

Ocs Inventory NG on MacOSX

What is this ?

I work as sysadmin and network engineer, fortunately networks where I work are not mine.

Often, in big companies, there is the necessity to inventory what is really attached to the network.

So, when I'm out for business I keep in my car just a couple of trashware computers, I install linux on-the-fly and I do the dirty work. In the past I worked also in a graphic lab where there were only macs, so I guess to install ocsinventory directly on a mac, without pcs."ok" I said, then I pointed to www.ocsinventory-ng.org and "ARGH, no Mac OSX support".

The challenge has started.

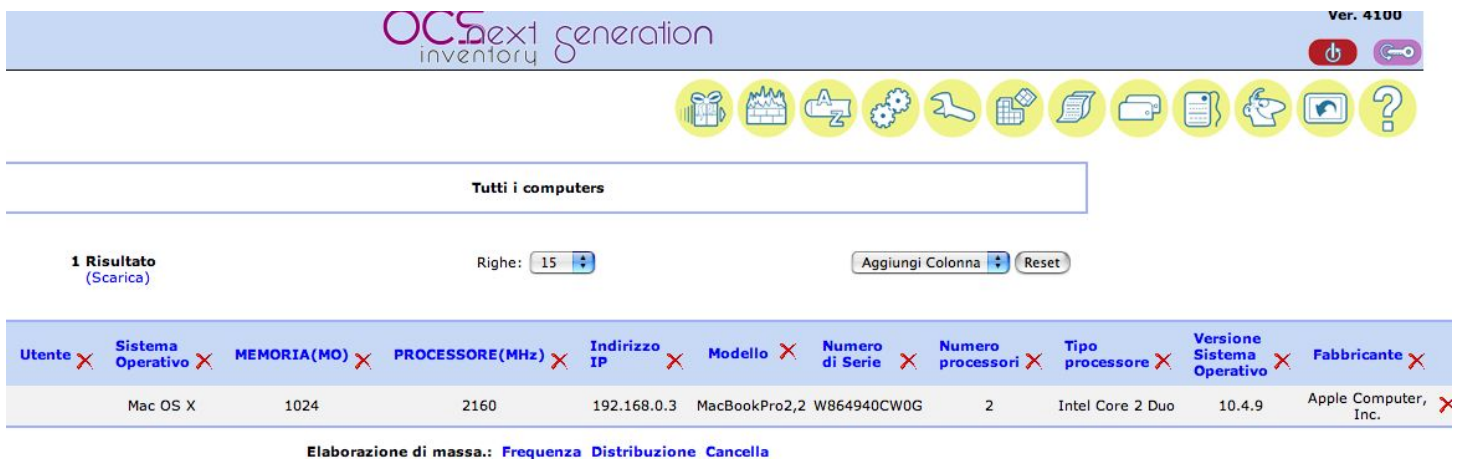
OcsInventory NG (<http://www.ocsinventory-ng.org>) is a software, developed by a french team that allows to inventory all the things that are attached to your network, PCs, Routers, Switches, Network Printers and everything that has a mac-address.

OcsInventory NG allows you to deeply analyze all the software installed on your clients, also in different networks.

And least but not last ocsinventory-ng is also capable to deploy other software, with a procedure of remote installation (not discussed here).

Obviously is possible to analyze machines not only with MacOsx installed, but also with different flavours of unix, and microsoft windows.

Figure 1, my Mac, in the inventory



The screenshot shows the OCS Inventory NG web interface. At the top, the logo "OCSnext generation inventory" is visible, along with the version "Ver. 4100" and a power button icon. Below the logo is a navigation bar with various icons representing different system components. The main content area is titled "Tutti i computers" and shows a search result for "Mac OS X". The table below lists the following details:

Utente	Sistema Operativo	MEMORIA(MO)	PROCESSORE(MHz)	Indirizzo IP	Modello	Numero di Serie	Numero processori	Tipo processore	Versione Sistema Operativo	Fabbricante
	Mac OS X	1024	2160	192.168.0.3	MacBookPro2,2	W864940CW0G	2	Intel Core 2 Duo	10.4.9	Apple Computer, Inc.

At the bottom of the table, there are options for "Elaborazione di massa.: Frequenza Distribuzione Cancella".

Ocs inventory is composed by two parts, a server and an agent.

The server-side of Ocs Inventory is built with a 3 tier technology.

The first tier is the communication server, that receives data from clients trough the agent.

Communication server takes the data and puts it into a database.

The second tier is the deployment server, basically a repository of files.

The last tier is the application server, the management interface of Ocs Inventory.

Look at the picture for further informations

Software requirements:

Ocs Inventory requires:

- MySQL version 4.1 or +
- PERL 5.6 or +
- Apache version 1.3.33 or +
- PHP 4.3.2 +
- A lot of perl modules.

Analyzing the problem:

At a glance, we can say the OCS Inventory ng is a software based on a AMP architecture (Apache, Mysql,Php) so, with a clean installations of MAMP, all the stuff should be work. Is not so.

MAMP has many problems with perl, because is redistributed with a versione of apache that does not have mod_perl

So we have to follow the Tafazzi way. (look at the picture, if you don't know who is Tafazzi, please document yourself on wikipedia)



Briefly,

- Use macosx built-in apache
- Manual installation of Php
- Manual installation of Mysql
- Mysql and PHP tricks
- Compilations of perl modules.

Project realization

Phase 1: Creation of an appropriate environment

some technical notes, when the font has the "terminal" style means that you have to digit lines into the Terminal.

At the end of each line there is "Return".

So just open Terminal, or a console and then let's start:

```
cd /etc/httpd/
```

then

```
sudo mkdir conf.d
```

if the system wants your password, please digit it, then

```
sudo chmod 755 conf.d
```

What we have done ?

We have created a new folder that our web server will read to load other configuration files, that I want to keep separated from httpd.conf, which is the main configuration file.

Phase 2: Modify httpd.conf

Now we have to open with write privileges the httpd.conf file, (I use vi) so from the Terminal

```
sudo vi httpd.conf
```

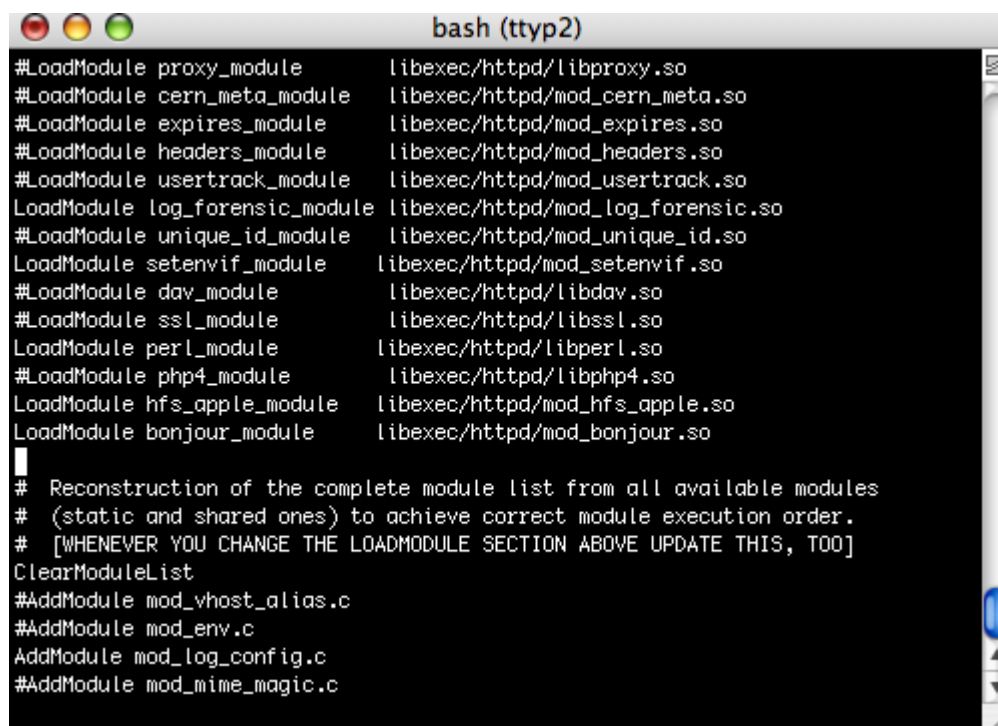
Let's comment the built-in php (if is already enables) so

we have to comment lines 240 and 284.

If you're using vi, it will be sufficient press the I key, insert the modifies and then press "ESC"

```
#LoadModule phnormale_module      libexec/httpd/libphnormale.so
```

```
#AddModule mod_phnormale.c
```

A terminal window titled "bash (tty2)" with a dark background and light text. It displays a list of Apache configuration commands for loading and adding modules. The commands are: #LoadModule proxy_module libexec/httpd/libproxy.so, #LoadModule cern_meta_module libexec/httpd/mod_cern_meta.so, #LoadModule expires_module libexec/httpd/mod_expires.so, #LoadModule headers_module libexec/httpd/mod_headers.so, #LoadModule usertrack_module libexec/httpd/mod_usertrack.so, LoadModule log_forensic_module libexec/httpd/mod_log_forensic.so, #LoadModule unique_id_module libexec/httpd/mod_unique_id.so, LoadModule setenvif_module libexec/httpd/mod_setenvif.so, #LoadModule dav_module libexec/httpd/libdav.so, #LoadModule ssl_module libexec/httpd/libssl.so, LoadModule perl_module libexec/httpd/libperl.so, #LoadModule php4_module libexec/httpd/libphp4.so, LoadModule hfs_apple_module libexec/httpd/mod_hfs_apple.so, LoadModule bonjour_module libexec/httpd/mod_bonjour.so. Below these is a comment: # Reconstruction of the complete module list from all available modules # (static and shared ones) to achieve correct module execution order. # [WHENEVER YOU CHANGE THE LOADMODULE SECTION ABOVE UPDATE THIS, TOO] followed by ClearModuleList, #AddModule mod_vhost_alias.c, #AddModule mod_env.c, AddModule mod_log_config.c, and #AddModule mod_mime_magic.c.

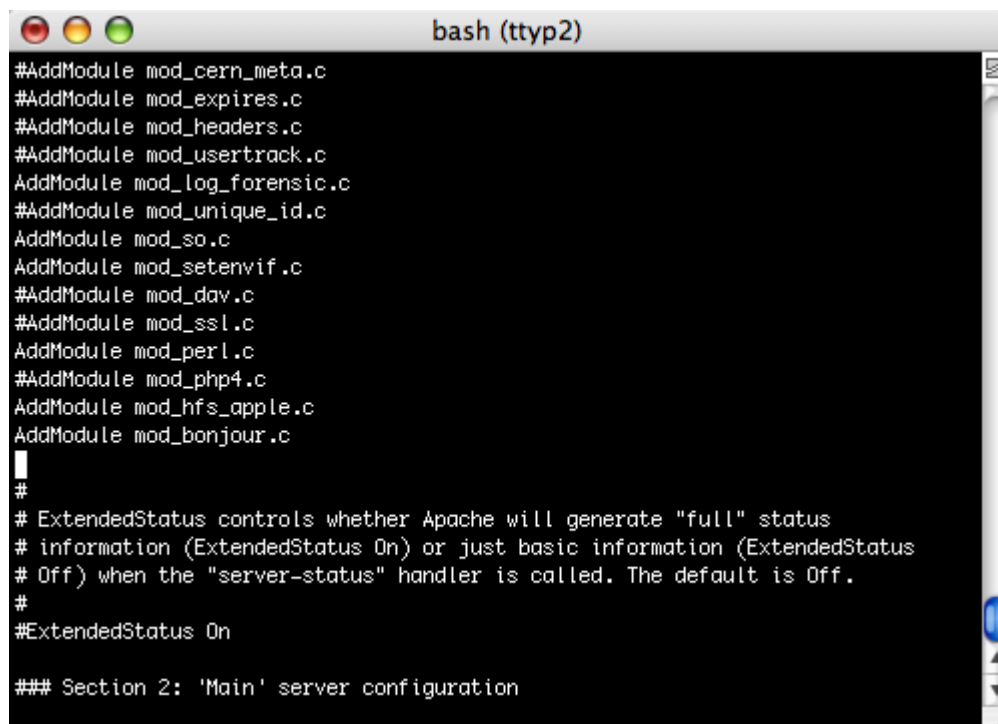
```
bash (tty2)
#LoadModule proxy_module      libexec/httpd/libproxy.so
#LoadModule cern_meta_module  libexec/httpd/mod_cern_meta.so
#LoadModule expires_module    libexec/httpd/mod_expires.so
#LoadModule headers_module    libexec/httpd/mod_headers.so
#LoadModule usertrack_module  libexec/httpd/mod_usertrack.so
LoadModule log_forensic_module libexec/httpd/mod_log_forensic.so
#LoadModule unique_id_module  libexec/httpd/mod_unique_id.so
LoadModule setenvif_module    libexec/httpd/mod_setenvif.so
#LoadModule dav_module        libexec/httpd/libdav.so
#LoadModule ssl_module        libexec/httpd/libssl.so
LoadModule perl_module        libexec/httpd/libperl.so
#LoadModule php4_module       libexec/httpd/libphp4.so
LoadModule hfs_apple_module   libexec/httpd/mod_hfs_apple.so
LoadModule bonjour_module     libexec/httpd/mod_bonjour.so
#
# Reconstruction of the complete module list from all available modules
# (static and shared ones) to achieve correct module execution order.
# [WHENEVER YOU CHANGE THE LOADMODULE SECTION ABOVE UPDATE THIS, TOO]
ClearModuleList
#AddModule mod_vhost_alias.c
#AddModule mod_env.c
AddModule mod_log_config.c
#AddModule mod_mime_magic.c
```

now we have to uncomment lines 239 and 283

If you're using vi, it will be sufficient press the "x" key

```
LoadModule perl_module      libexec/httpd/libperl.so
```

```
AddModule mod_perl.c
```

A terminal window titled "bash (tty2)" with a dark background and light text. It displays a list of Apache configuration commands for adding modules. The commands are: #AddModule mod_cern_meta.c, #AddModule mod_expires.c, #AddModule mod_headers.c, #AddModule mod_usertrack.c, AddModule mod_log_forensic.c, #AddModule mod_unique_id.c, AddModule mod_so.c, AddModule mod_setenvif.c, #AddModule mod_dav.c, #AddModule mod_ssl.c, AddModule mod_perl.c, #AddModule mod_php4.c, AddModule mod_hfs_apple.c, and AddModule mod_bonjour.c. Below these is a comment: # # ExtendedStatus controls whether Apache will generate "full" status # information (ExtendedStatus On) or just basic information (ExtendedStatus # Off) when the "server-status" handler is called. The default is Off. # #ExtendedStatus On. At the bottom, it shows ### Section 2: 'Main' server configuration.

```
bash (tty2)
#AddModule mod_cern_meta.c
#AddModule mod_expires.c
#AddModule mod_headers.c
#AddModule mod_usertrack.c
AddModule mod_log_forensic.c
#AddModule mod_unique_id.c
AddModule mod_so.c
AddModule mod_setenvif.c
#AddModule mod_dav.c
#AddModule mod_ssl.c
AddModule mod_perl.c
#AddModule mod_php4.c
AddModule mod_hfs_apple.c
AddModule mod_bonjour.c
#
#
# ExtendedStatus controls whether Apache will generate "full" status
# information (ExtendedStatus On) or just basic information (ExtendedStatus
# Off) when the "server-status" handler is called. The default is Off.
#
#ExtendedStatus On

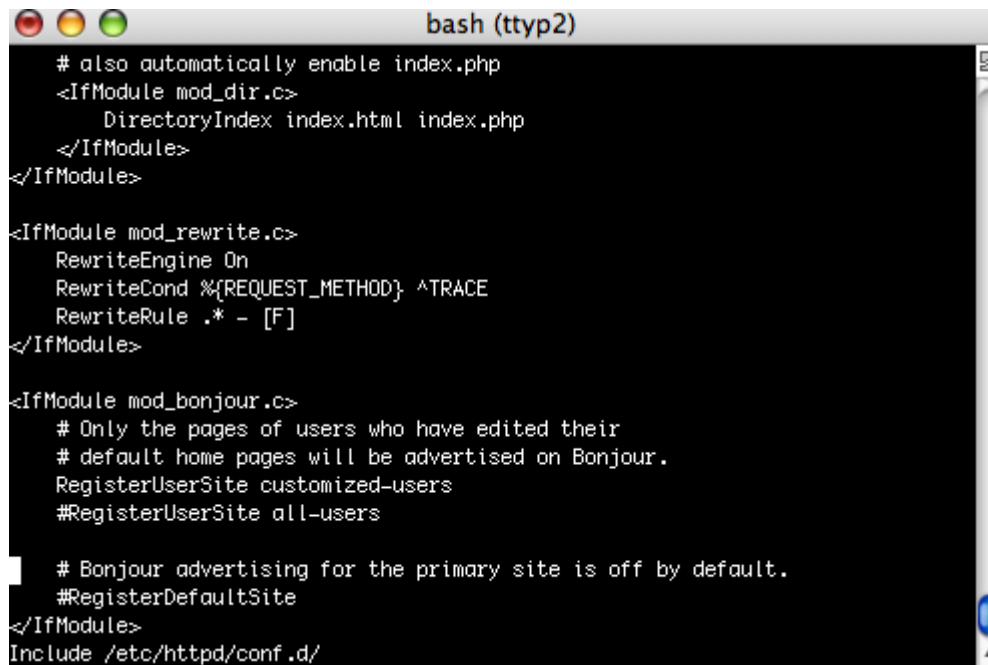
### Section 2: 'Main' server configuration
```

then, scroll down

and we have to add to the bottom of the file the following line

If you're using vi, it will be sufficient press the "i" key (insert), adding the line, and then press "esc"

```
Include /etc/httpd/conf.d/
```



```
bash (tty2)
# also automatically enable index.php
<IfModule mod_dir.c>
    DirectoryIndex index.html index.php
</IfModule>
</IfModule>

<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteCond %{REQUEST_METHOD} ^TRACE
    RewriteRule .* - [F]
</IfModule>

<IfModule mod_bonjour.c>
    # Only the pages of users who have edited their
    # default home pages will be advertised on Bonjour.
    RegisterUserSite customized-users
    #RegisterUserSite all-users

    # Bonjour advertising for the primary site is off by default.
    #RegisterDefaultSite
</IfModule>
Include /etc/httpd/conf.d/
```

save and exit.

With vi

```
:wq!
```

very good.

What we have done ?

We have disabled the built-in php, because this version of php has some problems, then we have specified that we want mod_perl enabled. Mod_perl is a piece of software that allow to interface directly perl scripts with the webserver.

Then we have added a line with and Include, to specifies where there are other configuration files.

Phase 3: Mysql download, installations and tricks

Let's download from Mysql site, the last version of Mysql for MacOSx version in pkg format.

<http://dev.mysql.com/downloads/mysql/5.0.html#macosx>

Mac OS X (package format) downloads (platform notes)

Mac OS X 10.3 (PowerPC, 32-bit)	5.0.41	65.3M	Download Pick a mirror MD5: 4f31c8f3eaf48a22fe71540882047a1f Signature
Mac OS X 10.4 (PowerPC, 32-bit)	5.0.41	61.6M	Download Pick a mirror MD5: 387e878e4273adee1abb55e988bbe7c6 Signature
Mac OS X 10.4, (PowerPC, 64-bit)	5.0.27	41.6M	Download Pick a mirror MD5: 32d22db810f517c07c86876cdd1b83f1 Signature
Mac OS X 10.4 (x86)	5.0.41	60.1M	Download Pick a mirror MD5: e5785f2e442355cb4075ef1c2889f636 Signature

Mac OS X (TAR packages) downloads (platform notes)

Mac OS X 10.3 (PowerPC, 32-bit)	5.0.41	65.2M	Download Pick a mirror MD5: 247b86c439d04fb4c5ef9e9f593be9d1 Signature
Mac OS X 10.4 (PowerPC, 32-bit)	5.0.41	61.9M	Download Pick a mirror MD5: ddbd6a06b9737281f918458e03126235 Signature
Mac OS X 10.4 (PowerPC, 64-bit)	5.0.27	41.9M	Download Pick a mirror MD5: c922ec57c9d12302704bfcd80044875e Signature
Mac OS X 10.4 (x86)	5.0.41	60.1M	Download Pick a mirror MD5: ba63239484e181f68ec5b67fc307d289 Signature

let's install Mysql, via pkg.

At this point I often use MysqlAdministrator.

<http://dev.mysql.com/downloads/gui-tools/5.0.html>

Editing Mysql:

Open Terminal

```
cd /usr/local/mysql/lib
```

```
sudo mkdir mysql
```

Insert your password

```
sudo cp ../* .
```

We have to do this due to a bug with paths, it's not elegant, but it works.

What we have done ?

Abbiamo installato una versione aggiornata di Mysql, e abbiamo copiato le librerie di mysql, in un'altra directory perchè un modulo di perl ha un bug nella gestione dei percorsi.

Phase 4: Download Ocsinventory-ng

Aye Aye, now you can go dinner. Or continue.

Now, we have to download OCSInventory-ng, linux version, of course.

http://prdownloads.sourceforge.net/ocsinventory/OCSNG_LINUX_SERVER_1.01.tar.gz?download

The packet is in tar.gz format

Let's decompress the packet, even on Desktop, and now we must modify the setup.sh script.

The script is quite simple, let's see.

Phase 5: Editing setup.sh

setup.sh verifies if the system is able to support OCSinventory-ng, then edits apache's files. The script also install administration console and communication server.

Let's open setup.sh with a text editor, possibly with coloured syntax; I use smultron, that is free, or if you prefer you can use textwrangler, textmate, pico and so on.

Let's change lines 91 and 92 where is specified mysql.

```
83 |
84 | echo
85 | echo "+-----+"
86 | echo "| Checking for database server properties... |"
87 | echo "+-----+"
88 | echo
89 | # Check mysql client distribution version
90 | echo "Checking for database server properties" >> $SETUP_LOG
91 | DB_CLIENT_MAJOR_VERSION=`eval /usr/local/mysql/bin/mysql -V | cut -d' ' -f6 | cut -d'.' -f1` >> $SETUP_LOG 2>&1
92 | DB_CLIENT_MINOR_VERSION=`eval /usr/local/mysql/bin/mysql -V | cut -d' ' -f6 | cut -d'.' -f2` >> $SETUP_LOG 2>&1
93 | echo "Your MySQL client seems to be part of MySQL version $DB_CLIENT_MAJOR_VERSION.$DB_CLIENT_MINOR_VERSION."
94 | echo "MySQL client distribution version $DB_CLIENT_MAJOR_VERSION.$DB_CLIENT_MINOR_VERSION." >> $SETUP_LOG
95 | # Ensure mysql distribution is 4.1 or higher
```

MySQL does not exist in the system path, because its binary is in another folder (/usr/local/mysql), so we have to specify this link in this figure.

At this point, we have to comment some lines, that modifies logrotate, this is not used by OSX.

So, in bash style, we have to comment lines from 795 to 818 as shown.

```

792 |     fi
793 |     echo "Configuring logrotate for Communication server."
794 |     echo "Configuring logrotate (ed logrotate.ocsinventory-NG)" >> $SETUP_LOG
795 |     # cp logrotate.ocsinventory-NG logrotate.ocsinventory-NG.local
796 |     ##
797 |     ## Now using perl to replace string in file instead of ed, not available by default in Mandriva Linux
798 |     ##
799 |     ## $ED_BIN logrotate.ocsinventory-NG.local << EOF >> $SETUP_LOG 2>&1
800 |     ##     1,$ g/^ *PATH_TO_LOG_DIRECTORY*/s#PATH_TO_LOG_DIRECTORY#$OCS_COM_SRV_LOG#
801 |     ##     w
802 |     ##     q
803 |     ##EOF
804 |     # $PERL_BIN -pi -e "s#PATH_TO_LOG_DIRECTORY#$OCS_COM_SRV_LOG#g" logrotate.ocsinventory-NG.local
805 |     # echo "***** Begin updated logrotate.ocsinventory-NG *****" >> $SETUP_LOG
806 |     # cat logrotate.ocsinventory-NG.local >> $SETUP_LOG
807 |     # echo "***** End updated logrotate.ocsinventory-NG *****" >> $SETUP_LOG
808 |     # echo "Writing communication server logrotate to file $LOGROTATE_CONF_DIR/ocsinventory-NG"
809 |     # echo "Writing communication server logrotate to file $LOGROTATE_CONF_DIR/ocsinventory-NG" >> $SETUP_LOG
810 |     # cp -f logrotate.ocsinventory-NG.local $LOGROTATE_CONF_DIR/ocsinventory-NG >> $SETUP_LOG 2>&1
811 |     # if [ $? != 0 ]
812 |     # then
813 |     #     echo "*** ERROR: Unable to configure log rotation, please look at error in $SETUP_LOG and fix !"
814 |     #     echo
815 |     #     echo "Installation aborted !"
816 |     #     exit 1
817 |     # fi
818 |     echo
819 |

```

save and exit setup.sh

Phase 6: Perl modules compilation

Now we can compile some perl modules:

There are 2 ways to do this:

- The first one is using cpan
- The second is using darwinports, macports, fink etc.

I choose to first

Let's explain a bit:

cpan (the command) is a command line utility that allow to download and install perl modules, directly from cpan repository. So the command cpan takes the latest version, downloads it, decompress it, configures it and then compiles and installs it.

We have to install some modules.

if the system ask for dependencies we must answer yes.

Little note on darwinports, fink etc:

I didn't choose this way because I'm not sure how those packet integrates with the system. Often those packet works in /opt/local/ and anyway at least one module is to be recompiled.

So let's get it started:

-Open Terminal

If you have already used cpan, there will be no problems, otherwise you may have to answer to a couple of questions, for the first startup.

```
sudo cpan -i Compress::Zlib
```

if the packet gives no errors, we can continue with

```
sudo cpan -i DBI
```

if the packet gives no errors, we can continue with

```
sudo cpan -i Net::IP
```

if the packet gives no errors, we can continue with

```
sudo cpan -i Apache::DBI
```

if the packet gives no errors, we can continue with

```
sudo cpan -i XML::Simple
```

XML probably ask for installation of XML::Namespaces, XML::Sax, XML::Sax-Expat, XML::Parser

Very well.

Now the harder part. Two modules a little bit more complicated: SOAP::Lite, and DBD::Mysql

first SOAP::Lite

```
sudo cpan -i SOAP::Lite
```

The system will work for a bit, if the system stops on make-test is because of some problems.

To resolve the problem, we have to force a bit the system.

```
sudo su
```

insert your password

****** You're working as a root be careful ! ******

Let's move in the folder `/var/root/.cpan/build/`

```
cd /var/root/.cpan/build/
```

searching for a folder named SOAP-Lite-x.xx

```
cd SOAP-Lite*
```

at this point digit "make"

```
make
```

followed by

```
make install
```

```
exit
```

Phase 7: Installation of DBD::mysql

At this point, I'll go to sleep

Let's manually download the module form here

<http://search.cpan.org/CPAN/authors/id/C/CA/CAPTTOFU/DBD-mysql-4.004.tar.gz>

in Terminal windows let's digit

```
sudo su
```

insert your password

```
**** ** You're working as a root be careful ! ** **
```

Let's move in the folder `/var/root/.cpan/build/`

```
cd /var/root/.cpan/build/
```

then we have to copy from desktop the file downloaded

```
cp /Users/<yourusername>/Desktop/DBD-mysql-4.004.tar.gz .
```

decompress

```
tar fxvz DBD-mysql-4.004.tar.gz
```

moving to the folder named `DBD-mysql-x.xx`

```
cd DBD-mysql*
```

what we are doing:

we're going to configure manually the packet, specifying where mysql libraries are.

```
perl MakeFile.PL --mysql_config=/usr/local/mysql/bin/mysql_config
```

followed by

```
make
```

now

make test

Phase 8 : Errors in the compilation phase

giving make test, there are two common errors:

- make test does not found mysql socket
- make test has socket problems,because of a pre-existent mysql installation with root password not-empty

* Resolution to Can't connect to local MySQL server through socket '/tmp/mysql.sock'

How to resolve this.

First of all, the sock file, is a file that is used by php,perl and other programs to connect at a database instance. the sock file is a sort of "database device" and indicates that the service is working, is active and we can connect to a service through this file.

from osx 10.4 there is a bit of confusion regarding where this file must be.

Some people have seen it in /var/mysql/mysql.sock , other say that must be in /tmp/mysql.sock

I added a soft-link between /var/mysql/mysql.sock and /tmp/mysql.sock so there are no problems. Is not so elegant but it works.

There is an interesting article on tevac about this.

<http://www.tevac.com/article.php/2006012513085364>

```
cd /etc/
```

```
sudo vi my.cnf
```

and then I insert in my.cnf

```
[mysqld]
```

```
socket=/var/mysql/mysql.sock
```

save and exit

at this point

```
sudo mkdir /var/mysql/
```

```
chmod 755 /var/mysql/
```

restart mysql, I use MySQL Administrator

```
cd /tmp
```

```
sudo ln -s /var/mysql/mysql.sock mysql.sock
```

* Resolution at a connection problem

make test want a mysql root user without password.

So temporary substitute the mysql root password, I use MySQL Administrator

If the make test gives no errors you can give

```
make install
```

very very well.

```
exit
```

Phase 9 : Packet Installation:

Form terminal

```
cd ~/Desktop/OCS*
```

and now we have to start setup.sh

the script could ask for apache's binary

```
/usr/sbin/httpd
```

the script could ask for apache's binary where is the apache's configuration file

```
/etc/httpd/httpd.conf
```

the script could ask for user, let's digit "www", and group "www"

the script could ask for apache's include directory

```
/etc/httpd/conf.d/
```

If the script stops for some missing perl modules, write down the name, then using cpan install it.

```
sudo cpan -i <modulename>
```

Answer yes to the other questions,when the script ask for DocumentRoot, digit

```
/Library/WebServer/Documents
```

the script ask for an apache restart.

At this point, let's install the php packet provided by Marc

<http://www2.entropy.ch/download/entropy-php-5.2.2-1.tar.gz>

stop Apache, stop mysql, restart all

Phase 10: First step with administration console

Open Safari and digit

<http://localhost/ocsreports/>

Let's start configuration.

Remember to set mysql root password now

Insert username and password

The system generates table, press enter.

**** Errors**

`/var/mysql/mysql.sock` not found

verify if the socket exist, and look for socket not found in this document.

the system has no permission to use the socket

check `/var/mysql/mysql.sock` permission

And here is the Administration Console (user: admin pass: admin)

Phase 11 : First Inventory

Let's install the MacOSx agent

thanks to codejanitor, we can use an agent for macosx

<http://codejanitor.com/wp/apps/ocs-mac-agent/>

Install the pkg, then with terminal we have to move into the folder `/etc/ocsinventory-client/`

```
cd /etc/ocsinventory-client/
```

let's digit

```
sudo vi ocsinv.conf
```

and we have to change the file in the line OCSFSERVER, substituting ocsinventory-ng with the ip-address or the dns name of the machine that has the server installed, in our case, we can write in this way.

```
<OCSFSERVER>localhost</OCSFSERVER>
```

or

<OCSFSERVER>192.168.0.1</OCSFSERVER>

or the ip-address correct.

let's force the packet to do an inventory, let's digit

```
sudo php /usr/local/sbin/ocs_mac_agent.php
```

The agent must respond with a 200 message.

Errors

server gives 500

```
verify /var/log/httpd/error.log
```

Probably is a problem of DBI , or DBD

GIYF

Phase 12: Results

If all things goes well

1) handshake yourself



2) digit <http://localhost/ocsreports> and you'll see your Mac

Ahr.

Other things:

Other ways:

There are many way to do what we have done.

This is essential:

- * Apache with mod_perl.
- * Mysql with include directory for correct compilation of DBD:mysql
- * Php, possibly the Entropy's one.

Debug:

files to check:

`/var/log/httpd/error.log`

`/var/log/ocsinventory-ng/ocsinventory-ng.log`

Thanks to:

If anything gives problem feel free to contact me.

I also want to thank my wife and myself.

And thanks to Tevac.

07/2007