OCS Inventory NG:Documentation - OCS Inventory NG

- [OCS Inventoiry NG Home]
- [OCS Inventory NG Forums]

OCS Inventory NG:Documentation

From OCS Inventory NG

Jump to: navigation, search

	Languages:	**English** • Español • Français

Contents

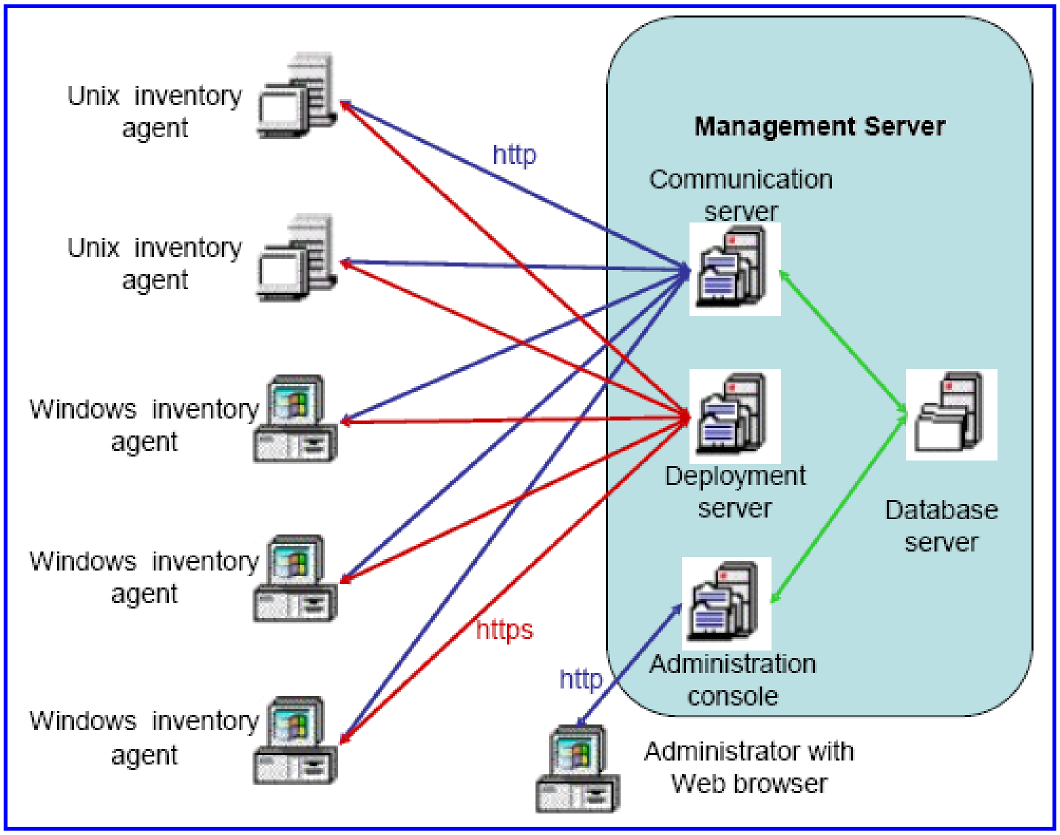
[hide]

Setting up management server.

Management server is made up of 4 main components:

1. **Database server**, which stores inventory information
2. **Communication server**, which handles HTTP communications between database server and agents.
3. **Administration console**, which allows administrators querying the database server using their favorite browser.
4. **Deployment server**, which stores all package deployment configuration (require HTTPS!)

These 4 components can be hosted on a single computer or on different computers to allow load balancing. Above 10000 inventoried computers, we recommend using at least 2 physical servers, one hosting database server + Communication server and the other one hosting a database replica + Administration server + Deployement server.



OCS Inventory NG communication architecture.

NB: If you want to use multiple computers to host OCS inventory NG management server, we recommend that you set it up on Linux servers. OCS Inventory NG server for Windows comes as an integrated package including all required components (apache, perl, php, mod_perl, mysql…).

Database server currently can only be MySQL 4.1 or higher with InnoDB engine active.

Communication server needs Apache Web Server 1.3.X/2.X and is written in PERL as an Apache module. Why? Because PERL scripts are compiled when Apache starts, and not at each request. This is better performance-wise. Communication server may require some additional PERL modules, according to your distribution.

Deployment server needs any Web Server with SSL enabled.

Administration console is written in PHP 4.1 (or higher) and runs under Apache Web Server 1.3.X/2.X. Administration console requires ZIP and GD support enabled in PHP in order to use package deployment.

Under Linux Operating System.

We assume that you have:

- MySQL database server running somewhere and listening on default port 3306 with TCP/IP communication enabled.
- Apache Web server installed and running for Communication server and Administration server.
- PHP and Perl installed and usable by Apache Web server for the Administration console.
- Perl and mod_perl installed and usable by Apache Web server for the Communication server.

Requirements.

- Apache version 1.3.33 or higher / Apache version 2.0.46 or higher.
  - Mod_perl version 1.29 or higher.
  - Mod_php version 4.3.2 or higher.

- PHP 4.3.2 or higher, with ZIP and GD support enabled.
- PERL 5.6 or higher.
  - Perl module XML::Simple version 2.12 or higher.
  - Perl module Compress::Zlib version 1.33 or higher.
  - Perl module DBI version 1.40 or higher.
  - Perl module DBD::Mysql version 2.9004 or higher.
  - Perl module Apache::DBI version 0.93 or higher.
  - Perl module Net::IP version 1.21 or higher.
  - Perl module SOAP::Lite version 0.66 or higher (not mandatory)

- MySQL version 4.1.0 or higher with InnoDB engine active.
- Make utility such as GNU make.

NB: OCS Inventory NG Server Setup will check for all these components and will exit if any is missing.

Installing Communication server required PERL modules.

The Web communication server requires Apache web server and Perl 5 scripting language and some additional modules for Perl 5 (see Requirements). It acts as an Apache module which handles HTTP OCS Inventory agents' requests to a virtual directory “/ocsinventory”.

You must have root privileges to set required perl modules up.

It is better for system integrity to use your distribution's precompiled packages when they are available.

On Fedora/Redhat like Linux, you can use “yum” to set required modules up:

- yum install perl-XML-Simple
- yum install perl-Compress-Zlib
- yum install perl-DBI
- yum install perl-DBD-MySQL
- yum install perl-Apache-DBI
- yum install perl-Net-IP
- yum install perl-SOAP-Lite

On Debian like Linux, you can use “apt-get” to set required modules up:

- apt-get install libxml-simple-perl
- apt-get install libcompress-zlib-perl
- apt-get install libdbi-perl
- apt-get install libdbd-mysql-perl
- apt-get install libapache-dbi-perl
- apt-get install libnet-ip-perl
- apt-get install libsoap-lite-perl

If a prepcompiled package is not available for your distribution, you can download the package source from http://search.cpan.org and build it on your system (make and C compiler must be available). For example:

- tar –xvzf package_name.tar.gz
- cd package_name
- perl Makefile.PL
- make
- make test
- make install

NB: if you are not using system perl interpreter, but another one such as the XAMPP/LAMPP perl interpreter, you must call this perl interpreter, not the system one, by specifying full path to your perl interpreter. For example:

/opt/lampp/bin/perl Makefile.PL

Installing Administration console required PHP modules

The Web Administration console requires Apache web server and PHP 4 scripting language and some additional modules for PHP (see Requirements).

You must have root privileges to set Administration console up.

You need to set ZIP support for PHP up.

On Fedora/Redhat like Linux, you can use “yum” to set it up:

- yum install php-pecl-zip

On Debian like Linux, you can use “apt-get” to set it up:

- apt-get install libphp-pclzip

Otherwise, the best way to do this is to use PHP PECL ZIP package. You must have PHP development libraries (php-devel package under RedHat or Fedora Core, under Linux Debian or Ubuntu) in order to have “phpize” command.

Then, if you have pear installed, just type

- pear install zip

If you don’t have pear installed, or no connection to Internet, download package “zip-1.3.1.tgz” from http://pecl.php.net/package/zip.

Install it (php devel package is required):

- tar –xvzf zip-1.3.1.tgz
- cd zip-1.3.1
- phpize
- ./configure
- make
- make install

You also need to set GD support for PHP up.

On Fedora/Redhat like Linux, you can use “yum” to set it up:

- yum install php-gd

On Debian like Linux, you can use “apt-get” to set it up:

- apt-get install php-gd

Installing management server.

You must have root privileges to set management server up.

NB: Ensure MySQL InnoDB engine is activated on your database server. Open my.cnf and ensure there is no line “skip-innodb” or this line is commented (begins with ‘#’).

Download “OCSNG_LINUX_SERVER_1.01.tar.gz” from OCS Inventory Web Site.

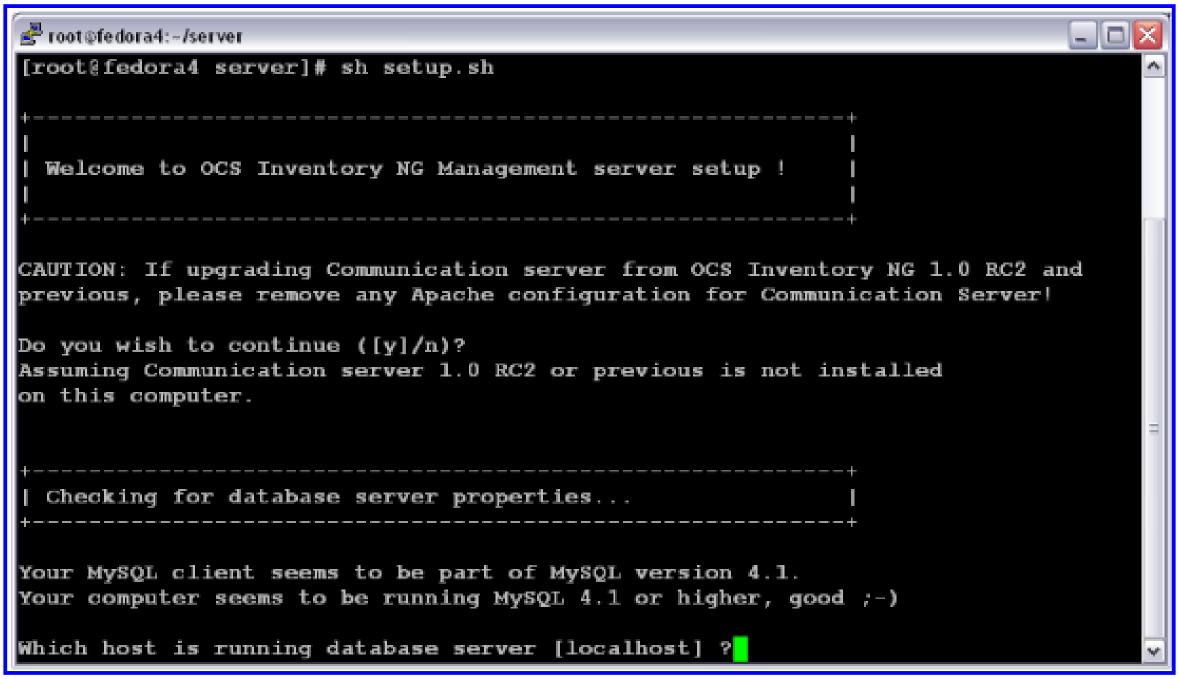
Unpack it.

- tar –xvzf OCSNG_LINUX_SERVER_1.01.tar.gz
- cd OCSNG_LINUX_SERVER_1.01

Run “setup.sh” installer. During the installer, default choice is presented between []. For example, [y]/n means that “y” (yes) is the default choice, and “n” (no) is the other choice.

- sh setup.sh

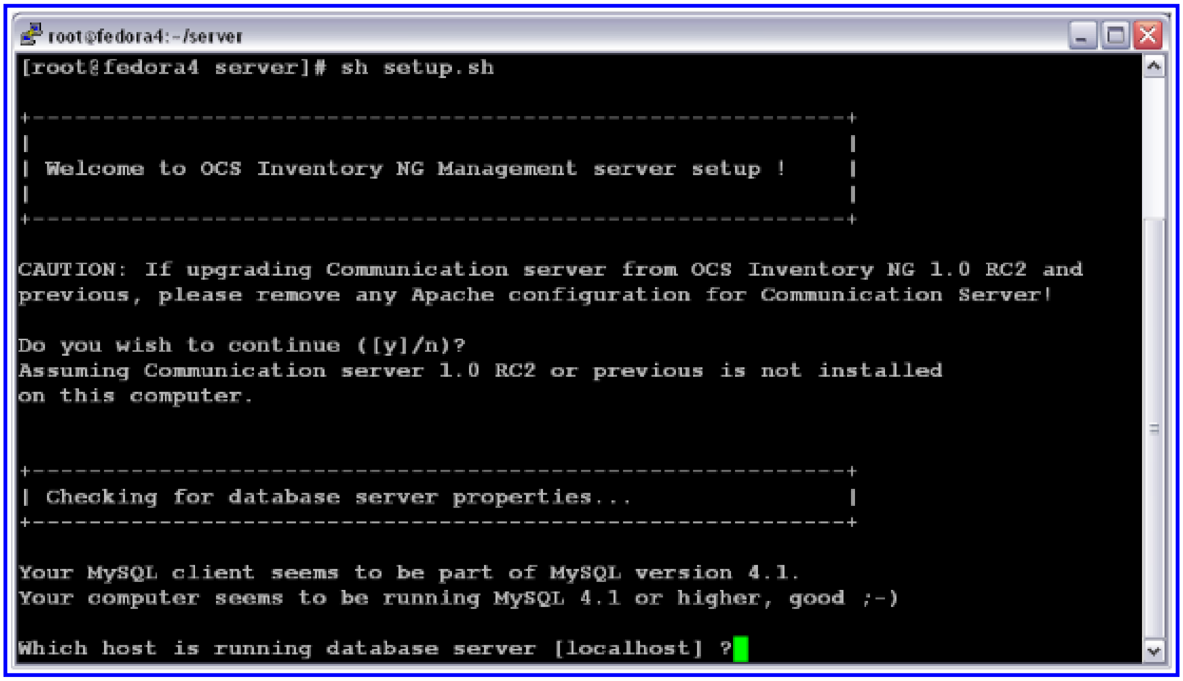
***NB:** installer writes a log file “ocs_server_setup.log” in the same directory. If you encounter any error, please refer to this log for detailed error message.*



```
[root@fedora4 server]# sh setup.sh

+----------------------------------------------------------+
|                                                          |
| Welcome to OCS Inventory NG Management server setup !    |
|                                                          |
+----------------------------------------------------------+

CAUTION: If upgrading Communication server from OCS Inventory NG 1.0 RC2 and
previous, please remove any Apache configuration for Communication Server!

Do you wish to continue ([y]/n)?
Assuming Communication server 1.0 RC2 or previous is not installed
on this computer.


+----------------------------------------------------------+
| Checking for database server properties...               |
+----------------------------------------------------------+

Your MySQL client seems to be part of MySQL version 4.1.
Your computer seems to be running MySQL 4.1 or higher, good ;-)

Which host is running database server [localhost] ?
```

CAUTION: *If you're upgrading from OCS Inventory NG 1.0 RC2 and previous, you must first remove any Apache configuration file for Communication server.*

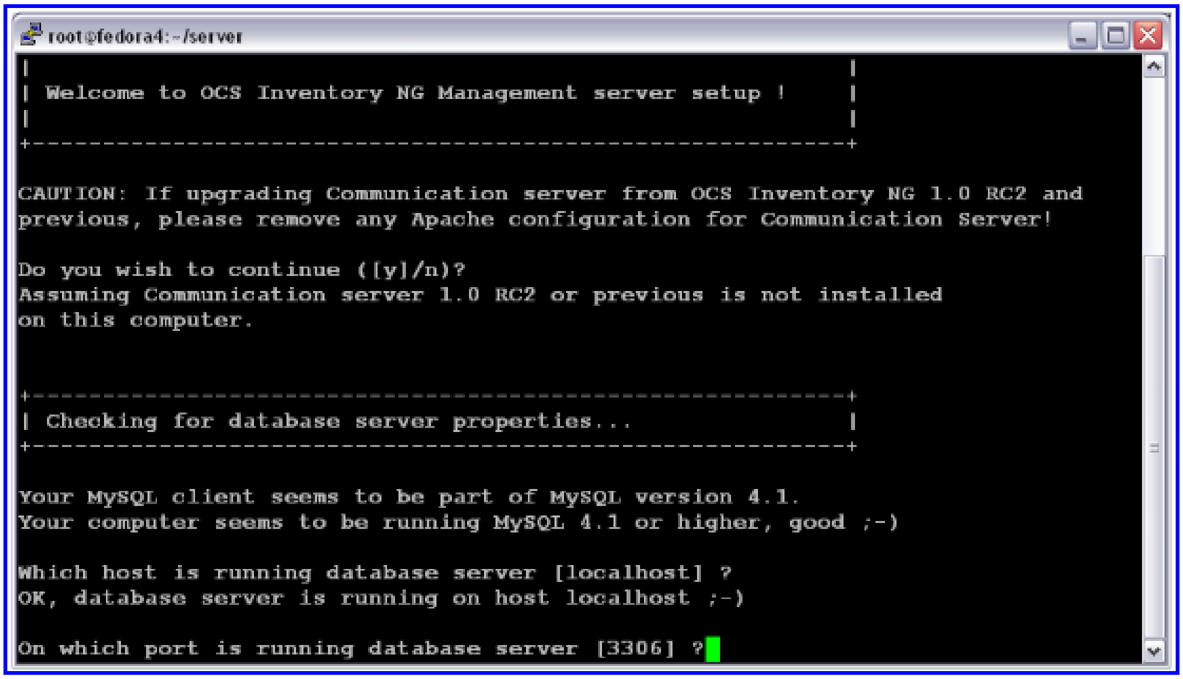
- Type "y" or "enter" to validate and, then enter MySQL server host address, in most cases localhost.



```
[root@fedora4 server]# sh setup.sh

+----------------------------------------------------------+
|                                                          |
| Welcome to OCS Inventory NG Management server setup !    |
|                                                          |
+----------------------------------------------------------+

CAUTION: If upgrading Communication server from OCS Inventory NG 1.0 RC2 and
previous, please remove any Apache configuration for Communication Server!

Do you wish to continue ([y]/n)?
Assuming Communication server 1.0 RC2 or previous is not installed
on this computer.


+----------------------------------------------------------+
| Checking for database server properties...               |
+----------------------------------------------------------+

Your MySQL client seems to be part of MySQL version 4.1.
Your computer seems to be running MySQL 4.1 or higher, good ;-)

Which host is running database server [localhost] ?
```

Then, setup checks for MySQL client binary files version 4.1 or higher. If not present, you will be prompted to continue or abort setup.

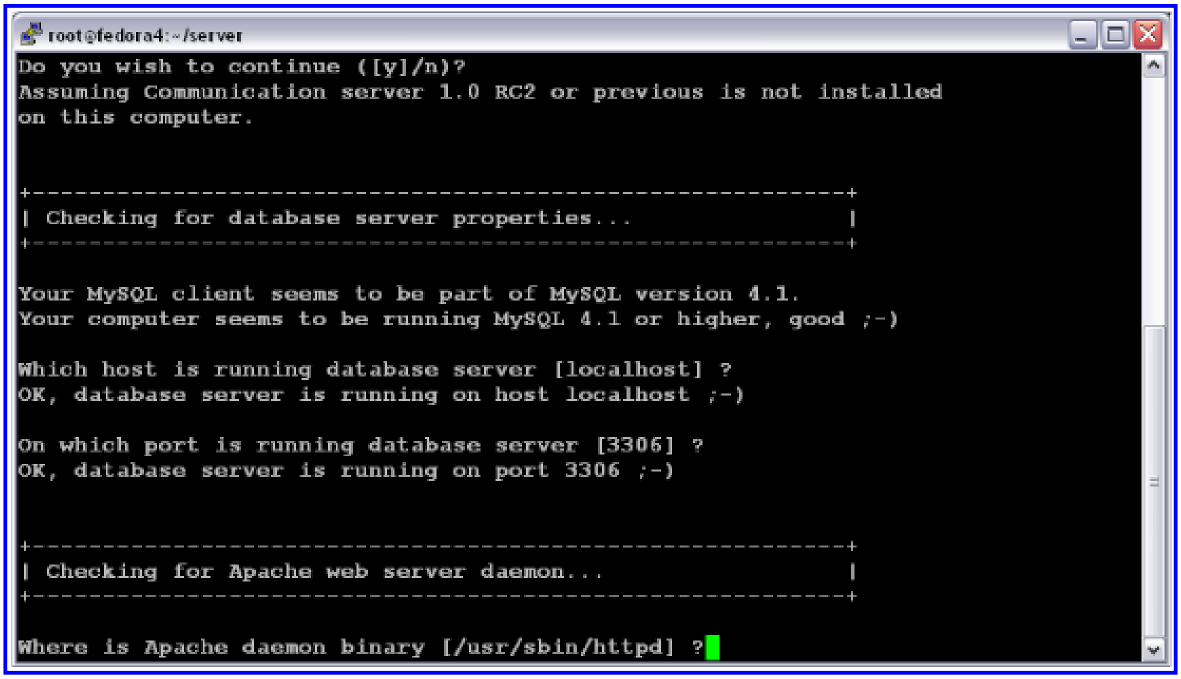
- If all is OK, enter MySQL server port, generally 3306.



```
|                                                            |
| Welcome to OCS Inventory NG Management server setup !      |
|                                                            |
+------------------------------------------------------------+

CAUTION: If upgrading Communication server from OCS Inventory NG 1.0 RC2 and
previous, please remove any Apache configuration for Communication Server!

Do you wish to continue ([y]/n)?
Assuming Communication server 1.0 RC2 or previous is not installed
on this computer.


+------------------------------------------------------------+
| Checking for database server properties...                 |
+------------------------------------------------------------+

Your MySQL client seems to be part of MySQL version 4.1.
Your computer seems to be running MySQL 4.1 or higher, good ;-)

Which host is running database server [localhost] ?
OK, database server is running on host localhost ;-)

On which port is running database server [3306] ?
```

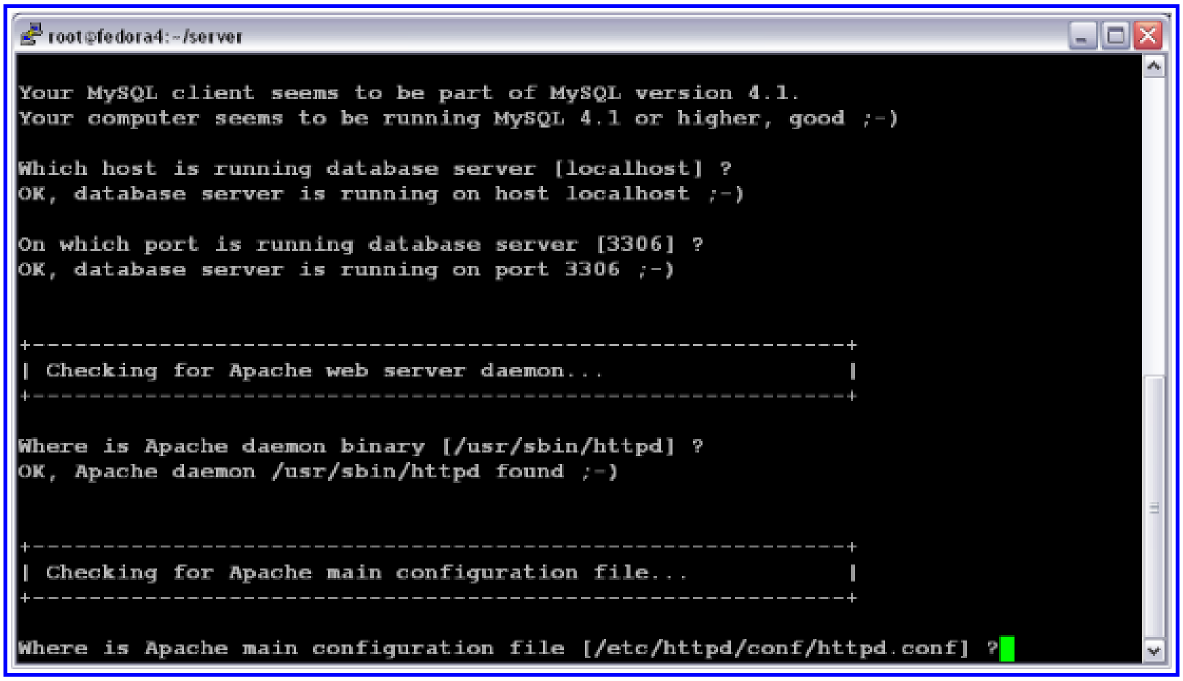
- Enter or validate path to Apache daemon binary, generally “/usr/sbin/httpd”. It will be used to find Apache configuration files.

***NB:** if you’re not using system Apache daemon, but another one like XAMPP/LAMPP Apache server, you must enter full path to your Apache daemon, not the system one.*



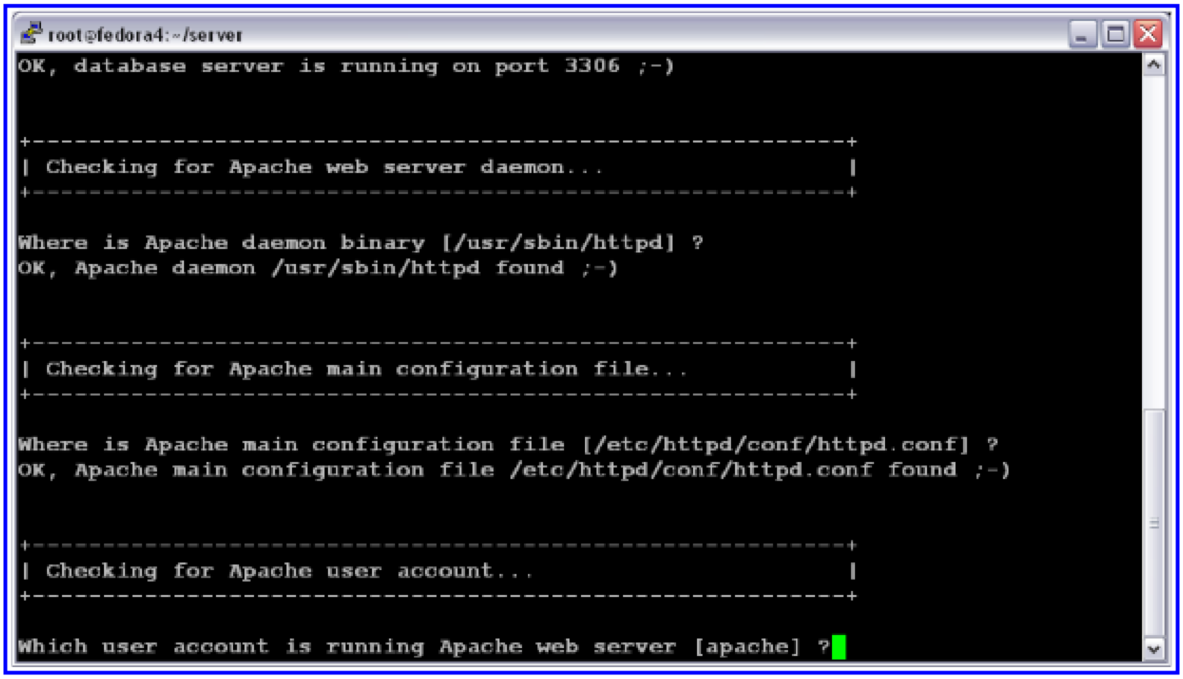
```
Do you wish to continue ([y]/n)?
Assuming Communication server 1.0 RC2 or previous is not installed
on this computer.


+------------------------------------------------------------+
| Checking for database server properties...                 |
+------------------------------------------------------------+

Your MySQL client seems to be part of MySQL version 4.1.
Your computer seems to be running MySQL 4.1 or higher, good ;-)

Which host is running database server [localhost] ?
OK, database server is running on host localhost ;-)

On which port is running database server [3306] ?
OK, database server is running on port 3306 ;-)


+------------------------------------------------------------+
| Checking for Apache web server daemon...                   |
+------------------------------------------------------------+

Where is Apache daemon binary [/usr/sbin/httpd] ?
```

- Enter or validate Apache main configuration file path, generally “/etc/apache/conf/apache.conf” or “/etc/httpd/conf/httpd.conf”.



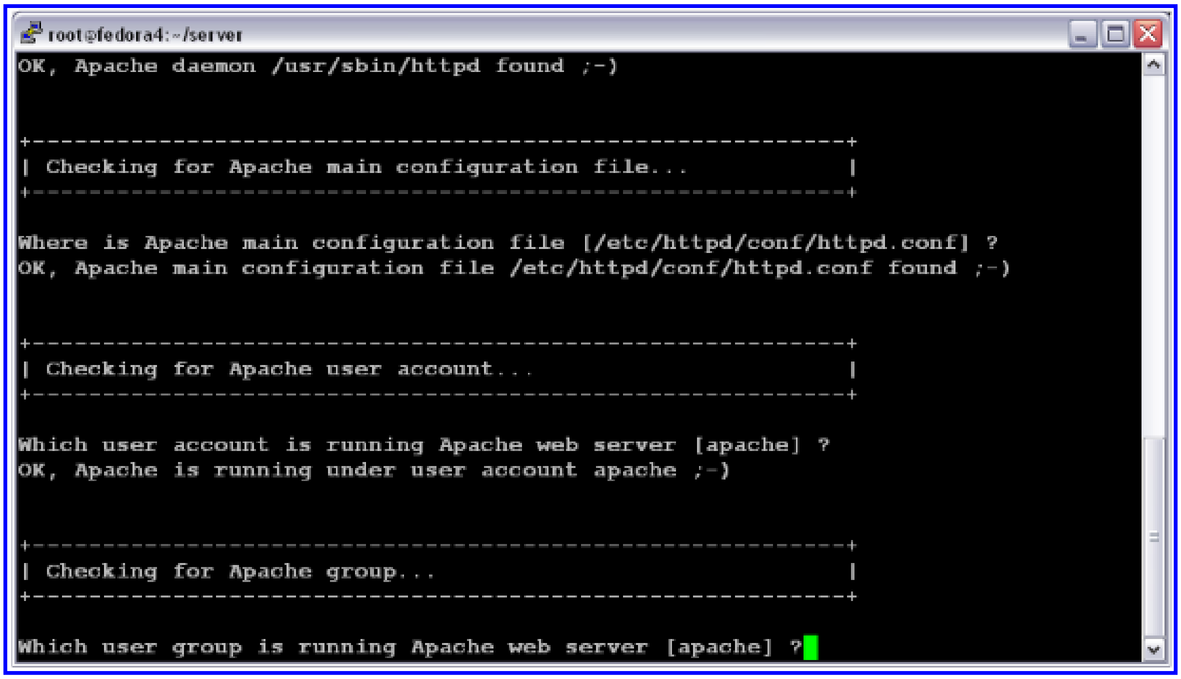
```
Your MySQL client seems to be part of MySQL version 4.1.
Your computer seems to be running MySQL 4.1 or higher, good ;-)

Which host is running database server [localhost] ?
OK, database server is running on host localhost ;-)

On which port is running database server [3306] ?
OK, database server is running on port 3306 ;-)


+----------------------------------------------------------+
| Checking for Apache web server daemon...                 |
+----------------------------------------------------------+

Where is Apache daemon binary [/usr/sbin/httpd] ?
OK, Apache daemon /usr/sbin/httpd found ;-)


+----------------------------------------------------------+
| Checking for Apache main configuration file...           |
+----------------------------------------------------------+

Where is Apache main configuration file [/etc/httpd/conf/httpd.conf] ?
```

- Enter or validate Apache daemon running user account, generally “apache” or “www”.



```
OK, database server is running on port 3306 ;-)


+----------------------------------------------------------+
| Checking for Apache web server daemon...                 |
+----------------------------------------------------------+

Where is Apache daemon binary [/usr/sbin/httpd] ?
OK, Apache daemon /usr/sbin/httpd found ;-)


+----------------------------------------------------------+
| Checking for Apache main configuration file...           |
+----------------------------------------------------------+

Where is Apache main configuration file [/etc/httpd/conf/httpd.conf] ?
OK, Apache main configuration file /etc/httpd/conf/httpd.conf found ;-)


+----------------------------------------------------------+
| Checking for Apache user account...                      |
+----------------------------------------------------------+

Which user account is running Apache web server [apache] ?
```

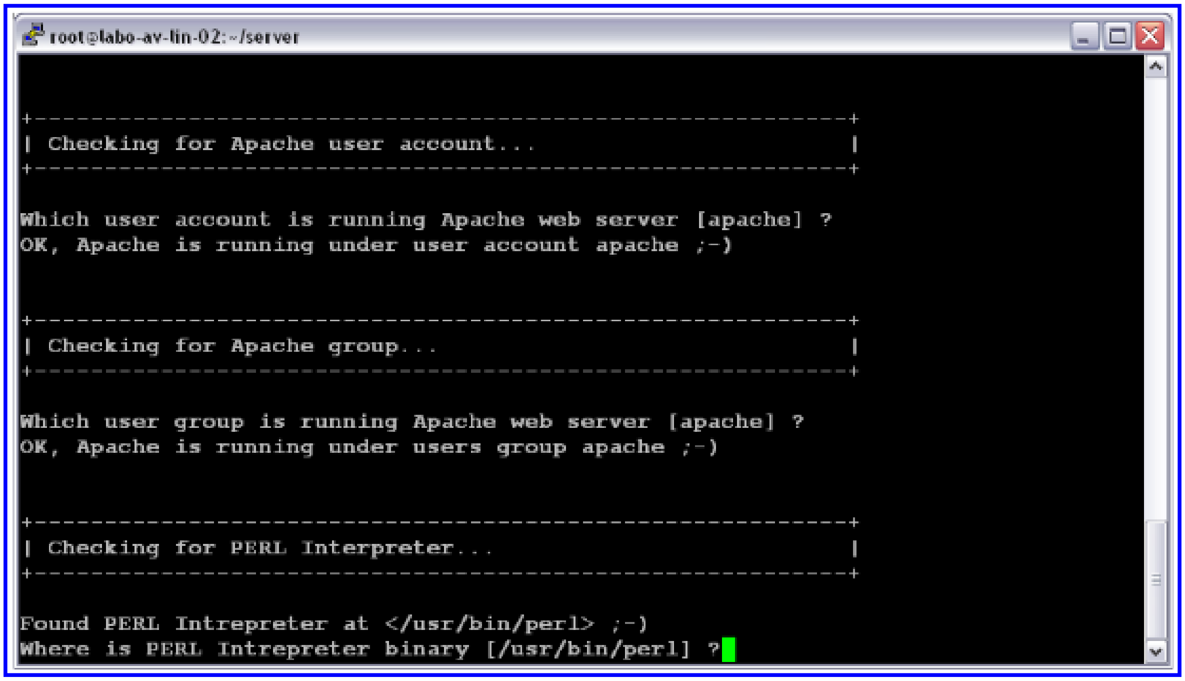
- Enter or validate Apache daemon user group, generally “apache” or “www”.



```
OK, Apache daemon /usr/sbin/httpd found ;-)


+----------------------------------------------------------+
| Checking for Apache main configuration file...           |
+----------------------------------------------------------+

Where is Apache main configuration file [/etc/httpd/conf/httpd.conf] ?
OK, Apache main configuration file /etc/httpd/conf/httpd.conf found ;-)


+----------------------------------------------------------+
| Checking for Apache user account...                      |
+----------------------------------------------------------+

Which user account is running Apache web server [apache] ?
OK, Apache is running under user account apache ;-)


+----------------------------------------------------------+
| Checking for Apache group...                             |
+----------------------------------------------------------+

Which user group is running Apache web server [apache] ?
```

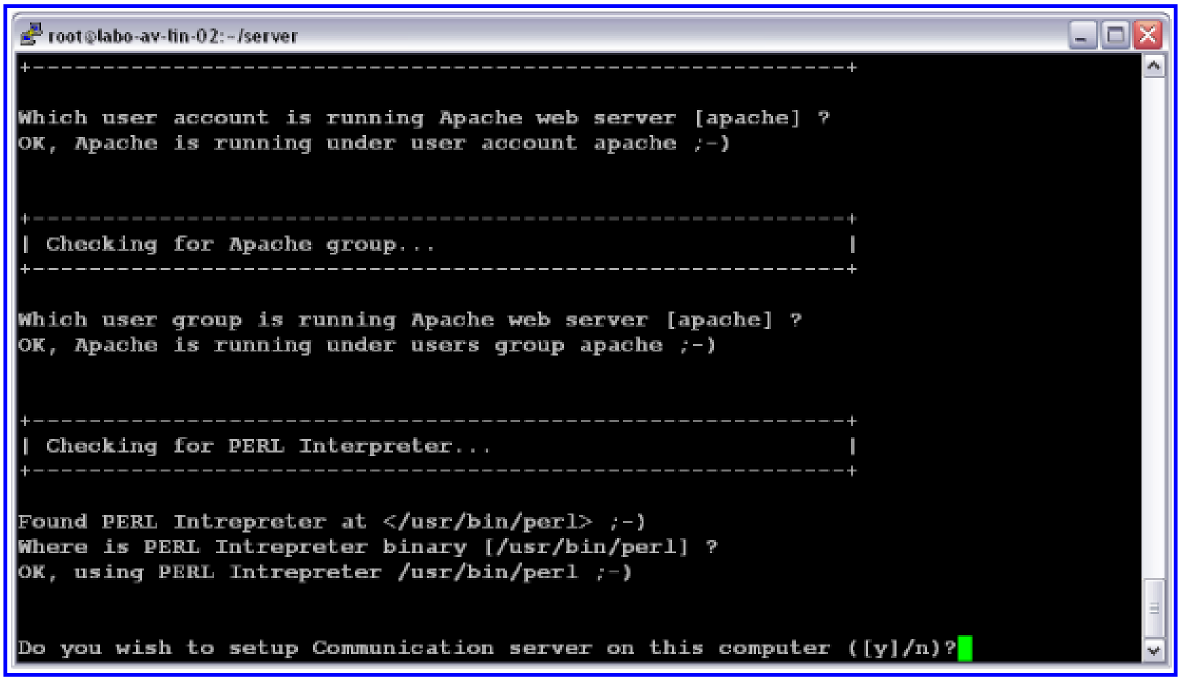
- Next, setup checks for PERL interpreter binaries. Enter or validate path to PERL interpreter.

NB: *if you're not using system perl interpreter, but another one like XAMPP/LAMPP perl interpreter, you must specify full path to this perl interpreter, not the default system one. (/opt/lampp/bin/perl generally used in XAMPP/LAMPP).*



```


+----------------------------------------------------------+
| Checking for Apache user account...                      |
+----------------------------------------------------------+

Which user account is running Apache web server [apache] ?
OK, Apache is running under user account apache ;-)


+----------------------------------------------------------+
| Checking for Apache group...                             |
+----------------------------------------------------------+

Which user group is running Apache web server [apache] ?
OK, Apache is running under users group apache ;-)


+----------------------------------------------------------+
| Checking for PERL Interpreter...                         |
+----------------------------------------------------------+

Found PERL Intrepreter at </usr/bin/perl> ;-)
Where is PERL Intrepreter binary [/usr/bin/perl] ?
```

- Common information for setting up Communication server or Administration console is now collected. Setup prompts you if you wish to set Communication server up on this computer. Enter "y" or validate to set Communication server up, "n" to skip Communication server installation.

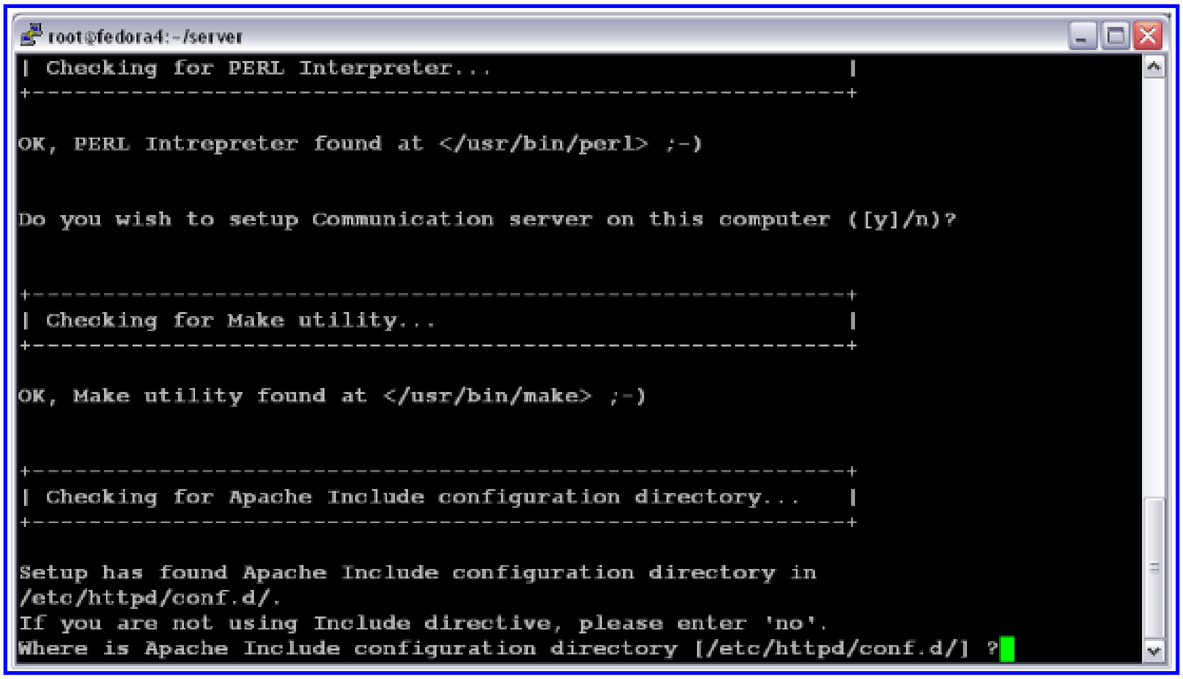


root@labo-av-lin-02:~/server

```
+----------------------------------------------------------+

Which user account is running Apache web server [apache] ?
OK, Apache is running under user account apache ;-)


+----------------------------------------------------------+
| Checking for Apache group...                             |
+----------------------------------------------------------+

Which user group is running Apache web server [apache] ?
OK, Apache is running under users group apache ;-)


+----------------------------------------------------------+
| Checking for PERL Interpreter...                         |
+----------------------------------------------------------+

Found PERL Intrepreter at </usr/bin/perl> ;-)
Where is PERL Intrepreter binary [/usr/bin/perl] ?
OK, using PERL Intrepreter /usr/bin/perl ;-)


Do you wish to setup Communication server on this computer ([y]/n)?
```

Setup will then try to find make utility. If it fails, setup will stop.

- Enter or validate path to Apache include configuration directory. This is the directory where is stored Apache configuration for specific modules. Generally, this directory is “/etc/httpd/conf.d” or “/etc/apache/conf.d”. If you are not using configuration directory, but having all configurations into Apache main configuration file, enter “no”.



root@fedora4:~/server

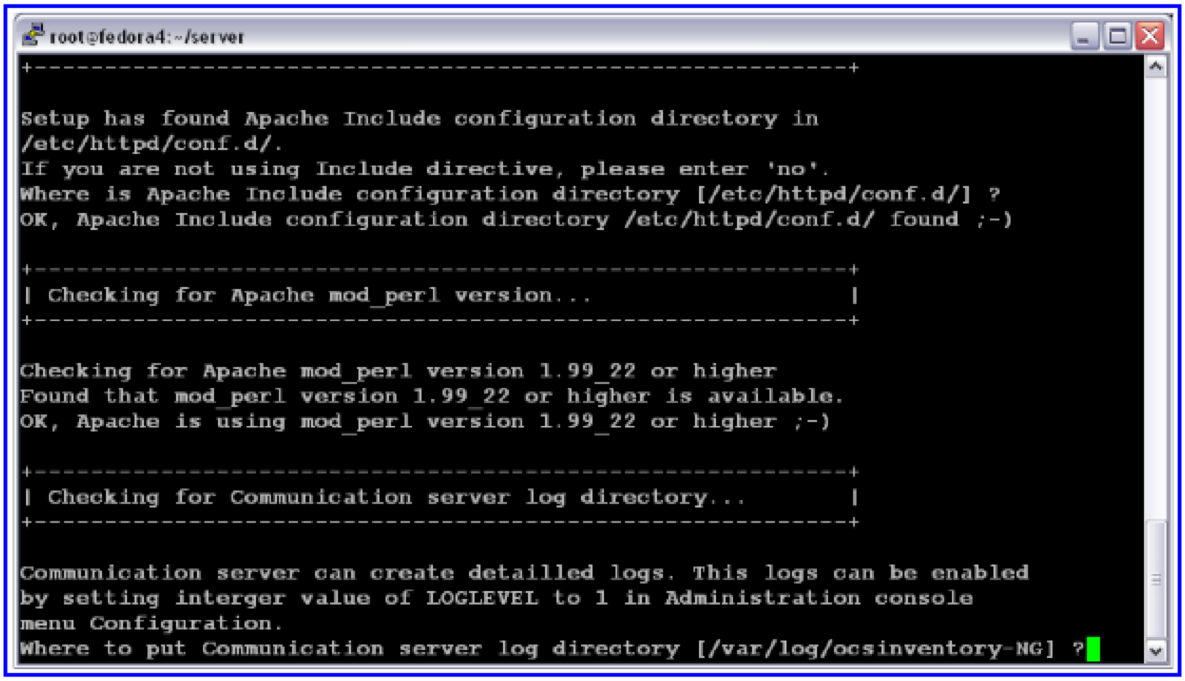
```
| Checking for PERL Interpreter...                         |
+----------------------------------------------------------+

OK, PERL Intrepreter found at </usr/bin/perl> ;-)


Do you wish to setup Communication server on this computer ([y]/n)?


+----------------------------------------------------------+
| Checking for Make utility...                             |
+----------------------------------------------------------+

OK, Make utility found at </usr/bin/make> ;-)


+----------------------------------------------------------+
| Checking for Apache Include configuration directory...   |
+----------------------------------------------------------+

Setup has found Apache Include configuration directory in
/etc/httpd/conf.d/.
If you are not using Include directive, please enter 'no'.
Where is Apache Include configuration directory [/etc/httpd/conf.d/] ?
```

Setup will next try to determine your Apache mod_perl version. If it is not able to determine mod_perl version, it will ask you to enter it.

NB: *You can check which version of mod_perl you are using by querying your server's software database.*

Under RPM enabled Linux distribution (RedHat/Fedora, Mandriva…), run “rpm –q mod_perl”. Under DPKG enabled Linux distribution (Debian, Ubuntu…), run “dpkg –l libapache-mod-perl*”.*

Next, it will prompt you to enter log directory where Communication server will store debugging/tuning logs. Validate or enter directory path. If it does not exist, this directory will be created.

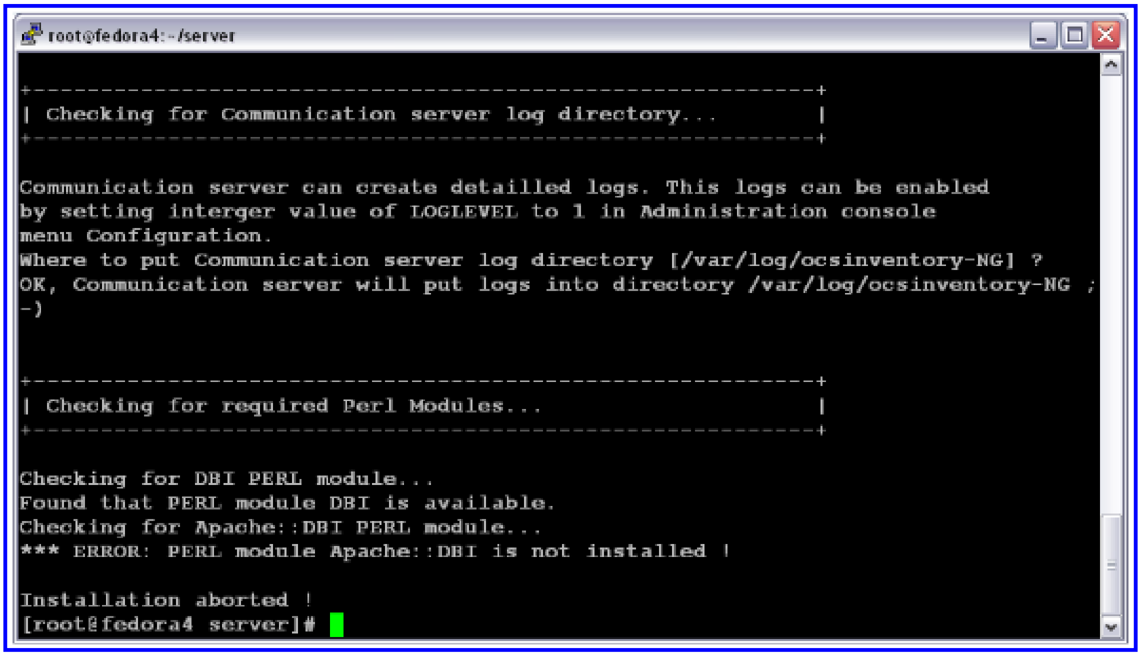


```
+------------------------------------------------------------+

Setup has found Apache Include configuration directory in
/etc/httpd/conf.d/.
If you are not using Include directive, please enter 'no'.
Where is Apache Include configuration directory [/etc/httpd/conf.d/] ?
OK, Apache Include configuration directory /etc/httpd/conf.d/ found ;-)

+------------------------------------------------------------+
| Checking for Apache mod_perl version...                    |
+------------------------------------------------------------+

Checking for Apache mod_perl version 1.99_22 or higher
Found that mod_perl version 1.99_22 or higher is available.
OK, Apache is using mod_perl version 1.99_22 or higher ;-)

+------------------------------------------------------------+
| Checking for Communication server log directory...         |
+------------------------------------------------------------+

Communication server can create detailled logs. This logs can be enabled
by setting interger value of LOGLEVEL to 1 in Administration console
menu Configuration.
Where to put Communication server log directory [/var/log/ocsinventory-NG] ?
```

Next, setup will check for required PERL modules (cf Requirements.):

- XML::Simple version 2.12 or higher
- Compress::Zlib version 1.33 or higher
- DBI version 1.40 or higher
- DBD::mysql version 2.9004 or higher
- Apache::DBI version 0.93 or higher
- Net::IP version 1.21 or higher
- SOAP::Lite version 0.66 or higher

If any of these modules is missing, setup will abort.



```
+------------------------------------------------------------+
| Checking for Communication server log directory...         |
+------------------------------------------------------------+

Communication server can create detailled logs. This logs can be enabled
by setting interger value of LOGLEVEL to 1 in Administration console
menu Configuration.
Where to put Communication server log directory [/var/log/ocsinventory-NG] ?
OK, Communication server will put logs into directory /var/log/ocsinventory-NG ;
-)


+------------------------------------------------------------+
| Checking for required Perl Modules...                      |
+------------------------------------------------------------+

Checking for DBI PERL module...
Found that PERL module DBI is available.
Checking for Apache::DBI PERL module...
*** ERROR: PERL module Apache::DBI is not installed !

Installation aborted !
[root@fedora4 server]#
```

If all is OK, setup will install Communication server:

- Configure Communication server PERL module.
- Build Communication server PERL module.

- Install Communication server PERL module into PERL standard library directories.
- Create Communication server log directory (/var/log/ocsinventory-NG by default).
- Configure daily log rotation for Communication server (file /etc/logrotate.d/ocsinventory-NG by default)
- Create Apache configuration file (ocsinventory.conf). If you are using Apache configuration directory, this file will be copied under this directory. Otherwise, you will be prompted to add content of this file to the end of Apache main configuration file. **Do not add content to apache main configuration file if it is not a fresh install! You must manually copy content of the “ocsinventory.conf.local” file created by setup into apache main configuration file, replacing existing configuration.**

```
################################################################################
#
# OCS Inventory NG Communication Server Perl Module Setup
#
# Copyleft 2006 Pascal DANEK
# Web: http://ocsinventory.sourceforge.net
#
# This code is open source and may be copied and modified as long as the source
# code is always made freely available.
# Please refer to the General Public Licence http://www.gnu.org/ or Licence.txt
################################################################################

# Which version of mod_perl we are using
# For mod_perl <= 1.999_21, replace VERSION_MP by 1
# For mod_perl > 1.999_21, replace VERSION_MP by 2
PerlSetEnv OCS_MODPERL_VERSION 1

# Where to write detailled logs
PerlSetEnv OCS_LOGPATH "/var/log/ocsinventory-NG"

# Database options
# Replace DATABASE_SERVER by hostname or ip of MySQL server, generally localhost
PerlSetEnv OCS_DB_HOST localhost
# Replace DATABASE_PORT by port where running MySQL server, generally 3306
PerlSetEnv OCS_DB_PORT 3306
# Name of database
PerlSetEnv OCS_DB_NAME ocsweb
PerlSetEnv OCS_DB_LOCAL ocsweb
# User allowed to connect to database
PerlSetEnv OCS_DB_USER ocs
# Password for user
PerlSetVar OCS_DB_PWD ocs

# The options below are overloaded if you are using ocs GUI
# Be careful: you must restart apache to have any effects
PerlSetEnv OCS_OPT_FREQUENCY 0
PerlSetEnv OCS_OPT_PROLOG_FREQ 24
PerlSetEnv OCS_OPT_DEPLOY 1
PerlSetEnv OCS_OPT_TRACE_DELETED 0
PerlSetEnv OCS_OPT_AUTO_DUPLICATE_LVL 7
PerlSetEnv OCS_OPT_LOGLEVEL 0
PerlSetEnv OCS_OPT_INVENTORY_DIFF 1
PerlSetEnv OCS_OPT_INVENTORY_TRANSACTION 1
PerlSetEnv OCS_OPT_PROXY_REVALIDATE_DELAY 3600

# Optional modules
PerlSetEnv OCS_OPT_IPDISCOVER 2
PerlSetEnv OCS_OPT_IPDISCOVER_MAX_ALIVE 7
PerlSetEnv OCS_OPT_IPDISCOVER_LATENCY 100
PerlSetEnv OCS_OPT_REGISTRY 0
PerlSetEnv OCS_OPT_UPDATE 0
PerlSetEnv OCS_OPT_DOWNLOAD 0
PerlSetEnv OCS_OPT_DOWNLOAD_FRAG_LATENCY 10
PerlSetEnv OCS_OPT_DOWNLOAD_CYCLE_LATENCY 0
PerlSetEnv OCS_OPT_DOWNLOAD_PERIOD_LATENCY 0
PerlSetEnv OCS_OPT_DOWNLOAD_TIMEOUT 30
PerlSetEnv OCS_OPT_WEB_SERVICE_ENABLED 0
```

```
############ DO NOT MODIFY BELOW ! #######################

# External modules
PerlModule Apache::DBI
PerlModule Compress::Zlib
PerlModule XML::Simple

# Ocs
PerlModule Apache::Ocsinventory
PerlModule Apache::Ocsinventory::Server::Constants
PerlModule Apache::Ocsinventory::Server::System
PerlModule Apache::Ocsinventory::Server::Communication
PerlModule Apache::Ocsinventory::Server::Inventory
PerlModule Apache::Ocsinventory::Server::Duplicate

# Options
PerlModule Apache::Ocsinventory::Server::Option::Registry
PerlModule Apache::Ocsinventory::Server::Option::Update
PerlModule Apache::Ocsinventory::Server::Option::Ipdiscover
PerlModule Apache::Ocsinventory::Server::Option::Download
# This module guides you through the module creation
# PerlModule Apache::Ocsinventory::Server::Option::Example
# This module adds some rules to filter some request sent to ocs server in the prolog and inventory stages
# PerlModule Apache::Ocsinventory::Server::Option::Filter

# Virtual directory for handling OCS Inventory NG agents communications
# Be carefull, do not create such directory into your web server root document !
#PerlTaintCheck On
<Location /ocsinventory>
order deny,allow
allow from all
```

Satisfy Any

SetHandler perl-script

PerlHandler Apache::Ocsinventory

</Location>

PerlModule Apache::Ocsinventory::SOAP;

<location /ocsinterface>

SetHandler perl-script

perlHandler "Apache::Ocsinventory::SOAP"

order deny,allow

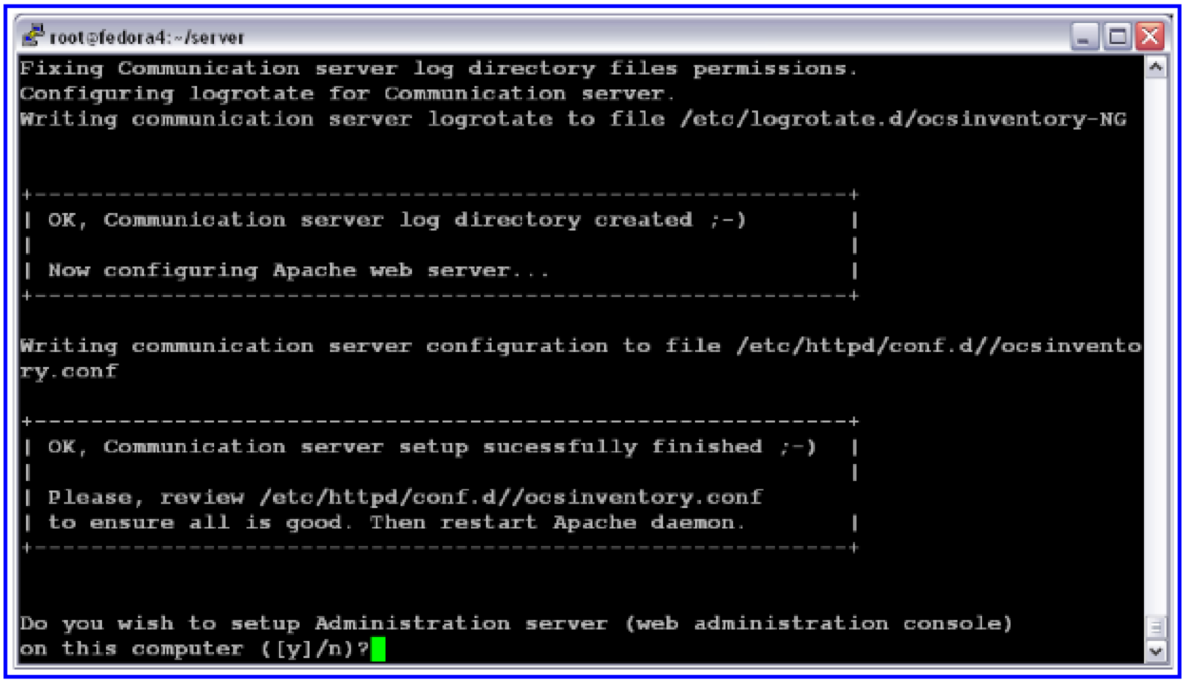
allow from all

Satisfy any

</location>

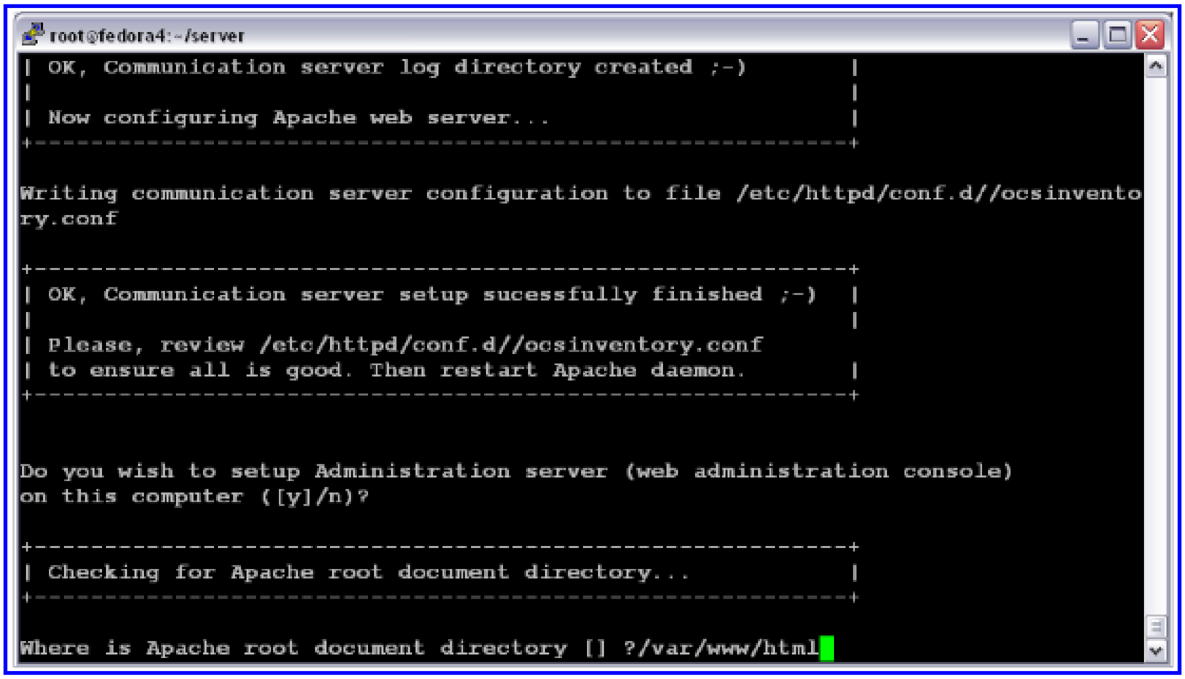
Figure 2: Apache configuration sample file

Communication server installation is now finished. You will be prompted to set Administration console up. Enter “y” or validate to set Administration console up, enter “n” to skip Administration console installation.



```
Fixing Communication server log directory files permissions.
Configuring logrotate for Communication server.
Writing communication server logrotate to file /etc/logrotate.d/ocsinventory-NG


+----------------------------------------------------------+
| OK, Communication server log directory created ;-)       |
|                                                          |
| Now configuring Apache web server...                     |
+----------------------------------------------------------+

Writing communication server configuration to file /etc/httpd/conf.d//ocsinvento
ry.conf

+----------------------------------------------------------+
| OK, Communication server setup sucessfully finished ;-)  |
|                                                          |
| Please, review /etc/httpd/conf.d//ocsinventory.conf
| to ensure all is good. Then restart Apache daemon.       |
+----------------------------------------------------------+


Do you wish to setup Administration server (web administration console)
on this computer ([y]/n)?
```

Setup will ask you to enter Apache root document directory, usually “/var/www/html” or “/var/www-data”.



root@fedora4:~/server

```
| OK, Communication server log directory created ;-)          |
|                                                              |
| Now configuring Apache web server...                         |
+--------------------------------------------------------------+

Writing communication server configuration to file /etc/httpd/conf.d//ocsinvento
ry.conf

+--------------------------------------------------------------+
| OK, Communication server setup sucessfully finished ;-)      |
|                                                              |
| Please, review /etc/httpd/conf.d//ocsinventory.conf
| to ensure all is good. Then restart Apache daemon.           |
+--------------------------------------------------------------+


Do you wish to setup Administration server (web administration console)
on this computer ([y]/n)?

+--------------------------------------------------------------+
| Checking for Apache root document directory...               |
+--------------------------------------------------------------+

Where is Apache root document directory [] ?/var/www/html
```

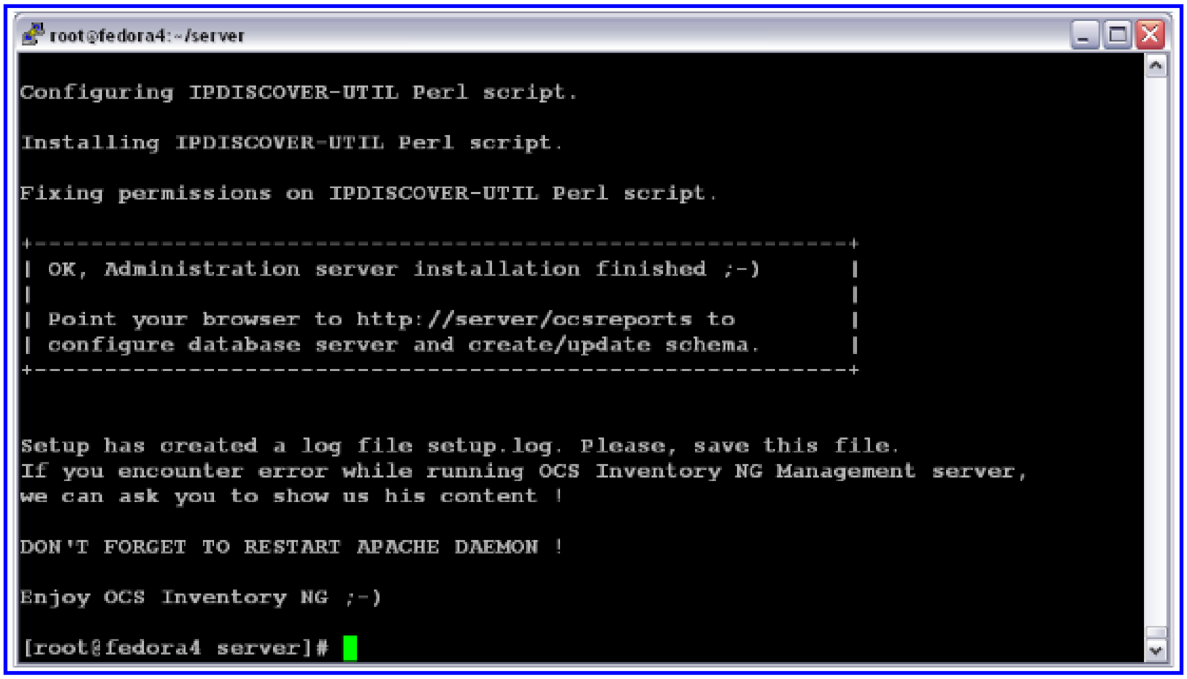
Next, setup will check for required PERL modules (cf Rquirements.):

- XML::Simple version 2.12 or higher
- DBI version 1.40 or higher
- DBD::Mysql version 2.9004 or higher
- Net::IP version 1.21 or higher

If any of these modules is missing, setup will abort.

If everything is OK, setup will install Administration console into the “ocsreports” subdirectory:

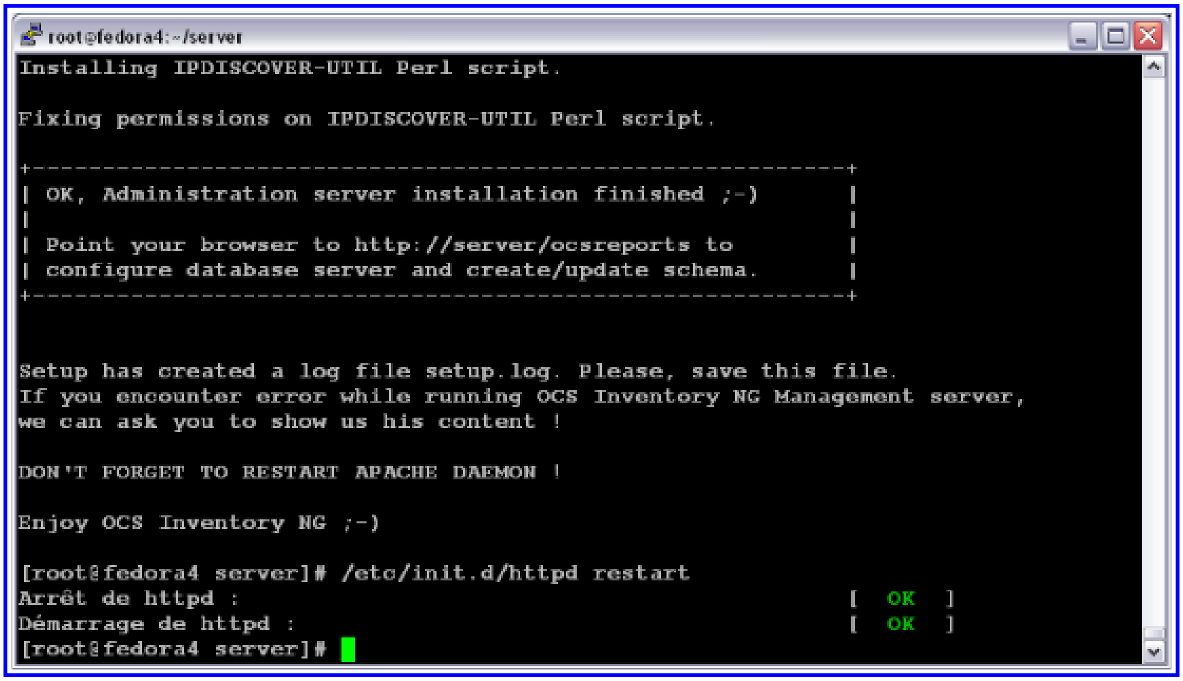
- Create /ocsreports directory structure.
- Create /download directory structure.
- Copy files into /ocsreports directory.
- Fix directories and files permissions to allow Apache daemon reading and writing to required directories (write access is required in /ocsreports, /ocsreports/ipd and /download, cf § 11.4 Files and directories permissions under Linux.).
- Configure PERL script ipdiscover-util.pl to access database and install it.



```
Configuring IPDISCOVER-UTIL Perl script.

Installing IPDISCOVER-UTIL Perl script.

Fixing permissions on IPDISCOVER-UTIL Perl script.

+----------------------------------------------------------+
| OK, Administration server installation finished ;-)      |
|                                                          |
| Point your browser to http://server/ocsreports to        |
| configure database server and create/update schema.      |
+----------------------------------------------------------+


Setup has created a log file setup.log. Please, save this file.
If you encounter error while running OCS Inventory NG Management server,
we can ask you to show us his content !

DON'T FORGET TO RESTART APACHE DAEMON !

Enjoy OCS Inventory NG ;-)

[root@fedora4 server]#
```

Now, you can restart Apache web server for changes to take effect.

- /etc/init.d/httpd restart or /etc/init.d/apache restart



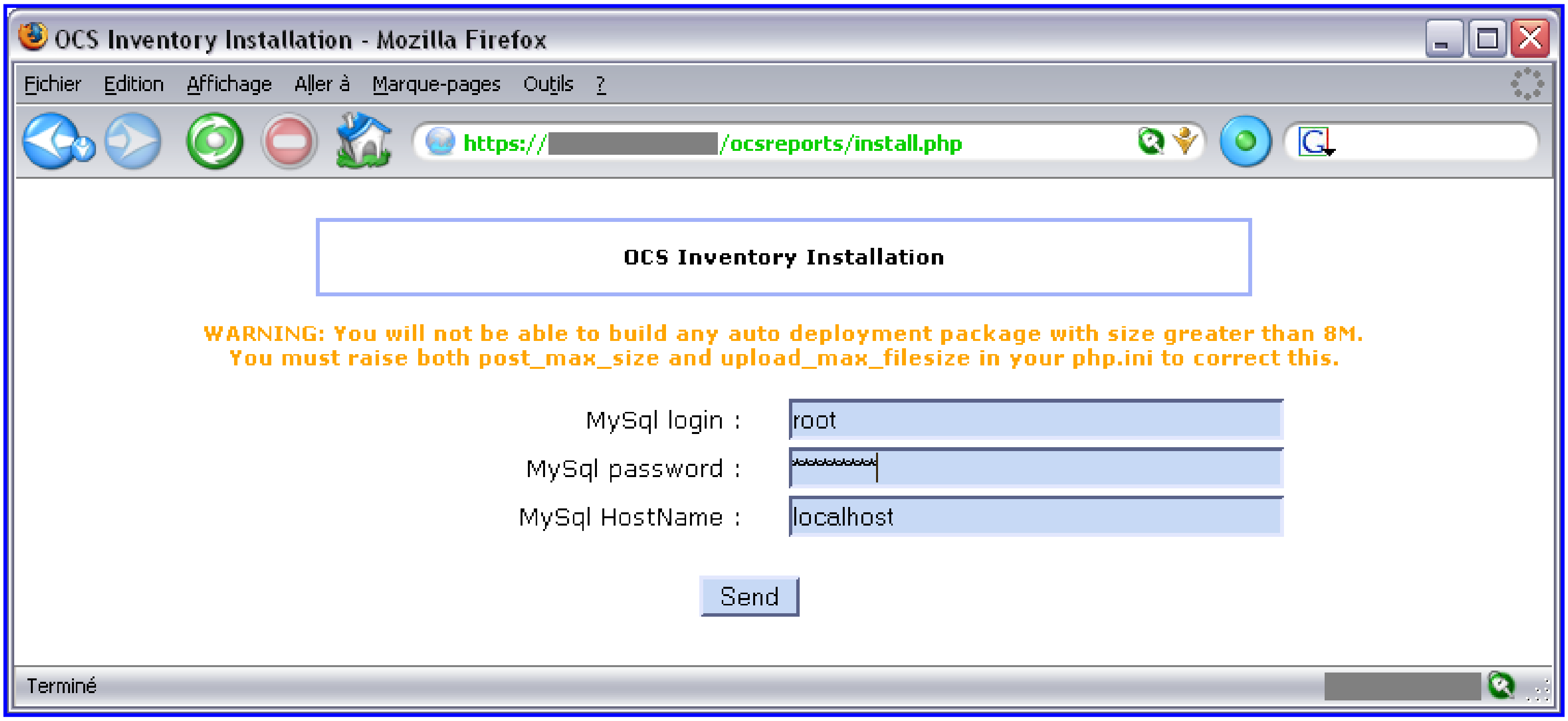
```
Installing IPDISCOVER-UTIL Perl script.

Fixing permissions on IPDISCOVER-UTIL Perl script.

+----------------------------------------------------------+
| OK, Administration server installation finished ;-)      |
|                                                          |
| Point your browser to http://server/ocsreports to        |
| configure database server and create/update schema.      |
+----------------------------------------------------------+


Setup has created a log file setup.log. Please, save this file.
If you encounter error while running OCS Inventory NG Management server,
we can ask you to show us his content !

DON'T FORGET TO RESTART APACHE DAEMON !

Enjoy OCS Inventory NG ;-)

[root@fedora4 server]# /etc/init.d/httpd restart
Arrêt de httpd :                                           [  OK  ]
Démarrage de httpd :                                       [  OK  ]
[root@fedora4 server]#
```

Configuring management server.

Open your favorite web browser and point it on URL "http://administration_console/ocsreports" to connect the Administration server.

As database is not yet created, this will begin OCS Inventory setup process. Otherwise, you can rerun configuration process by browsing http://administration_console/ocsreports/install.php URL (this must be used when upgrading OCS Inventory management server).

NB: *You will see warning regarding max size of package you will be able to deploy. Please, see Uploads size for package deployment.) to configure your server to match your need.*


OCS Inventory Installation - Mozilla Firefox
Fichier Edition Affichage Aller à Marque-pages Outils ?
Send
Terminé

https://[redacted]/ocsreports/install.php

OCS Inventory Installation

WARNING: You will not be able to build any auto deployment package with size greater than 8M.
You must raise both post_max_size and upload_max_filesize in your php.ini to correct this.

MySql login :	root
MySql password :	**********
MySql HostName :	localhost

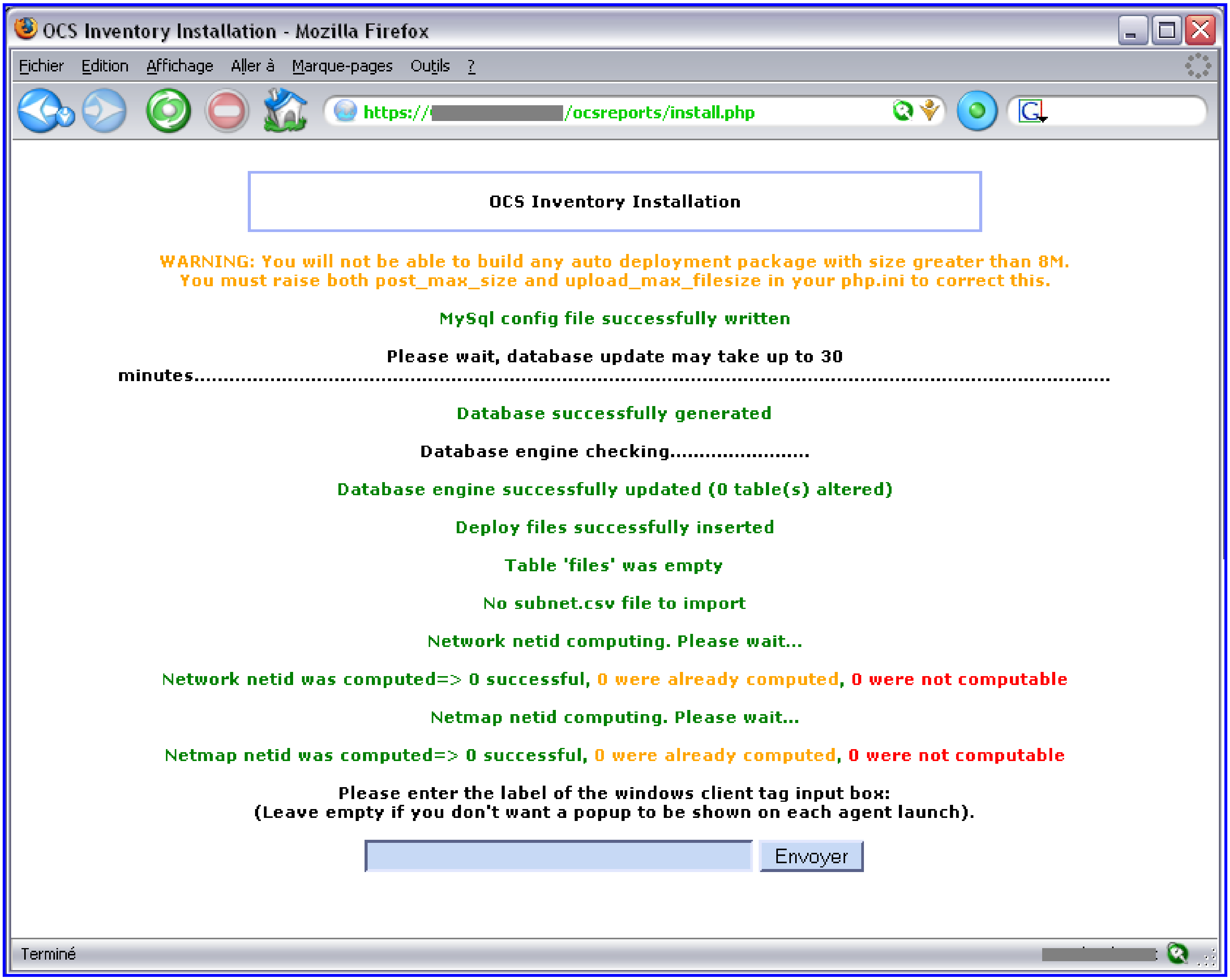
[redacted]

Fill in information to connect to MySQL database server with a user who has the ability to create database, tables, indexes, etc (usually root):

- MySQL user name
- MySQL user password
- MySQL hostname

NB: *Setup will create "ocsweb" database, and a MySQL user "ocs" with password "ocs". It will also grant to user "ocs" privileges* "Select | Insert | Update | Delete | Create | Drop | References | Index | Alter | Create temp | Lock" on database "ocsweb". *This user will be used by Administration server and Communication server to connect to the database. If you do not wish to use default MySQL user "ocs" with "ocs" password, you must update in the file "dbconfig.inc.php" PHP constants "COMPTE_BASE", which is MySQL user login, and/or "PSWD_BASE", which MySQL user password. Don't forget to also update Communication server configuration, especially in apache configuration file.*

Finally, you may fill in a text describing the TAG, a string displayed at first launch of the agent to ask user to enter the TAG Value. It's a generic data which allows you to sort the new computers (geographical site, first floor, john room....). If you don't want this functionality, just let it blank.


OCS Inventory Installation - Mozilla Firefox
Fichier Edition Affichage Aller à Marque-pages Outils ?
https://[redacted]/ocsreports/install.php
Envoyer
Terminé
[redacted]

OCS Inventory Installation

WARNING: You will not be able to build any auto deployment package with size greater than 8M.
You must raise both post_max_size and upload_max_filesize in your php.ini to correct this.

MySql config file successfully written

Please wait, database update may take up to 30 minutes...

Database successfully generated

Database engine checking.......................

Database engine successfully updated (0 table(s) altered)

Deploy files successfully inserted

Table 'files' was empty

No subnet.csv file to import

Network netid computing. Please wait...

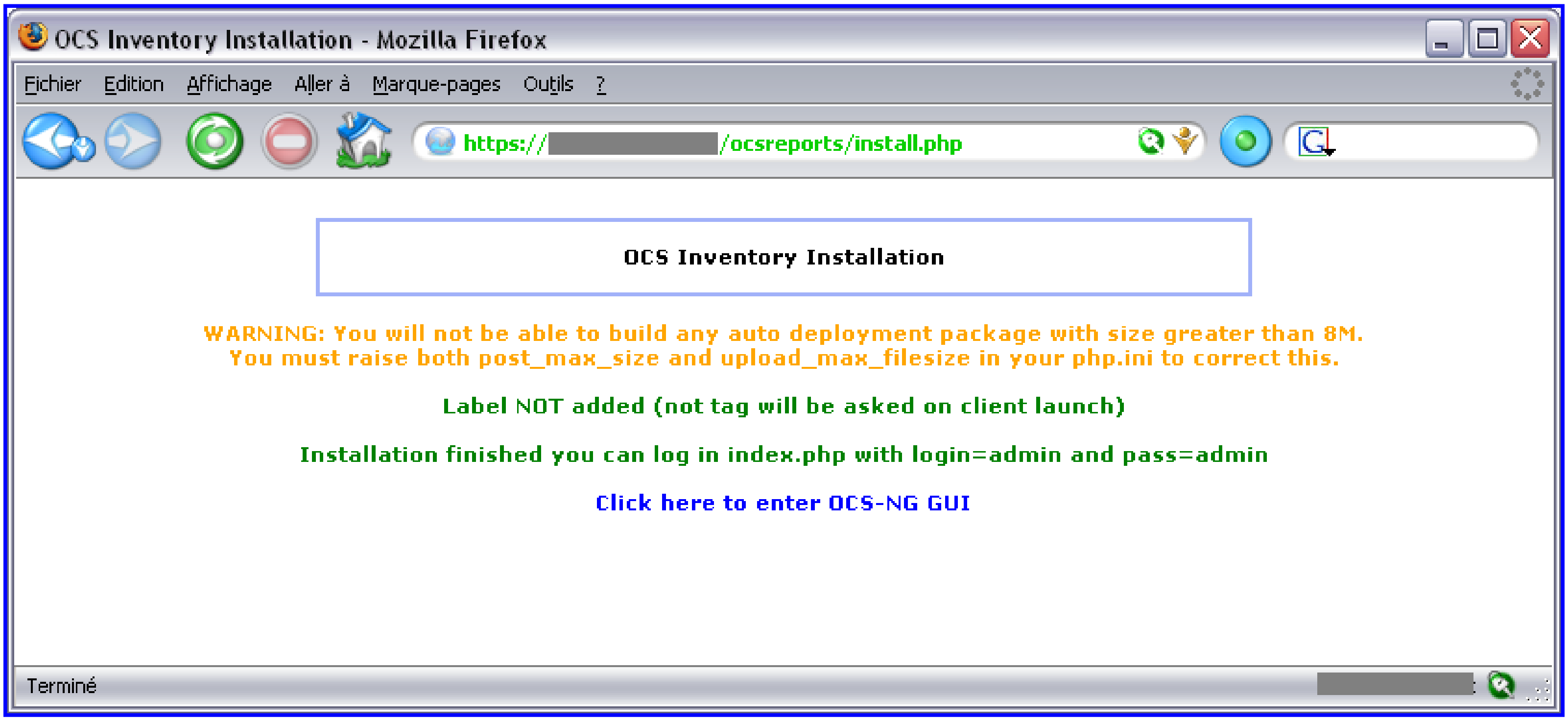
Network netid was computed=> 0 successful, 0 were already computed, 0 were not computable

Netmap netid computing. Please wait...

Netmap netid was computed=> 0 successful, 0 were already computed, 0 were not computable

Please enter the label of the windows client tag input box:
(Leave empty if you don't want a popup to be shown on each agent launch).

Configuration of Management server is now finished.


OCS Inventory Installation - Mozilla Firefox
Fichier Edition Affichage Aller à Marque-pages Outils ?
Terminé

https://[redacted]/ocsreports/install.php

OCS Inventory Installation

WARNING: You will not be able to build any auto deployment package with size greater than 8M.
You must raise both post_max_size and upload_max_filesize in your php.ini to correct this.

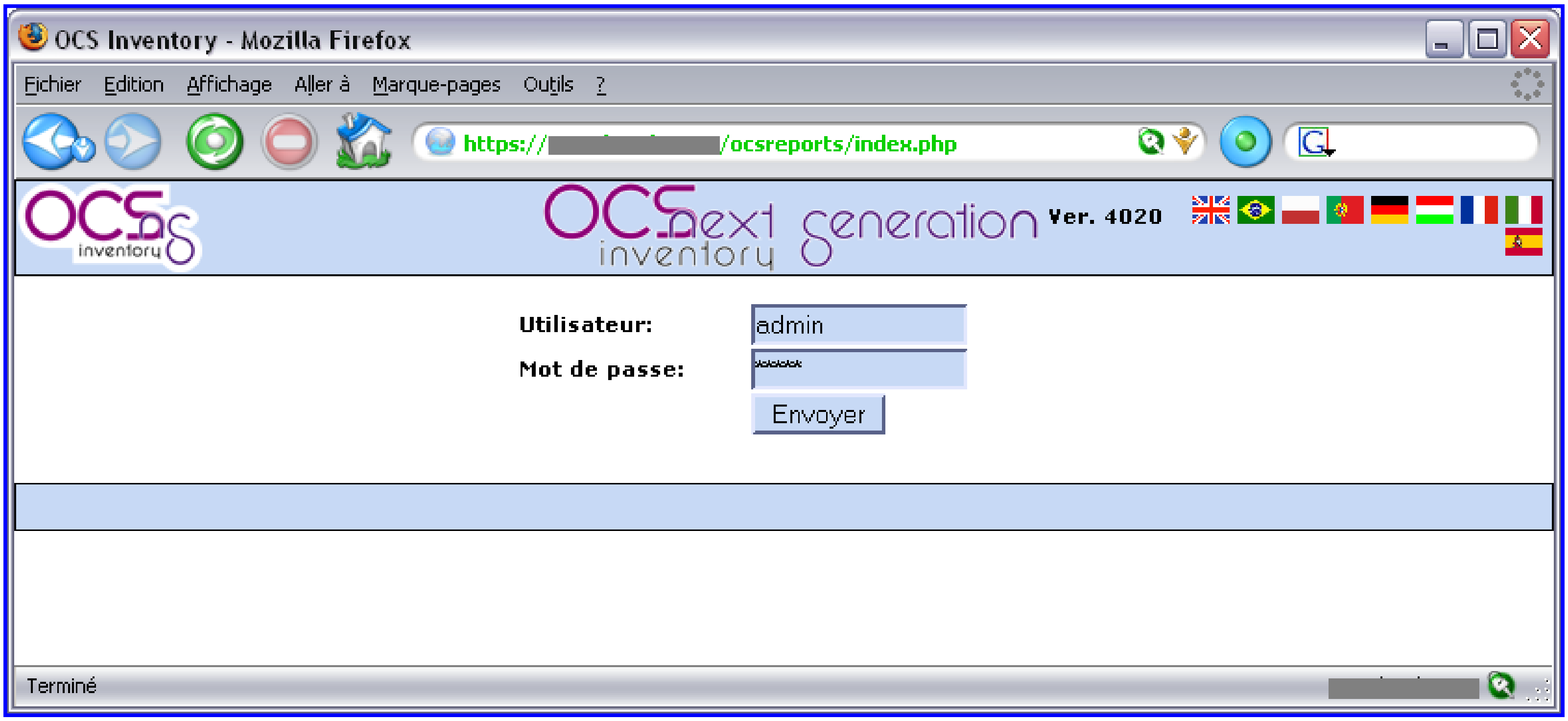
Label NOT added (not tag will be asked on client launch)

Installation finished you can log in index.php with login=admin and pass=admin

Click here to enter OCS-NG GUI

[redacted]

Just point your browser to the URL "http://administration_server/ocsreports" and login in with "admin" as user and "admin" as password.


OCS Inventory - Mozilla Firefox
Fichier Edition Affichage Aller à Marque-pages Outils ?
OCS ng inventory
OCS next generation inventory
Ver. 4020
Envoyer
Terminé

https://[redacted]/ocsreports/index.php

Utilisateur: admin

Mot de passe: *****

[redacted]

Upgrading management server.

When new versions of web communication server or web administration console are released, you must upgrade your installation.

NB: Ensure MySQL InnoDB engine is activated on your database server. Open my.cnf and ensure there is no ligne "skip-innodb" or this line is commented (begins with '#').

Backup your database before upgrading! If you encounter any error while upgrading, restore your database, and upgrade MySQL server to

version 4.1.20 or higher. Then, rerun upgrade procedure.

To upgrade web communication server and administration console, you must follow instructions as described in Installing management server. You don’t need to update Perl modules if not required in the release notes.

Then, just point your favorite browser to URL “http://administration_server/ocsreports” and it ill run the upgrade process to ensure that your database schema and default data are up to date. Upgrade process looks like configuration of management server as described in Configuring management server.

NB: You will see warning regarding max size of package you will be able to deploy. Please, see Uploads size for package deployment.) to configure your server to match your need.

[[Image:]]

Fill in MySQL administrator name (usually root) and password, and MySQL database server address and click “Send” button.

[[Image:]]

Finally, you may fill in a text describing the TAG if you wish to use it.

[[Image:]]

Under Windows Operating System.

We have chosen to package OCS inventory NG server for Windows as an integrated package containing all required components. As is, the 3 main components of Management server (database server, web communication server and web administration server) are installed on the same computer.

OCS Inventory NG server 1.0 for Windows is based on ApacheFriends XAMPP version 1.5.5 (http://www.apachefriends.org/index-en.html) which sets the following components up on a single computer:

- Apache 2.2.3
- MySQL 5.0.27
- PHP 5.2.0 + PHP 4.4.4 + PEAR
- PHP-Switch win32 1.0
- XAMPP Control Version 2.3 from www.nat32.com
- XAMPP Security 1.0
- SQLite 2.8.15
- OpenSSL 0.9.8d
- phpMyAdmin 2.9.1.1
- ADOdb 4.93
- Mercury Mail Transport System for Win32 and NetWare Systems v4.01b
- FileZilla FTP Server 0.9.20
- Webalizer 2.01-10
- Zend Optimizer 3.0.2
- eAccelerator 0.9.5 RC1 for PHP 5.1.6 (comment out in php.ini)
- Perl 5.8.8
- mod_perl 2.0.2

NB: Even if all these components are installed, you will be able to choose the components you want to automatically start.

Installing management server.

You must have Administrator privileges to set OCS Inventory NG server up under Windows NT4, Windows 2000, Windows XP or Windows Server 2003.

Download “OCSNG_WIN32_SERVER_1.01.zip” from OCS Inventory Web Site”, unpack it and launch “OcsWin32ServerSetup.exe.

[[Image:]]

If XAMPP components (server and perl addon) are not already installed, Setup will prompt you that you have to set them up. Otherwise, Setup will automatically install OCS Inventory Server into XAMPP directories.

[[Image:]]

[[Image:]]

Click “Next” button and accept License agreement.

[[Image:]]

Choose installation directory, by default “C:\Program Files\OCS Inventory NG”. You need 400 MB of free hard disk space if XAMPP components are not installed, otherwise, only 10MB are required.

NB: When upgrading, you must ensure that Setup detects the folder including XAMPP directory. See Upgrading management server.

[[Image:]]

Then, you have to validate components to install. Only “OCS Inventory NG Server” is required, if XAMPP components are already installed.

[[Image:]]

NB: OCS Inventory NG Server Setup now use standard XAMPP setup. So, it may be able to upgrade existing XAMPP installation. However, by default, Setup will not upgrade XAMPP components. See Upgrading management server.

Next, you have to choose the program group name in start menu, where OCS Inventory NG icons will be created and then click “Install” button to start installation.

[[Image:]]

[[Image:]]

If XAMPP setup selected, Setup will first launch XAMPP 1.5.5 setup in silent mode. This will create a folder “xampp” under destination folder, and a program group “Apache Friends” in start menu.

You will be prompted to start XAMPP Control Panel. Please, answer “No”.

[[Image:]]

Then, it will launch XAMPP perl addon setup in silent mode.

Last, Setup will install OCS Inventory NG Server files, configure XAMPP Apache and MySQL servers for OCS Inventory NG Server, and automatically start MySQL and Apache servers.

At the end of the process, Setup will launch your default browser to start OCS Inventory NG Server configuration (see Configuring management server.).

Setup is now finished and you can click “Close” button.

[[Image:]]

NB: OCS Inventory NG setup for Windows has installed XMAPP components under “xampp” subfolder of selected installation directory. Apache web server document root directory is located in the “htdocs” sub directory of XAMPP. This is here that “ocsreports” administration console files are installed.

Communication server files are now located into PERL standard libraries.

[[Image:]]

Apache logs (“access.log”, “error.log”, “phperror.log”) and communication server logs (“ocsinventory-NG.log”) are located in the sub-directory “Apache\Logs”.

Configuring management server.

Open your favorite web browser on the server and point it on URL “http://localhost/ocsreports” to connect the Administration server.

You will be prompted for information to connect to MySQL database server with a user who has the ability to create database, tables, indexes, etc:

- MySQL user name, “root” by default
- MySQL user password (empty password by default)

- MySQL hostname, “localhost”

[[Image:]]

NB: Setup will create “ocsweb” database, and a MySQL user “ocs” with password “ocs”. It will also grant to user “ocs” privileges “Select | Insert | Update | Delete | Create | Drop | References | Index | Alter | Create temp | Lock” on database “ocsweb”. *This user will be used by Administration server and Communication server to connect to the database. If you do not wish to use default MySQL user “ocs” with “ocs” password, you must update in the file “dbconfig.inc.php” PHP constants “COMPTE_BASE”, which is MySQL user login, and/or “PSWD_BASE”, which MySQL user password. Don’t forget to also update Communication server configuration, especially in apache configuration file.*

Finally, you may fill in a text describing the TAG, a string displayed at first launch of the agent to ask user to enter the TAG Value. It's a generic data which allows you to sort the new computers (geographical site, first floor, john room....). If you don't want this functionality, just let it blank.

[[Image:]]

Configuration of Management server is now finished.

[[Image:]]

Default Administrator login is “admin” as user and “admin” as password.

[[Image:]]

Updating security of XAMPP components.

By default, XAMPP is set up without security. MySQL root account do not have password, XAMPP web configuration interface is accessible by everybody without authentication…

You must update this.

Open your favorite web browser on the server and point it on URL “http://localhost/xampp/splash.php” to connect the XAMPP configuration GUI.

[[Image:]]

Click on the language you want to access the XAMPP main configuration menu.

[[Image:]]

Then, click “Security” on the left menu. As you will see, all is marked as unsecure or unknown for non started components.

[[Image:]]

You can change this by clicking the link “http://localhost/security/xamppsecurity.php”.

[[Image:]]

First of all, you must fill in MySQL root password and select phpMyAdmin authentication method.

NB: You can change this at any time by visiting the security web page of XAMPP server.

Validate your changes by clicking “Password changing” button.

[[Image:]]

You can then protect the access to XAMPP configuration menu by filling in user and password for XAMPP DIRECTORY PROTECTION. As is, this user and password will be asked to connect to XAMPP configuration menu through a web browser.

Validate your changes by clicking “Make safe the XAMPP directory” button.

[[Image:]]

Do not enable PHP safe mode, as you may encounter errors on Administration console.

Finally, you must restart Apache and MySQL services for changes to take effect.

Open XAMPP Control Panel from system tray or from “OCS Inventory NG” start menu folder, click “Stop” button for Apache, then “Start” button and do the same for MySQL.

You can now reselect “Security” on left side menu to see that all started services are now secured.

[[Image:]]

Upgrading management server.

To upgrade web communication server and administration console, you must follow instructions as described in § 3.2.1 Installing management server. Just ensure that setup detects old installation folder correctly.

You don’t need to update XAMPP components. Setup, by default, will not select XAMPP components install. If you do so, **backup your databases and web sites if you want to also upgrade XAMPP components !** See § 10 Backup/restore of OCS Inventory NG database.

At the end of the process, Setup will launch your default browser to run the upgrade process to ensure that your database schema and default data are up to date. Upgrade process looks like configuration of management server as described in § 3.1.5 Configuring management server.

NB: You will see warning regarding max size of package you will be able to deploy. Please, see § 11.2.4 Uploads size for package deployment.) to configure your server to match your need.

[[Image:]]

Fill in MySQL administrator name (usually root) and password, and MySQL database server address and click “Send” button.

[[Image:]]

Finally, you may fill in a text describing the TAG if you wish to use it.

[[Image:]]

Setting up agent on client computers.

There are 2 methods for inventorying a client computer using OCS Inventory NG agent:

- If the client computer cannot connect to the Communication server, inventory is done locally and is stored in a XML compressed file with ".ocs" extension. User can then send this file through email, USB disk or any other way to the administrator, which will import it in the database through the Administration server.
- If the client computer can reach using HTTP protocol the Communication server through the network, agent ask the Communication server for inventory parameters and send inventory results directly to the Communication server.

Under Windows Operating Systems.

OCS Inventory NG Agent for Windows is able to work as a Windows service, automatically started at computer startup. **However, we also provide a stand alone agent not running as a service,** which can be launched through login script, an Active Directory GPO, a scheduled task or a shortcut in Start menu.

NB: We recommend using the service version of Agent, especially if you plan to use package deployment feature.

Download and unzip OCSNG_WIN32_AGENT_1.01.zip. This package contains 3 files:

- **OcsAgentSetup.exe**, agent installer with Windows service included. We recommend using this package.
- **OcsAgent.exe**, to install standalone agent on a non network connected computer to allow running the inventory manually with /LOCAL command line switch (or if you do not want to use service).
- **OcsLogon.exe**, launcher of OCS Inventory NG agent to use when deploying agent through a login script or Active Directory GPO in the domain. If agent is already installed, it just runs the agent. Otherwise, it downloads agent's binaries from Communication server, set it up and launch it.

Which version of Windows Agent must I use ?

First of all, we have to explain how Agent and Service work.

How does Windows Agent work ?

When OCS Inventory NG Agent "OCSInventory.exe" is launched, it contacts Communication server using HTTP protocol to ask what it has to do. Server can answer "nothing" (not time for an inventory and no package to deploy), and so agent stops.

When agent is launched, it will generate and send an inventory only.

Otherwise, server may answer that Agent has to:

- **Send an inventory**: Agent retrieves all computer properties and send them using HTTP protocol to server. Server answers this only if last inventory date in the database is older than general option "FREQUENCY", specified in days (see § 6.2 Managing OCS Inventory NG general options.)
- **Discover the network**: Agent retrieves all computer properties, scan his sub network for active devices listening on the network, and sends these informations using HTTP protocol to server. Server answers this only if computer is elected to run IPDISCOVERY (see § 7 Using IP discovery feature.)
- **Deploy a package**: Agent contact deployment server using HTTPS protocol to get information file, download package fragments from repository, rebuild package and launch it.

NB: OCS Inventory NG Agent does not listen on the network. It always initiate communication to server. So you do not have to open port on personal firewall. But you must allow OCS Inventory NG agent file "OCSInventory.exe" to contact Communication Server or Deployement Server *using HTTP or HTTPS.*

Each time an inventory is done, Agent writes a configuration file “OCSInventory.dat” in his agent folder where it will put configuration options downloaded from the Communication server.

When launched for the first time, OCS Inventory NG agent will prompt user for the TAG value (if this feature is enabled). Help text displayed in the dialog-box is the one you have entered in Configuring management server. User may enter this value, or leave it blank (you will be able to update this value through the Administration server).

[[Image:]]

Then (or otherwise if TAG feature is not enabled), it will do the inventory and send in HTTP inventory results to Communication server.

How does Windows service work?

NB: You must have Administrator privileges to set up OCS Inventory NG Agent as a service, or you may use OCS Inventory NG Packager to create an installer able to run even if you do not have Administrator privileges. Refer to § 6.3Uploading Agent for deployement through launcher “OcsLogon.exe”. or OCS Inventory NG Packager documentation.

OCS Inventory NG Agent “OCSInventory.exe” is launched by service “OcsService.exe” every PROLOG_FREQ hours. It keeps trace of the countdown in seconds in file “service.ini” (value TTO_WAIT), so it is the time of EFFECTIVE run.

The number of hours to wait is randomized at install time and each time PROLOG_FREQ is changed in Administration Console.

It allows not having all agents contacting Communication Server at the same time. The randomization is between 0 and PROLOG_FREQ.

You can adjust these paramaters considering your server load.

When service launch agent, it calls it using the command line switches specified in value “Miscellaneous” of file “service.ini”.

```
[OCS_SERVICE]

NoProxy=1

Server=my-ocs-server.domain.tld

Pnum=80

Miscellaneous=/DEBUG /NP /server: my-ocs-server.domain.tld /pnum:80

PROLOG_FREQ=10

OLD_PROLOG_FREQ=10

TTO_WAIT=1505
```

Figure 3: Sample file "service.ini"

As you can see, Service is only a launcher which will run Agent regularly, even if nobody logs in the computer.

Do I have to use service or standalone agent ?

You want to have computers inventoried, even if nobody log in ? Use Service version.

You want to use package deployment feature ? Use service version. As is, package will be downloaded in background, and logged in user can continue to work.

You do not want to set service up or have anything appears in the registry ? Use standalone version. However, deployement of package may take a long time when users log in.

In other words, we recommend using service version.

Manually installing Service version of Agent.

Run “OcsAgentSetup.exe” on client computer and click “Next” button

[[Image:]]

Validate license agreement by clicking “I agree” button.

[[Image:]]

Fill in OCS Inventory NG Communication server address and port. If you do not wish to use Microsoft Internet Explorer proxy settings (because your proxy requires authentication for example), enable “No Proxy” checkbox. Miscellaneous field allow you to pass to agent other command line arguments (cf Agent’s command line switches). Then click “Next” button.

[[Image:]]

Choose destination folder, “C:\Program Files\OCS Inventory Agent” by default, and click “Install” button.

[[Image:]]

Click “Close” button to register OCS inventory NG agent service into System.

[[Image:]]

Agent is now installed as a service automatically started at system boot under account LocalSystem.

[[Image:]]

This is the interactive installation setup. However, this process requires that you launch it under all your computers. Hopefully, this installation can be scripted.

NB: Service Agent setup support in command line all Agent switches defined in § Agent's command line switches.

Service Agent setup "OcsAgentSetup.exe" may be called with some command line parameters:

- **/S** to run installation in silent mode, without any user interaction,
- **/UPGRADE** to upgrade an existing Service Agent installation,
- **/NOSPASH** to disable splash screen,
- and all Agent's command line switches, especially /SERVER to specify OCS Inventory NG Communication Server address when using silent installation (see § Agent's command line switches).

Manually installing standalone Agent (without service).

This way may be usefull on a non network connected computer.

Setup can be run by a normal user, or better by a system administrator.

You just have to run file "OcsAgent.exe /local" to launch OCS Inventory NG agent's setup. Setup will try to install OCS Inventory NG agent's files in the folder "C:\ocs-ng" or, if the locally connected user do not have permission to create folder in the root directory, in the folder "ocs-ng" in the user's temporary directory. Then, it will launch OCS Inventory NG agent.

Agent will then prompt user for folder where to store inventory results.

[[Image:]]

When launched for the first time, OCS Inventory NG agent will prompt the user for the TAG value User may enter this value, or leave it blank (you will be able to update this value through the Administration server).

[[Image:]]

When inventory is finished, agent will display a message showing where the inventory results file has been stored.

[[Image:]]

User then just has to send this file to administrator. Administrator will be able to import inventory results into the database through the Administration server.

If you want to run another inventory, you just have to rerun "Ocsinventory.exe /local" from OCS Inventory NG agent's installation folder.

NB: Standalone Agent setup support in command line all Agent switches defined in § Agent's command line switches.

Deploying Agent using launcher OcsLogon.exe through Login Script or Active Directory GPO.

Launcher "OcsLogon.exe" is a little tool able to run inside a login script or an Active Directory GPO. Its goal is to launch OCS Inventory NG Agent on client computers, and if Agent is not installed, to set it up.

Launcher "OcsLogon.exe" will try to connect by default to the Communication Server using a DNS name "ocsinventory-ng", as if you open your favorite web browser and enter the URL http://ocsinventory-ng/ocsinventory.

To use a different URL if you cannot add this DNS name, just rename “OcsLogon.exe” with the DNS name or IP address of the Communication Server (for example “ocsinventory.domain.tld.exe” if you have created for your server a DNS record “ocsinventory.domain.tld” or “192.168.1.2.exe” if your server's IP address is 192.168.1.2). Launcher will then try to connect to the DNS name or IP address you have given as its name (http://ocsinventory.domain.tld/ocsinventory or http://192.168.1.2/ocsinventory).

NB: Always use the latest version of OcsLogon.exe. You can get it from the latest package OCSNG_WIN32_AGENT_XX.zip.

Launcher will first check if OCS Inventory NG agent is installed, and if not, will contact Communication Server in HTTP to download latest agent binaries and set up locally on the computer:

- **Standalone Agent in the folder “C:\ocs-ng”** by default or, if the locally connected user does not have permission to create folder in the root directory, in the folder “ocs-ng” in the user’s temporary directory. If Standalone agent is already installed, launcher will just run the agent.
- **Service Agent in folder “C:\Program Files\OCS inventory Agent”** by default.

*NB: To deploy Standalone Agent with launcher**, you need to have uploaded Standalone Agent** file “ocsagent.exe” with Administration Console.*

*To deploy Service Agent with launcher, **you need to have uploaded Service Agent** file “ocspackage.exe” with Administration Console.*

Refer to § Uploading Agent for deployement through launcher “OcsLogon.exe”.

To choose between Standalone or Service agent, there is just a command line switch “/INSTALL” to specify if you want to use Service or not. If you add “/INSTALL” command line switch, launcher will use Service Agent. Otherwise, it will use Standalone Agent.

If you want to update Agent, you have to specify in launcher command line version of the new release using the switch /DEPLOY:XXXX, where XXXX is the version of agent, 4031 for version 1.01.

Deploying Agent through Active Directory GPO.

NB: We recommend using service version of Agent if you plan to use package deployment feature.

***To deploy Standalone Agent** with launcher**, you need to have uploaded Standalone Agent** file “ocsagent.exe” with Administration Console.*

***To deploy Service Agent** with launcher, **you need to have uploaded Service Agent** file “ocspackage.exe” with Administration Console **and to use “/INSTALL” command line switch** in launcher.*

Refer to § Uploading Agent for deployment through launcher “OcsLogon.exe”.

Open “Active Directory users and computers” tool.

[[Image:]]

Right click on your Active Directory domain or Organisational Unit and select “Properties”.

[[Image:]]

In “Group Policy” tab, create a new policy, or edit existing one.

You either use Computer policy, or User policy. Computer policy will run at computer startup or User policy will run at user login.

NB: In our example, we will use Computer policy and Startup script. Communication Server address is 192.168.1.2, because we choose to set up Service Agent version. If you choose to use Standalone Agent, it's better to use User policy and startup script, to allow agent running each time a user log in.

Expand "Computer configuration" tree in left pane and navigate as shown below to "Windows settings" and "Scripts". Then double click on "Startup" on right pane.

[[Image:]]

Click on "Show files" button to display script and executable files usable by computer startup scripts.

[[Image:]]

Copy launcher "Ocslogon.exe" or the renamed one (in our example "192.168.1.2.exe") into this folder to allow its use by computer startup scripts.

[[Image:]]

Next close "Startup" folder and click "Add" script button, click "Browse" button to select launcher "OcsLogon.exe" or the renamed one (in our example "192.168.1.2.exe"), and fill in launcher parameters (in our example "/S" for silent installation, "/DEBUG" to enable creating log files, "/NP" to disable use of Internet Explorer proxy settings, "/INSTALL" to deploy Service Agent, "/DEPLOY:4029" to force deployment of version 4029, "/SERVER:192.168.1.2" to use Communication Server at address 192.168.1.2).

[[Image:]]

Validate each window to activate Computer Startup script GPO.

[[Image:]]

When computer will start (or when user will log in if using User policy), launcher will set up and/or launch OCS Inventory NG agent.

NB: Launcher OcsLogon.exe may encounter problems accessing the Communication Server if you have configured a proxy with authentication in Microsoft Internet Explorer settings. You can force Launcher to not use proxy with "/NP" command line switch. You can also specify a different IP port to use for Communication server with "/PNUM:XX" command line switch, where XX is the IP port number to use. See § 4.1.5 Agent's command line switches.

Deploying Agent through login script.

NB: We recommend using service version of Agent if you plan to use package deployment feature.

***To deploy Standalone Agent** with launcher**, you need to have uploaded Standalone Agent** file "ocsagent.exe" with Administration Console.*

***To deploy Service Agent** with launcher, **you need to have uploaded Service Agent** file "ocspackage.exe" with Administration Console **and to use "/INSTALL" command line switch** in launcher OcsLogon.*

Refer to § Uploading Agent for deployement through launcher “OcsLogon.exe”.

Copy files “OcsLogon.exe” (or the renamed one) to a shared folder somewhere in your network. This folder must be readable by all your users. Then add a call to “OcsLogon.exe” (or to the renamed one) in your users' login script.

Here is a sample login script.

```
@echo of

echo Running system inventory, please wait…

REM Call to OCS Inventory NG agent for deployment

REM Using shared folder MY_SHARE on server MY_SERVER

REM Connect to Communication server at address 192.168.1.2

REM Enable debug log with /DEBUG to create OcsLogon.log and computer_name.log

REM Force setup agent version 4030 if agent not up to date

REM Deploy service version of agent using /INSTALL

“[////MY_SERVER/MY_SHARE/192.168.1.2.exe \\MY_SERVER\MY_SHARE\192.168.1.2.exe]” /DEBUG /NP /INSTALL /DEPLOY:4030

echo Done. Thanks a lot.
```

Sample login script for Windows domain

NB: To be compliant with Windows 9X, you must enclose path to renamed launcher between quotes in the script, otherwise Windows 9X will not use long filename, but sort name like “192~1.exe” and launcher will not be able to find the correct IP address or DNS name.

Put this script named “ocs.bat” for example on your Domain Controler in the folder “%WINDIR%\SYSVOL\Domain\Scripts”, where “%WINDIR%” is generally “C:\WINNT” or “C:\Windows”.

Next, you have to link login script with every users registered in your Active Directory domain. You can do this using “Active Directory users and computers” tool.

[[Image:]]

For each user, open his properties and in “Profile” tab, add login script name in “Session login script” field.

[[Image:]]

When user log in, launcher will set up and/or launch OCS Inventory NG.

NB: Launcher OcsLogon.exe may encounter problems accessing the Communication Server if you have configured a proxy with authentication in Microsoft Internet Explorer settings. You can force Launcher to not use proxy with “/NP” command line switch. You can also specify a different IP port to use for Communication server with “/PNUM:XX” command line switch, where XX is the IP port number to use. See § 4.1.5 Agent’s *command line switches.*

Agent’s command line switches

OCS Inventory NG Agent version 4.0.1.0 or higher includes some command line switches to allow detecting errors.

Once agent is installed, you can run it manually to diagnose problems. Use “C:\ocs-ng\OCSInventory.exe [options]” (with Standalone Agent) or “C:\Program Files\OCS Inventory Agent\OCSInventory.exe [options]” (with Service Agent) command line where [options] may be in the following command line switches.

Agent’s command line switch	Meaning
/server:[server name]	Tells agent to connect to server “[server name]”
/np	Disable use of proxy defined in Internet Explorer settings.
/pnum:X	Specify an IP port X for web communication server. By default, HTTP port 80 is used. You can force use of port 8080 for example with the argument /pnum:8080.
/local	run agent in local inventory mode. So agent does not try to connect to Communication server. A file “{hostname}.ocs”, containing inventory results in compressed XML, will be created in agent’s directory.
/file	Same as /local, but with interaction with Communication server.
/xml	Agent will create a non compressed XML file “{hostname}.xml”, containing inventory results, in his directory. If not used in conjunction with /LOCAL, agent tries to connect to Communication server.
/notag	Client does not ask user for TAG value.
/tag:”my tag value”	Specify TAG value in command line.
/hkcu	Force agent to search for installed software also under HKEY_CURRENT_USER registry hive.
/debug	Create a log file “{hostname}.log” in agent’s directory.
/force	Force agent to always send inventory results, independently of the FREQUENCY parameter.
/uid	Force agent to generate a new deviceid.
/dmi	If computer serial number cannot be retrieved with WMI, force agent to use DMI tables while running “BiosInfo.exe” tool. Otherwise agent tries first BIOS functions and then DMI tables.
/biosfunc	If computer serial number cannot be retrieved with WMI, force agent to use BIOS functions while running “BiosInfo.exe” tool. Otherwise agent tries first BIOS functions and then DMI tables.
/conf:[filename]	Tells agent to use “[filename]” as configuration file. Otherwise it will use default “Ocsinventory.dat” file.
/test	Agent only tests HTTP connection to communication server and write a file “ok.ok” if all is good. May be only used with /DEBUG, /NP et /PNUM switches (disable all other switches).
/ipdisc:[network number]	Force agent to run IPDISCOVERY feature on network numbered “[network number]” if server ask an inventory. May be used in conjunction with /force to ensure it will run. Cannot be used with /local.
/fastip	agent only scan 5 IPs if it is elected as IPDISCOVER host (only usable for debug or test, may not be used in production).

When using launcher “OcsLogon.exe” to deploy agent or run the agent, you can use the following command line switches.

NB: You can also add all switches supported by OCS Inventory NG Agent.

Launcher command line switch	Meaning
/DEPLOY:XXXX	Force deployment of a specific agent version XXXX. Use “/DEPLOY:4030” (lastest version) to upgrade agent to version 4.0.3.0.
/INSTALL	Tells launcher to set Service Agent up, instead of Standalone Agent.
/URL:[download_url]	Tells launcher to download Standalone Agent “ocsagent.exe” or Service Agent “ocspackage.exe” from “[download_url]” (without final filename). Usefull for deploying Agent in mulitple Active Directory domain, but using a single Communication Server.
/NP	Disable use of proxy defined in Internet Explorer settings.

/PNUM:X	Specify an IP port X for web communication server. By default, HTTP port 80 is used. You can force use of port 8080 for example with the argument /pnum:8080.
/LOCAL	run agent in local inventory mode. So agent does not try to connect to Communication server. A file "{hostname}.ocs", containing inventory results in compressed XML, will be created in agent's directory.
/NOTAG	Client does not ask user for TAG value.
/TAG:"my tag value"	Specify TAG value in command line.
/DEBUG	Create a log file "{hostname}.log" in agent's directory.
/FOLDER:[PATH]	Tells launcher to set agent up into "[PATH]" installation folder. MUST BE THE LAST COMMAND LINE ARGUMENT.

Under Linux Operating Systems.

OCS Inventory NG agent for Linux can only be set up locally. You cannot deploy the agent through the network as it's possible for Windows agent. However, you can choose during setup to activate auto-update of the agent if you've choosen HTTP inventory method.

NB: You must have root privileges to set Administration server up.

Requirements.

OCS Inventory NG Agent for Linux requires:

- dmidecode version 2.2 or higher
- PERL 5.6 or higher
  - Perl module XML::Simple version 2.12 or higher
  - Perl module Compress::Zlib version 1.33 or higher
  - Perl module Net::IP version 1.21 or higher
  - Perl module LWP::UserAgent version 5.800 or higher
  - Perl module Digest::MD5 version 2.33 or higher
  - Perl Module Net::SSLeay version 1.25 or higher
- Make utility
- C/C++ compiler like GNU GCC

NB: It's better for system integrity to use precompiled packages for your distribution if they are available.

***On Fedora/Redhat like Linux**, you can use "yum" tool to set required modules up like following:*

yum install perl-XML-Simple

yum install perl-Compress-Zlib

yum install perl-Net-IP

yum install perl-LWP

yum install perl-Digest-MD5

yum install perl-Net-SSLeay

***On Debian like Linux**, you can use "apt-get" tool to set required modules up:*

apt-get install libxml-simple-perl

apt-get install libcompress-zlib-perl

apt-get install libnet-ip-perl

apt-get install libwww-perl

apt-get install libdigest-md5-perl

apt-get install libnet-ssleay-perl

New installer script "setup.sh" is able to install these dependencies if they are not available. However, it will never upgrade an installed module. If one module has version lower than required once, you must upgrade yourself.

NB: installer does not set required components up for dependencies. For example, Net::SSLeay requires openssl to be installed. If not installed, setup of Net::SSLeay will fail and OCS Inventory NG agent setup will also fail.

Also, installer script produces a log file "setup.log". If you encounter any error while installing OCS Inventory NG agent, please refer to this file to have detailed error message.

Installing the agent interactively.

Download "OCSNG_LINUX_AGENT_1.01.tar.gz" from OCS Inventory Web Site.

Unpack it.

- tar –xvzf OCSNG_LINUX_AGENT_1.01.tar.gz

Run "setup.sh" installer. During the installer, default choice is presented between []. For example, [y]/n means that "y" (yes) is the default choice, and "n" (no) is the other choice.

- cd OCSNG_LINUX_AGENT_1.01
- sh setup.sh

NB: installer writes a log file "ocs_agent_setup.log" in the same directory. If you encounter any error, please refer to this log for detailed error message.

You will then have to choose between 2 methods for generating inventory:

1. http: computer is connected to the network and is able to reach the Communication server with HTTP protocol.
2. local: computer is not connected to the network and inventory will be generated in a file to manually send to OCS Inventory NG operator.

Enter "http" or validate if your computer can reach OCS Inventory NG Communication server, or enter "local" to enable local mode.

[[Image:]]

Enter OCS Inventory NG communication server address.

[[Image:]]

Enter OCS Inventory NG Communication server port, or validate if Communication server runs on standard HTTP port 80.

[[Image:]]

Enter a value for TAG.

[[Image:]]

Setup will check for PERL interpreter binary, C/C++ compiler and make utility. If one of these components is not found, setup will stop.

Setup will check for:

- dmidecode binary.
- Compress::Zlib PERL module
- XML::Simple PERL module
- Net::IP PERL module
- LWP::UserAgent PERL module
- Digest::MD5 PERL module
- Net::SSLeay PERL module

If not found, it will ask you if you wish to install it. Enter “y” or validate to enable install of required component. If you enter “n”, setup will stop here.

[[Image:]]

[[Image:]]

[[Image:]]

If all is OK, or you’ve chosen to install dependencies, setup will then do the following:

- Unpack, configure, build and install dmidecode if needed.
- Unpack, configure, build and install Compress::Zlib if needed.
- Unpack, configure, build and install XML::Simple if needed.
- Unpack, configure, build and install Net::IP if needed.
- Unpack, configure, build and install LWP (libwww-perl) if needed.
- Unpack, configure, build and install Digest::MD5 if needed.
- Unpack, configure, build and install Net::SSLeay if needed.
- Compile ipdiscover binary.
- Configure OCS Inventory NG agent PERL module.
- Build OCS Inventory NG agent PERL module.
- Install OCS Inventory NG agent PERL module into PERL standard library directories.
- Create a symbolic link “/usr/sbin/ocsinv” to run OCS Inventory NG agent manually.
- Create OCS Inventory NG agent’s log directory (/var/log/ocsinventory-NG by default).
- Configure daily log rotation for OCS Inventory NG agent (file /etc/logrotate.d/ocsinventory-client by default)
- Create OCS Inventory NG agent’s configuration file “ocsinv.conf” into “/etc/ocsinventory-client” directory
- Create OCS Inventory NG agent’s administrative information file “ocsinv.adm” into directory “/etc/ocsinventory-client” directory to store TAG and administrative data values.
- Create a cron task to launch OCS Inventory NG agent every day (default file “/etc/cron.d/ocsinventory-client”)

- Launch OCS Inventory NG agent to ensure all parameters are OK.

[[Image:]]

Here is a sample configuration file for OCS Inventory NG Linux agent.

```
<CONF>

<DEVICEID>computer.domain.tld-2006-02-27-13-59-47</DEVICEID>

<DMIVERSION>2.2</DMIVERSION>

<IPDISCOVER_VERSION>3</IPDISCOVER_VERSION>

<OCSFSERVER>my_ocs_com_server.domain.tld:80</OCSFSERVER>

</CONF>
```

Figure 5 : Sample agent’s configuration file ocsinv.conf for a network connected computer.

Deploying agent through scripted installation without user interaction.

Download “OCSNG_LINUX_AGENT_1.01.tar.gz” from OCS Inventory Web Site.

Unpack it.

- tar –xvzf OCSNG_LINUX_AGENT_1.01.tar.gz

Run “setup.sh” installer with the following command line arguments:

- cd OCSNG_LINUX_AGENT_1.01
- sh setup.sh <SETUP DEPENDENCIES> <SERVER ADDRESS> [<SERVER PORT> <TAG VALUE>]

where parameters values are:

- <SETUP DEPENDENCIES> must be “1” if want to enable automatic setup of missing dependencies, “0” to disable (setup will fail if there is missing dependency).
- <SERVER ADDRESS> must be the IP address or DNS name of OCS Inventory NG Communication server. If you plan to set agent up in local mode in a non network connected computer, you must set <SERVER ADDRESS> to “local”.
- <SERVER PORT> may be OCS Inventory NG Communication server port if you’re not using standard HTTP port 80.
- <TAG VALUE> may be the value of TAG, between quotes.

Command line parameters <SETUP DEPENDENCIES> and <SERVER ADDRESS> are required. The other parameters are optional, but if you wish to set <TAG VALUE>, you must also specify previous optional parameter <SERVER PORT>.

Example:

- **sh setup.sh 1 ocsng.domain.tld** will set agent up, installing missing dependencies if needed, and connecting to OCS Inventory NG Communication Server “ocsng.domain.tld”
- **sh setup.sh 0 ocsng.domain.tld 8080** will set agent up, without installing missing dependencies, and connecting to OCS Inventory NG Communication Server “ocsng.domain.tld” on port 8080

- **sh setup.sh 0 192.168.1.2 80 "my tag value"** will set agent up, without installing missing dependencies, and connecting to OCS Inventory NG Communication Server "192.168.1.2" on port 80, and setting TAG to "my TAG value".

NB: installer writes a log file "ocs_agent_setup.log" in the same directory. If you encounter any error, please refer to this log for detailed error message.

Agent's command line switches

If you encounter error, agent's produce a log file in directory "/var/log/ocsinventory-client".

However, agent's also support some command line switches. You can use them while launching the agent manually using "ocsinv" command:

Agent's command line switch	Meaning
-local	Runs the agent in local mode, without any connection to communication server. You will be prompted for a target directory where agent will put inventory results in XML compressed file with ".ocs" extension.
Agent's command line switch	Meaning
-xml	Agent will create a non compressed XML file with ".ocs" extension, containing inventory results. You will be prompted for a target directory where agent will put the file. If not used in conjunction with -local, agent tries to connect to communication server.
-nosoft	Do not search for installed software.
-tag="my tag value"	Set agent setting TAG value to "my TAG value".
-force	Force agent to always send inventory results, independently of the FREQUENCY parameter.
-info	Show a detailed output of agent runs.
-debug	Force agent to produce a more detailed log file, showing XML exchange with communication server.

Querying inventory results.

Point your browser to the URL "http://administration_server/ocsreports" and log into Administration console (default credentials are "admin" as login and "admin" as password).

[[Image:]]

You must use top left menus to run predefined queries.

[[Image:]]To view all computers

[[Image:]]To view Computer/TAG repartition

[[Image:]]To search computers using various criteria

Each of these queries' results can be customized by adding, removing columns, or changing the number of displayed lines per page. Theses customizations settings are saved per user between sessions. This means that when you come back to OCS Inventory NG Administration console, your settings are restored as they were during your last visit.

NB: Computers marked with a red bullet at beginning of line are computers whith specific customization parameters. They may have specific inventory frequency, ipdiscover status or have package deployment affected.

All computers.

This query will allow you to display all inventoried computers. Computers marked with a red bullet at the beginning of line are those which have specific constomization options.

Just click on a computer name to display its properties in a new browser window.

[[Image:]]

[[Image:]]

[[Image:]]

[[Image:]]

[[Image:]]

- Top banner - Display general informations for the current device
- Links section – Just click on the appropriate link to display the corresponding information.
- Bottom section – Use “show everything” to display all sections. To print the currently displayed information, use “print this page”.
- Special section Administrative data – Use this section to display the device's administrative data. This page fits with your settings in “admininfo” tab. Use the “update” button to change values.
- Special section Customization – Use this section to customize configuration option for individual computer. You can also select here package to deploy on a particular computer.

TAG / number of PC repartition.

This query allows you to display all machines grouped by TAG account info. Click on the computer count to retrieve the corresponding devices.

For example, if you’ve choosen to set your TAG information to reflect your geographical sites, this will display the number of computer in each different site.

[[Image:]]

Search with various criteria.

This query allows you to search for computers having specific feature.

[[Image:]]

You can add new parameters to the search query by dropping down the combo-box and selecting it in the list.

Default search parameters are:

- BIOS Manufacturer
- BIOS Version
- Computer name
- Customized IpDiscover

- Deployment Package
- Description
- Domain
- Free Disk Space
- Inventory Frequency
- Gateway
- IP Address
- IpDiscover Status
- Last Inventory
- MAC Address
- System Manufacturer
- Memory
- System Model
- Monitor Caption
- Monitor Manufacturer
- Monitor Serial Number
- Network Number
- Operating System
- Processor Speed
- Registry Key Value
- System Serial Number
- Software
- Tag value
- User logged in
- User agent (show OCS NG agent version)
- And all the administrative information you've defined.

For each parameter, you can use one of the following comparison operators, depending of the parameter you've selected:

- EQUAL
- DIFFERENT
- SMALLER
- BIGGER
- BETWEEN
- OUT OF

NB: Don't forget to enable the parameter in the search!

[[Image:]]

NB: You can search for as many software as you wish in one search query. This means that you can search for computers having software1 and software2 and… Unlike software field, all others fields can only be specified once in any given search query.

Administration of OCS Inventory NG.

Point your browser to the URL "http://administration_server/ocsreports" and login into Administration console (default credentials are "admin" as login and "admin" as password).

You may at least change the password of the default administrator, or better, add a new one and remove the default one.

NB: All these features are only available to OCS Inventory NG administrators.

Managing OCS Inventory NG Administration server users.

[[Image:]]Click on the toolbar “Users” menu to display all configured OCS Inventory NG Administration server users.

[[Image:]]

You can add new users by entering their name, password (user will be able to change it when logged in), and selecting their type. You can choose between:

- Administrator: user has the ability to configure all parameters of the product.
- User: user can only query the database and view results of inventory. It just has the top left Combo-box of menu toolbar to run general queries.

To delete a user, just click on red cross at the end of the corresponding line.

Managing OCS Inventory NG general options.

[[Image:]]Click on the toolbar “Config” menu to display all general options.

[[Image:]]

Here is the meaning of each option:

Configuration option	Meaning
AUTO_DUPLICATE_LVL	Choose what value you want to enable to detect double computers (renamed, reinstalled…). If you check multiple values, server will try to detect double comparing these values.
DEPLOY	Activates or not the automatic deployment of new agent release option.
DOWNLOAD	Activates or not package deployment feature. Turning off DOWNLOAD stop this functionnality on the server AND on the agents. With DOWNLOAD off, once agents will have contacted OCS server, they will stop current download WITHOUT cleaning packages.
DOWNLOAD_CYCLE_LATENCY	Time in seconds to wait between each downalod cycle (cf § 8.1 How does it work?)
DOWNLOAD_FRAG_LATENCY	Time in seconds to wait between each fragment download (cf § 8.1 How does it work?)
DOWNLOAD_PERIOD_LATENCY	Time in seconds to wait between each download period (cf § 8.1 How does it work?)
DOWNLOAD_PERIOD_LENGTH	Number of cycles per period (cf § 8.1 How does it work?)
DOWNLOAD_TIMEOUT	Validity in days of a package on an agent. If the time used to download a package is over DOWNLOAD_TIMEOUT days, package will be cleaned and ERR_TIMEOUT will be sent to ocs server.
FREQUENCY	Specify the frequency in days of inventories.
INVENTORY_DIFF	Enable or not differential inventory to speed up the server. With differential inventory, only changes are stored by the server, not full inventory.
INVENTORY_TRANSACTION	Enable or not transaction on server. With transaction, an inventory is stored only if all data have been processed correctly.
IPDISCOVER	Specify the number of agent to ask running IP discovery feature for each gateway (sub network). If you leave the default value 2, this mean that the Communication server will ask the 2 most active computers of each sub network to run IP discovery feature. If you set it to 0, IP discovery will be disabled.
IPDISCOVER_LATENCY	Time in seconds to wait between scan of each IP address. (cf § 7.2 How does it work?)
IPDISCOVER_MAX_ALIVE	Maximum number of days between two inventories for an IP Discovery enabled computer to hold his status of IP discovery computer. An IP discovery enabled computer will lose his status if it has not been seen by the Communication server for more days than the number of days defined in this setting. Another computer in the same sub network will then be designated.
LOCAL_PORT	IP port of OCS Inventory NG Communication Server.

LOCAL_SERVER	IP address or DNS name of OCS Inventory NG Communication Server.
LOGLEVEL	Enable or not detailed log for Communication server. If enabled, server will write logs to file “ocsinventory-NG.log” in directory “/var/log/ocsinventory-NG” for Linux and “…\xampp\apache\logs” for Windows.
PROLOG_FREQ	Number of hours between two run of the agent (useful for the service). Therefore, the agent will contact (not necessary send an inventory if inventory is not older than FREQUENCY days) ocs server every PROLOG_FREQ hours.
REGISTRY	Activates or not the registry query function (for Windows agent only).
TRACE_DELETED	Activates or not tracking of deleted/renamed computers for integration with GLPI. Enable this feature only if you use integration with GLPI asset management software.
UPDATE	Not used, always set to OFF.

Click “Update” button when you set all changes.

Uploading Agent for deployement through launcher “OcsLogon.exe”.

This feature works only for network connected computers able to connect to the Communication server.

OCS Inventory NG is able to automatically install agent on computers when launcher “OcsLogon.exe” is used through login script or GPO. Agent’s files are downloaded from the Communication server.

You just have to upload the agent package into the Administration console and to activate the deployment feature by setting “DEPLOY” general option to ON (see § 6.2 Managing OCS Inventory NG general options.).

Uploaded file must be one of the following:

- “ocsagent.exe” file for Windows agent, to deploy Agent without Windows service. This file is included in package OCSNG_WIN32_AGENT_XX.zip.
- “ocspackage.exe” file, created using OCS Inventory NG Packager, to deploy Windows service version of Agent, even if user connected does not have Administrator privileges.

To create “ocspackage.exe” file, just run OCS Inventory NG Packager, and fill in following informations:

- Path to file “OcsAgentSetup.exe”, installer of OCS Inventory NG service Agent, included in package OCSNG_WIN32_AGENT_XX.zip.
- Path to the Certificate file to use, for checking server certificate when using package deployement feature.
- Optionally, another file to include in setup.
- Command line parameters for running “OcsAgentSetup.exe” service installer, at least “/S” to run installer in silent mode, and “/SERVER: my_ocs_com_server_address” to specify “my_ocs_com_server” as address of OCS Inventory NG Communication Server.
- Username (**account@domain** for Active Directory account and **domain\account** for a NT4 account) and password of an Administrator account on client computers. “OcsAgentSetup.exe” will be run under this account on client computers, to allow installing service even if user connected does not have Administrator privileges.

[[Image:]]

NB: Refer to OCS Inventory NG Packager documentation for more informations on how to use Packager.

“OcsAgentSetup.exe” supports the following command line parameters (and all Agent’s parameters defined in § 4.1.5 Agent’s command line switches):

- **/S** Use quiet, silent installation
- **/SERVER:IP_ADDRESS** Use Commincation Server address “IP_ADDRESS”
- **/PNUM:XX** Use port XX instead of default HTTP port 80.

- **/NP** Do not use Microsoft Internet Explorer Proxy settings
- **/DEBUG** Produce a log file of OCS Inventory NG agent execution. This mode is automatically used under Windows to launch agent as a service through an entry in registry key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices.

This will create the file “ocspackage.exe” to upload into Administration console.

[[Image:]] Click “Agent” toolbar menu, browse your hard drive to select agent file and click “send” button.

[[Image:]]

NB: If you encounter error while uploading agent, refer to common errors § 11.2.3 PHP Requested content-length.)

Using Registry Query feature.

OCS Inventory NG agent for Windows is able to query the registry of inventoried computers for a value of a key or for all values of a key under registry hives HKEY_CLASSES_ROOT, HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, HKEY_USERS, HKEY_CURRENT_CONFIG (and HKEY_DYN_DATA for Windows 9X based computers).

You have to enable this feature in the general settings “REGISTRY”.

[[Image:]]You can then define your registry query by clicking “Registry” toolbar menu.

[[Image:]]

Click “Add” button to add a new query. Enter a name for this query, for example MS Office XP if you want to retrieve MS Office XP registration number, select the registry hive (HKEY_LOCAL_MACHINE in this example), enter the registry key (SOFTWARE\Microsoft\Office\10.0 \Registration\{9011040C-6000-11D3-8CFE-0050048383C9}) and the value name to query (ProductID) and validate. Put star (*) in field “Name of the key” to get all values of the key (This is useful to get all values of key “HKLM\Software\Microsoft\Windows\CurrentVersion\Run” to know which process are automatically started for example).

[[Image:]]

After that, each time a Windows client connects; it will retrieve and store all the values corresponding to these keys.

Managing duplicates computers.

OCS Inventory NG is able to detect a renamed computer, reinstalled computer…

Generally, it will handle that alone. But sometimes, it is impossible for the server to know whether two computers are the same or not, for example when it has no network adapter (no Mac address) or when the serial number was not properly set by the constructor (If you change a computer’s name, the application won’t be able to recognize it if it has no serial or no Mac address, and thus a duplicate fake computer may be created).

[[Image:]]This page accessible by clicking “Duplicates” toolbar menu is used to solve this problem.

[[Image:]]

You may choose the kind of comparison you want in the top right combo-box.

- Redundancy summary – Shows the number of redundant computers detected with each comparison method.
- “Hostname + Serial number”, “Hostname + Mac address”, “Mac address + Serial number” - These are the two criteria comparison methods, the more reliable. It returns all computers that have two criteria in common.
- “Hostname only”, “Serial only”, “Mac address only” - These are the one criterion comparison methods; it only shows all computers that shares one parameter.

Once computers are shown, it’s up to you to check whether several computers are redundant or not.

[[Image:]]

After that, you only have to select (checkbox) computers that look the same, and click “merge redundant computers”: all the data from checked computers will be merged.

The administrative data of the oldest computer are kept on the merged device.

NB: When those filters are applied, some values are filtered out (For example: 'NNNNNNN' or 'xxxxxxxxxxx' for serial number).

Editing administrative information.

OCS Inventory NG allows you to store custom information for each inventoried computers. For example, you can add administrative information to specify the owner of the computer, or the location of the computer.

This administrative information is stored both on the server and on the client to avoid any loss of data.

For example, if the database is lost, all clients will come back with their administrative information.

[[Image:]]You can define the administrative information you want for each computer by clicking “Admininfo” toolbar menu.

[[Image:]]

You can add new administrative information definition by entering their name (space or special characters are not allowed) and selecting their type. You can choose type between:

- Text.
- Integer.
- Real.
- Date.

To delete an administrative information definition, just click on red cross at the end of the corresponding line.

For example, we will add administrative information corresponding to the buy date of computer.

[[Image:]]

You can now browse all your computers to set the buy date for each.

Editing the label

If the “DEPLOY” option is activated, clients may get the “label” file. It is used to fill the popup shown on first start of a client.

This popup asks for the “TAG” value that is used to class computers (it may be, for example, a unit code, the name of one of your company’s section, a street number etc.)

[[Image:]]This “label” file is generated during the server install, and may be edited by clicking “Label” toolbar menu.

[[Image:]]

If you submit a blank label, the label file will be deleted, and no window will be displayed anymore on client’s computer.

Importing inventory for non network connected computers.

With Administration server through your web browser

For non network connected computers, you can import inventory results from a file created on the computer by the agent run in local mode (see § 3.2 We have chosen to package OCS inventory NG server for Windows as an integrated package containing all required components. As is, the 3 main components of Management server (database server, web communication server and web administration server) are installed on the same computer.). We assume that you’re able to save this file in your hard disk from your mailbox, or any other way.

NB: If Communication Server is not running on standard HTTP port 80, or on the same computer as Administration console, you must update General Options. See § 6.2 Managing OCS Inventory NG general options.)

[[Image:]]Click “Local import” toolbar menu, browse your hard drive to select the “.ocs” file created by agent release and click “send” button.

[[Image:]]

With Communication server through a Perl script

Communication server provides the ability to import inventory from a file created on the computer. This file has “.ocs” extension.

We assume that administrator can get the file from somewhere, his mailbox, a USB drive or any other place.

Go to the directory where you’ve installed Communication server, “/usr/local/ocsinventory-NG” for us, and run the script “Ocsinventory_local.pl” with path to file which contains inventory results as argument.

NB: With OCSInventory NG Server for Windows, import script “local_import.bat” is located in “INSTALLDIR\binutils” directory, where “INSTALLDIR” is the installation folder selected during server setup.

This will import inventory results into the database.

```
[root@l16753101aao ocsinventory-NG]# Ocsinventory_local.pl ST32491DL-2005-06-23-10-04-02.ocs

OK for ST32491DL-2005-06-23-10-04-02.ocs

--------------------------

Successly inventoried : 1

Errors : 0

:-)
```

Figure 6 : Sample import of a single inventory results file.

If you have multiple files to import, you can put them in a directory and them launch the script “Ocsinventory_local.pl” without argument, but within the directory where resides files to import.

```
[root@l16753101aao home]# Ocsinventory_local.pl

OK for ST32491DL-2005-06-23-10-04-02.ocs

--------------------------

Successly inventoried : 1

Errors : 0

:-)
```

Figure 7 : Sample of multiple inventory results files in directory home.

Using software dictionary for GLPI integration.

Software dictionary is used to categorize detected software. This feature is very useful when you use OCS Inventory NG combined with GLPI. As is, you can group software in a category and GLPI will import them with the name of category.

For example, you have many version of Microsoft Office (97, 2000, XP, 2003…), but you don’t care about version. You just want to manage total number of MS Office. You can create a category “Microsoft Office” and insert in this category all version of MS Office. In GLPI, you will only see “Microsoft Office”. Software dictionary acts as a rename or group utility for GLPI.

[[Image:]]Click “Dictionary” toolbar menu.

There are 3 categories by default:

- NEW: include all new or not yet categorized software.
- IGNORED: you can put in this category all software you don’t want to import in GLPI.
- UNCHANGED: you can put in this category all software you don’t want to “rename” in GLPI. These software will be imported as is.

[[Image:]]

You can search software with part of his name by fill in input and clicking “Search software EVERYWHERE” button.

If you manage many categories, you can search a category by fill in part of his name in input and clicking “Search category” button.

Click on a category name to display software included.

[[Image:]]

You can select software to move to new category (you must fill in the new category name) or to an existing category (select it in the dropdown).

If you check “Move all”, all software, even those not displayed, will be moved to the desired category, not only the selected ones.

You can select all displayed software by clicking “The page” and deselect all displayed software by clicking “Nothing”.

Using IP discovery feature.

IP discovery feature allow OCS Inventory NG to discover all network connected devices on the network.

For this, Communication server asks a number of most “active” computers running OCS Inventory NG agent to scan for MAC addresses in their sub network at each run. **They will not scan the entire wide area network, but only their local network defined by the couple IP address/subnet mask.**

Introduction.

Inventory software is very useful for administrator. It allows “enlightening” his computers stock. Today, with use of TCP/IP, we can say that enlightenment is done at the same time for all the enterprise network, especially if, like OCS Inventory NG, working is natively network based.

But, what about devices which do not send inventory, for many reasons like a forgetting, a lack of cooperation from users? What about from all “alive” devices which cannot run inventory agent (printers, switches, routers, WiFi access points…)? What about computers which do not have to be connected on your network and which are conspicuous for their discretion?

IPDISCOVER try to answer to those problematic. Even if it can work independently, it matches perfectly to OCS structure. As working is based on a communication between all hosts of information system and a central server, it’s easy for the server to order at his “subordinate” to make some little tasks, as getting registry key, doing an inventory or retrieving all devices answering on his IP segment.

How does it work?

Retrieving information.

OCS NG system is based on a dialog between agents installed on computers and Apache module on OCS NG server. Exchange is done in compressed XML and allows configuring agent tasks.

When a computer send an inventory result, server try to determine if it needs some other computers (number can be configured) to scan hosts in this sub network. Gateway IP address are used to cartography enterprise network.

If it’s needed, server estimate host quality and decide to activate or not host as an ipdiscover enabled computer. In this case, computer will send systematically an inventory, independently of general configuration parameter 'FREQUENCY'.

Election mechanism.

Once server has determined that there is a need for the selected gateway, it evaluates the following criteria:

- *OS*: operating system must be Windows XP or Windows 2000 (all versions) or Linux.
- *QUALITY*: this parameter means the host connection average to the server in days. It is evaluated dynamically by Communication server only when there is more than the number (defined by ‘IPDISCOVER’ option) of inventoried hosts for a gateway. If current computer sending inventory results has better quality than another IPDISCOVER enabled computer for this gateway, current computer will replace the other one. ‘IPDISCOVER’ option must be greater than zero to enable this feature!
- *FIDELITY*: total connection number to the server of the computer. This number must be at least 3, to allow QUALITY to be computed from representative data.
- *NETMASK*: sub network mask. It must describe a maximum of B class IP network (2 first bytes to 255 => 255.255.X.X).
- *LASTDATE*: when Communication server compute QUALITY, if it finds a host which hasn’t sent inventory results from number of days defined by ‘IPDISCOVER_MAX_ALIVE’ option, it will replace this host by a new one.

NB: You can customize agent’s ipdiscover settings for each computer from the administration console. You can totally disable ipdiscover or force ipdiscover on a specific network. These options are available on computer properties page, under “Customization” section. However, the election mechanism is the best way to do the network discovery. Use ipdiscover customization with care.

How do agents work?

Once agent has received order to proceed to discovery of his sub network, it identifies first network interface to use. It tries then to resolve through ARP all IP addresses answering on his segment (delay between 2 hosts acan may be defined using option “IPDISCOVER_LATENCY”, see § 6.2 Managing OCS Inventory NG general options.). All devices answering to the question are stored in XML inventory result and sent to server.

Server tuning

When doing a new installation of OCS NG, we understand easily that it requires some times to be ready to enable ipdiscovery feature. It requires some times to grab all gateways and to elect computers for ipdiscovery. As a computer can only be elected if it REALLY provides an inventory, it may be wiser to configure at the beginning ‘FREQUENCY’ option to zero, always generate an inventory. You can increase this value later, when infrastructure will be ready.

Analyzing errors (thought ipdiscover-util.pl used directly or from web interface) will allow you to detect potential problems. More the value of QUALITY will be lesser, more your ipdiscover information will be up to date.

System will be giving you the best in a domain, with a daily authentication and inventory. Tests done with this configuration on 20 000 hosts and 250 sub networks generate a DAILY actualization of 15 000 IP addresses.

To finish, value set for IPDISCOVER will tell to server how many computers are wished by gateway to run this task (if you set this value to zero, feature will be disabled)**.**

Working with results.

[[Image:]]You can view which computers run the IP discovery scans by clicking on toolbar “Security” menu (“IpDisc” menu if some languages).

[[Image:]]

Manage names of your networks.

You may define your sub networks by a name and a unique ID, to view results easily.

Click on “Config” menu and then on “Subnet names” menu to manage your sub networks.

[[Image:]]

Enter sub network name (ex DMZ), ID, IP address and IP network mask, then click “Send” button to validate.

To remove a sub network definition, just click red cross at the end of corresponding line.

Show list of networks

You can view the list of sub network configured in your network by selecting “Network information” menu.

[[Image:]]

For each network, you will be able to view how many inventoried hosts, non inventoried hosts, IPDISCOVER feature enabled hosts and identified hosts (known hosts manually registered in the database) are connected to this network.

Click on the number of needed column to view each type of devices.

Show inventoried hosts in the network.

You can view the list of inventoried hosts (computers with OCS Inventory NG agent installed) on your network by clicking on number in column “Inventoried” in the network list.

[[Image:]]

Show uninventoried network devices.

You can view the list of active network devices detected with IPDISCOVERY on your network by clicking on number in column “Non inventoried” (computers without OCS Inventory NG agent installed) in the network list.

[[Image:]]

If there is, in the list, devices you know as legitimate, you can register them so they will not be displayed next time. For this, just click the icon at the end of corresponding line. Before, you may want to register “Device type” to easily identify known hosts (see §7.3.8 Registering known hosts.).

You may also analyze this network by clicking “Analyze” button. IPDISCOVER-UTIL perl script will be used to determine, for each network device, his NetBios name or DNS name and the type of operating system.

NB: This feature uses IPDISCOVER-UTIL perl script available only under Linux Server which requires the following components.

- *nmap (tested on 3.75)*
- *nmblookup (part of the samba suite, tested on 3.0.7/3.0.10)*
- *Perl module Net::IP*
- *Perl module DBI*
- *Perl module DBD::mysql*
- *Perl module XML::Simple*

The following types are used:

- **Windows**: host seems to run one version of Microsoft Windows operating system.
- **Linux**: host seems to run under Linux operating system.
- **Network**: operating system cannot be determined, so it can be network device such as router, managed switch, printer, or host running Sun Solaris or IBM AIX… Maybe, host is running some firewall software?
- **Phantom**: host is not responding at this time. Maybe it is powered off or there is a firewall ?

[[Image:]]

If a network device is legitimate, you can register it by clicking on icon “Register” at the end of corresponding line. You will be able to enter a brief description and to select network device type while registering it (see § 7.3.8 Registering known hosts.).

Show IPDISCOVER enabled hosts.

You can view the list of hosts running IPDISCOVERY feature (computers with OCS Inventory NG agent installed, and this agent is elected by server to run discovery of his sub network) on your network by clicking on number in column “IpDiscover” in the network list.

[[Image:]]

Show known or identified hosts.

You can view the list of known hosts already identified by someone on your network by clicking on number in column “Identified” in the network list.

[[Image:]]

Managing known device types.

You can register devices (such as routers, switches, network printers, computers for which there is no inventory agent…) as you know they are legitimate. As is, they will not be displayed in the list of uninventoried devices, to allow you concentrate on suspicious devices.

You may first define some device types to identify them easily.

Click on “Config” menu to manage your device types.

[[Image:]]

You can add new device type by entering the “Type name” you want and clicking “Send” button.

You can remove any device type by clicking on the red cross at the end of corresponding line.

Registering known hosts.

You can add new devices by browsing list of uninventoried devices and clicking the icon at the end of corresponding line. This will bring you to page for adding or removing network devices and “MAC” field will be automatically fill in.

[[Image:]]

To remove a network device, just click on the red cross at the end of corresponding line.

Scanning an IP address

You can scan query specific IP address to obtain information about the host. Go to “Security” menu and click on “IP querying” menu and then enter IP address, network mask and click “Send” button.

NB: This feature uses IPDISCOVER-UTIL perl script available only under Linux Server which requires the following components.

- *nmap (tested on 3.75)*
- *nmblookup (part of the samba suite, tested on 3.0.7/3.0.10)*
- *Perl module Net::IP*
- *Perl module DBI*
- *Perl module DBD::mysql*
- *Perl module XML::Simple*

[[Image:]]

IPDISCOVER-UTIL perl script will use NMAP and NMBLOOKUP utilities to get information about the host (DNS name, NetBios name…) and also show if host is inventoried and/or discovered.

[[Image:]]

Deploying packages or executing commands on client hosts.

OCS Inventory NG includes package deployment feature on client computers. From the central management server, you can upload packages which will be downloaded through HTTP/HTTPS and launched by agent on client computer.

NB: This feature has been tested with OCS Inventory NG Agent for Windows service only. As software installation requires Administrator privileges, agent launched through a login script or shortcut in start menu under user account may not be able to launch software installation. Also, background download of package may take a long time, and may block login script. So, we do not recommend using package deployment feature using login script inventory.

How does it work?

A package has 4 main components:

- a priority,
- an action,
- optionnaly a ZIP or TAR.GZ file including how many files and directories you want,
- and optionnaly a command to launch.

There are 11 levels of priority, level 0 to 10. Level 0 is the highest priority and level 10 the lowest. Package of priority level 0 will be deployed before package of priority 1. Package of priority level 1 will be deployed before package of priority 2…

Action is associated with file to deploy and command to launch. This triplet may be one of the following:

- **Action Launch:** to deploy a ZIP or TAR.GZ file and launch with or without parameters an executable file **included** in ZIP or TAR.GZ file. ZIP or TAR.GZ file will be uncompressed into a temporary directory, and associated command (name of executable file without path!) will be launched into this temporary directory.**This action allows retrieving result code of launched command.**

- **Action Execute:** to deploy a ZIP or TAR.GZ file (optional), and launch with or without parameters an executable file **included or not** in ZIP or TAR.GZ file.If executable is not included in ZIP or TAR.GZ file, it must be part of software already installed on client computer. Typcally, it may be a Windows standard command like Windows Installer call, RPM or DPKG or TAR.GZ command on Linux.ZIP or TAR.GZ file will be uncompressed into a temporary directory, and associated command (name of executable file with path or parameters if needed) will be launched into this temporary directory.**This action does not allow retrieving result code of launched command.** However, this action allows you running command on client computers, without deploying any file. For example, you can use it to run specific operating system configuration command.

- **Action Store:** to deploy a ZIP or TAR.GZ file and only store his content on a folder of client computer.**There is no command associated with this action, only a path to specify where to store extracted files.**

NB: All packages you want want to deploy must be compressed with ZIP for Windows Agents and tar gzipped for Linux computers.

If you want to build your own installer, you may want to look at NullSoft Installer System (http://nsis.sourceforge.net) or Inno Setup (http://www.jrsoftware.org). These tools are GPL installer for Windows able to create one file self extracting installer.

For example, this feature allows you to create a ZIP package including Media Player Classic executable, a sub directory including some MP3 files and a play list for Media Player Classic referencing these MP3 into sub directory. Associated command will be a call to Media Player Classic with command line switch to launch play list. Once this package will be downloaded on Windows clients, users will have Media Player Classic launched and playing MP3 from play list. Beautifull, isn't it ;-)

You create through administration console your deployment package. It is automatically described by:

- A reference in database, used by Communication server to ask agent to download the package.
- An information file, named "info". It is an XML file describing the package and action agent will have to launch,
- 0 or more data fragment files. File you will upload (if there is one) will be splitted in small parts to allow agents downloading parts by parts, and then easely resuming a failed download. If download of a fragment fails, only this fragment will be downloaded another time, not all the package. You will be able to choose fragment size according to your network capabilities.

NB: as you will upload your package through Administration console, you may configure PHP and Apache to allow uploading large files. See § 11.2.4 Uploads size for package deployment. to know how to configure this.

Once package is built, you must activate it. It indicates where is located SSL enabled web server (i.e. deployment servers) where agent will able to download information file and fragment files.

Finaly, you must select on which computer you will deploy the package.

From now, agent is able deploying the package.

When agent send an inventory to Communication server, Communication server tell the agent if he has one or more packages to deploy, with the level of priority of each package, and where it can find information files.

Agent then begins a download period. A period is composed of cycles, defined by configuration option “DOWNLOAD_PERIOD_LENGTH”. By default, a period contains 10 cycles.

At each cycles, it compute “cycle’s number modulo package priority”. If it equals to 0, it download package fragment files. After each fragment, it will wait “DOWNLOAD_FRAG_LATENCY” (configuration option set to 10 seconds by default) before downloading the next fragment.

When all fragment of package are downloaded, it will launch package command and wait “DOWNLOAD_CYCLE_LATENCY” (configuration option set to 60 seconds by default) before beginning a new cycle and incrementing cycle number.

When all cycle of a period have been processed, it waits “DOWNLOAD_PERIOD_LATENCY” (configuration option set to 0 seconds by default).

If all packages have been successfully downloaded and installed, it stops. If not, it begins a new period of cycles.

CAUTION: Priority level 0 is a special level. All packages with priority 0 will be downloaded before all others packages with greater priority at the beginning of each cycle. If download fails, agent will retry to download failed packages of priority 0, without checking others package. So it can completely stop deployments. USE PRIORITY LEVEL 0 WITH CARE!

You may use these settings to customize your network bandwith usage. By increasing latency options, you will increase time to download fragments and reduce network use average.

By increasing period length option, you will delay new download of failed fragments, but also, by decreasing period length to a value lower than 10, you can stop downloading package with priority level higher than this value.

Requirements.

Deployement server storing information files must have SSL enabled, as downloading the deployement information file is very critical. This information file contains description of package and command to launch. So, if somebody can usurp your deployement server, he may launch any command he wants on your computers. That’s why deployment server must use SSL to allow agents authenticating the server and ensuring this is the real deployment server.

Agent must have a certificate to validate deployment server authentication. This certificate must be stored in a file named “cacert.pem” in OCS Inventory NG agent’s folder under Windows, and in directory “/etc/ocsinventory-client” under Linux.

Under Windows, you can use OCS Inventory NG Packager (see Uploading Agent for deployement through launcher “OcsLogon.exe”.) to create an agent installer which include certificate, or you can use the following sample login script to copy certificate file in agent’s folder (we assume that agent is installed under “C:\Program Files\OCS Inventory Agent” and certificate file is available on a share “MYSHARE” on server “MYSERVER”).

```
@echo off

REM Check if CA file exists

if exist “C:\Program Files\OCS Inventory Agent\cacert.pem” goto CA_END

REM CA file does not exists, install it
```

Copy [//MYSERVER/MYSHARE/cacert.pem \\MYSERVER\MYSHARE\cacert.pem] “C:\Program Files\OCS Inventory Agent\cacert.pem”

CA_END

If you have a Public Key Infrastructure, you must create a valid server certificate for your deployment server and copy your Authority certificate file into file “cacert.pem”.

If you do not have a Public Key Infrastructure, you can use a self signed certificate for your deployement server, and copy server certificate into file “cacert.pem”.

Refer to § 8.8 Using SSL certificates in Package deployment. For more informations.

Creating packages.

First of all, you must build your package.

[[Image:]]Point your mouse on “Deployment” menu and select “Build”.

[[Image:]]

Enter a name for your package.

Select operating system for this package. You can choose between Windows and Linux.

Select download protocol for this package. At this time, only HTTP protocol is available.

Select priority on this package. You can choose level 0 to 10 for priority. Package with lesser priority will be downloaded before package of greater priority, except if download fails (cf § 8.7 Deployment statistics and success validation.).

You may also choose to warn user that something is being launched on his computer. Set “Warn user” dropdown list to “YES”, fill in text to display to user, how long to display the text before auto validating package installation (set 0 to wait indefinitely) and if user can cancel deployment or delay deployement to next inventory.

You may also specify if package deployment needs a user interaction by setting dropdown list “Installation completion need user action” to “YES”, for example, if setup needs that user fill in a informations on a dialog to terminate.

Last, you can select your action in “Action” dropdown list. Here are some samples describing what kind of package you can build.

Deploying package through “Launch” command.

Package you want to deploy has one or more files, with at least an executable file for launching package setup. Compress theses files using ZIP tool if package addresses Windows computers, using tar and gzip if package addresses Linux computers.

Choose action “Launch” and click “Browse” button to select your ZIP or TAR.GZ file.

In field “Command”, just fill in name of executable file without path, but with, optionally, parameters. It’s this command which will be launched on client computer once package will be downloaded and uncompressed to a temporary directory.

In our following example, we deploy a new release of OCS Inventory NG Agent for Windows, using silent installation, specifying Communication server address my_ocs_com_srv, disabling use of IE proxy settings and enabling debugging mode. So, ZIP file only include file “OcsAgentSetup.exe” and the “File name” field contains:

- OcsAgentSetup.exe /S /SERVER:my_ocs_com_srv /NP /DEBUG

[[Image:]]

Click “Send” button to upload package to Administration console.

Next, you must specify the size of each fragment of package to allow agent downloading package by small parts. This will allow download resuming. If download of a fragment fails, only this fragment will be downloaded another time, not all the package. So choose fragment size according to your network capabilities.

Administration console will then split package in fragments and store them in a folder named as package timestamp in directory “download” of apache web server root directory. It will also create in the same directory the package information file named “info”, an XML file describing the package and action agent will have to launch.

[[Image:]]

Deploying package through “Execute” command.

Package you want to deploy has one or more files, with optionally an executable file for launching package setup. Compress theses files using ZIP tool if package addresses Windows computers, using tar and gzip if package addresses Linux computers.

Choose action “Execute” and click “Browse” button to select your ZIP or TAR.GZ file.

In field “Command”, just fill in path of executable file to launch with parameters (full path is not required as application executable is listed on system search path, or is included in package). It’s this command which will be launched on client computer once package will be downloaded.

NB: Environnement variables are expanded in “Command”. It enables you to use things such as %SystemDrive%, %SystemRoot%, %windir%, %ProgramFiles%, %CommonProgramFiles% ...etc.

In our following example, we deploy software using silent Windows Installer installation. So, ZIP file only include file “software.msi” and the “Command” field contains:

- msiexec.exe /i software.msi /quiet

[[Image:]]

Click “Send” button to upload package to Administration console.

Next, you must specify the size of each fragment of package to allow agent downloading package by small parts. This will allow download resuming. If download of a fragment fails, only this fragment will be downloaded another time, not all the package. So choose fragment size according to your network capabilities.

Administration console will then split package in fragments and store them in a folder named as package timestamp in directory “download” of apache web server root directory. It will also create in the same directory the package information file named “info”, an XML file describing the package and action agent will have to launch.

[[Image:]]

Command through “Execute” command.

Package you want to deploy is only a command launch.

Choose action “Execute” and leave field “File” empty.

In field “Command”, just fill in name of command with, optionally, parameters. It’s this command which will be launched on client computer once package will be downloaded.

NB: Environnement variables are expanded in “Command”. It enables you to use things such as %SystemDrive%, %SystemRoot%, %windir%, %ProgramFiles%, %CommonProgramFiles% ...etc.

In our following example, we deploy a command to specify proxy address to use for System Applications under Windows. So the “Command” field contains:

- Proxycfg.exe /p 192.168.1.1

[[Image:]]

Click “Send” button to upload package to Administration console.

As you’ve not selected to upload a file, screen to configure fragement size is not displayed. Administration console will only create, in a folder named as package timestamp in directory “download” of apache web server root directory, the package information file named “info”, an XML file describing the package and action agent will have to launch.

Stored package through “Store” command.

Package you want to deploy has one or more files, to be stored in a specific folder on client computers. Compress theses files using ZIP tool if package addresses Windows computers, using tar and gzip if package addresses Linux computers.

Choose action “Store” and click “Browse” button to select your ZIP or TAR.GZ file.

In field “Path”, just fill in path where agent will store extracted files once package will be downloaded.

NB: Environnement variables are expanded in “Command”. It enables you to use things such as %SystemDrive%, %SystemRoot%, %windir%, %ProgramFiles%, %CommonProgramFiles% ...etc.

Also, if provided folder path does not exist, it will be recursively created.

In our following example, we deploy a file to store in folder “C:\My Folder”:

[[Image:]]

Click “Send” button to upload package to Administration console.

Next, you must specify the size of each fragment of package to allow agent downloading package by small parts. This will allow download resuming. If download of a fragment fails, only this fragment will be downloaded another time, not all the package. So choose fragment size according to your network capabilities.

Administration console will then split package in fragments and store them in a folder named as package timestamp in directory “download” of apache web server root directory. It will also create in the same directory the package information file named “info”, an XML file describing the package and action agent will have to launch.

[[Image:]]

Activating package

Once package have been created, you must specify where agent can download it.

Agent will first download package information file. As this file is very critical, this download must be done using HTTP over SSL (HTTPS) to ensure that agent can authenticate deployment server. Next, download of package fragment described in information file will be done using standard HTTP.

NB: If you do not want to use Administration server as deployment server, you must first copy folder “download/package_timestamp” from Administration server Apache document root directory to another web server. You may want to use a directory synchronization utility like rsync (http://samba.anu.edu.au/rsync) to automatically do this task; otherwise, we will have to do it manually.

You may also choose to host information file on a different web server than the one which hosts fragment files. For example, if you have multiple geographical sites with only one central Communication server, you may want to host information files on Communication server, and fragment files on a web server on each site. For this, you need to activate a package per site, and for each package, information file will be hosted on Communication server and fragment files on site web server. This will dramatically decrease intersite network bandwidth use.

[[Image:]] Point your mouse on “Deployment” menu and select “Activate”. You will view here all built package, and also ALL activated package.

You can click the red cross to delete a built package. This will delete package reference from database and also delete information file and fragment files from Administration console download directory. So, deleted package will be unavailable for activation, all activated packages using this package will be deleted, and also unaffected from computers.

[[Image:]]

Click “Active” button on the line corresponding to the package you want to activate.

In field “HTTPS url”, enter URL for download in HTTPS package information file.

In field “HTTP url”, enter URL for downloading in HTTP package fragment files.

NB: Do not enter localhost as server address in URL! Remember that these URLs will be processed by agents.

If your HTTPS or HTTP deployement server works on non standard ports, you can specify working port using the standard notation “server_address:server_port/folder”. For example, your deployement server works on port HTTP 8080 and HTTPS 4343 on server 192.168.1.1, and packages are located under /download durectory. You must fill in

https url: 192.168.1.1:4343/download

http url: 192.168.1.1:8080/download

In our case, we’ve choosen to use Administration server as deployment server for both package information file and package fragments.

So we have filled in in both fill something like “ocs-admin-srv.domain.tld/download”.

[[Image:]]

Click send button. Administration console will ensure that both information file and package fragment files are available on specified URLs.

“Non notified” column shows you the number of computers which haven’t yet been notified they have corresponding package to deploy.

“Success” column shows you the number of computers which have successfully deploy corresponding package.

“Errors” column shows you the number of computers which encounter errors deploying corresponding package.

The “Stats” icon allows you to view percentile of computers which are waiting for notification, those which are notified (server ask them to deploy the package), and those which have finished deploying with result code (SUCCESS or ERROR).

Affecting packages to computers.

You can affect package to computer one by one, by displaying computer properties, selecting “Customization” icon and adding the package”. However, this is not the best way if you would like to affect package to many computers.

The best way is to use “Search with various creteria” functions to search for computers you want, and to affect package to all these computers in one time.

In the following example, we will affect package we’ve created to all Windows XP computers.

So first, we search for Windows XP computers.

[[Image:]]

This search returns 59 computers on 4 pages.

[[Image:]]

Just click “Deploy” to deploy on all computers returned by the search, not only on visible ones.

[[Image:]]

Click on “Affect” icon of package line to affect this package to all selected computers.

Agents on computers will be notified at next Communication Server contact they have this package to deploy. So, while agent do not contact Communication server, computer will appear in console with status “WAITING NOTIFICATION”. Once agent has contacted Communication server, the status will be “NOTIFIED”.

Unactivating packages.

[[Image:]] Point your mouse on “Deployment” menu and select “Activated”. You will see here all packages available for deployments on computers.

[[Image:]]

The red cross icon will unaffect package for ALL computers, and unactivate package. So package is still referenced in database, information and fragment files are still available on Administration server download directory. However, it has the same status as if you have just built it and can be activated again.

Deployment statistics and success validation.

As package may have been activated and then unactivated, deployment statics are in “Activate” menu.

You can show deployement statics by clicking “Stats” icon for a package.

Since you’ve affected package at least to one computer, you will have graphical stats showing deployment notification status.

[[Image:]]

Status may be once of the following:

Status code	Meaning
WAITING NOTIFICATION	Server is waiting for agent communication to notify there is something to download.
NOTIFIED	Agent has been notified there is something to download. Now waiting for result code.
SUCCESS [code]	Agent has successfully download package and launch command or stored extracted data. With “Launch” action, this status may be completed with command execution return code.
ERR_ALREADY_SETUP	Package was previously installed successfully on this computer.
ERR_BAD_ID	Agent is unable to download package because it cannot find package ID on deployment server.
ERR_BAD_DIGEST	Downloaded data are has bad digest, so agent does not execute associated command.

ERR_DOWNLOAD_PACK	Agent was unable to uncompress download ZIP or TAR.GZ file.
ERR_BUILD	Agent was unable to rebuild package fragments.
ERR_EXECUTE	Agent was unable to execute associated package command.
ERR_CLEAN	Agent was unable to clean downloaded package.
ERR_TIMEOUT	Agent was unable to download package during DOWNLOAD_TIMEOUT days.
ERR_ABORTED	User canceled package command execution (you've choosen to notify him, and allowed him to cancel).
ERR_EXECUTE_PACK	Not used

"Validating Success" will clear statistics of computers which have successfully deployed package.

"Unaffect not notified" will unaffect package from computers which do not have contacted server since you've affected package to computers. Package will not be deleted, only computers which do not have yet receive order to deploy this package will have this order cancelled.

"Validate all" will clear all statistics, and unaffect package from non notified computers. It is the same as "Validate Success" + "Unaffect not notified".

NB: You MUST validate deployment status once deployement ended to clear database deployement status for computer. Otherwise, database will grow up and speed down !

You can click on number for each status line to display computers having this deployment status.

Using SSL certificates in Package deployment.

Package deployment infrastructure is too much powerfull, so it requiress SSL access to validate server before trying to download something from. So you need some SSL certificates for use with your deployment server.

Certificate definition from http://en.wikipedia.org/wiki/Public_key_certificate

"In cryptography, a ***public key certificate*** *(or* ***identity certificate****) is a certificate which uses a digital signature to bind together a public key with an identity — information such as the name of a person or an organization, their address, and so forth. The certificate can be used to verify that a public key belongs to an individual.*

In a typical public key infrastructure (PKI) scheme, the signature will be of a certificate authority (CA). In a web of trust scheme, the signature is of either the user (a self-signed certificate) or other users ("endorsements"). In either case, the signatures on a certificate are attestations by the certificate signer that the identity information and the public key belong together."

You can use a quick, easy but limited way**, self signed certificate,** or a more secure and reliable tool**, a PKI with a Certificate Authority.**

Apache web server comes with OpenSSL cryptographic library, which allow creating and managing certificates.

Using self signed certificates.

NB: Take care about certificate validity period, as web server self signed certificate must be installed on each client computer running the agent. When certificate will expire, you will have to generate and deploy new certificate on each client computer!

With OCS Inventory NG Server for Linux.

Usually, Apache or mod_ssl packages come with sample scripts to generate certificates, especially test certificates.

However, we provide below a sample script using OpenSSL for generating a self signed certificate for use in Apache.

```
#!/bin/sh

#
# First, generate apache server certificate request
#
# Generate 1024 bits RSA key, store private key in a
# no password protected PEM file server.key, using
# system default openssl configuration file.
#
echo
echo Generating Apache server private key...
echo
openssl genrsa -out server.key 1024

#
# Next, sign the apache server certificate with the apache
# server key
#
# Sign with PEM certificate server.crt, using PEM file
# server.key for server private key, using system default
# openssl configuration file.
#
# The produced certificate will be valid for 1825 days (about 5 years)
#
echo
echo Generating Apache server self signed certificate...
echo
```

```
openssl req -outform PEM -new -key server.key -x509 -days 1825 -out server.crt
```

Figure 8: Sample apache_generate_cert.sh script

This script generates a RSA private key in file “server.key” and an X.509 self signed certificate in file “server.crt”.

First, launch this script using command:

- sh apache_generate_cert.sh

It will generate private key, and prompt you for certificate properties:

- Country code, usually required
- State or province name, usually required
- City, usually required
- Organisation or company name, usually required
- Organisational Unit name, usually optional
- Common name (this is the DNS name or IP address of your server), required
- An email address, usually optional

[[Image:]]

In our sample, we’ve generated self signed certificate for our server name “ocs.domain.tld”.

Next, you just have to copy server certificate file “server.crt” and server private key file “server.key” files into appropriate directories and update Apache/mod_ssl configuration files to use these files.

Here is a sample and minimalist Apache/mod_ssl configuration for using SSL under CentOS/Fedora/RedHat Linux. (server certificate is stored under “/etc/httpd/conf/ssl.crt” directory and server key is stored under “/etc/httpd/conf/ssl.key” directory).

NB: Generally, Apache/mod_ssl configuration is provided for your system. So, do not use following configuration if your system already has a configuration file for mod_ssl !

```
#

# This is the Apache server configuration file providing SSL support.

# It contains the configuration directives to instruct the server how to

# serve pages over an https connection. For detailing information about these

# directives see <URL:http://httpd.apache.org/docs-2.0/mod/mod_ssl.html>

#

# For the moment, see <URL:http://www.modssl.org/docs/> for this info.

# The documents are still being prepared from material donated by the

# modssl project.
```

```
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#

LoadModule ssl_module modules/mod_ssl.so

# Until documentation is completed, please check http://www.modssl.org/
# for additional config examples and module docmentation. Directives
# and features of mod_ssl are largely unchanged from the mod_ssl project
# for Apache 1.3.

#
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
#
# To allow connections to IPv6 addresses add "Listen [::]:443"
#
Listen 0.0.0.0:443

#
# Some MIME-types for downloading Certificates and CRLs
#
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl

# Pass Phrase Dialog:
# Configure the pass phrase gathering process.
# The filtering dialog program (`builtin' is a internal
# terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog builtin
```

```
##
## SSL Virtual Host Context
##

<VirtualHost _default_:443>

# Use separate log files:
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# SSL Cipher Suite:
# List the ciphers that the client is permitted to negotiate.
# See the mod_ssl documentation for a complete list.
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP

# Server Certificate:
# Point SSLCertificateFile at a PEM encoded certificate. If
# the certificate is encrypted, then you will be prompted for a
# pass phrase. Note that a kill -HUP will prompt again. A test
# certificate can be generated with `make certificate' under
# built time. Keep in mind that if you've both a RSA and a DSA
# certificate you can configure both in parallel (to also allow
# the use of DSA ciphers, etc.)
SSLCertificateFile /etc/httpd/conf/ssl.crt/server.crt

# Server Private Key:
# If the key is not combined with the certificate, use this
# directive to point at the key file. Keep in mind that if
# you've both a RSA and a DSA private key you can configure
# both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile /etc/httpd/conf/ssl.key/server.key

# SSL Engine Options:
# StdEnvVars:
# This exports the standard SSL/TLS related `SSL_*' environment variables.
# Per default this exportation is switched off for performance reasons,
# because the extraction step is an expensive operation and is usually
# useless for serving static content. So one usually enables the
# exportation for CGI and SSI requests only.
SSLOptions +StdEnvVars

# SSL Protocol Adjustments:
# The safe and default but still SSL/TLS standard compliant shutdown
# approach is that mod_ssl sends the close notify alert but doesn't wait for
# the close notify alert from client. When you need a different shutdown
# approach you can use one of the following variables:
# o ssl-unclean-shutdown:
# This forces an unclean shutdown when the connection is closed, i.e. no
# SSL close notify alert is send or allowed to received. This violates
# the SSL/TLS standard but is needed for some brain-dead browsers. Use
# this when you receive I/O errors because of the standard approach where
# mod_ssl sends the close notify alert.
# o ssl-accurate-shutdown:
# This forces an accurate shutdown when the connection is closed, i.e. a
# SSL close notify alert is send and mod_ssl waits for the close notify
# alert of the client. This is 100% SSL/TLS standard compliant, but in
# practice often causes hanging connections with brain-dead browsers. Use
# this only for browsers where you know that their SSL implementation
# works correctly.
# Notice: Most problems of broken clients are also related to the HTTP
# keep-alive facility, so you usually additionally want to disable
# keep-alive for those clients, too. Use variable "nokeepalive" for this.
# Similarly, one has to force some clients to use HTTP/1.0 to workaround
# their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
# "force-response-1.0" for this.
SetEnvIf User-Agent ".*MSIE.*" \
nokeepalive ssl-unclean-shutdown \
downgrade-1.0 force-response-1.0

# Per-Server Logging:
# The home of a custom SSL log file. Use this when you want a
# compact non-error SSL logfile on a virtual host basis.
CustomLog logs/ssl_request_log \
"%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>
```

Figure 9: Sample Apache/mod_ssl configuration file

Once you’ve configured your Apache web server, don’t forget to restart Apache daemon for changes to take effect.

Last, you have to install server certificate file “server.crt” on each client computer into OCS Inventory Agent installation directory, under the name “cacert.pem”.

With OCS Inventory NG Server for Windows.

XAMPP Apache distribution comes with a script “makecert.bat” for generating self signed certificates. This script is located under “INSTALL_PATH\xampp\apache” directory (where INSTALL_PATH is the installation folder of OCS Inventory NG Server)

```
@echo off
set OPENSSL_CONF=./bin/openssl.cnf

if not exist .\conf\ssl.crt mkdir .\conf\ssl.crt
if not exist .\conf\ssl.key mkdir .\conf\ssl.key
```

```
bin\openssl req -new -out server.csr

bin\openssl rsa -in privkey.pem -out server.key

bin\openssl x509 -in server.csr -out server.crt -req -signkey server.key -days 365


set OPENSSL_CONF=

del .rnd

del privkey.pem

del server.csr


move /y server.crt .\conf\ssl.crt

move /y server.key .\conf\ssl.key


echo.

echo -----

echo Das Zertifikat wurde erstellt.

echo The certificate was provided.

echo.

Pause
```

Figure 10: XAMPP "makecert.bat" default script

This script generate self signed certificate usable for 365 days. If you want to increase certificate validity, you must update directive “-days 365” to specify in days new validity period (1825 days, about 5 years must be up a good value ;-).

Just double run script “makecert.bat”. It will generate a RSA private key and ask you for a password (at least 4 characters).

Enter password and confirm it.

[[Image:]]

Next, you will be prompted for certificate properties:

- Country code, usually required
- State or province name, usually required
- City, usually required
- Organisation or company name, usually required
- Organisational Unit name, usually optional
- Common name (this is the DNS name or IP address of your server), required
- An email address, usually optional

- A challenge password (must be empty, just press enter)
- An optional company name

Finally, you will be prompted for private key password.

[[Image:]]

Now, self signed certificate is created and installed. Just restart Apache2 service for changes to take effect.

Last, you have to install server certificate file “INSTALL_PATH\xampp\apache\conf\ssl.crt\server.crt” on each client computer into OCS Inventory Agent installation directory, under the name “cacert.pem”.

Using PKI with Certificate Authority.

We assume that you’re already using an internal PKI or commercial one like Verisign.

However, if you don’t have an internal PKI, and don’t want to pay for certificates, you can use services provided by cacert.org (http://www.cacert.org), a free worldwide PKI provider. Using cacert.org services require that you register your email and DNS domain name, before to be able to request server certificate. See cacert.org manuals.

You may take a look at Pablo Iranzo Gómez excellent article (http://alufis35.uv.es/OCS-Inventory-Package-Deployment.html) for more detailled instructions about using cacert.org certificates in OCS Inventory NG.

With OCS Inventory NG Server for Linux.

Usually, Apache or mod_ssl packages come with sample scripts to generate certificates request to submit to a PKI provider.

However, we provide below a sample script using OpenSSL for generating a certificate request for use in Apache.

```
#!/bin/sh

#

# Generate server certificate request

#

# Generate 1024 bits RSA key, store private key in a

# no password protected PEM file server.key, store certificate

# request in a PEM file server.csr, using system default

# configuration file

#

# The produced key will be valid for 1825 days (5 years)

#
```

```
echo

echo Generating server private key and certificate request...

echo

openssl req -newkey rsa:1024 -outform PEM -out server.csr -keyout server.key -keyform PEM \

-days 1825 -nodes
```

Figure 11: Sample apache_request_cert.sh script

This script generates a RSA private key in file “server.key” and a certificate request in file “server.csr”.

First, launch this script using command:

- sh apache_request_cert.sh

It will generate private key, and prompt you for certificate request properties:

- Country code, usually required
- State or province name, usually required
- City, usually required
- Organisation or company name, usually required
- Organisational Unit name, usually optional
- Common name (this is the DNS name or IP address of your server), required
- An email address, required to receive certificate generated by Certificate Authority.
- An optional challenge password
- An optional company name

[[Image:]]

In our sample, we’ve generated certificate request for our server name “ocs.domain.tld”.

Next, you must transmit your certificate request “server.csr” to your PKI Certificate Authority.

Once you’ve received your server certificate from Certificate Authority, you just have to copy server certificate file “server.crt” and server private key “server.key” files into appropriate directories, and update Apache/mod_ssl configuration files to use these files.

You must also retreive Certificate Authority root certificate into file “ca_root.crt” to specify it in Apache configuration.

Here is a sample and minimalist Apache/mod_ssl configuration for using SSL under CentOS/Fedora/RedHat Linux. (server certificate is stored under “/etc/httpd/conf/ssl.crt” directory and server key is stored under “/etc/httpd/conf/ssl.key” directory).

NB: Generally, Apache for Win32 comes with a predefined Apache/mod_ssl configuration file. So, do not use following configuration if your system already has a configuration file for mod_ssl !

```
#
```

```
# This is the Apache server configuration file providing SSL support.
# It contains the configuration directives to instruct the server how to
# serve pages over an https connection. For detailing information about these
# directives see <URL:http://httpd.apache.org/docs-2.0/mod/mod_ssl.html>
#
# For the moment, see <URL:http://www.modssl.org/docs/> for this info.
# The documents are still being prepared from material donated by the
# modssl project.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#

LoadModule ssl_module modules/mod_ssl.so

# Until documentation is completed, please check http://www.modssl.org/
# for additional config examples and module docmentation. Directives
# and features of mod_ssl are largely unchanged from the mod_ssl project
# for Apache 1.3.

#
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
#
# To allow connections to IPv6 addresses add "Listen [::]:443"
#
Listen 0.0.0.0:443

#
# Some MIME-types for downloading Certificates and CRLs
```

```
#
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl

# Pass Phrase Dialog:
# Configure the pass phrase gathering process.
# The filtering dialog program (`builtin' is a internal
# terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog builtin

##
## SSL Virtual Host Context
##

<VirtualHost _default_:443>

# Use separate log files:
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# SSL Cipher Suite:
# List the ciphers that the client is permitted to negotiate.
# See the mod_ssl documentation for a complete list.
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP

# Server Certificate:
# Point SSLCertificateFile at a PEM encoded certificate. If
# the certificate is encrypted, then you will be prompted for a
# pass phrase. Note that a kill -HUP will prompt again. A test
# certificate can be generated with `make certificate' under
# built time. Keep in mind that if you've both a RSA and a DSA
# certificate you can configure both in parallel (to also allow
# the use of DSA ciphers, etc.)
SSLCertificateFile /etc/httpd/conf/ssl.crt/server.crt

# Server Private Key:
# If the key is not combined with the certificate, use this
# directive to point at the key file. Keep in mind that if
# you've both a RSA and a DSA private key you can configure
# both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile /etc/httpd/conf/ssl.key/server.key

# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)
# Note: Inside SSLCACertificatePath you need hash symlinks
# to point to the certificate files. Use the provided
# Makefile to update the hash symlinks after changes.
#SSLCACertificatePath /etc/httpd/conf/ssl.crt
SSLCACertificateFile /usr/share/ssl/certs/ca_root.crt

# SSL Engine Options:
# StdEnvVars:
# This exports the standard SSL/TLS related `SSL_*' environment variables.
# Per default this exportation is switched off for performance reasons,
# because the extraction step is an expensive operation and is usually
# useless for serving static content. So one usually enables the
# exportation for CGI and SSI requests only.
SSLOptions +StdEnvVars
```

```
# SSL Protocol Adjustments:
# The safe and default but still SSL/TLS standard compliant shutdown
# approach is that mod_ssl sends the close notify alert but doesn't wait for
# the close notify alert from client. When you need a different shutdown
# approach you can use one of the following variables:
# o ssl-unclean-shutdown:
# This forces an unclean shutdown when the connection is closed, i.e. no
# SSL close notify alert is send or allowed to received. This violates
# the SSL/TLS standard but is needed for some brain-dead browsers. Use
# this when you receive I/O errors because of the standard approach where
# mod_ssl sends the close notify alert.
# o ssl-accurate-shutdown:
# This forces an accurate shutdown when the connection is closed, i.e. a
# SSL close notify alert is send and mod_ssl waits for the close notify
# alert of the client. This is 100% SSL/TLS standard compliant, but in
# practice often causes hanging connections with brain-dead browsers. Use
# this only for browsers where you know that their SSL implementation
# works correctly.
# Notice: Most problems of broken clients are also related to the HTTP
# keep-alive facility, so you usually additionally want to disable
# keep-alive for those clients, too. Use variable "nokeepalive" for this.
# Similarly, one has to force some clients to use HTTP/1.0 to workaround
# their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
# "force-response-1.0" for this.
SetEnvIf User-Agent ".*MSIE.*" \
nokeepalive ssl-unclean-shutdown \
downgrade-1.0 force-response-1.0

# Per-Server Logging:
```

```
# The home of a custom SSL log file. Use this when you want a

# compact non-error SSL logfile on a virtual host basis.

CustomLog logs/ssl_request_log \

"%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"


</VirtualHost>
```

Figure 12: Sample Apache/mod_ssl configuration file

Once you’ve configured your Apache web server, don’t forget to restart Apache daemon for changes to take effect.

Last, you have to install Certificate Authority root certificate file “ca_root.crt” on each client computer into OCS Inventory Agent installation directory, under the name “cacert.pem”.

With OCS Inventory NG Server for Windows.

We provide below a sample script using OpenSSL for generating a certificate request for use in XAMPP Apache.

```
@echo off

REM

REM Generate server certificate request

REM

REM Generate 1024 bits RSA key, store private key in a

REM no password protected PEM file server.key, store certificate

REM request in a PEM file server.csr, using system default

REM configuration file

REM

REM The produced key will be valid for 1825 days (5 years)

REM

echo.

echo Generating server private key and certificate request...

echo.

set OPENSSL_CONF=./bin/openssl.cnf

bin\openssl req -newkey rsa:1024 -outform PEM -out server.csr -keyout server.key -keyform PEM -days 1825 -nodes
```

Figure 13: Sample apache_request_cert.sh script

This script generates a RSA private key in file “server.key” and a certificate request in file “server.csr”.

First, copy this script into “INSTALL_PATH\xampp\apache” directory (where “INSTALL_PATH is the installation folder of OCS Inventory NG) and launch it.

It will generate private key, and prompt you for certificate request properties:

- Country code, usually required
- State or province name, usually required
- City, usually required
- Organisation or company name, usually required
- Organisational Unit name, usually optional
- Common name (this is the DNS name or IP address of your server), required
- An email address, required to receive certificate generated by Certificate Authority.
- An optional challenge password
- An optional company name

[[Image:]]

In our sample, we’ve generated certificate request for our server name “ocs.domain.tld”.

Next, you must transmit your certificate request “server.csr” to your PKI Certificate Authority.

Once you’ve received your server certificate from Certificate Authority, you just have to copy server certificate file “server.crt” to directory “INSTALL_PATH\xampp\apache\conf\ssl.crt” and server private key “server.key” files into to directory “INSTALL_PATH\xampp\apache\conf\ssl.key”.

You must also retreive Certificate Authority root certificate into file “ca_root.crt” to specify it in Apache configuration. Store this file into directory “INSTALL_PATH\xampp\apache\conf\ssl.crt” under name “ca-bundle.crt”.

Update Apache/mod_ssl configuration by editing file ““INSTALL_PATH\xampp\apache\conf\extra\httpd-ssl.conf”, and uncommenting line 132 (remove # character at beginning) as follow.

```
# Certificate Authority (CA):

# Set the CA certificate verification path where to find CA

# certificates for client authentication or alternatively one

# huge file containing all of them (file must be PEM encoded)

# Note: Inside SSLCACertificatePath you need hash symlinks

# to point to the certificate files. Use the provided

# Makefile to update the hash symlinks after changes.

#SSLCACertificatePath conf/ssl.crt
```

SSLCACertificateFile conf/ssl.crt/ca-bundle.crt

Figure 14: Sample Apache/mod_ssl configuration file

Once you’ve configured your Apache web server, don’t forget to restart Apache2 service for changes to take effect.

Last, you have to install Certificate Authority root certificate file “ca_root.crt” on each client computer into OCS Inventory Agent installation directory, under the name “cacert.pem”.

Example: Deploying new version of Service agent for Windows.

Create a ZIP “OcsAgentSetup.zip” including file “OcsAgentSetup.exe”.

Next, connect to Administration console and go to menu “Deployment / Build”.

- **Fill in package name**, for example “Ocs Agent Service 4031”,
- **select target operating system** “Windows”,
- **select protocol** “HTTP”,
- **select priority** “5”,
- **browse to select ZIP file**,
- **select action** “Launch”
- **and fill in file name** with Service Agent setup command line switches, for example “OcsAgentSetup.exe /S /NOSPASH /UPGRADE /NP /DEBUG /SERVER:my_ocs_server.domain.tld” (/S to run installer in silent mode, /NOSPASH to disable installer spash screen, /UPGRADE to indicate that you’re upgrading an already installed Service Agent, /NP to disable use of IE proxy settings, /DEBUG to enable creation of log files, /SERVER to indicate that agent must connect to server at address “my_ocs_server.domain.tld”).

***NB**: Don’t forget /UPGRADE command line switch to allow upgrading an existing OCS Inventory NG agent installed as a service.*

[[Image:]]

And validate by clicking “Send” button..

Next choose fragment size by moving slider, for example 500 Kb and click “Send” button.

[[Image:]]

Now, deployement package is created. You have to activate it.

Go to menu “Deployment / Activate”.

[[Image:]]

Click on button “Activate” in the corresponding line.

- Fill in HTTPS url where metadata file INFO can be downloaded by agents using HTTPS.
- Fill in HTTP url where fragment files can be downloaded by agent using HTTP.

And click “Send” button.

[[Image:]]

Now, package is ready to be affected to computers.

Go to “Search” menu, search for computers having Operating System eaquels to “Windows (ALL)” and click “Search” button.

[[Image:]]

Next, click “Deploy” on “Mass processing” line.

[[Image:]]

To fiinish, click on “Affect” button in the corresponding line to the package you want.

[[Image:]]

That’s all folks !

Management server tuning.

OCS Inventory NG management server needs some tuning to support the load of a large number of inventoried computers. Performances are just limited by the hardware configuration (especially the amount of RAM, processor is not very loaded) of computer hosting the 3 main components:

- MySQL database server.
- Communication server.
- Administration server.
- Deployment server.

For example, our production server manages more than 70 000 clients. For this, we have 3 servers running Linux Debian Sarge, one for the database server and the Communication server, another one for Administration console and a replica of database server (we choose to replicate database on Administration server to avoid Administration console SQL queries using CPU and MySQL connexions of database used by Communication server) and the last one for deployement server. Hardware configuration for servers is the following:

- 1 Intel Pentium Xeon 2,8 GHz.
- 3 GB RAM

Because of the amount of available RAM, we have to limit the number of simultaneous HTTP connection to Communication and Administration server to 400.

You must keep an eye on Apache web server logs for Communication server to detect any problems. Also, check Communication server log file in directory “/var/log/ocsinventory-NG”.

If you want to upgrade the number of simultaneous connections, you must update the “MaxClients” directive in Apache configuration file, usually “/etc/httpd/conf/httpd.conf”.

Refer to Apache tuning guidelines on Apavhe web site (http://httpd.apache.org) for more information.

Also, MySQL database server is limited by default to 100 simultaneous connections. So, if the Communication server handles more than 100 simultaneous requests for inventory, it will not be able to answer all. You can upgrade this value by updating the “max_connections” MySQL variable for mysqld daemon.

Here is sample recommdations found on MySQL web site, using server with different amount of physical memory.

Parameter	800 MB	1.7 GB	2.4 GB
Table_cache	64M	64M	64M
Key_buffer	128M	256M	256M
Sort_buffer_size	2M	2M	2M
Read_buffer_size	2M	2M	2M
Read_rnd_buffer_size	4M	4M	4M
Myisam_sort_buffer_size	64M	64M	64M
Query_cache_size	128M	128M	128M
InnoDB_buffer_pool_size	384M	1024M	1700M
InnoDB_additional_mem_pool_size	20M	20M	20M
InnoDB_log_buffer_size	8M	8M	8M

Refer to MySQL tuning guidelines on MySQL web site (http://dev.mysql.com/tech-resources/articles/) for more information.

Backup/restore of OCS Inventory NG database.

You can use tools like phpMyAdmin or MySQL Administrator to backup/restore MySQL databases.

However, MySQL Server provide standard tools usable in command line.

Backuping OCS Inventory NG database.

Tool “mysqldump” allows you to dump all content of a database to a single file. These tool is available in directory “INSTALL_PATH\xampp\mysql\bin” where INSTALL_PATH is the installation oath you choosen for OCS Inventory NG Server for Windows, and generally can be directly called under Linux.

Simply run the following command to backup OCS Inventory NG database:

- Mysqldump --add-drop-table --complete-insert --extended-insert --quote-names --host=localhost --user=”root” --password=”root password” ocsweb > mysqldump_ocsweb.sql

This will save database content of OCS Inventory NG database “ocsweb” into file “mysqldump_ocsweb.sql”.

Restoring OCS Inventory NG database.

Tool “mysql” allows you to launch SQL queries on a database. These tool is available in directory “INSTALL_PATH\xampp\mysql\bin” where INSTALL_PATH is the installation oath you choosen for OCS Inventory NG Server for Windows, and generally can be directly called under Linux.

Just run MySQL command line interpreter to import saveset previously created “ocsweb” database.

- Mysql –u root –p ocsweb
- Source “path_to_saveset”
- exit

NB: You will be prompted from root password. If you haven’t yet set root password, do use use –“p” command line switch.

[[Image:]]

[[Image:]]

Common errors.

Check FAQ on OCS Inventory web site for updates.

Troubleshouting agent’s execution.

Windows launcher OcsLogon.exe does not download Agent.

When I launch launcher OcsLogon.exe, agent installation files are not downloaded.

Launcher "OcsLogon.exe" must be renamed with communication server IP address or DNS name (ex : 192.168.1.12.exe or ocs_com.domain.tld.exe). If this is already done, you may have configured a proxy in Internet Explorer, and OCS is using this configuration. Try to disable use of proxy by lauching OcsLogon with "/NP" command line switch. In any case, launch OcsLogon with "/DEBUG" command line switch and take a look at log file "C:\ocs-ng\ocslogon.log" and "C:\ocs-ng\computer_name.log".

Here is a typical OcsLogon.log content of faulting launcher:

```
OCS server port number : Default (80)

Install folder : C:\Ocs-ng

OCSserver is set to: a.b.c.d

Internal Ocslogon version: 4.0.1.4


Testing: C:\ocs-ng\BIOSINFO.EXE

Ocs Inventory NG () was not previously installed.

Start deploying OCS


http://a.b.c.d/ocsinventory/deploy/ocsagent.exe : HTTP/1.1 500 Internal Server Error

http://a.b.c.d/ocsinventory/deploy/label : HTTP/1.1 500 Internal Server Error

End Deploying
```

```
Testing ocsagent.exe version:0000

Proxy use.

Launching : C:\ocs-ng\OCSInventory.exe /debug /server:a.b.c.d

Cmdline option is :\\server_share\a.b.c.d.exe /debug
```

As you can see, launcher is using proxy settings from IE, and there is "HTTP/1.1 500 Internal Server Error" error when downloading file "ocsagent.exe". This means that launcher is not able to download agent installation file. So try disabling use of proxy by adding « /NP » to agent’s command line launch, and then take a look at § 11.1.4 Agent HTTP errors., and then at § 11.3 Communication server errors.

NB: same error for file "label" is not blocking one. This means you aren't using TAG.

Windows agent does not send inventory to server.

I have set OCS Inventory NG server up as per the guide, but when I launch agent, nothing appears in Administration console.

On Windows client computer, launch "INSTALL_FOLDER\ocsinventory.exe /server:communication_server_ip /debug". A log file "computer_name.log" is created in directory "C:\ocs-ng", which will help you finding problem. Generally, you will see something like: "...is not a well configured ocs server" and an http error (see FAQ Windows agent HTTP errors).

Here is a typical "computer_name.log" content of faulting agent:

```
OCS INVENTORY ver. 4014 Starting session for Device <COMPUTER_NAME>

on Friday, February 24, 2006 15:34:27...

Command line parameters: </np /debug /server:a.b.c.d>

WMI Connect: Trying to connect to WMI namespace root\cimv2 on device <Localhost>...OK.

Registry Connect: Trying to connect to HKEY_LOCAL_MACHINE on device <Localhost>...OK.

SetupAPI Connect: Trying to connect to SetupAPI on device <Localhost>...OK.

CHECKINGS: No ocsinventory.dat file found !

IpHlpAPI GetNetworkAdapters...

IpHlpAPI GetNetworkAdapters: Calling GetIfTable to determine network adapter properties...OK

IpHlpAPI GetNetworkAdapters: Calling GetAdapterInfo to determine IP Infos...OK

IpHlpAPI GetNetworkAdapters: OK (1 objects).

DID_CHECK: Mac changed new:<00:40:63:D8:BC:61> old:<>, hname changed new:<COMPUTER_NAME> old:<>

Generating Unique ID for device <COMPUTER_NAME>...OK (COMPUTER_NAME-2006-02-24-15-34-27)
```

```
CHECKINGS: write <COMPUTER_NAME-2006-02-24-15-34-27> and <00:40:63:D8:BC:61>
in ocsinventory.dat
HTTP SERVER: Connection WITHOUT proxy
HTTP SERVER: Creating CInternetSession to get inventory parameters...OK.
HTTP SERVER: Connecting to server a.b.c.d 80...OK.
HTTP SERVER: Sending prolog query...
HTTP SERVER: The server <a.b.c.d> is not a well configured OCS server
HTTP ERROR:
<...>

<h1>Server error!</h1>
<p>The server encountered an internal error and was unable to complete your request.
Either the server is overloaded or there was an error in a CGI script.
</p>

<p>If you think this is a server error, please contact the <a href="mailto:admin@localhost">webmaster</a>.
</p>

<h2>Error 500</h2>
<address>
<a href="/">a.b.c.d</a><br />
<span>24.02.2006 15:40:10<br />
Apache/2.2.0 (Win32) DAV/2 mod_ssl/2.2.0 OpenSSL/0.9.8a mod_autoindex_color PHP/5.1.1 mod_perl/2.0.2 Perl/v5.8.7
</span>
</address>

HTTP SERVER: Closing HTTP connection
WMI Disconnect: Disconnected from WMI namespace.
SetupAPI Disconnect: Disconnected from SetupAPI.
Execution duration: 00:00:00.
```

As you can see, agent is not using IE proxy settings ("HTTP SERVER: Connection WITHOUT proxy "), and there is error "HTTP SERVER: The server is not a well configured OCS server" followed by "Error 500". So take a look at § 11.1.4 Agent HTTP errors., and then at § 11.3

Communication server errors.

Linux agent does not send inventory to server.

I have set OCS Inventory NG server up as per the guide, but when i launch Linux agent, nothing appears in Administration console.

On Linux client computer, you can use "ocsinv –debug" or "ocsinv –info" to obtain a trace.

Generally, you will see something like: « ...is not a well configured ocs server » and an http error. So take a look at § 11.1.4 Agent HTTP errors., and then at § 11.3 Communication server errors.

Agent HTTP errors.

I see in agent logs http errors. What do they mean ?

- **500**: server encountered an internal error. You must take a look at apache log files, especially file "error.log", generally located under "/var/log/httpd".
- **404**: URL "/ocsinventory" cannot be found on server. You have made a mistake in Apache configuration. Do you have included in apache configuration content of apache_config file ? Do you have updated this content to match your need ?
- **301**: This error means that you already have a directory named "/ocsinventory" in your apache server, and this name conflicts with apache <location> directive of apache_config file. You must change name of "/ocsinventory" directory.

In all case, you must also take a look at Communication Server log files and check § 11.3 Communication server errors.

Administration console errors.

MySQL Max_allowed_packet error.

If you encounter an error message with “max_allowed_packet” MySQL error, you must update your MySQL configuration to increase the maximum size of packet accepted by MySQL. We recommend setting the value to 4 MB.

[[Image:]]

Open the file “my.cnf” (usually available in “/etc” directory under Linux, in “C:\OCSinventoryNG\xampp\mysql\bin” under Windows) and add the line “max_allowed_packet=4M” in “[mysqld]”, “[mysql.server]” or “[safe_mysqld]” section.

[mysqld]

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

max_allowed_packet=4M

[mysql.server]

user=mysql

basedir=/var/lib

```
max_allowed_packet=4M

[safe_mysqld]

err-log=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid

max_allowed_packet=4M
```

Figure 15 : Sample my.cnf MySQL configuration file.

Then, restart MySQL server.

- /etc/rc.d/init.d/mysql restart

MySQL Client does not support authentication protocol.

If you encounter an error message with “Client does not support authentication protocol requested by server; consider upgrading MySQL client” MySQL error, you must enable support for old password storage method in your MySQL configuration.

[[Image:]]

There is 2 way to do this.

1. Add directive “old-passwords” to the file “my.cnf” (usually in directory “/etc” under Linux and in “C:\OCSinventoryNG\xampp\mysql\bin” under Windows), in the section corresponding to your MySQL server.

```
# The MySQL server

[mysqld]

old-passwords

port = 3306

socket = mysql
```

Figure 16 : Sample my.cnf MySQL configuration file.

1. Add switch “--old-password” to the command line launching MySQL server.

Then, restart MySQL server.

- /etc/rc.d/init.d/mysql restart

Next, you may have to update ‘root’ password with the following commands:

- Connect to MySQL database “mysql –u root –p mysql” as root to update his password.

- Then, run the update statement “update user set password=OLD_PASSWORD(‘root_password’) where user=’root’;”
- Once terminated, exit mysql command interpreter by entering “exit” command.

```
[root@linux root]# mysql -u root -p mysql

Enter password:

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A


Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 19 to server version: 4.1.7-standard


Type 'help;' or '\h' for help. Type '\c' to clear the buffer.


mysql> update user set password=OLD_PASSWORD('admin123') where user='root';

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0


mysql> exit

Bye

[root@linux root]#
```

Figure 17 : Sample MySQL root password update.

PHP Requested content-length.

If you see in apache error log (file error_log or ssl_error.log) **message like following**:

```
[Mon Sep 05 18:30:03 2005] [error] [client XXX.XXX.XXX.XXX] Requested content-length

of 831148 is larger than the configured limit of 524288, referer: http://administration_server/ocsreports/?multi=8
```

That’s because Apache directive “LimitRequestBody” is used to limit size of HTTP requests.

To fix this, open Apache configuration file “httpd.conf”, usually in directory “/etc/httpd/conf” (under some distributions, Apache configuration for PHP may also resides in include directory, usually “/etc/httpd/conf.d”).

Find the directive “LimitRequestBody” and ensure that the size is at least 4 MB (4194304 bytes) and not the default 512 KB (524288 bytes).

```
#

# PHP is an HTML-embedded scripting language which attempts to make it
# easy for developers to write dynamically generated webpages.

#


LoadModule php4_module modules/libphp4.so


#

# Cause the PHP interpreter handle files with a .php extension.

#

<Files *.php>

SetOutputFilter PHP

SetInputFilter PHP

LimitRequestBody 4194304

</Files>
```

Figure 18 : Sample Apache configuration for PHP.

Update this value if needed and restart Apache daemon.

- /etc/rc.d/init.d/httpd restart

Uploads size for package deployment.

All the configuration settings for your installation are contained in the “php.ini” file or Apache configuration files. Sometimes these setting might be overridden by directives in apache .htaccess files or even with in the scripts themselves. However you cannot over ride the settings that effect file uploads with .htaccess directives in this way. So let's just concentrate on the ini file.

You can call the phpinfo() function to find the location of your “php.ini” file, it will also tell you the current values for the following settings that we need to modify:

- file_uploads
- upload_max_filesize
- max_input_time
- memory_limit
- max_execution_time
- post_max_size

The first one is fairly obvious if you set this off, uploading is disabled for your installation, so you will not be able to upload software for deployment. We will cover the rest of the configuration settings in detail below.

Remenber to restart Apache web server for changes to take effect.

upload_max_filesize and post_max_size.

Files to deploy are *POST*ed to the webserver in a format known as 'multipart/form-data'. The post_max_size sets the upper limit on the amount of data that a script can accept in this manner. Ideally this value should be larger than the value that you set for upload_max_filesize.

It's important to realize that upload_max_filesize is the sum of the sizes of all the files that you are uploading. post_max_size is the upload_max_filesize plus the sum of the lengths of all the other fields in the form plus any mime headers that the encoder might include. Since these fields are typically small you can often approximate the upload max size to the post max size.

If you want to deploy files up to 200 MB, you must set in “php.ini” file

- upload_max_filesize = 200M
- post_max_size = 201M

memory_limit.

When the PHP engine is handling an incoming POST, it needs to keep some of the incoming data in memory. This directive has any effect only if you have used the *--enable-memory-limit* option at configuration time. Setting too high a value can be very dangerous because if several uploads are being handled concurrently all available memory will be used up and other unrelated scripts that consume a lot of memory might effect the whole server as well.

So we recommend using the following value in “php.ini” file:

- memory_limit = 16M

max_execution_time and max_input_time

These settings define the maximum life time of the script and the time that the script should spend in accepting input. If several mega bytes of data are being transfered max_input_time should be reasonably high.

So we recommend disabling these limits using the following value in “php.ini” file:

- max_execution_time = -1
- max_input_time = -1

Additonal Comments

The apache webserver has a *LimitRequestBody* configuration directive that restricts the size of all POST data regardless of the web scripting language in use. Some RPM installations sets limit request body to 512Kb. You will need to change this to a larger value or remove the entry altogether.

If you want to deploy files up to 200 MB, you must comment in apache main configuration file.

- #LimitRequestBody

Communication server errors.

I see "Unknown directive PerlRequire...." in Apache log files.

This means that mod_perl for Apache is not installed, or loaded at startup by Apache. Install mod_perl and then update apache configuration file "httpd.conf" to load mod_perl.so extension.

You can check which version of mod_perl is installed by launching following command:

- On RPM based Linux: rpm -q mod_perl
- On Debian package based Linux: dpkg –l libapache*-mod-perl*

```
[root@linux conf.d]# rpm -q mod_perl
mod_perl-1.99_16-4.centos4
[root@linux conf.d]#
```

If mod_perl is installed, check Apache configuration files to ensure mod_perl is enabled. You may found something like this:

```
#
# Mod_perl incorporates a Perl interpreter into the Apache web server,
# so that the Apache web server can directly execute Perl code.
# Mod_perl links the Perl runtime library into the Apache web server
# and provides an object-oriented Perl interface for Apache's C
# language API. The end result is a quicker CGI script turnaround
# process, since no external Perl interpreter has to be started.
#

LoadModule perl_module modules/mod_perl.so
```

If needed, uncomment line (remove # at line begin) and restart Apache.

Please, refer to Apache manual for more details.

I see "Can't locate [Perl module name], cannot resolve handler Ocsinventory.pm..." in Apache log files.

Perl module [Perl module name] is not installed on your server. Please, refer to OCS Inventory NG guide or to Perl manual to install missing module.

If this error concerns Perl module "compat.pm" like below

```
[Thu Mar 02 11:43:56 2006] [error] [client client_ip] failed to resolve handler
`Ocsinventory': Can't locate Apache/compat.pm in @INC (@INC contains:...)
```

You probably have installed Communication server for use with Apache mod_perl version 1.999_21 or previous. Open OCS Inventory NG apache configuration file « ocsinventory.conf » and set variable « OCS_PERL_VERSION » to 2 enable use of mod_perl version 1.999_22 or higher.

```
# Which version of mod_perl we are using
# For mod_perl <= 1.999_21, replace VERSION_MP by 1
# For mod_perl > 1.999_21, replace VERSION_MP by 2
PerlSetEnv OCS_MODPERL_VERSION 2
```

Then restart Apache web server.

If this error concerns Perl module "Apache2/connection.pm" like below

```
[Sun Mar 05 12:46:09 2006] [error] [client client_ip] failed to resolve handle

r `Ocsinventory': Can't locate Apache2/Connection.pm in @INC (@INC contains:...)
```

You probably have installed Communication server for use with Apache mod_perl 1.999_22 and newer. Open OCS Inventory NG apache configuration file « ocsinventory.conf » and set variable « OCS_PERL_VERSION » to 1 to enable use of mod_perl version 1.999_21 or previous.

```
# Which version of mod_perl we are using

# For mod_perl <= 1.999_21, replace VERSION_MP by 1

# For mod_perl > 1.999_21, replace VERSION_MP by 2

PerlSetEnv OCS_MODPERL_VERSION 1
```

Then restart Apache web server.

I see "Cannot open log file: ..." in Apache logs. Communication server is not able to write his logs.

- Does directory « /var/log/ocsinventory-NG » (or any other path you've fill in at setup) exists ?
- Can Apache web server write in this directory ?

Files and directories permissions under Linux.

We assume that Apache web server is running under account "apache" and group "apache", and that you've set OCS Inventory NG management server up as described previously:

- Administration console files are in directory "/var/www/html/ocsreports"
- Deployment server files are in directory "/var/www/html/download"
- Log for Communication server are in directory "/var/log/ocsinventory-NG"

Directory	File	Owner	Group	Permissions
/var/www/html/download		root	apache	-rwxrwxr-x
	All directories	apache	apache	-rwxrwxr-x
	All files	apache	apache	-rw-rw-r--
/var/www/html/ocsreports		root	apache	-rwxrwxr-x
	dbconfig.inc.php	apache	apache	-rw-rw-r--
	All others	root	root	-rw-r--r--
/var/www/html/ocsreports/ipd		root	apache	-rwxrwxr-x
	All files	apache	apache	-rw-rw-r--
/var/www/html/ocsreports/css		root	root	-rwxr-xr-x
	All files	root	root	-rw-r--r--
/var/www/html/ocsreports/files		root	root	-rw-r--r--
	All files	root	root	-rw-r--r--
/var/www/html/ocsreports/image		root	root	-rwxr-xr-x
	All files	root	root	-rw-r--r--
/var/www/html/ocsreports/js		root	root	-rwxr-xr-x

	All files	root	root	-rw-r--r--
/var/www/html/ocsreports/languages		root	root	-rwxr-xr-x
	All files	root	root	-rw-r--r--
/var/log/ocsinventory-NG		root	apache	-rwxrwx-r-x
	All files	apache	apache	-rw-rw-r--

Getting help in forums.

If you are unable to diagnose yourself the problem, you can get help using OCS Inventory NG web site forums (http://ocsinventory.sourceforge.net/index.php?page=Forums).

If you do so, please provide us:

- Server operating system
- OCS Inventory NG server version and patch level
- Agent’s operating system
- OCS Inventory NG agent version
- Agent execution logs
    - Run “AGENT_INSTALL_FOLDER\ocsinventory.exe /NP /DEBUG /SERVER:you_server_address” under Windows. Log file is created on folder “AGENT_INSTALL_FOLDER” under name “your_computer_name.log”.
    - Run “ocsinv –debug > ocsinv.log” under Linux. Log file is “ocsinv.log”.
- Apache server error.log file, located for Windows under “SERVER_INSTALL_FOLDER\xampp\apache\logs\error.log” and for Linux under “/var/log/httpd/*error.log”.
- OCS Inventory NG Server log file, located for Windows under “SERVER_INSTALL_FOLDER\xampp\apache\logs\ocsinventory-NG.log” and for Linux under “/var/log/ocsinventory-NG/ocsinventory-NG.log” under Linux.

Thanks by advance.

Retrieved from "http://wiki.ocsinventory-ng.org/index.php?title=OCS_Inventory_NG:Documentation"

Views

- Project page
- Discussion
- View source
- History

Personal tools

- Log in / create account

Navigation

- Main Page
- Documentation
- Admin center
- Tutos
- Links
- Recent changes
- Random page
- Help
- Donations

Search

Go Search

Toolbox

- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- Print as PDF

- This page was last modified 08:02, 8 March 2008.
- This page has been accessed 4,200 times.

- Privacy policy
- About OCS Inventory NG
- Disclaimers
- SOURCEFORGE.NET®