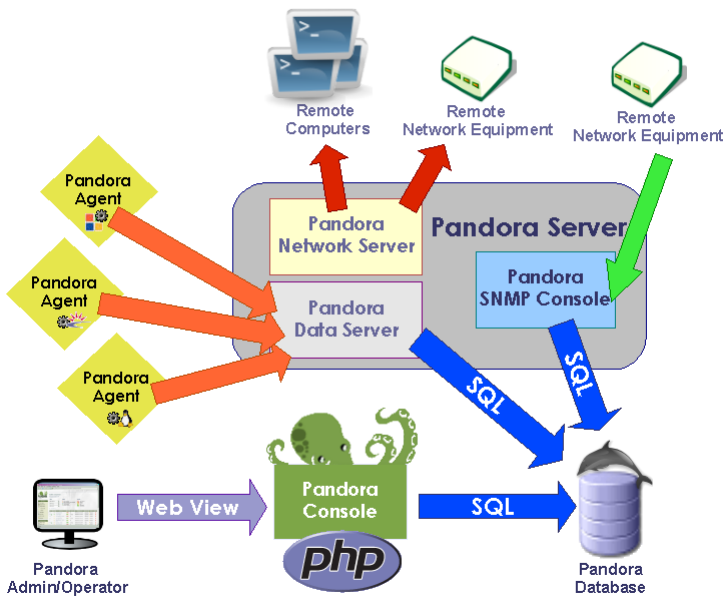
1.1. Pandora. The Free Monitoring System

Pandora FMS is a monitoring application to watch systems and applications. Pandora allows to know the status of any element of your bussiness systems. Pandora watch for your hardware, your software, your multilayer system and of course your Operating System. Pandora could detect a network interface down and the movement of any value of the NASDAQ new technology market. If you want, Pandora could sent a SMS message when your systems fails... or when Google value low below US$ 330.

Pandora FMS will adjust, like an octopus, to your systems and requirements, because it has been designed to be open, modular, multiplattform and easy to customize.

1.2. Introducing Pandora FMS.

Pandora is a monitoring tool that allows a system administrator to visually analise the status and efficiency of Operating Systems, Servers, Applications and Hardware Systems - such as firewalls, proxies, databases, Web servers, tunnelling servers, routers, switches, processes, services, remote access servers, etc. - all integrated into an open and distributed architecture. Pandora can be implemented over any operating system, with specific agents for each platform. Pandora can also monitor any TCP/IP hardware system, as load balancers, routers, switches, printers, etc.



Pandora architecture is formed of four main components:

- *Web Console*: Pandora's user interface. The user controls and operates the system with it. Several Web consoles can be implemented in a single system. The Web console is written in PHP, and it is over a database and a Web server. It is compatible with any platform - GNU/Linux, Solaris, Win2000, AIX, etc.T official supported platform is GNU/Linux, though

  The console permits the user to control the status of the agents, view statistical information, generate graphs and data tables, keep a system incident control,moreover it is able to generate reports and change the alerts, agents, and user profile settings.

- *Server*: In Pandora 1.2 there are three different servers:

  The core server is the receptor of the data packages and generates the alerts - it is the brain of the system. Several servers can work alongside for larger systems. The core server accesses Pandora database, which is shared with the Web server, and stores the processed data packages. Server executes as daemon, and processes the packages stored in its file system. Data is generated by the system agents. Despite the server's low system resources comsumption and simple installation and operation, the core server is the most critical element of the system. The core server receives and processes the produced data, and fires the alerts and the events.

  The Network Servers monitorize remote systems using network resources like ICMP, TCP, UDP or SNMP Queries. Network Servers are acting itself like "Network Agents". This server fires the alerts and the events for this modules.

  The SNMP Server receives and processes the snmp traps, and fires the alerts associated to it.

- *Central Database*: At the moment the system only supports MySQL. The central database keeps all the information Pandora needs to work - agent data, settings, user information, incidents, system settings, etc. The system can use a MySQL cluster to store the information, or a High Availability (HA) solution for larger sytems.

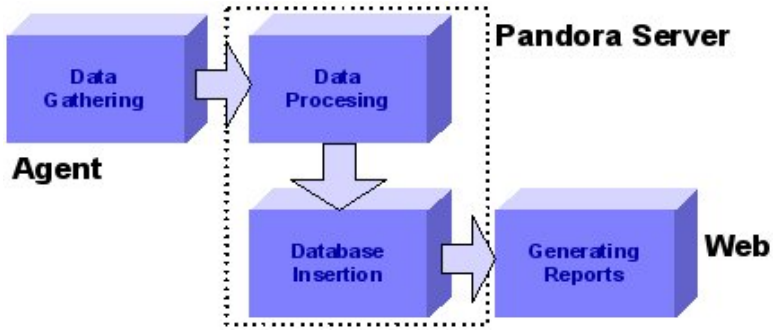
  This database can work with any of the platform officially supported by MySQL. Pandora can be implemented with MySQL versions from 3.0 to 5.0, although the latest is recommended.

- *Pandora Agents*: They collect all the system's data. They are executed in each local system, although they can also collect remote information by intalling monitoring sytems for the agent in several different machines - called satellite agents.

  They have been developed to work under a specific platform, making use of the specific tools of the used language: ShellScripting for Unix - which includes GNU/Linux, Solaris, AIX, HP-UX and BSD, as well as the Nokia's IPSO. Pandora agents can be developed in virtually any language, given its

simple API and being open source. Windows agent are developed in a free development environment for C++ and uses the same interface and modularity than Unix agents.

The old agent for Windows plattforms was developed on VBS Scripting language, and is deprecated with the new Pandora 1.2 windows agent.



1.3. What kind of systems/ services can be monitored?

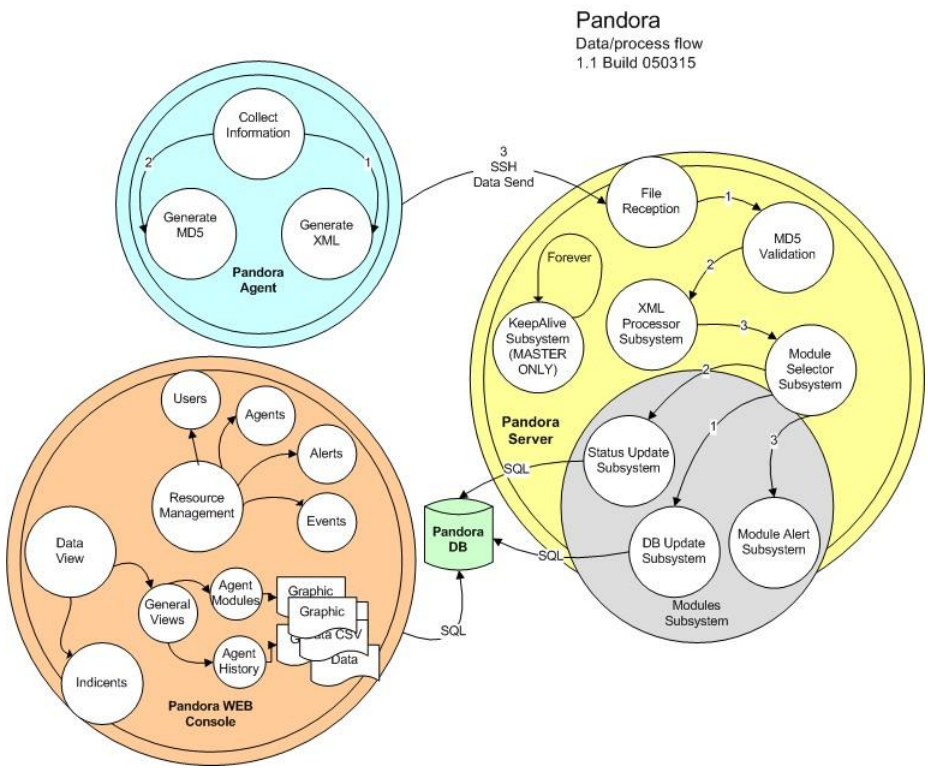
At present, with Pandora any process or system that through a command returns a value can be monitored, as well as any value in any Operating System log file or similar. Some examples of already existing implementations below:

```
Number of connections (sessions) of Checkpoint FW-1
Number of NAT sessions of Checkpoint FW-1
Number of connections of Linux NetFilter / IPTables firewall
Number of FW-1 logged packets
Number of FW-1 dropped packets
Number of FW-1 accepted packets
State of High Availability in FW1 NG
Last policy installed in a Firewall-1 module
Synchronization state of the modules in FW1 NG
CPU of the system: idle, user and system
Number of processes of the system
Temperature of the CPU of a system
Value of a MS Windows registry entry
Queued jobs in a generic dispatcher
Memory of the system: free, swap, kernel Fw-1, cache
Percentage of free space on disc (for different partitions)
Messages processed by a mail gateway
Existence of a string in a text file
IP traffic (filtering based on the connections of the firewall)
Hits of pages in HTTP Servers (Apache, iPlanet, IIS, Netscape)
```

```
Percentage of erroneous packets in a Gateway
Connections established in a Remote Access Server (RAS)
Size of a file
Open sessions by a VPN server
MySQL Performance: Threads, queries, sessions...
Snort system state
Reported events by IDS (Snort) up to six levels of priority
Network load
Number of local Connections (TCP, UDP, Unix sockets)
Detected viruses by a Web Antivirus Gateway
ICMP latency time towards a host
Rate of average transference in a file transfer tool
Number of DNS requests attended by a server (including types)
Number of FTP sessions attended by a FTP server
(Generic) State of any active process / service in the system
(Generic) State of any countable parameter of the system
```

1.3.1. Global architecture

Pandora 1.2 has changed many things from 1.1 version, but this graph representing Pandora architecture is very useful to understand in a single graph, all components.



1.4. Information gathering with Pandora agents

Pandora agents are based on native languages in every platform: scripts that can be written in any language. It's possible to reproduce any agent in any programming language and can be extended without difficulty the existing ones in order to cover aspects not taken into account up to the moment.

These scripts are formed by modules that each one gathers a "chunk" of information. Thus, every agent gathers several "chunks" of information; this one is organized in a data set and stored in a single file, called data file.

The process of transferring the data file from the agent to the server is made regularly at a defined time interval in the agent configuration file, pandora_agent.conf. It's possible to modify that parameter in order to do not fill the database with non-relevant information, either load the network or affect the system performance. The default interval is 300 (seconds), which is equivalent to five minutes. Minor values of 100 (seconds) are not recommended since host performance could be affected, besides loading excessively Database and the Operating System of Pandora Server. Pandora is not a real time system; it's an applications and systems general monitoring system in environments that are not critical at real time.

Packets transfers are made via SSH, with DSA authentication (although also RSA can be used). The process is completely safe since neither any password nor unencrypted confidential information is sent. Confidentiality, integrity and authentication of the connections between the agent and the server are ensured. In the Agents and Server Installation and Configuration guides, the process of generation of keys to do the automatic SCP transfer is detailed.

Also the transfer via FTP or any other file transfer system could be made, although SSH has been chosen for security and compatibility with most of the systems in the market.

Pandora Agents are thought to be executed from the agent from which they gather information, although the agents can gather information of accessible machines from the host where they are installed. In this case those agents are called "Satellite Agents". These Satellite Agents can use Telnet, SNMP or any other commands to get the information.

We can also have a host with several agents: Some that gather information from the accessible machines (acting as "satellite agents") and the Standard Agent that monitors the host where it's running.

1.4.1. XML Data files

The data file has the following syntax:

```
hostname.serialnumber.data
```

This is an XML file, and its name is the combination of the hostname where the agent runs, a different serial number for every data package and the extension .data that indicates that it's a data file.

We also have a control file for every data file:

```
hostname.serialnumber.checksum
```

This file has .checksum extension and contains a MD5 hash of the data file. This allows checking that the information has not been changed before being processed.

The XML data file generated by every agent is the core of Pandora. This file has the information gathered by the Agent. Its easy structure allows that any user could create its own developments to be processed in Pandora, or use the included ones. An example of the information included into the data file below:

```
<agent_data os_name="SunOS" os_version="5.8" timestamp="300"
agent_name="pdges01" version="1.0">
  <module>
    <name>SSH Daemon</name>
    <type>generic_proc</type>
    <data>1</data>
  </module>
  <module>
    <name>FTP Daemon</name>
    <type>generic_proc</type>
    <data>0</data>
  </module>
  <module>
    <name>DiskFree</name>
    <type>generic_data</type>
    <data>5200000</data>
  </module>
  <module>
    <name>UsersConnected</name>
    <type>generic_data_inc</type>
    <data>119</data>
    <min>1</min>
    <max>250</max>
    <description>Users currently connected</description>
  </module>
  <module>
    <name>LastLogin</name>
    <type>generic_data_string</type>
```

```
    <data>slerena</data>
  </module>
</agent_data>
```

1.4.2. Pandora servers

With Pandora 1.2 version, you have three different types of servers:

- *Pandora Data Server*. This is a PERL application that that processes the information sent by the agents. The agents send the XML data file via SSH and the server periodically verifies if it has new data files waiting to be processed. You can setup different data servers in different systems or in the same host (that will be different virtual servers).
- *Pandora Network Server*. This is a PERL application that execute network tasks like sending pings, TCP requests, SNMP requests and UDP request. When you assign an agent to a server, you are assigning to a network server, not a data server, so, this is very important that machines running network servers have "network visibility" to hosts assigned in network modules.

  For example, if you create a module to make a ping check to 192.168.1.1 and assign this agent/module to a server in a 192.168.2.0/24 network without access to 192.168.1.0/24 module will always report DOWN.

- *Pandora SNMP Server*. This is a PERL application that parse output from standard snmptradp (we provide one binary for snmptrapd, but it is possible that you need to replace it with a binary that runs better in your system). This daemon receives SNMP traps, and Pandora SNMP Server stores in database and fire alerts assigned in Pandora SNMP Console.

Data are extracted from the data file, identifying origin, type and category. Once it's classified, the data are inserted into the Database by the same Perl script.

Pandora Server can work in High Availability and/or Load Balancing. In a very big architecture, several Pandora Servers can be arranged simultaneously in order to be able to manage big volumes of information distributed by geographical or functional zones.

Pandora Server is always running (as a daemon) and permanently verifies if some element causes to fire an alarm. If so, it executes the action defined in the alarm, as to send a SMS, an email, even activates the execution of a SCRIPT or to send an HTTP form.

We could have several simultaneous servers, one of them is the Main Server or "Master Server " and the rest of servers are "Slave Servers". The Master Server is the only one that verifies the alarms if any agent goes down. The server which receives the data file from the agent always fires the rest of alarms, defined in the agents' modules. This is also important if this server changes (due to configurations of high availability, load balancing or clustering).

1.4.3. Pandora console

The Web Console is a web application that allows to see graphical reports, state of every agent, also to access to the information sent by the agent, to see every monitored parameter and to see its evolution throughout the time, to form the different nodes, groups and users of the system. It is the part that interacts with the final user, and that will allows you to administer the system.

The Web Console is written in PHP and no plug-in, Flash, Java or ActiveX is needed to access the console, only a browser that supports HTML and CSS (IE5+ or Mozilla 4+). Pandora Web Console can run in several servers, the only thing you need is to be allow to access Pandora Database, where Pandora stores all the information.

1.4.4. Pandora database

Pandora uses a SQL Database to store all the information. Pandora maintains an asynchronous database with all the received data, making a temporary cohesion of everything it is receives and normalizing all the information from the different sources. Every Agent data module generates an entry of information for every data bundle, which implies that a real production system can have of the order of ten million of data, or information atoms.

This information is managed automatically from Pandora, carrying out a periodic and automatic maintenance of the database. This means that there is no operator either manager required to run tasks as database administration ones. This is possible thanks to a periodic purge of the past information over a date (by default 90 days), as well as a data which is older, by default, 30 days.

1.4.4.1. Compacting data

Data stored by Pandora are useful to see evolutions through the time, in order to: make statistics, generate reports and to do capacity planning, as well as other statisticals tasks. To do that it isn't necessary to have all the data, but it's enough to have a representative sample, of smaller resolution, enough to carry out the task that is needed.

With that philosophy the compaction system has been constructed. For instance, If we have a sample of 9.000 elements, distributed during 90 days, Pandora will take the data of last month, which would be 3.000 elements and will compress it in 300. In the graphs they will practically be equal, and it will be

usfel for the reports, statistics and other tasks. This is made thanks to a interpolation in temporary strips, in a totally automatic and periodic way, there is not user or the administrator needed to do this.

1.5. Pandora 1.2 new features

Alert system. Now it is possible to define a "minimun" and "maximum" limit to fire an alert, just to delete "noisy" data that fires false positives.

Network Subsystem. Now it is possible to monitor and analyze data using remote network tools, without using agents, from the new Pandora Network Server component. All management are made from Pandora Console, and now you will be able to make ICMP checks (Ping), size network latency, get all types of SNMP values (including scanning MIB), and makes TCP/UDP connections to check ports, and test text applications, sending texts and waiting for a specific response.

*Module groups.*Modules now could be grouped using a new "module groups".

Network data refresh on demand. Could be for each module or using a "global group refresh", forcing Pandora Network Servers to refresh all network modules inside a group.

Online contextual help, for Pandora WEB Console.

New Pandora server infraestructure.

New SNMP trap console to receive SNMP traps and assigning alerts.

Internal messaging system, to notify events to Pandora users.

Agent detail view autorefresh

New main agent group view

Improved database management system, that allows to manage much more data.

1.6. About Pandora

Pandora is a project initiated and mainly developed by Sancho Lerena, at present other people is working on it: Raul Mateos, David Villanueva, Esteban Sanchez, Jose Navarro and Jonathan Barajas. We want to

thank many other people who help us with translation, graphic design, bugs reporting and interesting ideas.

Pandora is Free Software, and is published under GPL Licence. In order to know the last features, go to the official web site of the project in http://pandora.sourceforge.net.

Chapter 2. *Pandora* installation

2.1. Prerequisites

Pandora is not only a single app, it is made up by several shellscript files (Unix/Linux Agents), a WEB application in PHP (Console), some code in C++ (Windows Agent), some code in PERL5 (Server) and some structure and data in SQL (Database), so, to get all this running you need to have some pieces of software installed in your system. This is a list of packages, libraries and software you need before install *Pandora*.

2.2. Pandora Servers

Pandora FMS 1.2 has three kind of servers: Data server, Network Server and SNMP Server/Trap console. All of them could be installed in the same machine or in different machines, also, you could setup many of them in a High Availability environment or using it to manage highs loads of data.

2.2.1. Pandora Data Server

To build *Pandora Data Server* you need to have the following perl modules and software installed in your machine. This packages could be installed using your distribution packaging system or using CPAN.

- XML::Simple, useful XML functions
- Digest::MD5, MD5 generation
- Time::Local, Date and Time basic manipulation
- DBI, DB interface with MySQL
- Date::Manip, needed to manipulate Date and Time formats of input, output and compare

You can find them at http://www.cpan.org or install using your default package instalation system. These packages are in the default distribution of Suse 9.1 and Debian 3.0 GNU/Linux. Also available for Solaris in CPAN repository. Next, you need to set the TZ (Time Zone) environment variable.

2.2.2. Pandora Network Server

Requires SSH Server and Perl v5.8 or higher and the next Perl Modules:

- IO::Socket, manage and manipulation of TCP/UDP sockets

- Time::HiRes, needed for ICMP times
- Time::Local, Date and Time basic manipulation
- SNMP, for SNMP management
- Date::Manip, needed to manipulate Date and Time formats of input, output and compare
- Net::Ping, to calculate latency times (it's required that the server runs as root user).

To use SNMP fuctions it's needed also to have installed the net-snmp package. It's worth to say that to run modules of GENERIC_ICMP_DATA type (calculate ICMP latency time) Pandora Network Server must run with root privileges.

2.2.3. Pandora SNMP Server

You need to install the NET-SNMP package which is included in all GNU/Linux distributions. You have to use the snmptrapd binary and copy or link it to $HOME_PANDORA/util, where $HOME_PANDORA is the instalation directory of Pandora.

This binary gets the SNMP traps, generating a log that is parsed by the Pandora Server.

2.2.4. Installing Pandora Server

Create the `/opt/pandora` directory and "gunzip" and "untar" here the `pandora_server_1.2.tar.gz` file.

Create an user pandora in OS. Usually you do that in GNU/Linux with commands:

```
useraddd pandora -d /home/pandora
mkdir /home/pandora
chown pandora /home/pandora
```

This user will be used by the SSH transfers to the server, so this user will need a strong password.

In the file `/home/pandora/.ssh/authorized_keys` we will add the public key of each agent which send data to Pandora Server. These keys must be SSH v2, OpenSSH DiffieHellman (DF) or RSA. To convert between keys you can use the ssh-keygen tool.

Pandora Server will check and parse XML files sent by Pandora Agents and will insert the data into the Database.

Check launch scripts (pandora_network, pandora_server, pandora_snmp) and check for pathnames in the first two variables in script. roa Server. This usually is `/opt/pandora_server`

2.2.5. Configuring your new Pandora Server setup

After install Pandora Server in, you will need to edit the file `pandora_server.conf`, where are defined the variables of the server configuration. File `pandora_server.conf` is a text file, you could edit with your prefer text editor, like emacs. This configuration file is common to all kinds of Pandora Server (Data server, SNMP Server, Network server), you also could have different copies of configuration file for each Pandora Server you have.

Edit configuration file of Pandora Server, usually `/opt/pandora/conf/pandora_server.conf` and take a look at the lines:

```
dbuser pandora
dbpass pandora
dbhost localhost
```

Please change them to your own data. For security reasons isn't recommended use the default values.

These are default values, and all must be existing directory and filename and valid username, password and hostname.

Remember: you need to create the directory /opt/pandora/data_in where Pandora Server will read and write data, sent by remote agents using ssh/scp. This directory must be owned or with permissions to write for user "pandora". If you don't have a "pandora" user yet, create it.

You can run Pandora Server with an user without privilegues, you can use the user "pandora", it only needs to run /usr/bin/perl and access to `/opt/pandora` and `/opt/pandora/data_in` directories.

This is true with all the components but with Pandora SNMP Console needs root user to open UDP port 161 (this can be solved setting SUID0 to the snmptrapd binary) and running the rest of the Server using an user without privileges.

Also Pandora Network Server can be run using an user without privileges, but the GENERIC_ICMP_DATA type won't work, as root privileges are required to get ICMP latency times.

Check the MySQL connection with the user and password before running the server

Pandora Server distribution tarball includes a Posix/System V start/stop script for "daemonize" Pandora Server. It is possible that you need to customize, but its runs smoothly on GNU/Linux (debian, Suse) and Solaris 8 systems. It has start|stop|restart parameters to include it in your default init level directory and it creates a logfile defined in $log_file variable (by default is `/opt/pandora/pandora.log`):

```
/etc/init.d/pandora_server start
```

2.2.5.1. Setting up SSH configuration

Pandora, uses SSH protocol to copy XML data packets, generated by the agents, to the server. You need to generate a SSH2 key in every agent, and copy the public key in `/home/pandora/.ssh/authorized_keys`, so you need to create a user called "pandora" withour privileges. This user will be used by agents to copy data into Pandora Data Server `/opt/pandora/data_in` directory.

Please BE SURE that user "pandora" exists (if not, create with useradd), and `/home/pandora/.ssh/authorized_keys` exists and ownership of this file and directory is for pandora user, and permissions set to 600.

Please be sure that directory `/opt/pandora/data_in` exists and pandora user is able to write in.

2.3. Pandora Console and Pandora database

2.3.1. Pandora database install

Please look at MySQL install and management guide (http://dev.mysql.com/doc) to obtain information about how to create a MySQL database, how to manage mysql users and give him/her privileges to read/write in Pandora database. Remember that you must write the password of the root user in MySQL database to enter mysql command line. This user is not the same of the Operating System. The root password in MySQL is in blank by default (within almost all distributions), you must changed this password with the MySQL command `mysqladmin`. Please be careful with this.

You need a database with name "pandora", you could rename it, but you need to reconfigure in server too.

To create the structure of Pandora database in MySQL Server you have the SQL script "pandoradb.sql".

It creates tables and indexes needed to insert information into Pandora database.

You MUST populate database with SQL script "pandoradb_data.sql", it inserts data needed to run Web Console and default user (login: admin, pass: pandora) to access Pandora Web Console.

First create a database called "pandora", and set an user to be able to access this database:

```
mysql> create database pandora;
```

Later, execute the next commands using a user with enough privileges to create tables and indexes for pandora Database into your MySQL Server:

```
cat pandoradb.sql | mysql -D pandora -u root -p
cat pandoradb_data.sql | mysql -D pandora -u root -p
```

You can also use the source command, if you are connected to MySQL, from the MySQL prompt:

```
mysql> use babel
mysql> source path_to_babel_dbstruct.sql
mysql> source path_to_babel_dbdata.sql
```

This example is valid using root user in MySQL

Now we will create an user "pandora" and will be given to it privileges from the localhost:

```
mysql> grant all on pandora.* to 'pandora'@'localhost'
identified by 'pandora';
```

Keep in mind that users need access from Pandora WEB Console and from Pandora Server, if your deployment has many subcomponents in different physical machines, you need to setup a MySQL user with privileges to access from different locations.

If you get the error "Warning: mysql_connect() [function.mysql-connect]: client does not support authentication protocol requested by server; consider upgrading" when authenticating Web Console, you have to change the way the password is stored into the database:

```
mysql> set password for 'pandora'@'localhost' = old_password('pandora');
```

Please note this user will be used by several pandora subcomponents (Pandora Server, Pandora Web Console) to access database.

2.3.2. Pandora console install.

Prior to install Pandora console, you need the following dependencies and software needed:

- Web server. Apache2 is recommended.
- PHP 4.3.x, or PHP 5.x. Both has been tested for Pandora 1.2

- PHP Modules for MySQL, GD, session management and SNMP.
- JpGraph, it is necesary to generate graphics. It has an open source license, you can download it in http://www.aditus.nu/jpgraph/

To install Pandora Console, simply untar in your HTTP server publishing directory and set perms to www-data or http user.

To setup Pandora Console, you only need to modify a file, `include/config.php`, where the following variables are included in .php code:

```
$dbname="pandora";   // name of database for pandora)
$dbuser="pandora";   // mysql user to access db
$dbpassword="pandora";   // Password for mysql user
$dbhost="pandora";   // Hostname or IP of mySQL server
```

If database is defined and was correctly installed, you can now access:

```
http://hoste:port/installdir/index.php
```

The first time you log there is a default admin user "admin" and password "pandora". It's worth to say that *YOU MUST CHANGE CREDENTIALS BEFORE LOGIN FIRST TIME*, change it or create another account, give it administrator privileges, and disable this one.

If you cannot see a screen like this, it's possible that you have problems with PHP instalation. When you installed the Web, please check that PHP engine its running. Fist try to access to the server IP with a browser. You must see the Welcome Apache page.

Remember that alter installing the PHP and the PHP module for Apache you must stop and start the Server Apache. As an example, Ubuntu with Apache2:

```
/etc/init.d/apache2 stop
/etc/init.d/apache2 start
```

To verify the PHP and Apache integration you can create the file `test.php` with the following lines:

```
<?PHP
echo "<h1>TEST</h1>";
phpinfo();
?>
```

Now, copy this file in the Apache HTTPDOC directory. This directory depend of the Operating System or Linux Distribution, for example in Ubuntu this directory is `/var/www` and in SUSE is `/srv/www/htdocs`).

To check this integration, please use your browser to open the following URL:

```
http://IP/test.php
```

Where IP is IP Address of your Apache server. If the integration is correct you will see in the browser a text string with big font: "TEST" and a big table with a lot of info about your PHP installation.

2.3.2.1. Graphic reporting instalation

For correct graphic generation, you need to enter the full path to a TrueType font installed in your system. By default a free truetype font is distributed with Pandora Console package, and placed in `./reporting/FreeSans.ttf` file. Please check that setup directive `$config_fontpath` is well configured.

Pandora 1.2 uses JpGraph for viewing graphics. JpGraph is a different project and has no relationship with Pandora, so you need to install it. You can find at `http://www.aditus.nu/jpgraph/`. Download last version (2.x), and place all `.php` files from src directory into `reporting/jpgraph` Pandora Console directory.

2.4. Pandora Agents

2.4.1. Introduction

Pandora agents collect all system's data. They are executed in each local system, although they can also collect remote information by installing monitoring systems for the agent in several different machines - called satellite agents.

They are developed to work under a given platform, making use of the specific tools of the language being used: VBSCript/Windows Scripting for Microsoft platforms (Win2000, WinXP y Win2003), ShellScripting for UNIX - which includes Linux, Solaris, AIX, HP-UX and BSD, as well as the Nokia's IPSO. Pandora agents can be developed in virtually any language, given its simple API system and being open source. There are branches of the Pandora project started for the creation of agents in Posix C, Perl and Java for those systems requiring closed agents.

Pandora Agents are Free Software, i.e., the way agents collect and sent information is documented. An agent can be recreated in any programming language, and can be upgraded easily, to improve aspects of the program not covered so far.

This document describes the installation of agents in machines running over Windows and Unix operating systems.

2.4.2. Generic role of the agents

Regardless the platform an agent is running on, this is formed of the following elements:

A script (or binary application in Windows) that collects and sends the data to the server. For UNIX machines the script is called pandora_agent.sh and is executed directly from the Pandora agent folder.

One or several configuration files where the values to be collected are defined. The file is called pandora_agent.conf both for Windows and Unix machines.

This simple structure makes it easy the customisation of an agent. There is no need to code again the agent to modify the way it works, as the configuration file holds most of the parameters needed to do so.

2.4.3. Main Script

The main script is the executable file that collects the data specified in the configuration file. It sends the data to the server in XML. In Windows machines application is installed as a service and is executed at

the time intervals set in the configuration file. In machines running over UNIX the main script is run through a special script called `pandora_agent_daemon`, and runs continuously in the machine as a process.

2.4.4. Configuration File

The data collection in the host system is the gathering of independent data units, which are defined in the `pandora_agent.conf` file. The pandora_agent.conf file is divided in two parts:

- *General parameters*: Configure general options about server location, agent name, interval, and other general options.
- *Module definitions*: Configure and define the method of extraction for each piece of information that will be extracted from local host and sent to Pandora Server.

2.4.4.1. General parameters

The general parameters of the agent configuration are defined in this section. Some of these parameters are common for all systems and others specific for Windows or UNIX. The general parameters are:

- *server_path*: (Shared parameter) The server path is the full path of the folder where the server stores the data sent by the agent. It is usually `/opt/pandora/data_in`.
- *server_ip*: (Parameter shared by Windows and Unix agents) The server IP is the IP address or the host name of the Pandora server, where the data will be stored. The host must be reachable and must be listening to port 22 (SSH).
- *temporal*: (Shared parameter) This is the full path of the folder where the agent stores the data locally, before it is sent to the server. It must be said that the data packages are deleted once the agent tries to contact Pandora server, no matter if the communication was successful or not. This is done to avoid over flooding hard drive of the host system where the agent runs. The location of the local folder varies with the architecture of the host system. In Unix systems this is usually `/opt/pandora/data_out`, and in Windows systems `C:\pandora\data_out`.
- *interval*: (Shared parameter) This is the time interval in seconds in which the agent will collect data from the host system and send the data packages to the server. The recommended value ranges from 300 (5 minutes) to 600 (10 minutes). This number could be larger, although it is important to consider the impact of a larger number on the database.
- *debug*: (Unix only) This parameter is used to test the connection between agent and server and the correct working condition of the agent. The process consists of a loop, data collection and data transfer. It does not delete any data when the process is finished. The activity is written in a log file, stored in the Pandora root folder. The file is named pandora_agent.log. This log file can be used to test the system and to investigate potential issues.
- *agent_name*: (Shared parameter) This is an alternative host name. This parameter is optional as if it is not declared the name is obtained directly from the system.

- *pandora_path*: (Unix exclusive parameter) This is then path of the folder where the files of the Pandora agent are stored. This is usually `/opt/pandora` or `/opt/pandora_agent`.
- *checksum*: (Shared parameter). This parameter can take two values. If the value is 1, the checksums are performed through MD5. If the value is 0, the checksum is not performed at all. This may be useful for systems where a MD5 tool cannot be implemented. If the checksum is deactivated in the agent it must be also disconnected in the server. Otherwise it could create problems.

An example of the general parameters from a Unix configuration would be.

```
server_ip Pandora_Server
server_path  /opt/pandora/data_in
pandora_path /opt/pandora
temporal /opt/pandora/data_out
interval 300
agent_name  satellite_agent
debug 1
checksum 1
```

2.4.4.2. Module definition

Each data item that is to be collected must be defined precisely in each module, using the exact syntax. As many values as necessary can be set to be collected, adding at the end of the general parameters as many modules as the number of values to collect. Each module is made of several directives. Following is a descriptive relation of all module marks available for Unix agents (almost all of them are applicable to Windows Agent too).

2.4.4.2.1. module_begin

Defines the beginning of the module.

2.4.4.2.2. module_name name

Name of the module. This is the id for this module, choose a name without blank spaces and not very long. There is no practical limitation (max of 250 chars) but will be more easy to manage if you use short names. This name CANNOT be duplicated with a similar name in the same agent. This name could be duplicated with other modules in other agents.

2.4.4.2.3. module_type type

Data type the module will handle. There are four data types for agents:

- Numeric (generic_data). Simple numeric data, float or integer. If the values are of the float type, they will be truncated to their integer value.
- Incremental (generic_date_inc). Integer numeric data equal to the differential between the actual value and the previous one. When this differential is negative the value is set to 0.
- Alphanumeric (generic_string). Text strings up to 255 characters.
- Monitors (generic_proc). Stores numerically the status of the processes. This data type is called monitor because it assigns 0 to an "Incorrect" status and any value above 0 to any "Correct" status.

2.4.4.2.4. module_exec command

This is the generic "*command to execute*" directive. Both, for Unix and Windows agents there is only one directive to obtain data in a generic way, executing a single command (you could use pipes for redirecting execution to anoter command). This directive executes a command and stores the returned value. This method is also available on Windows agents. This is the "general purpose method" for both kind of agents.

For a Windows agent there are more directives to obtain data, who are described following this lines.

2.4.4.2.5. module_service service (Win32 Only)

Checks if a given service name is running in this host. Remember to use " " characters if service name contains blank spaces.

2.4.4.2.6. module_proc process (Win32 Only)

Checks if a given processname is running in this host. If the process name contains blank spaces *do not* use " ". Also notice that the process name must have the .exe extension. The module will return the number of process running with this name.

2.4.4.2.7. module_freedisk drive_letter: (Win32 Only)

Checks free disk on drive letter (do not forget ":" after drive letter.

2.4.4.2.8. module_cpuusage cpu id (Win32 Only)

Returns CPU usage on CPU number cpu. If you only have one cpu, use 0 as value.

2.4.4.2.9. module_freememory (Win32 Only)

Return free memory in the whole system.

2.4.4.2.10. module_min value

This is the minimum valid value for the data generated in this module. If the module has not yet been defined in the web console this value will be taken from this directive. This directive is not compulsory. This value does not override the value defined in the agent if the module does not exist in the management console. It is created automatically when working on learning mode.

2.4.4.2.11. module_max value

It is the maximum valid value for the data generated in this module. If the module has not been defined in the web console this value will be taken from this directive. This directive is not compulsory and is not supported by the Windows agent. This value does not override the value defined in the agent if the module does not exist in the management console. This is created automatically when working on learning mode.

2.4.4.2.12. module_description text

This directive is used to add a comment to the module. This directive is not compulsory. This value does not override the value defined in the agent if the module does not exist in the management console. This is created automatically when working on learning mode.

2.4.4.2.13. module_interval factor

Pandora 1.2 introduces this new feature. You can, for each module, setup its own interval. This interval its calculated as a multiply factor for agent interval. For example, if your agent has interval 300 (5 minutes), and you want a module only be calculated each 15 minutes, you could add this line: `module_interval 3`. So this module will be calculated each 300sec x 3 = 900sec (15 minutes).

2.4.4.2.14. module_end

Ends module definition

2.4.4.2.15. Examples

An example of a Windows module, checking if EventLog service is alive, would be:

```
module_begin
module_name ServicioReg
module_type generic_proc
module_service Eventlog
module_description Eventlog service availability
module_end
```

An example of a Unix module would be:

```
module_begin
module_name cpu_user
module_type generic_data
module_exec vmstat | tail -1 | awk '{ print $14 }'
module_min 0
module_max 100
module_description User CPU
module_end
```

2.4.5. Agent types

It is possible to monitor virtually any system with Pandora. This can be done either with a local agent collecting data directly from the system to be monitored, using a a satellite agent collecting data from a system by SNMP or using the new Pandora 1.2 agents, the remote agents, who can chack using remote network polling (TCP, UCP, ICMP/PING and SNMP) remote services, from the Pandora Network Server.

The local agents can be either Windows or Unix agents. The satellite agents can be implemented using any of the agents above. The modules are configured to collect data from the external system by, for example, an SNMPGET tool.

2.4.5.1. UNIX agents

2.4.5.1.1. Introduction to Unix agents

The in-built UNIX applications and tools make the agents running on this system be very simple. There are also agents developed for AIX, Linux, Solaris and BSD platforms, some of them very similar but not identical. Requirements for the installation of Pandora Agents on UNIX

2.4.5.1.1.1. AIX

MD5 signatures are used to guarantee the integrity of the generated data packages. The MD5 package is integrated in AIX 5.1 and above. There is a freeware package for AIX 4.3 but it has several issues and might not work correctly. In the case of having problems with the AIX agents the checksum system used to validate the integrity of the data can be disabled.

2.4.5.1.1.2. Solaris

The MD5 package is necessary to execute the Solaris agent correctly. This package is available from http://sunfreeware.com . It can be also downloaded for Solaris 8 from the following URL:

ftp://ftp.sunfreeware.com/pub/freeware/sparc/8/md5-6142000-sol8-sparc-local.gz

MD5 Package installation on Solaris

```
root@stest:/tmp:> gzip -d md5-6142000-sol8-sparc-local.gz
root@stest:/tmp:> pkgadd -d ./md5-6142000-sol8-sparc-local

The following packages are available:
1  SMCmd5     md5
(sparc) 6142000

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1
```

Solaris SSH

The suggested SSH client is OpenSSH. If any other SSH client is to be used it must be considered that each piece software may have different ways to generate or manage keys. For example, if F-Secure SSH is used, the public key must be in OpenSSH format when the keys are generated. The format can be changed from IETF to OpenSSH with F-Secure SSH, using the following command:

```
ssh-keygen -i -f file_ietf_pubkey
```

2.4.5.1.1.3. GNU/Linux

SSH and MD5 should be installed in Linux by default, but if they are not they can be installed using the tools available in each distribution.

2.4.5.1.1.4. BSD (IPSO)

SSH and MD5 should be installed by default. If they are not, it is necessary to install them.

2.4.6. Pandora Unix Agent install

The software comes in a .tar.gz file. First of all the file needs to be extracted into a folder, usually /opt/pandora_agent, although any other folder may be used. If a different folder is used, the daemon launcher must be modified by changing route to $PANDORA_HOME.

There is hardly any difference between AIX, Solaris and Linux, and they all work around the hash MD5 generation binaries.

This is the structure of the installation in /opt/pandora_agent/ once the files have been extracted:

`/opt/pandora_agent/data_out`, folder where the data collected by the agents is stored.

`/opt/pandora_agent/doc`, folder with information about the agent and its license.

`/opt/pandora_agent/pandora_agent.conf`, file where the data to be collected is defined, along side the command to be executed for the data collection. This is the system core, as it defines the main data to be collected in any Firewall.

`/opt/pandora_agent/pandora_user.conf`, file where several of the parameters to collect data from the monitored system are defined in more detail.

`/opt/pandora_agent/pandora_agent.sh`, this is the actual Pandora agent. This file is a shellscript that collects the data configured in the pandora_agent.conf and pandora_user.conf files. It also transfers the data packages to the Pandora server.

`/opt/pandora_agent/pandora_agent_daemon`, start and stop script. It makes a call to pandora_agent.sh. It offers two options, start and stop.

`/opt/pandora_agent/pandora.log`, text file where the activity of the Pandora agent is saved, when the agent is executed in debugging mode.

2.4.6.1. Key generation

The SSH keys generated must be:

- SSSH version2 keys
- Open SSH format keys
- DiffieHellman (DH) format keys

To generate the keys the command ssh-keygen is executed followed by the specific parameters for our operating system. Please, create key WITHOUT password.

The public key must be copied into the `/home/.pandora/authorized_keys` file in the Pandora server. Before starting the Pandora agent the SSH authentication must be checked. To do this the following command must be executed on the agent machine:

```
$ ssh pandora@pandora_server
```

The system must connect successfully BEFORE launching the Pandora agent.

2.4.6.2. First running of the Unix agent

To start the agent it is only necessary to execute pandora_agent_daemon start from /opt/pandora_client. Pandora Agent creates a file (/var/run/pandora.pid) with the PID number of the process when it is started.

For IPSO systems the agent will be started with a nice -10 priority, so it becomes the process with the lowest priority over the system CPU. It will be executed when no other processes with a higher priority are waiting in the system CPU queue.

In BSD systems the maximum priority is +20 and the lowest -20.

To stop agent, execute pandora_agent_daemon stop from /opt/pandora_agent.

2.4.6.3. Advanced configuration for Unix Agent

The real power of Pandora resides in the capability of the agents to run user defined scripts. This could be used to collect specific data or to perform an operation to return any desired value. This is the purpose of pandora_user.conf.

This file is executed every in agent loop. It is a shell-script in which any command can be executed, as long as the output is in the XML format the agent uses to send data to the server. The XML structure would be:

```
<module>
<name>NAME</name>
<type>TYPE</type>
<data>DATA</data>
</module>
```

Where NAME, TYPE and DATA are the variables already defined in previous sections. The XML must be built manually, usually using echo commands.

For example, this would be the script a customized agent would use for Checkpoint FW1 in IPSO agents:

```
#!/bin/sh
# Pandora User-Defined acquisition script
# This code is under GPL licence
# Please refer documentation for more example and a more
# depth usage instructions

# mbuf clusters usados (%)
MBUF_TOTAL=`netstat -m |grep "mbuf cluster" | tr -s "/" " " |awk '{ print $2 }'`
MBUF_USED=`netstat -m |grep "mbuf cluster" | tr -s "/" " " |awk '{ print $1 }'`
MBUF_USED_PER=`echo $MBUF_TOTAL $MBUF_USED |  awk '{ print $2 / ($1 / 100) }

echo "<module>"
echo "<name>MBUF_CLUSTER_USED_PER</name>"
echo "<data>$MBUF_USED_PER</data>"
echo "<type>generic_data</type>"
echo "</module>"
```

A more complex example could be:

```
#!/bin/sh
# Pandora User-Defined acquisition script
# This code is under GPL licence
# Please refer documentation for more example and a more
# depth usage instructions

# Calculating the number of packages generated by ETH2,
# if nothing is generated
# within 20 seconds an alert is rosen
# Perform the calculation between 8 to 23h. Return ok for times
# outside this range

echo "<module>"
echo "<name>Packet_Generator_Check</name>"
echo "<type>generic_proc</type>"
UNO=`ifconfig eth2 | grep "TX packets" | cut -f 2 -d ":" | grep -o -e "[0-9]*"`
sleep 20
```

```
    DOS=`ifconfig eth2 | grep "TX packets" | cut -f 2 -d ":" | grep -o -e "[0-9]*"`
    HORA=`date "+%k"`
    if [ "$HORA" -lt "8" ] && [ "$HORA" -gt "11" ]
    then
    # Time out of range, no checking, everything OK
    # Fuera de hora, no compruebo, esta OK
       echo "<data>1</data>"
    else
   if [ "$UNO" == "$DOS" ]
   then
          echo "<data>0</data>"
   else
       echo "<data>1</data>"
   fi
 fi
   echo "</module>"
```

2.4.6.4. Implementation examples for Unix Agents

Example #1: calculate the number of HITS of the main page of an Apache Web server:

```
    module_begin
    module_name WEB_Hits
    module_type generic_data_inc
    module_exec cat /var/log/apache/access.log | grep "index" | wc -l
    module_end
```

Example: check if the process of the DNS server (named) is active or fell over:

```
    module_begin
    module_name DNS_Daemon
    module_type generic_proc
    module_exec ps -Af | grep named | grep -v "grep" | wc -l
    module_end
```

Complete example of the configuration of an agent for Linux

```
    # General Parameters
    # ==================
```

```
server_ip   192.168.100.45
server_path  /opt/pandora/data_in
pandora_path  /opt/pandora_ng/
temporal     /opt/pandora_ng/data_out
interval     300
hostname  linuxbox01
debug   0
checksum  1

# Module Definition
# =================

module_begin
module_name cpu_user
module_type generic_data
module_exec vmstat 1 2 | tail -1 | awk '{ print $14 }'
module_end

module_begin
module_name cpu_sys
module_type generic_data
module_exec vmstat 1 2 | tail -1 | awk '{ print $14 }'
module_end

module_begin
module_name disk_root_free
module_type generic_data
module_exec df -kh / | tail -1 | awk '{ print 100 - $5 }'
module_end

module_begin
module_name disk_store_free
module_type generic_data
module_exec df -kh /store  | tail -1 | awk '{ print 100 - $5 }'
module_end

module_begin
module_name memfree
module_type generic_data
module_exec cat /proc/meminfo  | grep MemFree | cut -c 10-23
module_end

module_begin
module_name memused
module_type generic_data
module_exec cat /proc/meminfo | grep "Active"  | cut -c 8- | cut -f 1 -d "k"
module_end

module_begin
module_name proctotal
module_type generic_data
module_exec ps -A | wc -l
```

```
module_end

module_begin
module_name sshd
module_type generic_proc
module_exec ps -Af | grep sshd | grep -v "grep" | wc -l

module_begin
module_name WEB_Hits
module_type generic_data_inc
module_exec cat /var/log/apache/access.log | grep "index.php" | wc -l
module_end

module_begin
module_name eMails_proc
module_type generic_data_inc
module_exec cat /var/log/mail/mail.log | grep "message-id" | wc -l
module_end

module_begin
module_name FTP_sessions
module_type generic_data_inc
module_exec cat /var/log/syslog | grep "FTP session opened" | wc -l
module_end

module_begin
module_name eMails_SPAM
module_type generic_data_inc
module_exec cat /var/log/mail/mail.log | grep "identified spam" | wc -l
module_end
```

2.4.7. Pandora Windows Agents

2.4.7.1. Build Windows Agent from sources

In order to build from sources, you will need the latest Dev-Cpp IDE version, with the MinGW tools. Download from http://www.bloodshed.net/devcpp.html

Open PandoraService.dev with Dev-Cpp and construct the project. Everything should compile fine in a default installation.

2.4.7.2. Windows Agent installation

Before running or installation of Pandora Windows service, you must create the configuration directory and extract the PandoraBin.zip file into it. It doesn't matter where it is installled, because Pandora Agent will adapt to any local directory. In the examples, the application will be installed in `C:\Pandora\`

This directory will hold the configuration files, which are:

```
c:\Pandora\pandora_agent.conf   :: Pandoramain configuration
c:\Pandora\id_dsa              :: Private SSH key
c:\Pandora\id_dsa.pub          :: Public SSH key
```

Notice: At this moment, the installation of the Pandora Windows Agent must be done manually. We are working in a auto-install package.

To install the Pandora Windows Agent execute this sentence in a Windows command line:

```
PandoraService.exe  --install
```

The Agent will be installed into the Windows services system. You can check it on Control Panel -> Administrative tools -> Services.

To run the Agent open the "Services" dialog (Control Panel -> Administrative tools-> Services), search the "Pandora Service" service and run it clicking the play button. To stop the service, open the "Services" dialog, search the "Pandora Service" and click the stop button.

To uninstall the Pandora Windows Agent, execute this sentence in a Windows command line:

```
PandoraService.exe --uninstall
```

2.4.7.3. Windows Agent testing

You can check the Pandora Windows Agent output in the `C:\babel\babel-debug.dbg` file, that is a plain text file and includes info about the execution flow of the Agent.

To test that SSH is working correctly, you can use the --test-ssh parameter in the executable file. This force babel to conect using internal SSH and copy a file called "ssh.test".

2.4.7.4. Windows Agent configuration

All setup is made in `babel_agent.conf`. This file is a list of keys/values pairs. Here is an example of this file.

```
# General Parameters
# ==================

server_ip 127.0.0.1
server_path /opt/pandora_server/data_in
temporal "D:\temp"
interval 1
agent_name localhost

# Module Definition
# =================


# Counting OpenedConnections (check the language string)
module_begin
module_name OpenNetConnections
module_type generic_data
module_exec netstat -na | grep ESTAB | wc -l | tr -d " "
module_description Conexiones abiertas (interval 2)
module_interval 2
module_end

# Is Schedule service running ?
module_begin
module_name ServicioProg
module_type generic_proc
module_service Schedule
module_description Servicio Programador de tareas
module_end

# Is Eventlog service running ?
module_begin
module_name ServicioReg
module_type generic_proc
module_service Eventlog
module_description Servicio Registro de sucesos
module_end

# Is lsass.exe process alive ?
module_begin
module_name Proc_lsass
module_type generic_proc
module_proc lsass.exe
module_description LSASS.exe process.
module_end
```

```
# Received packets.
       # Please notice that "Paquetes recibidos" string must be replaced by
       # the correct string in your Windows system language.
module_begin
module_name ReceivedPackets
module_type generic_data
module_exec netstat -s | grep  "Paquetes recibidos  "|
            tr -d " " | cut -f 2 -d "=" | tr -d "\n"
module_description Conexiones abiertas (interval 2)
module_end

# Free space on disk
module_begin
module_name FreeDiskC
module_type generic_data
module_freedisk C:
module_description Free space on drive C:
module_end

# CPU usage percentage
module_begin
module_name CPUUse0
module_type generic_data
module_cpuusage 0
module_description CPU#0 usage
module_end

module_begin
module_name FreeMemory
module_type generic_data
module_freememory
module_description Amount of free memory.
module_end
```

Chapter 3. Migrating from Pandora 1.1

3.1. Migrate Pandora 1.1 to Pandora 1.2

You need a script called pandoradb_1.1_to_1.2.sql provided in Pandora Console Package for 1.2 version. Migrate 1.1 to 1.2 versions includes:

- Full replacement of Pandora Console PHP Code (backup your original one to replace values in /include/config.php for your login, password, hostname, and database parameters.
- Full replacement of Pandora Server PERL Code (backyou your original one to replace values in pandora_server.conf. The new pandora_server.conf is located at /conf directory. It's possible that you need to change incoming dir, or parameters into daemon scripts. Please recheck Pandora Server 1.2 documentation before try to run it.
- Read documentation about new features of Pandora 1.2. Migration process doesnt delete any data, you don't loss any agente, config, or environment data, but please, read _carefully_ all documentation about this process before trying it.

3.2. Migration steps

0. You need to stop Pandora Server before change anything in database. Think that you don't lose any data in the process of migration because agent data will be stored in incoming directory. Move this data to the new incoming directory before launch the new server, and you process that data without miss anything.

1. DUMP all tagente_datos records to a safe file, for example

```
mysqldump --no-create-info -u root -p pandora \
          tagente_datos > /tmp/pandora.conv.tmp
```

2. Run migration SQL script, for example:

```
cat pandoradb_1.1_to_1.2.sql | mysql -u root -p -D pandora
```

PLEASE NOTE THAT this script WILL DROP your data table, so please DONT SKIP step #1.

3. Reimport data writen in first step, for example:

```
cat /tmp/pandora.conv.tmp | mysql -u root -p -D pandora
```

4. Delete temporal files, for example:

```
rm -Rf /tmp/pandora.conv.tmp
```

3.3. Code upgrade

Simply backup your Pandora Console and Pandora Server install and proceed as if Pandora 1.2 was a new installation. Use your configuration values your the new configuration files in Pandora and Server components. Database migration is the "heavy" task, and need to be made before using new code.

3.4. Final steps

Start your new Pandora components. Enter the WEB console and check that a new "Network Server" has been created. Please, edit each agent you have and assign this new server for each agent. This process could be automated using SQL, and you only need to do once. This only needs to be if you want to use Network Servers.

Chapter 4. Pandora FMS advanced section

4.1. Pandora FMS High Availability features

You may setup Pandora for use HA in several scenarios:

- *Database Clustering for HA*. You need to setup a MySQL5 Cluster. In support forums / wiki are information to make this, you only need to convert DB schema in a MySQL Cluster compatible tables. This scenario has been tested and works fine, but you need some advanced knowledge about MySQL Clustering administration.
- *Multiple Pandora Console*. It's easy, you only need to setup another one. No locking problems or incompatibility has been detected in several Pandora FMS deployments.
- *Multiple Pandora Data Servers for HA* . This is the more complex scenario, because you don't need to know anything special about Pandora Server setup, and you need to use of another tool to implement Network HA, like VRRP or Keepalive. For Pandora Data server you need to setup two identical machines, with the same public keys for all agents connecting to server (and duplicate server SSH host key). And setup Network HA to point one of them. If one fails, VRRP or Keepalive "promote" the other server up, and Pandora Agents, will connect it for the next data packets. There is no need to change anything in each of Pandora Data server, only need ensure that Pandora Server name is the same on both machines (in pandora server setup, not in the system hostname).
- *Multiple Pandora Network Servers for HA* . This is more easy. You need to setup multiple network servers in several machines across your network (or all of them in the same segment), and assign modules to the same server. If this servers fails, and there are other Network Servers active, marked as "primary" server, automatically, the first network server available marked as "Primary" will launch the network module query. If you have many servers marked as "primary", any of them could launch query.
- *Multiple Pandora Network Servers for load balancing.* . You need to setup multiple network servers in several machines across your network (or all of them in the same segment), and assign agent/modules to different servers, balancing load between all servers available.

4.2. Pandora virtual servers

An special case for implement more processing power in servers could be to implement "virtual" servers. Using virtual servers (another instance of the same server in the same machine) is used when Pandora Server cannot process all information without delay too much. Pandora 1.2 it's using a limited number of threads to process information (this will change in future versions), so you can install another instance of Pandora Network or Pandora Data server (with another data_in directory!), to be able to process more information with the same machine.

4.3. Pandora Database design (and redesign from 1.1)

First Pandora versions, from 0.83 until 1.1 was based on a simple idea: one data, one database insertion. This was very easy to develop and allowed to program easily searches, insertions and other operations.

This had many advantages and a big problem: the scalability. This system has a limit defined in maximum number of modules that could support in a "easy" way, from that number of modules the management was too slow.

Solutions based on MySQL cluster was difficult and cames with some problems and they did not offer either a solution in the long term.

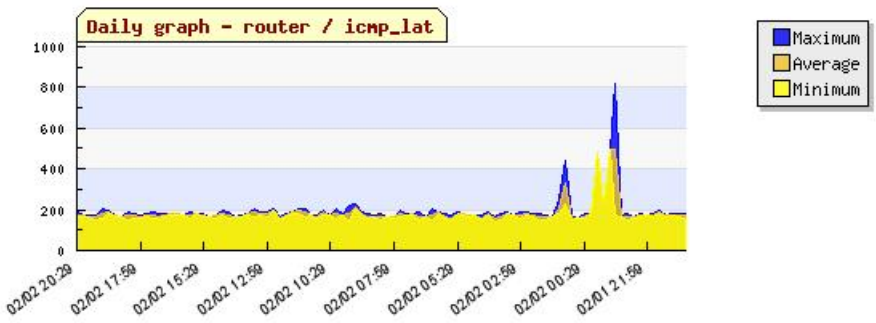
Data compression based on interpolation and data purge, makes a smaller database, but this was not enough. Production systems has a limit based on 100 agents, with about ten modules each one. This was not a high limit for large environments.

This problem was very important for Pandora future, so we changed the way Pandora store its data. New data management system store only "new" data. If a duplicate value enter the system, it won't be stored in database. It's very useful to keep database small. This works for all pandora data modules: numerical, incremental, boolean and string.

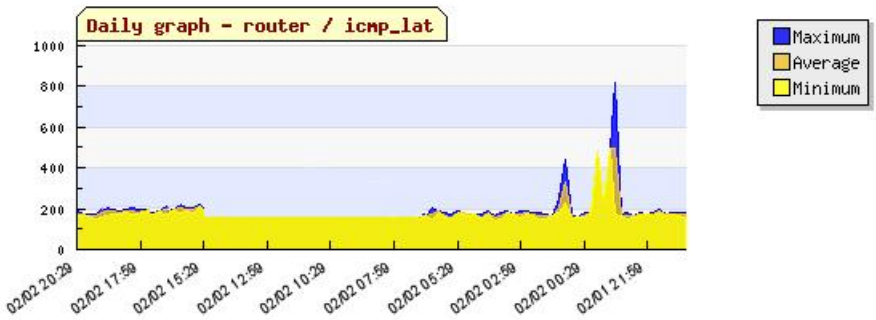
This solves part of scalability problem reducing considerably database usage, in about 40%-70%. We also have another solution for scalability problems: total segregation of components in Pandora and a built-in method to implement High Availability solutions on Pandora components. You may have many Pandora servers (network, data or SNMP), Pandora Consoles, and Pandora Database (in a MySQL5 Cluster setup).

Changes come with some different ways to reading data. With new version, if an agent cannot communicate with Pandora, and Pandora Server doesn't receive data from agent, this "no data" cannot have a graphical representation, for module graph there will be no changes. You will have a graph with a perfect horizontal line. Pandora, if doesn't receive new values, thinks that there are no new values, and everything seems to be as in the last notification.

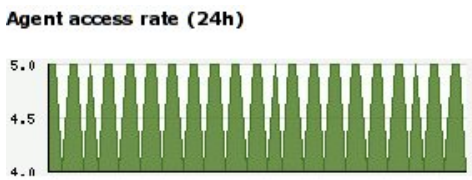
This graph, for example, shows changes for each data, received every 180 seconds.



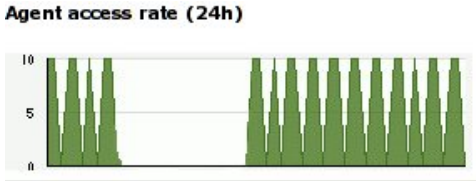
This would be the equivalent graph for the same data, except a connection failure, from 05:55 to 15:29 aproximately.



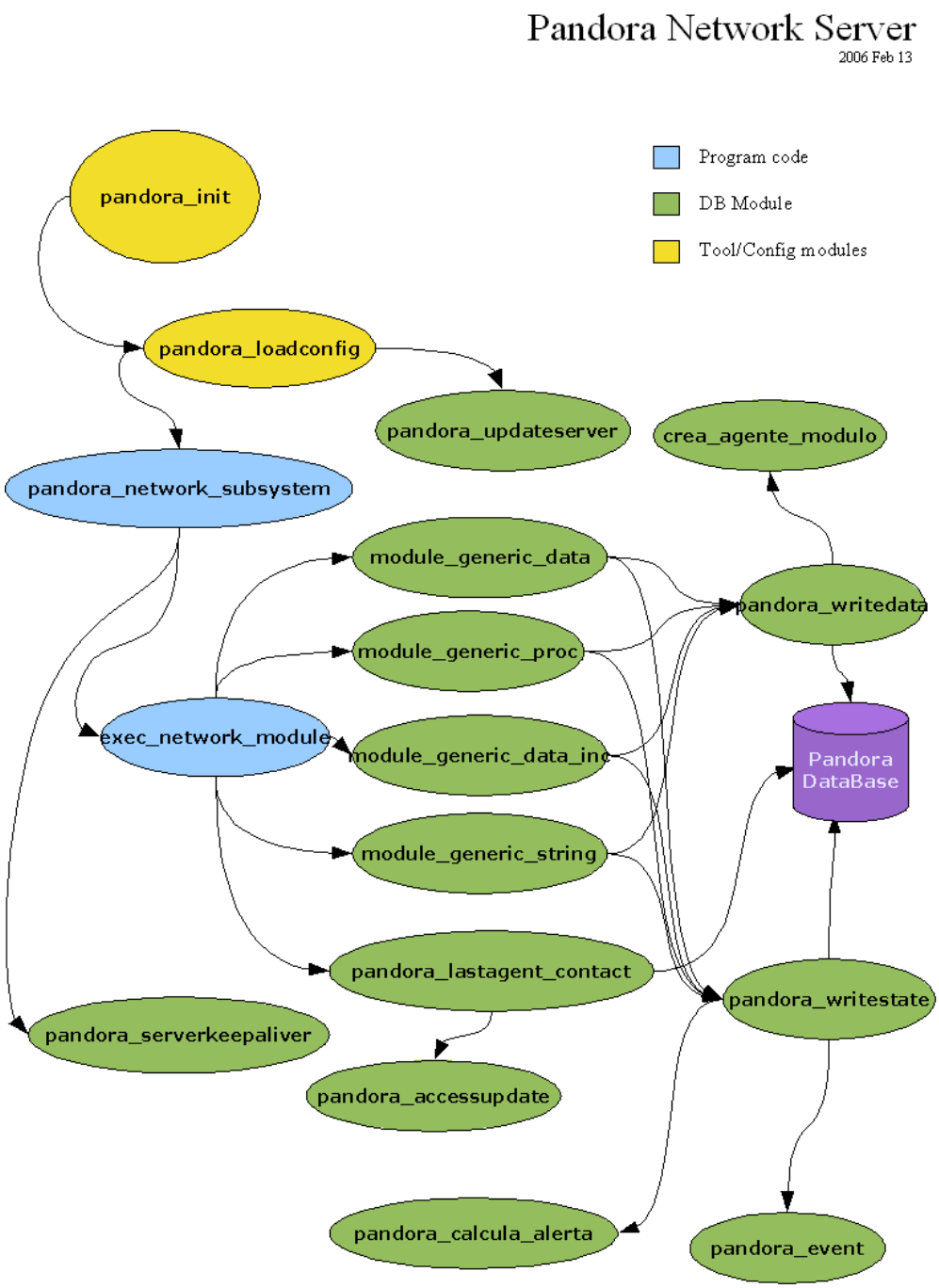
In Pandora 1.2 we introduce a new general agent graph for show connectivity. It reflects access from modules to this agent. This graph complements all other graphs showing when agent has activity and it's receiving data. This is an example of an agent connecting regulary to server:

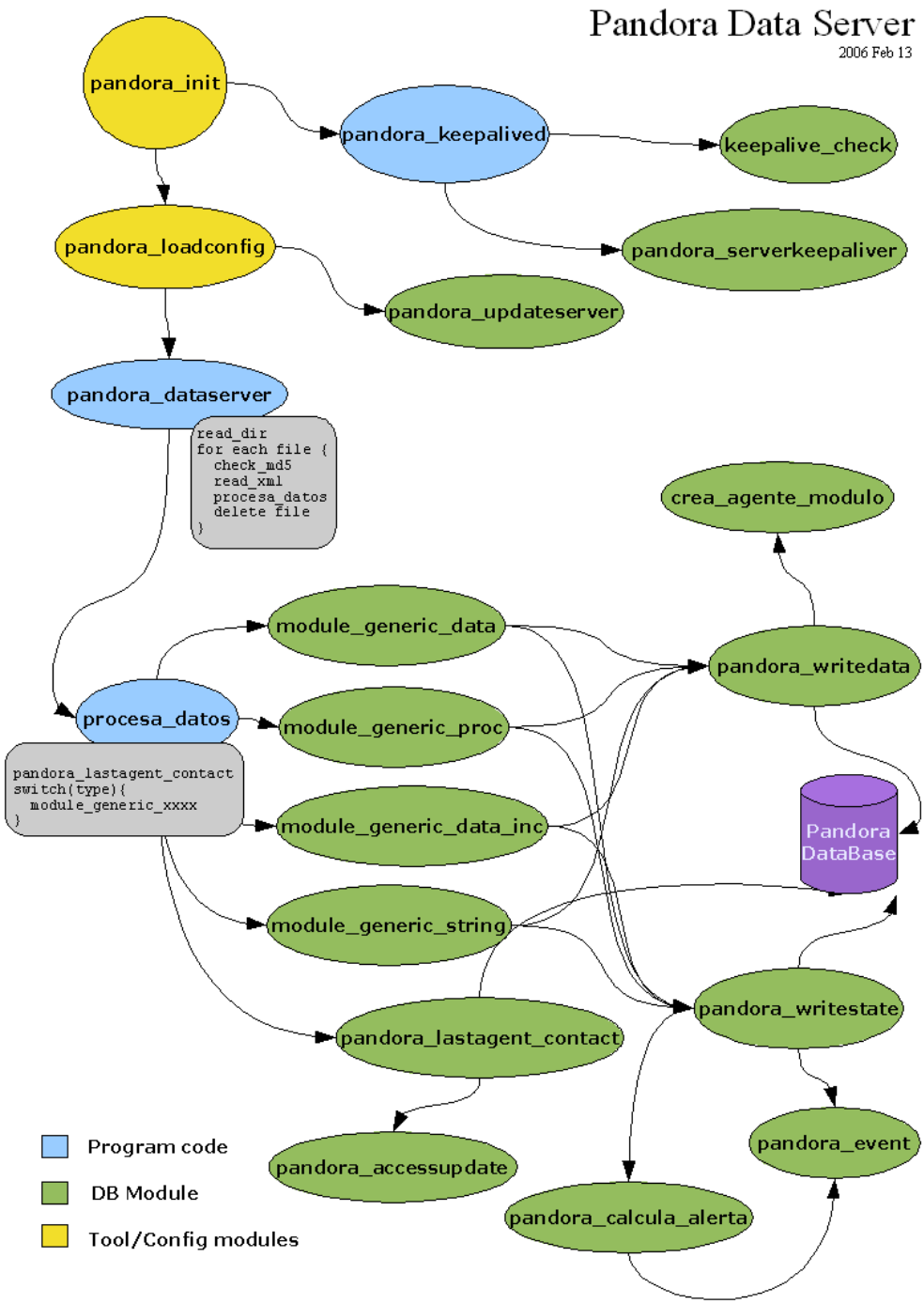


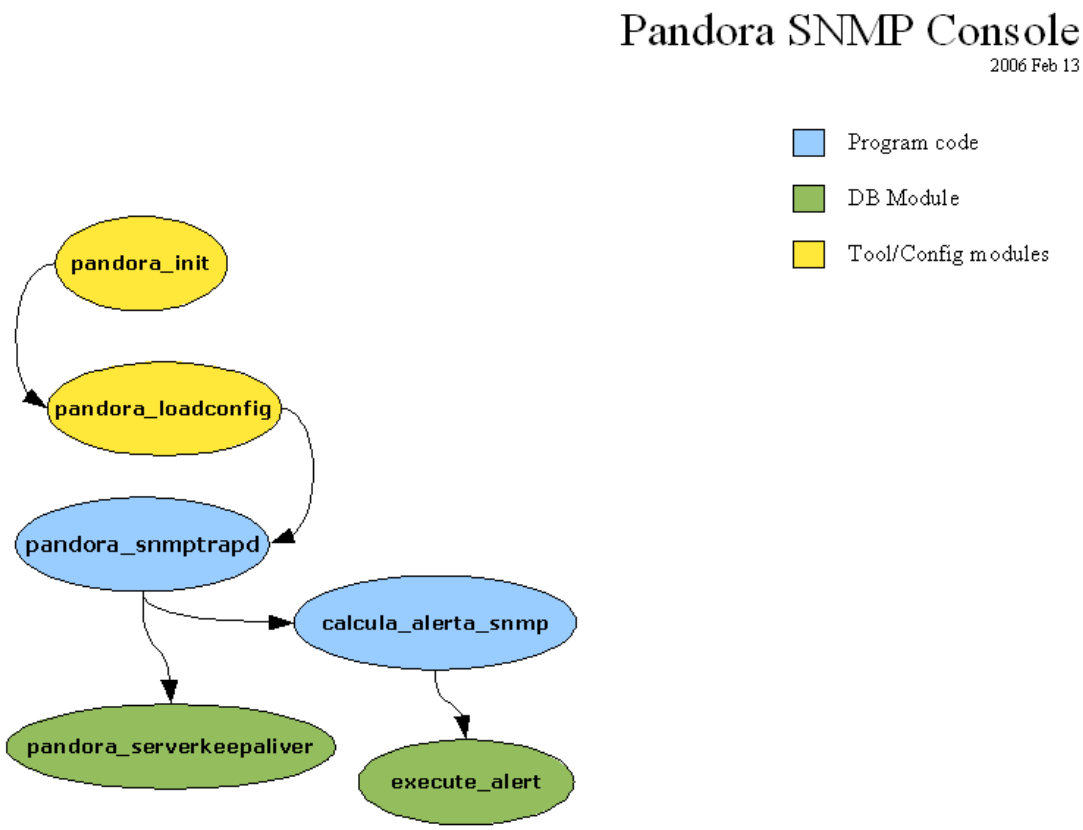
If you have low leaks in this graph, you may have some problems or slow connections in Pandora Agent connectivity with Pandora Server. This graph with previous example could have an aspect similar to this:

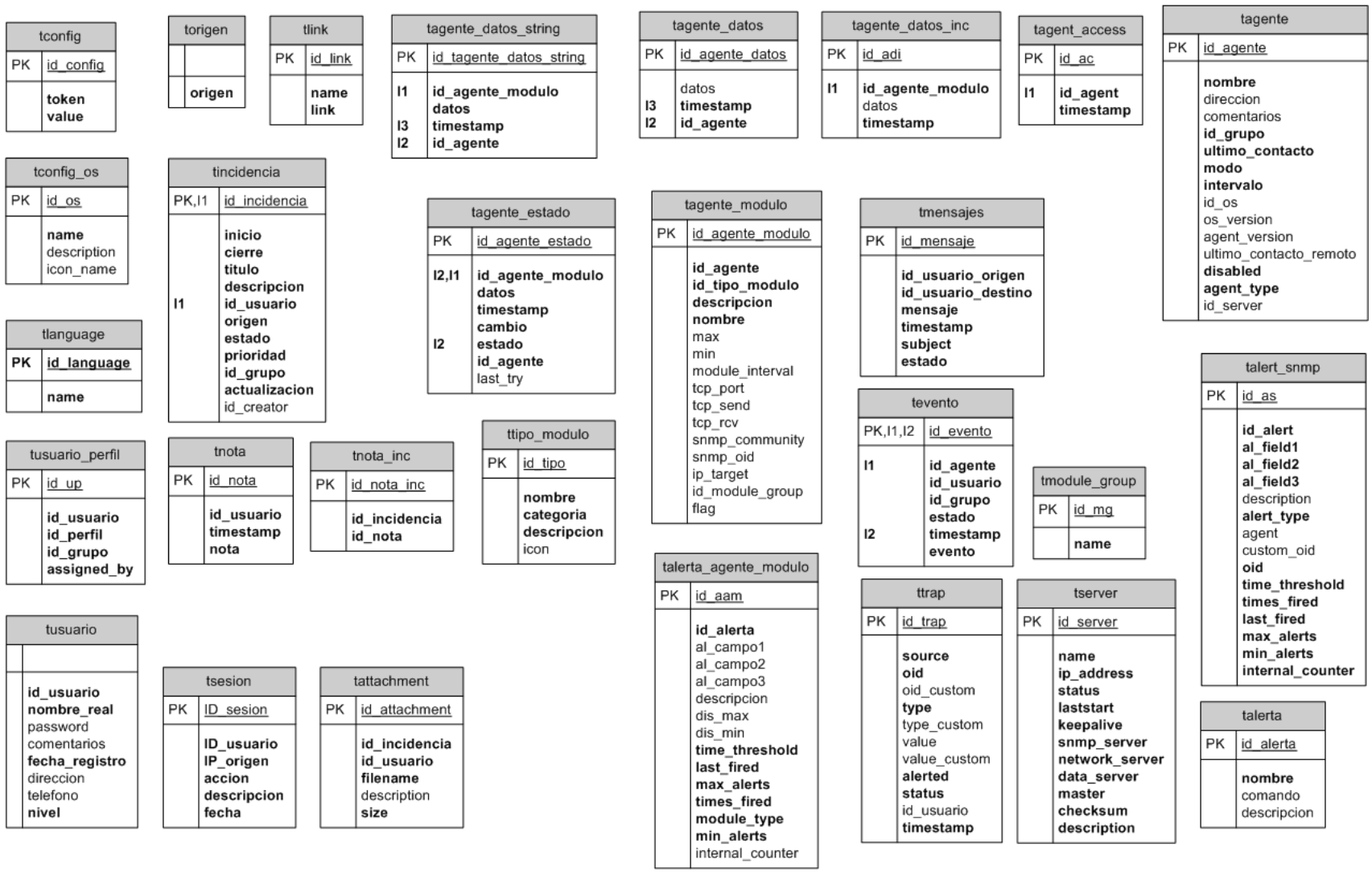

Agent access rate (24h)
10
5
0

4.4. Programmers guide to Pandora architecture




Pandora Data Server
2006 Feb 13
pandora_init
pandora_keepalived
keepalive_check
pandora_loadconfig
pandora_serverkeepaliver
pandora_updateserver
pandora_dataserver
read_dir
for each file {
check_md5
read_xml
procesa_datos
delete file
}
crea_agente_modulo
module_generic_data
pandora_writedata
procesa_datos
module_generic_proc
pandora_lastagent_contact
switch(type){
module_generic_xxxx
}
module_generic_data_inc
Pandora
DataBase
module_generic_string
pandora_writestate
pandora_lastagent_contact
pandora_event
pandora_accessupdate
pandora_calcula_alerta
Program code
DB Module
Tool/Config modules


Pandora SNMP Console
2006 Feb 13
Program code
DB Module
Tool/Config modules
pandora_init
pandora_loadconfig
pandora_snmptrapd
calcula_alerta_snmp
pandora_serverkeepaliver
execute_alert


tconfig
PK id_config
token
value
torigen
origen
tlink
PK id_link
name
link
tagente_datos_string
PK id_tagente_datos_string
I1 id_agente_modulo
datos
I3 timestamp
I2 id_agente
tagente_datos
PK id_agente_datos
datos
I3 timestamp
I2 id_agente
tagente_datos_inc
PK id_adi
I1 id_agente_modulo
datos
timestamp
tagent_access
PK id_ac
I1 id_agent
timestamp
tagente
PK id_agente
nombre
direccion
comentarios
id_grupo
ultimo_contacto
modo
intervalo
id_os
os_version
agent_version
ultimo_contacto_remoto
disabled
agent_type
id_server
tconfig_os
PK id_os
name
description
icon_name
tincidencia
PK,I1 id_incidencia
I1 inicio
cierre
titulo
descripcion
id_usuario
origen
estado
prioridad
id_grupo
actualizacion
id_creator
tagente_estado
PK id_agente_estado
I2,I1 id_agente_modulo
datos
timestamp
cambio
I2 estado
id_agente
last_try
tagente_modulo
PK id_agente_modulo
id_agente
id_tipo_modulo
descripcion
nombre
max
min
module_interval
tcp_port
tcp_send
tcp_rcv
snmp_community
snmp_oid
ip_target
id_module_group
flag
tmensajes
PK id_mensaje
id_usuario_origen
id_usuario_destino
mensaje
timestamp
subject
estado
talert_snmp
PK id_as
id_alert
al_field1
al_field2
al_field3
description
alert_type
agent
custom_oid
oid
time_threshold
times_fired
last_fired
max_alerts
min_alerts
internal_counter
tlanguage
PK id_language
name
tevento
PK,I1,I2 id_evento
I1 id_agente
id_usuario
id_grupo
estado
I2 timestamp
evento
tusuario_perfil
PK id_up
id_usuario
id_perfil
id_grupo
assigned_by
tnota
PK id_nota
id_usuario
timestamp
nota
tnota_inc
PK id_nota_inc
id_incidencia
id_nota
ttipo_modulo
PK id_tipo
nombre
categoria
descripcion
icon
tmodule_group
PK id_mg
name
talerta_agente_modulo
PK id_aam
id_alerta
al_campo1
al_campo2
al_campo3
descripcion
dis_max
dis_min
time_threshold
last_fired
max_alerts
times_fired
module_type
min_alerts
internal_counter
ttrap
PK id_trap
source
oid
oid_custom
type
type_custom
value
value_custom
alerted
status
id_usuario
timestamp
tserver
PK id_server
name
ip_address
status
laststart
keepalive
snmp_server
network_server
data_server
master
checksum
description
tusuario
id_usuario
nombre_real
password
comentarios
fecha_registro
direccion
telefono
nivel
tsesion
PK ID_sesion
ID_usuario
IP_origen
accion
descripcion
fecha
tattachment
PK id_attachment
id_incidencia
id_usuario
filename
description
size
talerta
PK id_alerta
nombre
comando
descripcion

Appendix A. GNU Free Documentation License

A.1. 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2. 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A.3. 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. 3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material

on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- **C.** State on the Title Page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.

- **I.** Preserve the section entitled “History”, and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** In any section entitled “Acknowledgements” or “Dedications”, preserve the section’s title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section entitled “Endorsements”. Such a section may not be included in the Modified Version.
- **N.** Do not retitle any existing section as “Endorsements” or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version .

A.6. 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms

defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled “History” in the various original documents, forming one section entitled “History”; likewise combine any sections entitled “Acknowledgements”, and any sections entitled “Dedications”. You must delete all sections entitled “Endorsements.”

A.7. 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and dispbibute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document , on account of their being thus compiled, if they are not themselves derivative works of the Document. If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

A.9. 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

A.10. 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document 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.

A.11. 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation (http://www.gnu.org/fsf/fsf.html) may publish new, revised versions of the GNU Free Documentation 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. See http://www.gnu.org/copyleft/ (http://www.gnu.org/copyleft).

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.12. Addendum

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME.

> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License (http://www.gnu.org/copyleft/gpl.html), to permit their use in free software.

Appendix B. GNU General Public License

B.1. 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.

B.2. TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

B.2.1. Section 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.

B.2.2. Section 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.

B.2.3. Section 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.

B.2.4. Section 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.

B.2.5. Section 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.

B.2.6. Section 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.

B.2.7. Section 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.

B.2.8. Section 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.

B.2.9. Section 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.

B.2.10. Section 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.

B.2.11. Section 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.

B.2.12. NO WARRANTY Section 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.

B.2.13. Section 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

B.3. How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

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

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

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

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

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

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary

applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.