HAAGA-HELIA
University of Applied Sciences

Open PC Server Integration 4.0.1

Cédric Zwimpfer

System documentation

BiTE

2011

HAAGA-HELIA
University of Applied Sciences

Abstract

Date of presentation

Degree programme

Author or authors Cédric Zwimpfer	**Group or year of entry** 2009
Title of thesis Open PC Server Integration	**Number of pages and appendices** 70+45
Supervisor or supervisors Markku Somerkivi	

Imagine a great tool built to make the installation and administration of multiple Windows clients a piece of cake, which is also open source and free of charge. This tool exists for Linux servers and is called Open PC and Server Integration – abbreviated OPSI.

The main purpose of this thesis is to describe the installation process and basic functions of OPSI. The core task was to create system documentation on OPSI, which describes several aspects from evaluation, planning, realisation, management and finally to testing. The secondary objective was to create a user documentation that will support the execution of most common OPSI functions.

The basic functions of OPSI are vividly depicted and described. To ensure the quality of the OPSI environment, a detailed test plan has been elaborated for this project. The component testing was done with two Windows clients. Additionally this thesis picks up subjects, which deal with OPSI server security, server configuration and monitoring.

Currently OPSI is widely used in Germany. Unfortunately it isn't as wide spread in the rest of the world. The goal is to point out the advantages of OPSI to make it attractive for Finnish companies and overall the globe.

Keywords
OPSI; Automated Software distribution, OS Installation, Inventory

Table of contents

1 Management Summary

1.1 Project definition

All Haaga-Helia UAS degree students write an academic Bachelor’s Thesis (15 ECTS). The topic has to be IT related and is evaluated by a group of experts and assessed on the scale 1 (satisfactory) – 5(excellent). My thesis is done as an individual effort and related to a research work.

1.2 Documentation structure

The documentation of my thesis is divided in following parts:

Document	Content
System documentation	Project Management, Evaluation, Planning, Preparation, Realisation, Monitoring, Summary
User documentation	Basic OPSI management operations
Work journal	Record daily workload
Test plan	Testing, execute different tests
Project plan	Project planning
Summary report	Sum up the thesis

1.3 Initial Situation

Usually middle sized to big companies are struggling with their client management. Often the client installation processes aren’t standardised and moreover the companies lack a software distribution system. Especially update a periodic survey of all goods and IT components in stock is a tedious job to do if it isn’t automated.

Three months ago I came across a powerful open source software called “Open PC Server Integration”. It caught my attention since “OPSI” provides many solutions to the client management problems mentioned in previous paragraph. Furthermore it combines features like installation, inventory, software distribution, backup and restore of clients, etc. Nevertheless it is free of charge. All these great features and opportunities made me curious to explore more of that interesting Linux based desktop/client management system called "OPSI".

Obviously the benefits of such an extraordinary system are immense. Not only will you save time, stress and prevent human errors with an automated client management system moreover it's a great way to save money for companies. The efficiency of administration and installation of clients is much higher whereas at the same time you have control over your IT inventory.

1.4 Objectives

The target of this project is to create a working OPSI environment, which will be documented and tested properly. OPSI should adopt the inventory at its final stage, installation and software distribution of the clients. The goal is to install an OPSI server, which is able to accomplish those tasks.

1.5 Scope / Workload

- Introduction (Doc structure, Scope, Project def., What is OPSI)
- Preparations (What to do before start)
- Planning (Methods, Expected Results, Starting-Point)
- Evaluation (Systems, Software, Hardware, Decision)
- Realisation (Installation, Configuration, Administration, Testing)
- Monitoring (Milestones, Reached goals)
- Summary (Result, Conclusion, Outlook, etc.)

1.6 Execution Overview

This is just a rough overview about the planned working steps. I have divided the project in 3 realisation stages:

Stage 1

- OPSI depot server installation
- OPSI configuration

Stage 2

- Windows installation
- Packages configuration

Stage 3

- Create test plan
- Testing
- Documentation

1.7 Expected Results

- Properly working OPSI environment
- System documentation, which explains installation and management
- User documentation, which explains the common tasks
- Test plan, which covers the functionality

1.8 Used Symbols

⚠	Attention
⛔	Error
ℹ	Information
>_	Console commands
↗	References

1.9 Abbreviations

API	Application Programming Interface
CPU	Central Processing Unit
DB	Database
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FQDN	Fully Qualified Domain Name

GB	Gigabyte
GHZ	Giga Hertz
GPL	General Public License
GUI	Graphical User Interface
HW	Hardware
IP	Internet Protocol
LAN	Local Area Network
MB	Megabyte
NIC	Network Interface Controller/Card
OPSI	Open PC Server Integration
OS	Operating System
PXE	Preboot Execution Environment
RAM	Random Access Memory
SSH	Secure Shell
SW	Software
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
URL	Uniform Resource Locator
VM	Virtual Machine
VPN	Virtual Private Network
WAN	Wide Area Network
wInst	Windows Installer
WLAN	Wireless Local Area Network
WMI	Windows Management Instrumentation

2 Introduction

2.1 What is OPSI?

OPSI (Open PC Server Integration) is a software distribution and management system for Windows Clients, based on Linux servers.
The key features of OSPI are:

- Automated operating system installation (OS deployment)
- Software distribution
- Patch management
- Inventory (hardware and software)
- License Management / Software Asset Management

OPSI supports following 32/64 –bit client operating systems:

- Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows Server 2008R2

The OPSI depotserver can be installed on following Linux distributions:

- Debian
- Ubuntu
- Univention Corporate Server
- OpenSuse
- SLES
- CentOS
- RedHat

2.2 How does OPSI work?

The OPSI depotserver is the backbone, which is providing basic services. The server administrates the configuration data of the clients and provides methods to access that data. The storage of that data is usually saved in a file or MySQL database. On each

Windows Client, which is administrated by OPSI, is the "OPSI Client Agent" installed. The depotserver provides software packages (depots), which are called by the clients (OPSI Client Agent) for the installation process. The client doesn't have to be in a Windows Domain to be part of the software distribution process. Additionally it is possible to integrate a DHCP server as well as a TFTP Server to allocate boot images for OS installation. The management of OPSI is done with help of a graphical user interface, which can also be accessed in a browser. Moreover it is possible to manage OPSI only via command line. In general you can describe the OPSI functions as followed:

1. **Automated OS- Installation**

   During the boot process a PXE- Image is loaded on the windows client, which starts a hardware inventory. All necessary installation files and drivers will be transmitted from the OPSI depot server to the windows client. Afterwards an unattended windows installation will be executed.

2. **Automated Software Distribution**

   The OPSI Client Agent is responsible for the automated software distribution on the clients. This service, which is installed on the client side, connects to the OPSI depotserver each system start and checks if there are any new packages available for the client. In case there are some packages ready to be installed, a small open source program called "Windows Installer" (wInst) will install those packages automatically. The graphical user interface (GUI) is locked to the user during the whole process.

3. **Hard- and Software Inventory**

   The OPSI inventory is based on the Windows Management Instrumentation (WMI). All client data is gathered by special software packages and saved in MySQL database. All gathered information is accessible through the OPSI administration interface.

OPSI.org, Features, 2011

2.3 OPSI Extensions

The core functions of OPSI are licensed under GPL. GPL stands for General Public License and is the most widely used free software license. Therefore all core functions of OPSI are free to use without any license. Usually this includes having the range of clients between 100-200. This corresponds to small or middle-sized companies. However, bigger companies with more than 1000 clients need more capabilities for maintenance. OPSI provides extensions for those companies, which are liable to pay costs.

ℹ	In my research I will not include any of those extensions. Following extensions are available:
↗	Detlef Oertel, Techcast, CeBit 2011

MySQL Backend

This is a database backend for higher performance and larger installations. Usually used for more than 500 clients.

Hierarchical grouping of clients and products

This describes the possibility to create groups of clients and products. Therefore it's easier to browse and sort a huge amount of clients later on.

Link-up of clients via WAN/VPN

This describes the possibility to integrate clients with slow network connections like home office.

Load balancing/system stability

This extension is meant to be for bigger installations to tune up the server with multi processors and extra hardware specifications. It guarantees a stable running OPSI server.

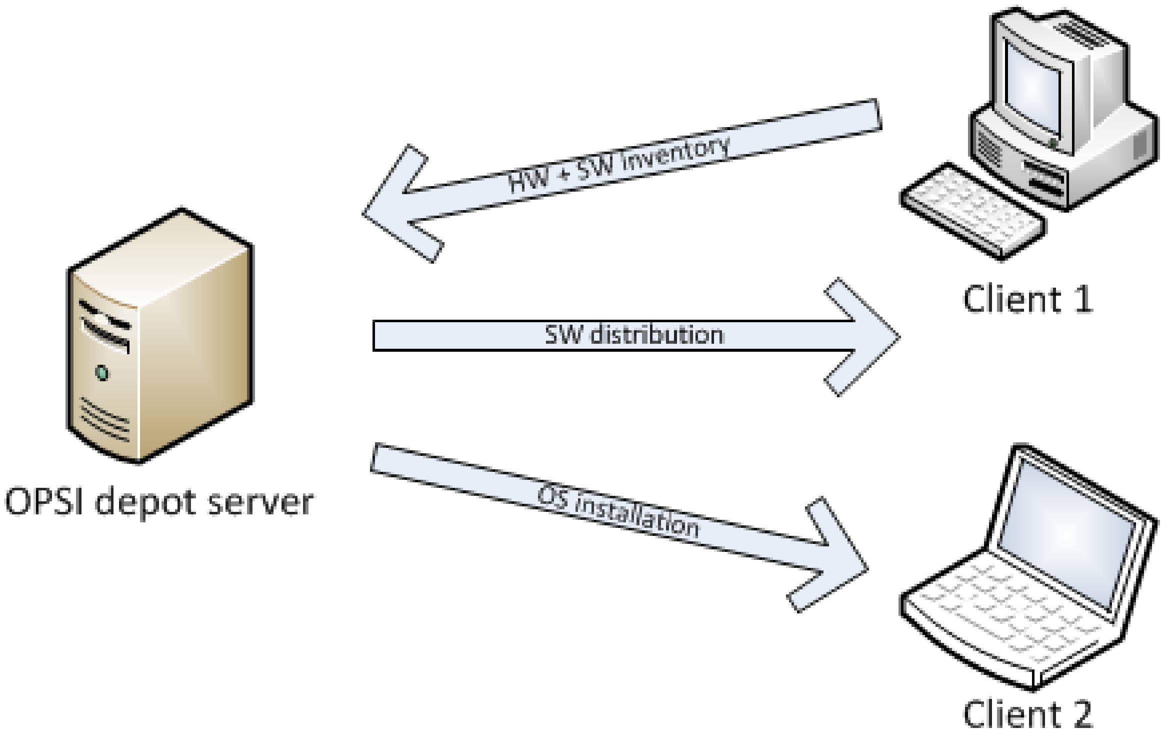
Dynamic depots/packages assignment

Dependant on the location of the client, it can recognise dynamically which depot server is the closest.

Software On Demand /Kiosk Mode

With this extension it is possible for administrator to mark software which can be installed by the user himself. Therefore the user can decide which marked eligible software he would like to install.

2.4 Picture



3 Evaluation

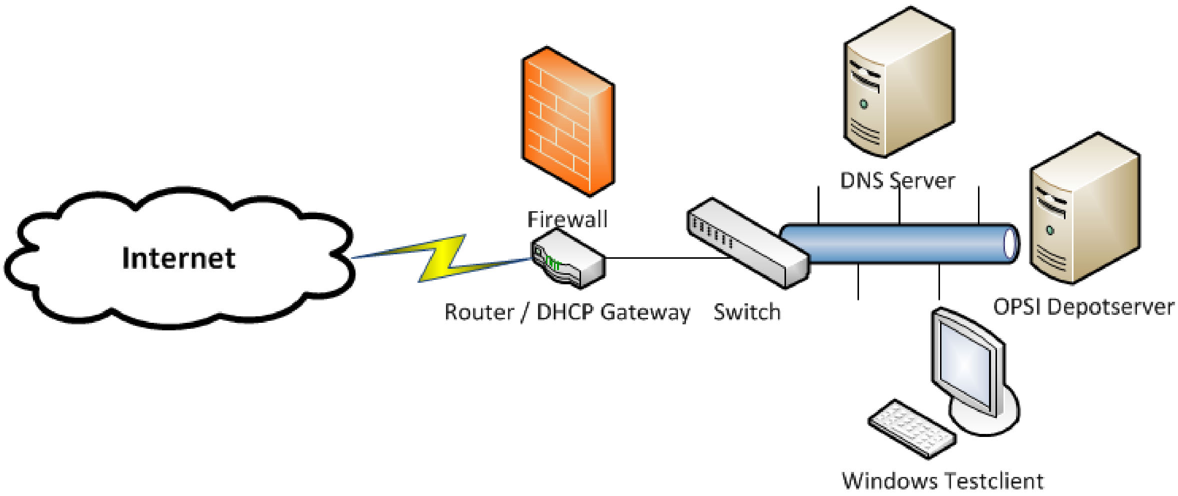
3.1 Preliminary work

- Setting up environment
- Installing Ubuntu Server
- Installing Windows 2003 Server
- Installing Windows Test Clients
- Creating the network
- Join Servers and Clients to network
- Save OPSI documentation on disk

3.2 Preliminary knowledge

- Basic Linux (Ubuntu) knowledge
- Basic Windows knowledge
- Network Services (DNS, DHCP)
- PXE Boot
- Unattended installation

3.3 Environment



3.4 Security

Firewall

My used Router has an integrated firewall. No additional ports are opened besides of the standard ones like 80, 25 and so on. The firewall is protecting the network from unauthorised access from the Internet.

Network

The network was built for the only purpose of connecting the necessary servers and clients. No additional machines are allowed to join the network. The home network is using 10/100 Mbps.

Updates

All machines, servers and clients, are up to date. They will receive all patches and updates automatically.

Antivirus

The windows client has free antivirus software called “AVG Free Antivirus 2011” installed. The virus definitions are up to date by all means.

3.5 Server platform installation options

To install the OPSI depotserver there are many options to do so. In this chapter I explain the different possibilities to host OPSI.

3.5.1 Option 1: OPSI depotserver installation as a VMWare machine

An OPSI depotserver can be installed as a virtual machine. The depotserver is predestined to run on a VM machine since it only requires little system resources and therefore the computer speed can be low.

This method is very convenient because there is already a corresponding VMware machine prepared on the official OPSI Internet site. In order to run this virtual machine a VM Player is needed. Fortunately there are several free of charge VM Players on the Net.

Besides the many advantages virtual machines have (low cost, low energy consumption, low administrative work…), a virtual OPSI depotserver is also dedicated to be mobile. With copy and paste you can carry your personal OPSI depotserver even on a memory stick in your pockets.

Pro	Contra
Quick and easy set up	No control over installation process
No installation	No knowledge over process
Automated	
Virtual	

3.5.2 Option 2: OPSI depotserver installation from the OPSI-Server-CD (Debian)

This method requires the possession of an OPSI CD, which can be acquired online from the official OPSI Internet site.

⚠	After inserting the CD, the system will be completely reinstalled. Therefore the hard disk will be formatted and the system will implement a basic Linux (Debian) operating system.

Afterwards the system will install and configure all required packets from the CD. Unfortunately due to the automated process, many additional and unnecessary programs, which aren’t related to OPSI, will also be installed. To keep away the ballast, which slows the system needlessly down, all extra software (Firefox, VNC, …) and GUI (Gnome) has to be removed manually.

Pro	Contra
Quick and easy installation	No control over installation process
All functions immediately available	Plenty of ballast
Automated	

3.5.3 Option 3: OPSI depotserver installation from the packet manager (Ubuntu, Debian)

With this method the entire installation has to be done manually. Obviously it is very time consuming to load and install every single packet via packet manager. Moreover the configuration hast to be adjusted to ensure OPSI works as intended. On the other hand this opens many possibilities to configure and control the system and has the advantage of customizing the system to your needs.

ℹ	The installation process is done solely in a command shell. Therefore it's crucial to have some Linux know-how since you need root rights.

The benefit of this method is obvious. No extra ballast will be installed, which would slow your system down as described in 3.4.2.

Pro	Contra
Full control over installation process	Time consuming
Slick installation	Easy to make mistakes during the process
No extra ballast	

3.5.4 Option 4: OPSI depotserver installation on a openSUSE system

OPSI also works on Linux openSUSE distribution. OPSI 4.0 is tested and released for openSUSE 11.3. There is no guarantee that OPSI will work on different openSUSE versions.

OPSI has been designed for Debian systems therefore the packet management seems to work more flawless on a Debian system. Due to the different distribution architecture the installation process is more complicated. However the functionality is like in paragraph 3.4.3.

Personally I don't like openSUSE and furthermore I am more familiar with Debian systems. Even the UIB GmbH (Distributor) recommends using Debian with OPSI.

Pro	Contra
Full control over installation process	Time consuming
No extra ballast	Easy to make mistakes during the process
	To me unknown distribution
	Better packet management

3.6 Windows platform installation options

There are basically two possibilities to install a windows client via OPSI depotserver.

3.6.1 Option 1: Installation Image

OPSI has a tool included, which enables the creation and distribution of images to clients. The creation of those images files is very fast and easy. Additionally it is possible to put those files on a data storage medium such as CD to create bootable image disk. The downside of this method is the inflexible driver integration. Since an image file has been created it cannot be changed anymore. When new patches for a client arise, it is necessary to create a new image file. Finally you would need a different image file for every a different client.

Pro	Contra
Fast and easy IMG file creation	Image file cannot be edited

3.6.2 Option 2: Unattended Installation

This Installation is done with an unattended file. An unattended file is editable which makes it easy to include patches and hotfixes afterwards. An unattended installation only copies those files to the destination client, which are required. Therefore you save bandwidth and unmatched driver problems. However, the creation of an unattended file is rather time consuming.

Pro	Contra
Fast and easy IMG file creation	Image file cannot be edited

3.7 Conclusion

3.7.1 OPSI depotserver installation

In my opinion the best solution is method 3.4.3 (OPSI depotserver installation from the packet manager (Ubuntu, Debian)). The reason for that are the high configurability, stability and my know-how. However due to personal reasons (travel to my home country) I need the OPSI depotserver to be mobile. Therefore I decide to realise the project in a mixture of method 3.4.3 and 3.4.1, which allows me to have the OPSI depotserver as a virtual machine. Nevertheless I will document the installation process of OPSI on an Ubuntu Lucid Lynx (10.04 LTS)

3.7.2 Windows installation

If you don't have more than 5 clients with different hardware configuration you are good to go with option 1, namely an installation image. Unfortunately you have to create for each hardware constellation a different boot image DVD. Therefore it is obvious that option 2 (unattended installation) is more advisable if you have a lot of clients with different hardware setup. Both options have their advantages and disadvantages. I decided to include the unattended installation in my project since in my opinion it is the more slick way.

4 Planning

4.1 Initial Point

The main purpose of this writing is to examine OPSI and its process from installation to administration. Since the installation of OPSI is one of the main parts in my thesis, my starting point is set exactly there. I will not consider the pre-work like the installation of Ubuntu nor Windows 2003 Server in my documentation although it is calculated in the time schedule. However I will emphasise on the subject OPSI and its features. The initial point is set to a working environment. This includes both necessary servers (Ubuntu and Windows 2003) as well as the test clients (Windows clients). The initial point of the Ubuntu server is a clean installation with all necessary updates. The initial point of the Windows 2003 server is a clean installation with all necessary updates and efficient DNS service as domain controller and likewise the windows client just without DNS service running. All machines are on the same network and properly connected and working. Thereby the goal is now to install OPSI and its features on the Ubuntu server and test the functions thoroughly and moreover document the process.

4.2 Study Methods

During my research I am going to invest quite a lot of time reading the official "OPSI manual" and the "OPSI getting started" guide. The background idea is to get more theory experience of OPSI. Especially the official OPSI manual provides detailed knowledge to master OPSI and all its features whereas the "OPSI getting started" guide deals with the first steps of OPSI and tackles mainly the installation part. Combined, the manual and the guide are excellent reference materials. In addition to this literature-based research, other approaches to get more reliable information are used, including Google, video material and practical work. Unfortunately I haven't found any good books about OPSI and its features.

4.3 Planned Hardware / OS

The hardware parts used during my thesis are solely private owned. The whole equipment is gathered and built together by myself. Therefore the used hardware is not up to date anymore since I didn't want to put extra money in expensive new hardware.

4.3.1 Server Configuration

Ubuntu OPSI depot server

CPU:	1.2GHZ	
RAM:	512MB	
Harddisk	60GB	
Hostname	depotserver	
Network	IP: 192.168.1.80 Subnet Mask: 255.255.255.0 Gateway: 192.168.1.1 DNS: 192.168.1.70 Domain: DOMAIN.OPSI	
OS	Debian Ubuntu 10.04 Lucid Lynx	
OPSI- Version	4.0.1	

Windows 2003 Server Domain Controller / DNS server

CPU:	AMD Sempron 1GHZ	
RAM:	512MB Ram	
Harddisk	15GB	
Hostname	win2003server	
Network	IP: 192.168.1.70 Subnet Mask: 255.255.255.0 Gateway: 192.168.1.1 DNS: 192.168.1.70 Domain: DOMAIN.OPSI	
OS	Windows Server 2003 Standard Edition SP2	

4.3.2 Client Configuration

Windows XP test client 1

CPU:	1 GHZ
RAM:	512MB Ram
Harddisk	20GB
Hostname	opsiclient
Network	IP: 192.168.1.90 Subnet Mask: 255.255.255.0 Gateway: 192.168.1.1 DNS: 192.168.1.70 Domain: DOMAIN.OPSI
OS	Windows XP Professional SP1

Windows XP test client 2

CPU:	1 GHZ
RAM:	512MB Ram
Harddisk	20GB
Hostname	opsiclient2
Network	IP: 192.168.1.91 Subnet Mask: 255.255.255.0 Gateway: 192.168.1.1 DNS: 192.168.1.70 Domain: DOMAIN.OPSI
OS	Windows XP Professional SP1

4.3.3 Network Configuration

OPSI Hostname	depotserver.domain.opsi
IP Address	192.168.1.80
DNS- Server	win2003server.domain.opsi
Location	Helsinki, Finland

4.4 Planned User Rights

Name	Login name	Password
root	root	hydra1
MySQL-Admin	root@mysql	python1
Pcpatch-User	pcpatch	medusa1
OPSI- Database User	adminuser	sulfid1
Windows Admin User	administrator	hydra1

4.5 Detailed Roadmap

This is the planned and more detailed project activity of the realisation part. The working steps are planned as followed:

1. Server installations (Ubunut, Windows)
2. Services installation
3. Test the configuration (network, settings, …)
4. OPSI installation 4.0.1
5. OPSI installation documentation
6. Backup the OPSI server
7. Testing (software packages, inventory and automated OS installation)
8. Testing documentation
9. Create user manual

It is possible to divide those steps in three phases with milestones as mentioned in the thesis project plan. This table is more detailed than in the thesis project plan.

Phase 1: Pre-work

Working step:	- OS installation - Server configuration - Test the environment configuration
Date	1.8.2011
Milestone	Properly working configuration and environment.

Phase 2: OPSI installation

Working step:	- Prepare OPSI installation - Install OPSI 4.0.1 - Configure OPSI services - Test the OPSI installation - Document the OPSI installation
Date	15.9.2011
Milestone	The OPSI services are working properly and the documentation is finished

Phase 3: Testing

Working step:	- Create a test plan - Test the automated OS installation - Test the software packages distribution - Test the OPSI inventory - Create the user manual
Date	15.10.2011
Milestone	The user manual and testing is finished

This leaves 15 days of finalizing my thesis project. In these 15 last days I will cover the monitoring and summary part.

4.6 Detailed Execution Overview

Task	**Description**
OPSI depot server installation	Installation and configuration of the OPSI server and all needed packages: - OS Installation (Ubunut) - Gnome GUI - Static IP address - DNS Name = depotserver.domain.opsi - Remote administration via Putty or Terminal - Samba service - TFTP service - MySQL (Service installation, create OPSI user, inventory DB) - OPSI configuration - Server configuration - Installation documentation
Windows test client preparation	Test Client 1: - OS installation (Windows XP) - OS update - Antivirus installation - Join to domain - Integrate in OPSI - HW/SW Inventory - Testing - Documentation Test Client 2: - PXE ready boot - Automated OS installation via OPSI (Windows XP) - Antivirus installation - Automated SW Package installation via OPSI (7Zip) - Testing

	- Documentation
Testing	Following functions of OPSI will be tested on the test clients - Inventory - Administration - Automated OS Installation - Automated Software Installation

5 Realisation

The realisation is based on the official OPSI documentation

uib gmbh, OPSI Getting Started, 2011

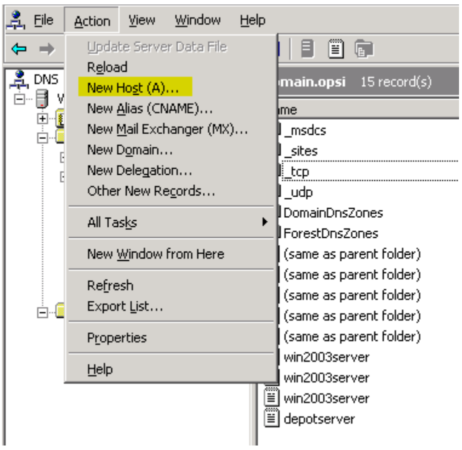
5.1 DNS Settings

The following settings have to be executed from a windows client or on the Windows 2003 itself. Possible required software:

- Putty
- Windows Server administration program (AMINPAK.MSI)
- Admin rights in the domain

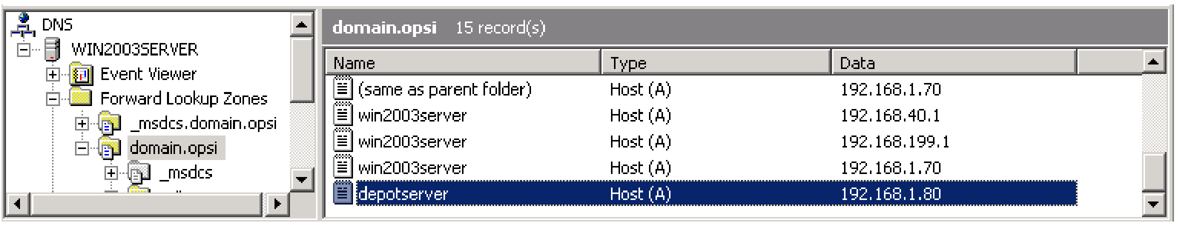
To reach the OPSI depotserver via its hostname it is necessary to make following entry on the DNS-Server (Win2003Server).

Screenshot: New Host



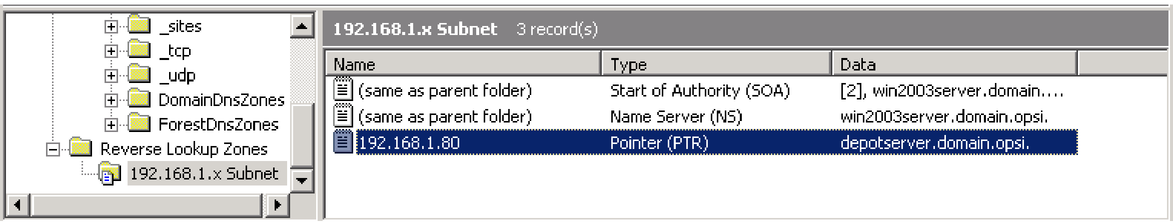
"New Host (A)" will add two new entries in the DNS; one entry in the forward lookup zone and the other one in the reverse lookup zone.

Screenshot: DNS Entry for Forward-Lookup Zone



Name	Type	Data
(same as parent folder)	Host (A)	192.168.1.70
win2003server	Host (A)	192.168.40.1
win2003server	Host (A)	192.168.199.1
win2003server	Host (A)	192.168.1.70
depotserver	Host (A)	192.168.1.80

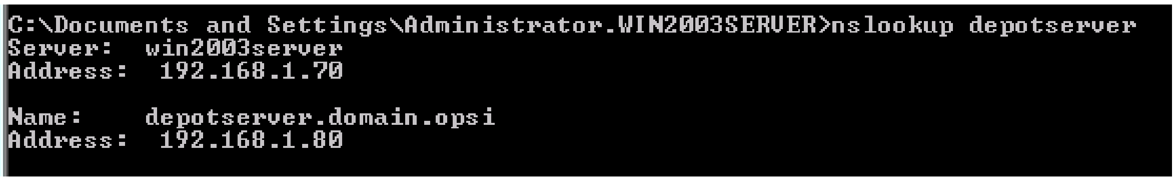
Screenshot: DNS Entry for Reverse-Lookup Zone



Name	Type	Data
(same as parent folder)	Start of Authority (SOA)	[2], win2003server.domain....
(same as parent folder)	Name Server (NS)	win2003server.domain.opsi.
192.168.1.80	Pointer (PTR)	depotserver.domain.opsi.

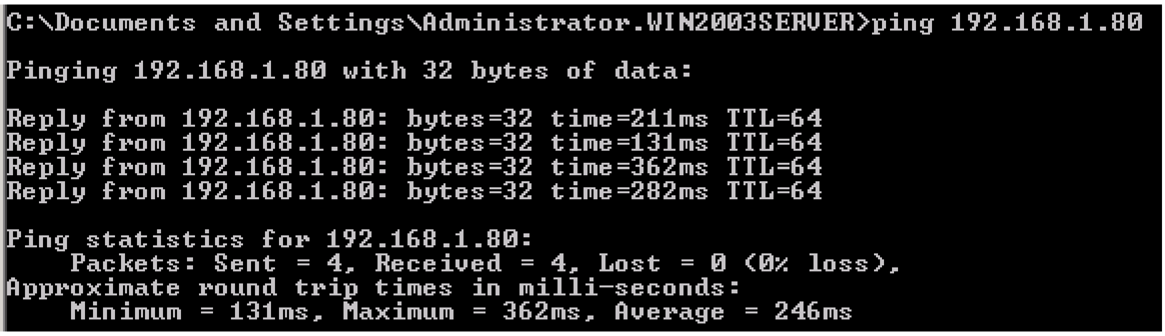
To test the DNS entries it is advisable to PING the OPSI depotserver and execute a NSLOOKUP.

Screenshot: Connection test with NSLOOKUP



```
C:\Documents and Settings\Administrator.WIN2003SERVER>nslookup depotserver
Server:  win2003server
Address:  192.168.1.70

Name:    depotserver.domain.opsi
Address:  192.168.1.80
```

Screenshot: Connection test with PING



```
C:\Documents and Settings\Administrator.WIN2003SERVER>ping 192.168.1.80

Pinging 192.168.1.80 with 32 bytes of data:

Reply from 192.168.1.80: bytes=32 time=211ms TTL=64
Reply from 192.168.1.80: bytes=32 time=131ms TTL=64
Reply from 192.168.1.80: bytes=32 time=362ms TTL=64
Reply from 192.168.1.80: bytes=32 time=282ms TTL=64

Ping statistics for 192.168.1.80:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 131ms, Maximum = 362ms, Average = 246ms
```

NSLLOKUP shows the DNS hostname resolution from hostname to IP and vice versa. PING shows that the Server is at the moment reachable.
Since everything looks perfect, the OPSI configuration can start.

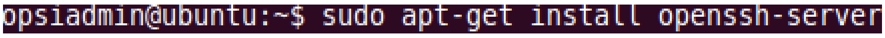
5.2 Server Configuration

Before the OPSI installation there are some settings adjustments which have to be done on the OPSI depotserver.

5.2.1 SSH Installation

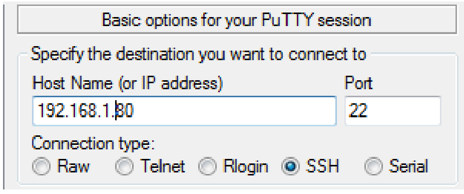
Since it is more practical and comfortable to configure the OPSI depotserver remotely it is advisable to install SSH. The combination of SSH and Putty lets you control, operate and administrate the server remotely. Additionally another advantage is bypassing the Ubuntu GUI which makes the work just slower. Therefore I decided to set up SSH and work remotely.

Start SSH installation:

>_	 `opsiadmin@ubuntu:~$ sudo apt-get install openssh-server`
↗	NIXCRAFT, nixCraft Blog, 2006

After the installation process has been finished I was able to log into the OPSI depotserver via Putty.

Screenshot: Putty Login



Basic options for your PuTTY session

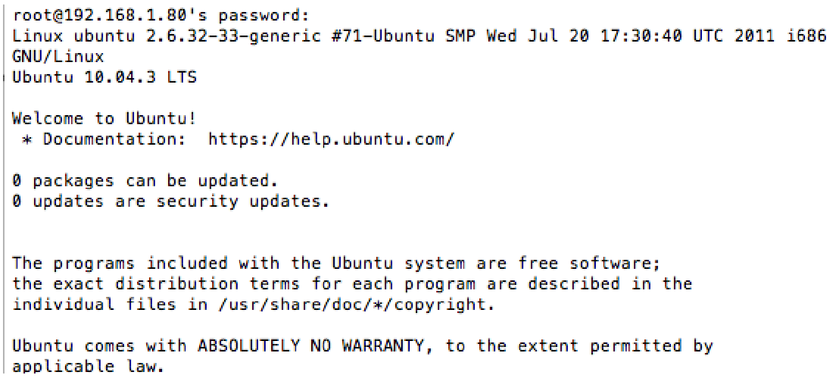
Specify the destination you want to connect to

Host Name (or IP address): 192.168.1.80

Port: 22

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Screenshot: Welcome Screen after successful login



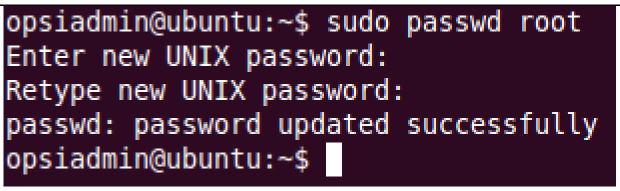
```
root@192.168.1.80's password:
Linux ubuntu 2.6.32-33-generic #71-Ubuntu SMP Wed Jul 20 17:30:40 UTC 2011 i686
GNU/Linux
Ubuntu 10.04.3 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

0 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

5.2.2 Activate root account

The activation of the root account is not necessary for a successful OPSI installation however you save the permanent entry of the “sudo” command.

ℹ	Security freaks can let the root account be deactivated.

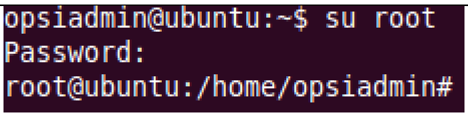
Root password change:



```
opsiadmin@ubuntu:~$ sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
opsiadmin@ubuntu:~$ █
```

VIVEK, nixCraft Blog, 2006

Root login:

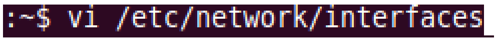


```
opsiadmin@ubuntu:~$ su root
Password:
root@ubuntu:/home/opsiadmin#
```

5.2.3 Network verficiation

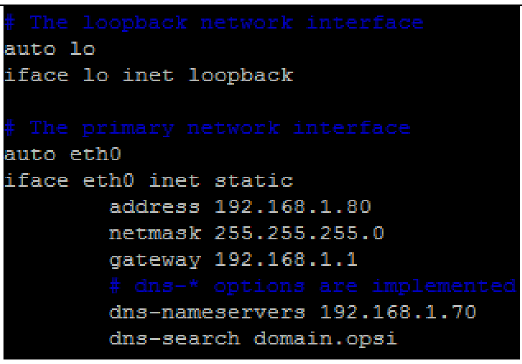
Ubuntu saves all network settings in the file /etc/network/interfaces. It is good to check this file and all entries for any errors.

Opened file interfaces in an editor (vi):



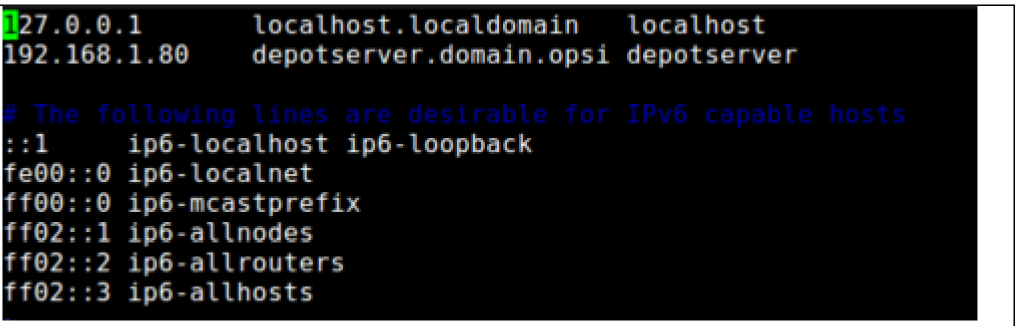
```
:~$ vi /etc/network/interfaces_
```

The entries of the primary network adapters should look like this:



```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.1.80
        netmask 255.255.255.0
        gateway 192.168.1.1
        # dns-* options are implemented
        dns-nameservers 192.168.1.70
        dns-search domain.opsi
```

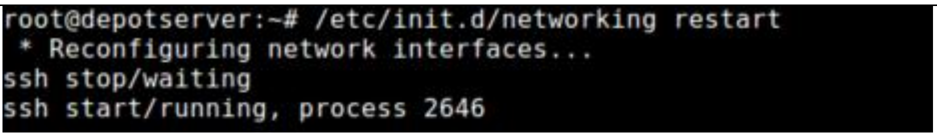
Moreover Ubuntu sets the hostname and IP address of the server in the /etc/hosts file This is how the content of the hosts file should look like:



```
127.0.0.1       localhost.localdomain   localhost
192.168.1.80    depotserver.domain.opsi depotserver

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Any distinctions should be adjusted. This applies for both files (interfaces and hosts)

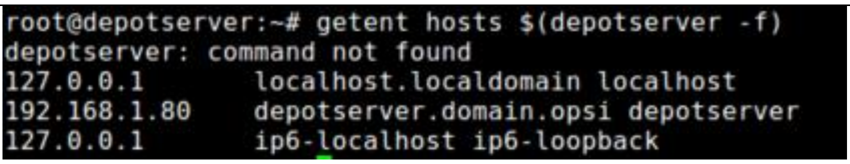
To save all changes, the network services have to be rebooted



```
root@depotserver:~# /etc/init.d/networking restart
 * Reconfiguring network interfaces...
ssh stop/waiting
ssh start/running, process 2646
```

VIVEK, nixCraft Blog, 2007

With following command I tested the entries for the depotserver. The result should look like this:

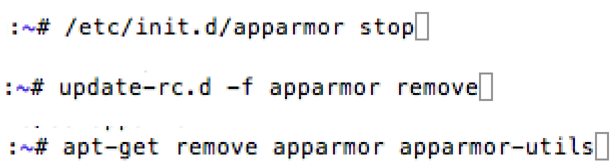


```
root@depotserver:~# getent hosts $(depotserver -f)
depotserver: command not found
127.0.0.1       localhost.localdomain localhost
192.168.1.80    depotserver.domain.opsi depotserver
127.0.0.1       ip6-localhost ip6-loopback
```

If the result doesn't look like this, there is still an error in the hosts file or the name resolution is wrong.

5.2.4 Remove AppArmor

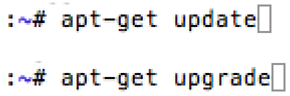
AppArmor is a security extension for Ubunut. Unfortunately usually it brings more disadvantages than advantages and therefore should be deactivated.



```
:~# /etc/init.d/apparmor stop
:~# update-rc.d -f apparmor remove
:~# apt-get remove apparmor apparmor-utils
```

5.2.5 Server Update

Before the OPSI installation it is wise to bring the server up-to-date. Just with two commands the server will automatically load all required updates from the internet.



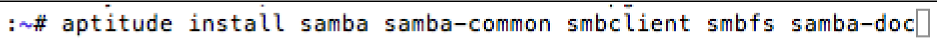
```
:~# apt-get update
:~# apt-get upgrade
```

5.3 Software Installation

First of all OPSI requires some software packages which have to be pre-installed. Those software packages include "Samba", "MySQL", "Python" and "Java".

Samba Installation

Samba allows file and print sharing between computers running Windows and computers running Unix.

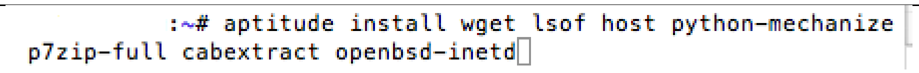


```
:~# aptitude install samba samba-common smbclient smbfs samba-doc
```

Wikipedia, Samba (software), 2011

Python Installation

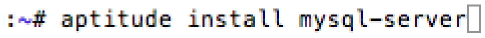
OPSI requires this installation of python-mechanize.



```
:~# aptitude install wget lsof host python-mechanize
p7zip-full cabextract openbsd-inetd
```

MySQL Installation

If you want to use MySQL as Backend for License Management or Inventory, you require the MySQL-Server. With this command I installed the MySQL-Server



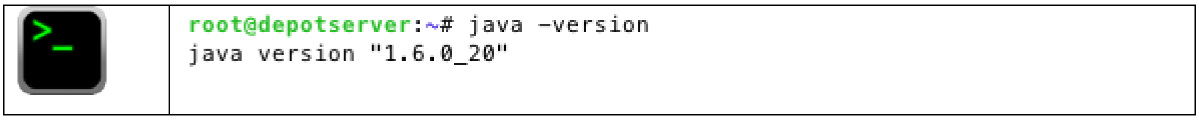
```
:~# aptitude install mysql-server
```

Unfortunately MySQL as backend is nowadays out of the test phase and from now on a non-free module.

During the SQL-Server installation I was prompted to provide the password for the root user.

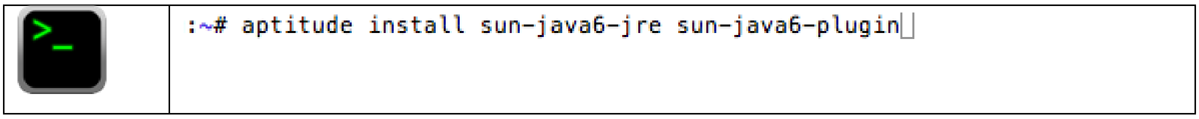
Java Installation

The administration of the OPSI depotserver and the connected clients is done with help of a small program called “opsi-configed”. This program is written in Java and requires at least Java version 1.6.0. Java should be already installed on Ubuntu 11.04. Checked my Java Version:



```
root@depotserver:~# java -version
java version "1.6.0_20"
```

Installed Java if version < 1.6.0



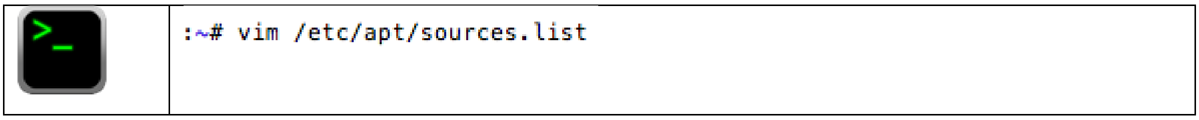
```
:~# aptitude install sun-java6-jre sun-java6-plugin
```

5.4 OPSI installation

5.4.1 Update Sources.list

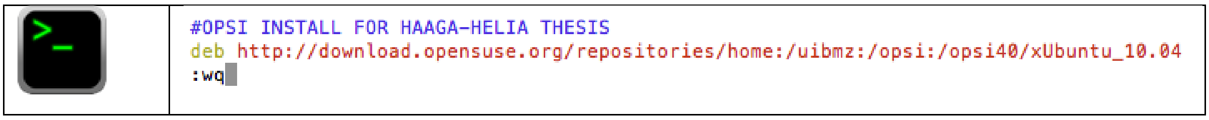
Before the OPSI packages can be installed, it was necessary to add following command into the /etc/apt/sources.list file.

Opened the file with vim or vi



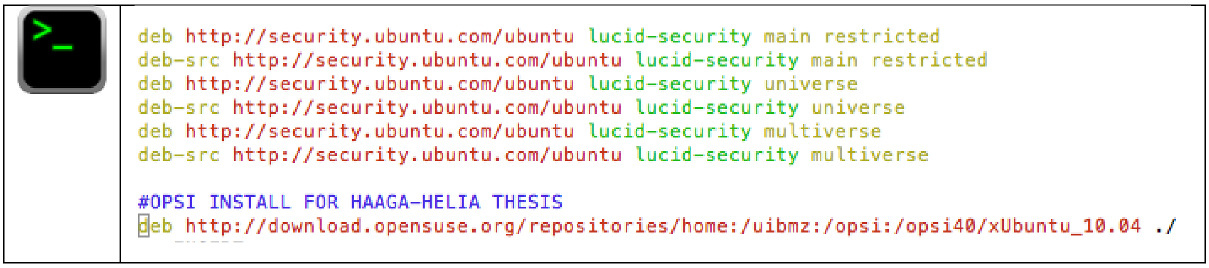
```
:~# vim /etc/apt/sources.list
```

Added this line:



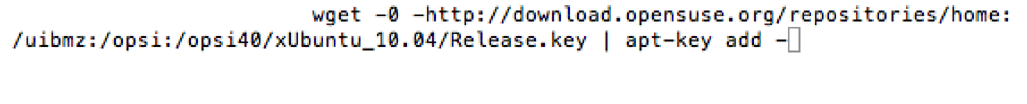
```
#OPSI INSTALL FOR HAAGA-HELIA THESIS
deb http://download.opensuse.org/repositories/home:/uibmz:/opsi:/opsi40/xUbuntu_10.04
:wq
```

The file looks like this:



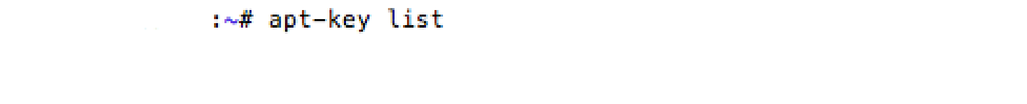
```
deb http://security.ubuntu.com/ubuntu lucid-security main restricted
deb-src http://security.ubuntu.com/ubuntu lucid-security main restricted
deb http://security.ubuntu.com/ubuntu lucid-security universe
deb-src http://security.ubuntu.com/ubuntu lucid-security universe
deb http://security.ubuntu.com/ubuntu lucid-security multiverse
deb-src http://security.ubuntu.com/ubuntu lucid-security multiverse

#OPSI INSTALL FOR HAAGA-HELIA THESIS
deb http://download.opensuse.org/repositories/home:/uibmz:/opsi:/opsi40/xUbuntu_10.04 ./
```

Imported the signatures of the repositories with following command:



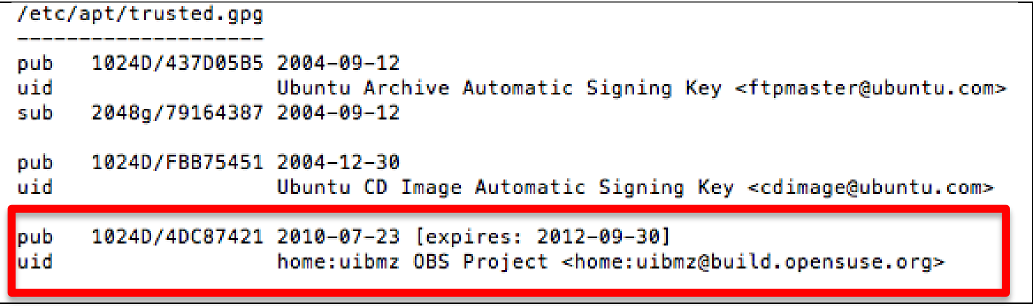
```
                    wget -0 -http://download.opensuse.org/repositories/home:
/uibmz:/opsi:/opsi40/xUbuntu_10.04/Release.key | apt-key add -
```

Checked if the import was successful:



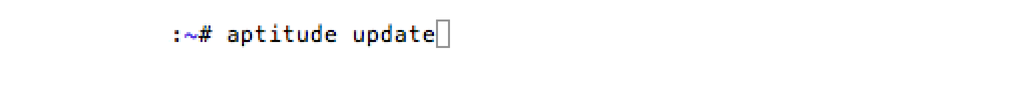
```
          :~# apt-key list
```

The output should look like this:



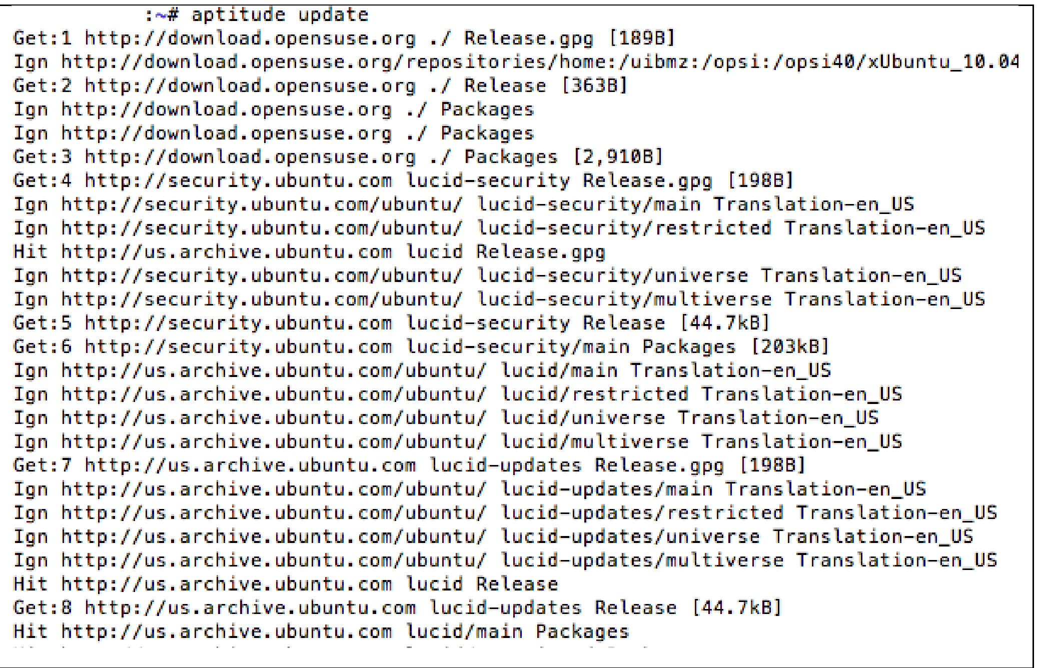
```
/etc/apt/trusted.gpg
--------------------
pub   1024D/437D05B5 2004-09-12
uid                  Ubuntu Archive Automatic Signing Key <ftpmaster@ubuntu.com>
sub   2048g/79164387 2004-09-12

pub   1024D/FBB75451 2004-12-30
uid                  Ubuntu CD Image Automatic Signing Key <cdimage@ubuntu.com>

pub   1024D/4DC87421 2010-07-23 [expires: 2012-09-30]
uid                  home:uibmz OBS Project <home:uibmz@build.opensuse.org>
```

I executed following commands to load the installation files previously added to the sources.list file:



```
             :~# aptitude update
```

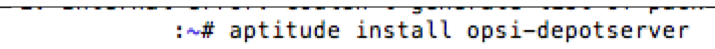
Result may look like this:



```
           :~# aptitude update
Get:1 http://download.opensuse.org ./ Release.gpg [189B]
Ign http://download.opensuse.org/repositories/home:/uibmz:/opsi:/opsi40/xUbuntu_10.04
Get:2 http://download.opensuse.org ./ Release [363B]
Ign http://download.opensuse.org ./ Packages
Ign http://download.opensuse.org ./ Packages
Get:3 http://download.opensuse.org ./ Packages [2,910B]
Get:4 http://security.ubuntu.com lucid-security Release.gpg [198B]
Ign http://security.ubuntu.com/ubuntu/ lucid-security/main Translation-en_US
Ign http://security.ubuntu.com/ubuntu/ lucid-security/restricted Translation-en_US
Hit http://us.archive.ubuntu.com lucid Release.gpg
Ign http://security.ubuntu.com/ubuntu/ lucid-security/universe Translation-en_US
Ign http://security.ubuntu.com/ubuntu/ lucid-security/multiverse Translation-en_US
Get:5 http://security.ubuntu.com lucid-security Release [44.7kB]
Get:6 http://security.ubuntu.com lucid-security/main Packages [203kB]
Ign http://us.archive.ubuntu.com/ubuntu/ lucid/main Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ lucid/restricted Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ lucid/universe Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ lucid/multiverse Translation-en_US
Get:7 http://us.archive.ubuntu.com lucid-updates Release.gpg [198B]
Ign http://us.archive.ubuntu.com/ubuntu/ lucid-updates/main Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ lucid-updates/restricted Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ lucid-updates/universe Translation-en_US
Ign http://us.archive.ubuntu.com/ubuntu/ lucid-updates/multiverse Translation-en_US
Hit http://us.archive.ubuntu.com lucid Release
Get:8 http://us.archive.ubuntu.com lucid-updates Release [44.7kB]
Hit http://us.archive.ubuntu.com lucid/main Packages
```

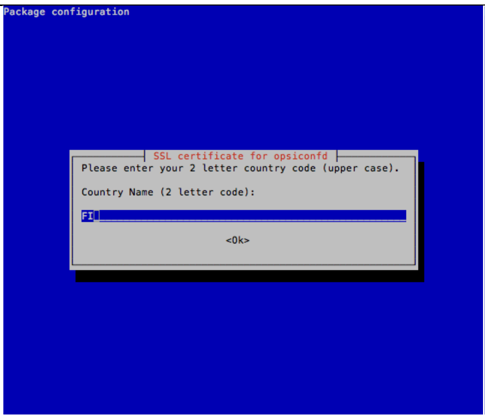
5.4.2 Install the OPSI deptoserver

Depotserver install command:

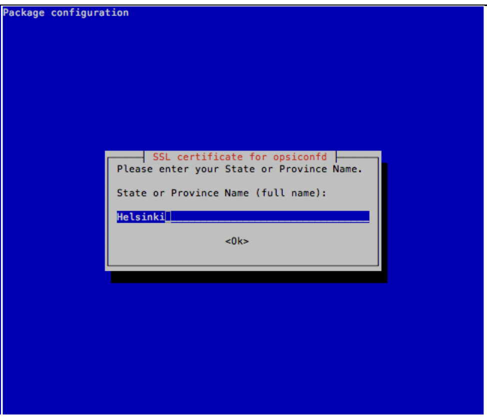


```
:~# aptitude install opsi-depotserver
```

Package Configuration Country Code and State or Province Name:

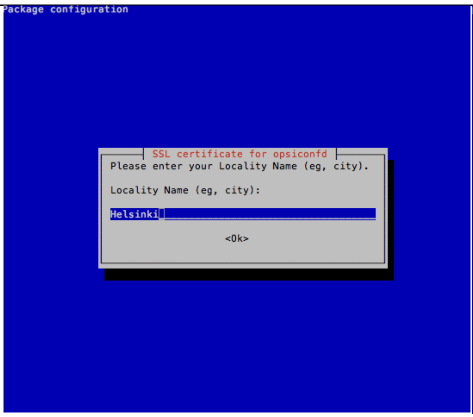


```
SSL certificate for opsiconfd
Please enter your 2 letter country code (upper case).

Country Name (2 letter code):

FI

<Ok>
```

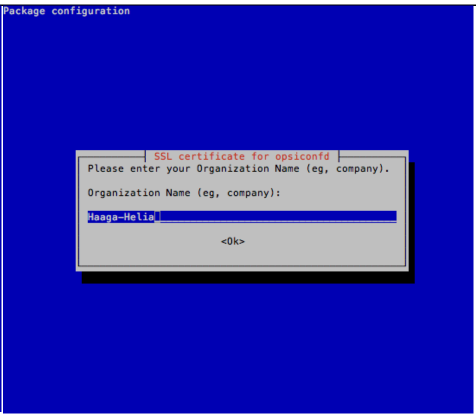


```
SSL certificate for opsiconfd
Please enter your State or Province Name.

State or Province Name (full name):

Helsinki

<Ok>
```

Package Configuration Locality Name and Organization Name

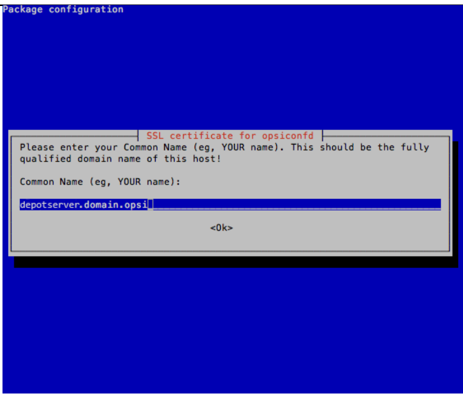


```
SSL certificate for opsiconfd
Please enter your Locality Name (eg, city).

Locality Name (eg, city):

Helsinki

<Ok>
```

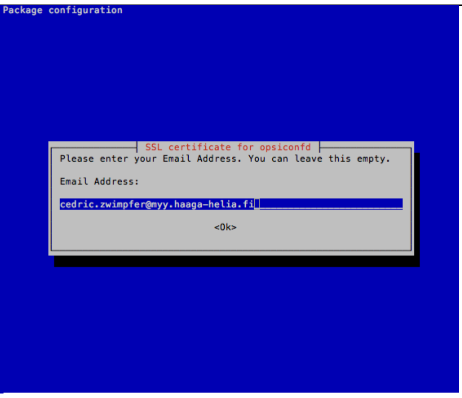


```
SSL certificate for opsiconfd
Please enter your Organization Name (eg, company).

Organization Name (eg, company):

Haaga-Helia

<Ok>
```

Package Configuration FQDN of Host and Email Address

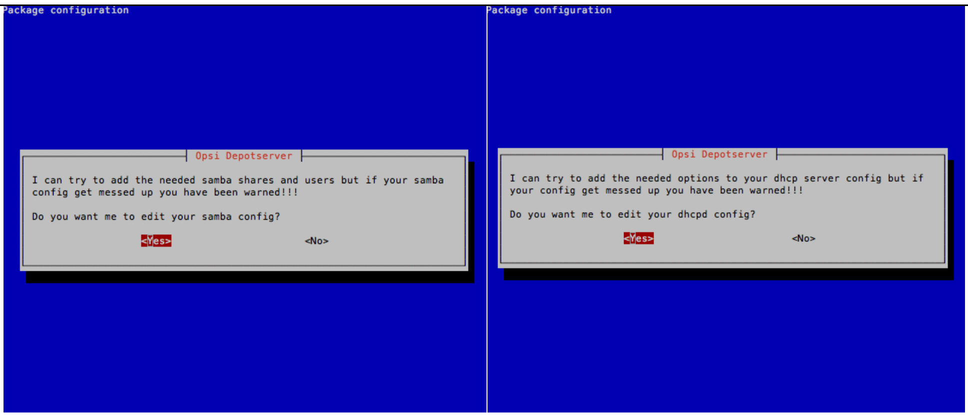


```
SSL certificate for opsiconfd
Please enter your Common Name (eg, YOUR name). This should be the fully
qualified domain name of this host!

Common Name (eg, YOUR name):

depotserver.domain.opsi

<Ok>
```



```
SSL certificate for opsiconfd
Please enter your Email Address. You can leave this empty.

Email Address:

cedric.zwimpfer@myy.haaga-helia.fi

<Ok>
```

During the configuration I was prompted to agree to patch the dhcp.conf and smb.conf; answered both with “Yes”.



```
Package configuration

┤ Opsi Depotserver ├

I can try to add the needed samba shares and users but if your samba
config get messed up you have been warned!!!

Do you want me to edit your samba config?

            <Yes>                    <No>
```

```
Package configuration

┤ Opsi Depotserver ├

I can try to add the needed options to your dhcp server config but if
your config get messed up you have been warned!!!

Do you want me to edit your dhcpd config?

            <Yes>                    <No>
```

After the installation is complete it is advisable to provide a password for the user pcpatch I assigned the password according to the list planned user rights.

```
root@depotserver:~# opsi-admin -d task setPcpatchPassword
Password:
```

pcpatch is a pseudo user. Later on this user is permited to log on the clients for the software distribution. pcpatch has access to the installation files which are stored on the share. pcpatch is simultaniously system-user, samba-user and opsi-user.

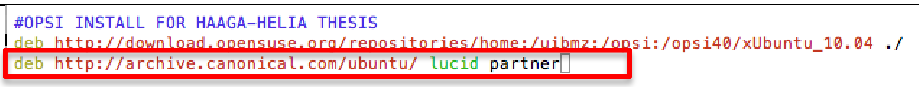
Furthermore I had to execute following command to install the management interface:

```
:~# aptitude install opsi-configed
```

If you plan to run the management interface straight on the server then you need the java plugin from Sun/Oracle. To install that I had to edit once again the sources.list file. I opened the sources.list with (vi):

```
~# vi /etc/apt/sources.list
```

Added following line:



```
#OPSI INSTALL FOR HAAGA-HELIA THESIS
deb http://download.opensuse.org/repositories/home:/uibmz:/opsi:/opsi40/xUbuntu_10.04 ./
deb http://archive.canonical.com/ubuntu/ lucid partner
```

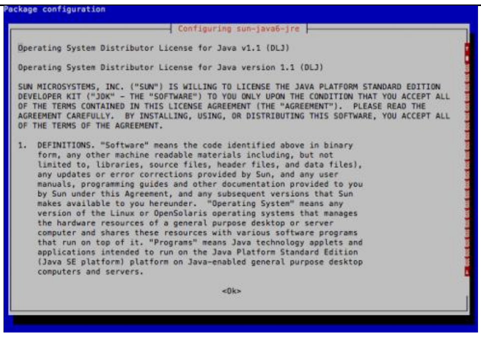
Updateed the list:

```
:~# aptitude update
```

Installed the Java plugin

```
:~# aptitude install sun-java6-jre sun-java6-plugin
```

Agreed to the terms:



```
Package configuration

                  ┤ Configuring sun-java6-jre ├

Operating System Distributor License for Java v1.1 (DLJ)

Operating System Distributor License for Java version 1.1 (DLJ)

SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THE JAVA PLATFORM STANDARD EDITION
DEVELOPER KIT ("JDK" - THE "SOFTWARE") TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL
OF THE TERMS CONTAINED IN THIS LICENSE AGREEMENT (THE "AGREEMENT").  PLEASE READ THE
AGREEMENT CAREFULLY.  BY INSTALLING, USING, OR DISTRIBUTING THIS SOFTWARE, YOU ACCEPT ALL
OF THE TERMS OF THE AGREEMENT.

1.   DEFINITIONS. "Software" means the code identified above in binary
     form, any other machine readable materials including, but not
     limited to, libraries, source files, header files, and data files),
     any updates or error corrections provided by Sun, and any user
     manuals, programming guides and other documentation provided to you
     by Sun under this Agreement, and any subsequent versions that Sun
     makes available to you hereunder.  "Operating System" means any
     version of the Linux or OpenSolaris operating systems that manages
     the hardware resources of a general purpose desktop or server
     computer and shares these resources with various software programs
     that run on top of it. "Programs" means Java technology applets and
     applications intended to run on the Java Platform Standard Edition
     (Java SE platform) platform on Java-enabled general purpose desktop
     computers and servers.

                              <Ok>
```

5.4.3 Remove TFTPD / Install ATFTPD

TFTPD is a file transfer protocol. OPSI requires the Advanced TFTPD therefore I had to uninstall the basic version with following command:

```
:~# aptitude remove tftpd

:~# update-inetd --remove tftpd
```

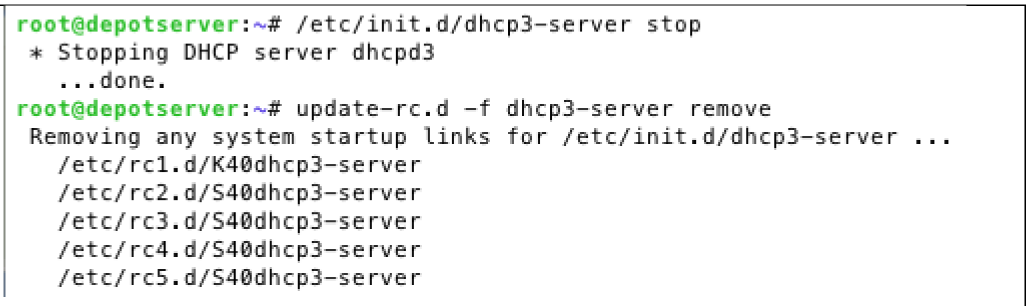
Wikipedia, Trivial File Transfer Protocol, 2011

Now I installed ATFTPD:

```
:~# aptitude install opsi-atftpd
```

5.4.4 Remove DHCP

Since I am using my router's DHCP service which doesn't support the PXE boot I am forced to skip this step. If you have a Windows or Linux DHCP-Server you don't need the integrated DHCP service in OPSI. With following commands you stop and remove the OPSI DHCP service:



```
root@depotserver:~# /etc/init.d/dhcp3-server stop
 * Stopping DHCP server dhcpd3
   ...done.
root@depotserver:~# update-rc.d -f dhcp3-server remove
 Removing any system startup links for /etc/init.d/dhcp3-server ...
   /etc/rc1.d/K40dhcp3-server
   /etc/rc2.d/S40dhcp3-server
   /etc/rc3.d/S40dhcp3-server
   /etc/rc4.d/S40dhcp3-server
   /etc/rc5.d/S40dhcp3-server
```

5.5 Backend Configuration

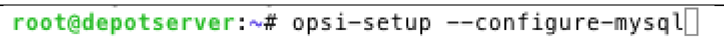
OPSI supports different backends of data storage. The most common ones are:

- file (data storage in files)
- ldap (data storage in LDAP database)
- mysql (data storage in a MySQL database)

Besides of that there are backends for special purposes:

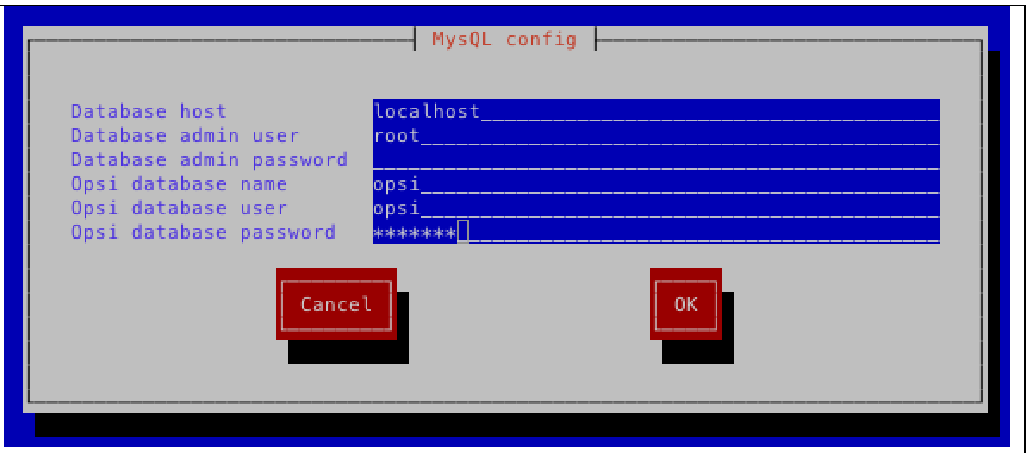
- opsipxeconfd (service for the OPSI pxe-boot)
- dhcp (for the communication with the dhcp-server on the opsi-server)
- JSON-RPC (for forwarding all queries to another server)

This is how I configured it:

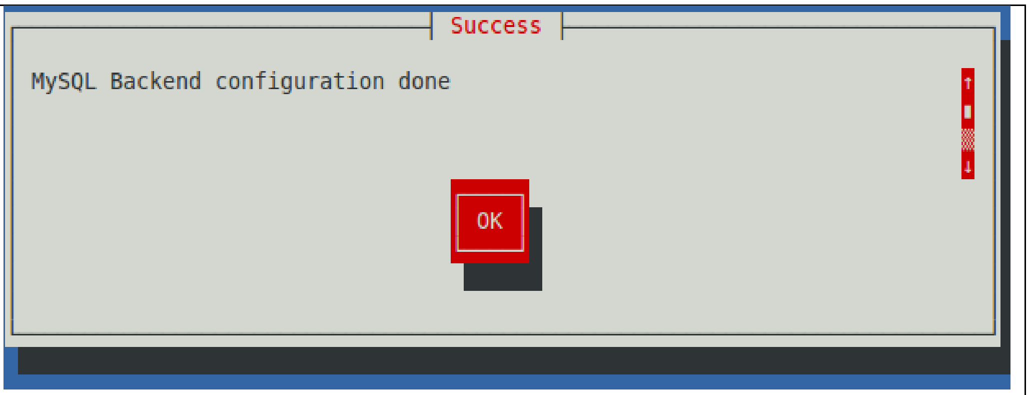


```
root@depotserver:~# opsi-setup --configure-mysql
```

All prompts can be approved with “enter” beside of the password entries:



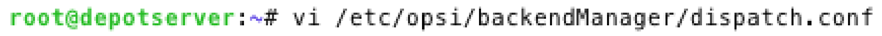
```
Database host             localhost
Database admin user       root
Database admin password
Opsi database name        opsi
Opsi database user        opsi
Opsi database password    *******
```



MySQL Backend configuration done

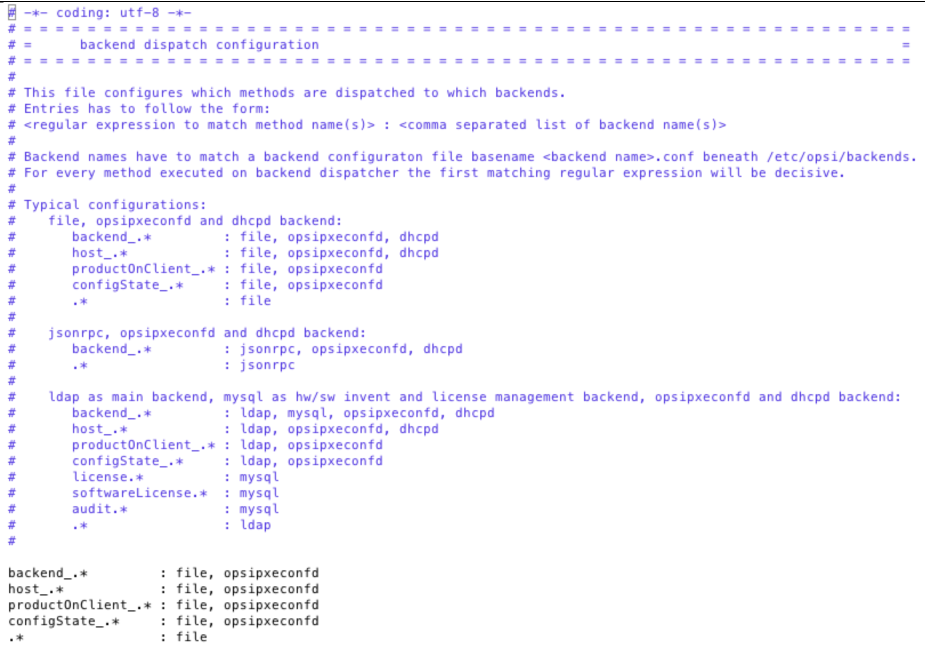
Different data can be stored in different backends. For special processes it is necessary to state multiple backends. In the file /etc/opsi/backendManager/dispatch.conf you can assign methods to backends.

With following command I opened the dispatch.conf file:



```
root@depotserver:~# vi /etc/opsi/backendManager/dispatch.conf
```

This is how the original dispatch.conf file looks like:

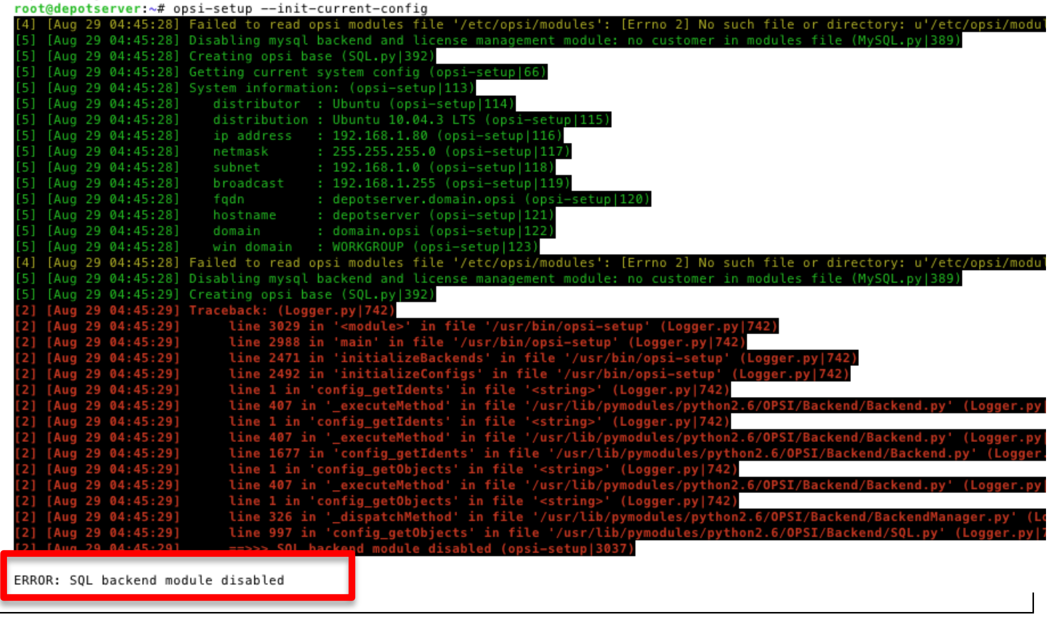


```
# -*- coding: utf-8 -*-
# = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
# =      backend dispatch configuration                                                                          =
# = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
#
# This file configures which methods are dispatched to which backends.
# Entries has to follow the form:
# <regular expression to match method name(s)> : <comma separated list of backend name(s)>
#
# Backend names have to match a backend configuraton file basename <backend name>.conf beneath /etc/opsi/backends.
# For every method executed on backend dispatcher the first matching regular expression will be decisive.
#
# Typical configurations:
#    file, opsipxeconfd and dhcpd backend:
#       backend_.*         : file, opsipxeconfd, dhcpd
#       host_.*            : file, opsipxeconfd, dhcpd
#       productOnClient_.* : file, opsipxeconfd
#       configState_.*     : file, opsipxeconfd
#       .*                 : file
#
#    jsonrpc, opsipxeconfd and dhcpd backend:
#       backend_.*         : jsonrpc, opsipxeconfd, dhcpd
#       .*                 : jsonrpc
#
#    ldap as main backend, mysql as hw/sw invent and license management backend, opsipxeconfd and dhcpd backend:
#       backend_.*         : ldap, mysql, opsipxeconfd, dhcpd
#       host_.*            : ldap, opsipxeconfd, dhcpd
#       productOnClient_.* : ldap, opsipxeconfd
#       configState_.*     : ldap, opsipxeconfd
#       license.*          : mysql
#       softwareLicense.*  : mysql
#       audit.*            : mysql
#       .*                 : ldap
#

backend_.*         : file, opsipxeconfd
host_.*            : file, opsipxeconfd
productOnClient_.* : file, opsipxeconfd
configState_.*     : file, opsipxeconfd
.*                 : file
```

I changed the values from “file” to “mysql”:

```
backend_.*         : mysql, opsipxeconfd
host_.*            : mysql, opsipxeconfd
productOnClient_.* : mysql, opsipxeconfd
configState_.*     : mysql, opsipxeconfd
.*                 : mysql
```

I wanted to use the MySQL-backend for the inventory. However later I found out that the MySQL backend is a fee required module and not working without payment. It allows you to configure it but later on won’t establish a connection to the DB.

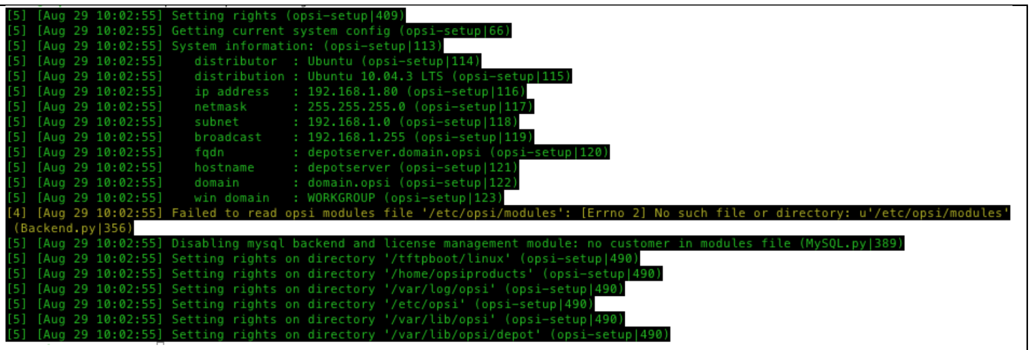


```
root@depotserver:~# opsi-setup --init-current-config
[4] [Aug 29 04:45:28] Failed to read opsi modules file '/etc/opsi/modules': [Errno 2] No such file or directory: u'/etc/opsi/modu
[5] [Aug 29 04:45:28] Disabling mysql backend and license management module: no customer in modules file (MySQL.py|389)
[5] [Aug 29 04:45:28] Creating opsi base (SQL.py|392)
[5] [Aug 29 04:45:28] Getting current system config (opsi-setup|66)
[5] [Aug 29 04:45:28] System information: (opsi-setup|113)
[5] [Aug 29 04:45:28]    distributor  : Ubuntu (opsi-setup|114)
[5] [Aug 29 04:45:28]    distribution : Ubuntu 10.04.3 LTS (opsi-setup|115)
[5] [Aug 29 04:45:28]    ip address   : 192.168.1.80 (opsi-setup|116)
[5] [Aug 29 04:45:28]    netmask      : 255.255.255.0 (opsi-setup|117)
[5] [Aug 29 04:45:28]    subnet       : 192.168.1.0 (opsi-setup|118)
[5] [Aug 29 04:45:28]    broadcast    : 192.168.1.255 (opsi-setup|119)
[5] [Aug 29 04:45:28]    fqdn         : depotserver.domain.opsi (opsi-setup|120)
[5] [Aug 29 04:45:28]    hostname     : depotserver (opsi-setup|121)
[5] [Aug 29 04:45:28]    domain       : domain.opsi (opsi-setup|122)
[5] [Aug 29 04:45:28]    win domain   : WORKGROUP (opsi-setup|123)
[4] [Aug 29 04:45:28] Failed to read opsi modules file '/etc/opsi/modules': [Errno 2] No such file or directory: u'/etc/opsi/modu
[5] [Aug 29 04:45:28] Disabling mysql backend and license management module: no customer in modules file (MySQL.py|389)
[5] [Aug 29 04:45:29] Creating opsi base (SQL.py|392)
[2] [Aug 29 04:45:29] Traceback: (Logger.py|742)
[2] [Aug 29 04:45:29]      line 3029 in '<module>' in file '/usr/bin/opsi-setup' (Logger.py|742)
[2] [Aug 29 04:45:29]      line 2988 in 'main' in file '/usr/bin/opsi-setup' (Logger.py|742)
[2] [Aug 29 04:45:29]      line 2471 in 'initializeBackends' in file '/usr/bin/opsi-setup' (Logger.py|742)
[2] [Aug 29 04:45:29]      line 2492 in 'initializeConfigs' in file '/usr/bin/opsi-setup' (Logger.py|742)
[2] [Aug 29 04:45:29]      line 1 in 'config_getIdents' in file '<string>' (Logger.py|742)
[2] [Aug 29 04:45:29]      line 407 in '_executeMethod' in file '/usr/lib/pymodules/python2.6/OPSI/Backend/Backend.py' (Logger.py
[2] [Aug 29 04:45:29]      line 1 in 'config_getIdents' in file '<string>' (Logger.py|742)
[2] [Aug 29 04:45:29]      line 407 in '_executeMethod' in file '/usr/lib/pymodules/python2.6/OPSI/Backend/Backend.py' (Logger.py
[2] [Aug 29 04:45:29]      line 1677 in 'config_getIdents' in file '/usr/lib/pymodules/python2.6/OPSI/Backend/Backend.py' (Logger.
[2] [Aug 29 04:45:29]      line 1 in 'config_getObjects' in file '<string>' (Logger.py|742)
[2] [Aug 29 04:45:29]      line 407 in '_executeMethod' in file '/usr/lib/pymodules/python2.6/OPSI/Backend/Backend.py' (Logger.py
[2] [Aug 29 04:45:29]      line 1 in 'config_getObjects' in file '<string>' (Logger.py|742)
[2] [Aug 29 04:45:29]      line 326 in '_dispatchMethod' in file '/usr/lib/pymodules/python2.6/OPSI/Backend/BackendManager.py' (L
[2] [Aug 29 04:45:29]      line 997 in 'config_getObjects' in file '/usr/lib/pymodules/python2.6/OPSI/Backend/SQL.py' (Logger.py|7
[2] [Aug 29 04:45:29]      ==>>> SQL backend module disabled (opsi-setup|3037)

ERROR: SQL backend module disabled
```

It is crucial to execute following commands every time the content of this file dispatch.conf has changed. It is even advisable to run those commands after the first launch of the server:

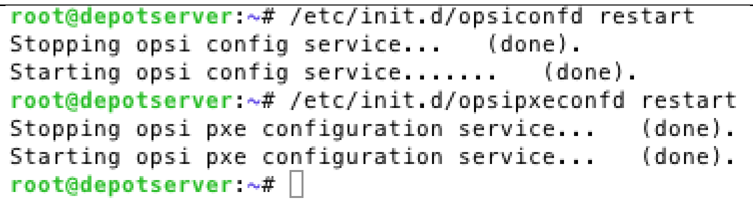
```
root@depotserver:~# opsi-setup --init-current-config

root@depotserver:~# opsi-setup --set-rights
```

This is how the output looks like:



```
[5] [Aug 29 10:02:55] Setting rights (opsi-setup|409)
[5] [Aug 29 10:02:55] Getting current system config (opsi-setup|66)
[5] [Aug 29 10:02:55] System information: (opsi-setup|113)
[5] [Aug 29 10:02:55]    distributor  : Ubuntu (opsi-setup|114)
[5] [Aug 29 10:02:55]    distribution : Ubuntu 10.04.3 LTS (opsi-setup|115)
[5] [Aug 29 10:02:55]    ip address   : 192.168.1.80 (opsi-setup|116)
[5] [Aug 29 10:02:55]    netmask      : 255.255.255.0 (opsi-setup|117)
[5] [Aug 29 10:02:55]    subnet       : 192.168.1.0 (opsi-setup|118)
[5] [Aug 29 10:02:55]    broadcast    : 192.168.1.255 (opsi-setup|119)
[5] [Aug 29 10:02:55]    fqdn         : depotserver.domain.opsi (opsi-setup|120)
[5] [Aug 29 10:02:55]    hostname     : depotserver (opsi-setup|121)
[5] [Aug 29 10:02:55]    domain       : domain.opsi (opsi-setup|122)
[5] [Aug 29 10:02:55]    win domain   : WORKGROUP (opsi-setup|123)
[4] [Aug 29 10:02:55] Failed to read opsi modules file '/etc/opsi/modules': [Errno 2] No such file or directory: u'/etc/opsi/modules'
 (Backend.py|356)
[5] [Aug 29 10:02:55] Disabling mysql backend and license management module: no customer in modules file (MySQL.py|389)
[5] [Aug 29 10:02:55] Setting rights on directory '/tftpboot/linux' (opsi-setup|490)
[5] [Aug 29 10:02:55] Setting rights on directory '/home/opsiproducts' (opsi-setup|490)
[5] [Aug 29 10:02:55] Setting rights on directory '/var/log/opsi' (opsi-setup|490)
[5] [Aug 29 10:02:55] Setting rights on directory '/etc/opsi' (opsi-setup|490)
[5] [Aug 29 10:02:55] Setting rights on directory '/var/lib/opsi' (opsi-setup|490)
[5] [Aug 29 10:02:55] Setting rights on directory '/var/lib/opsi/depot' (opsi-setup|490)
```

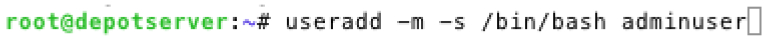
opsiconfd and opsipxeconfd have to be restarted after the update:



```
root@depotserver:~# /etc/init.d/opsiconfd restart
Stopping opsi config service...   (done).
Starting opsi config service.......   (done).
root@depotserver:~# /etc/init.d/opsipxeconfd restart
Stopping opsi pxe configuration service...   (done).
Starting opsi pxe configuration service...   (done).
root@depotserver:~# 
```

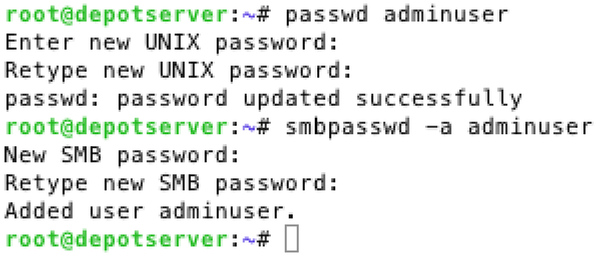
5.6 User and Groups administration

The administration of OPSI is only permitted by members of the opsi-admin group. Since only the user root is a member of the opsi-admin group it is recommended to add another user called adminuser:



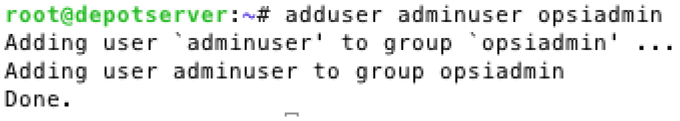
```
root@depotserver:~# useradd -m -s /bin/bash adminuser
```

Afterwards I set a Unix and Samba password for the created adminuser:



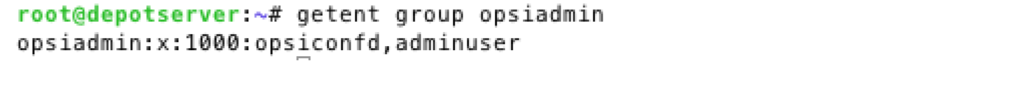
```
root@depotserver:~# passwd adminuser
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@depotserver:~# smbpasswd -a adminuser
New SMB password:
Retype new SMB password:
Added user adminuser.
root@depotserver:~# 
```

Then I added the freshly created adminuser to the group opsiadmin:



```
root@depotserver:~# adduser adminuser opsiadmin
Adding user `adminuser' to group `opsiadmin' ...
Adding user adminuser to group opsiadmin
Done.
```

Finally it is wise to double-check the group membership of adminuser with following command:

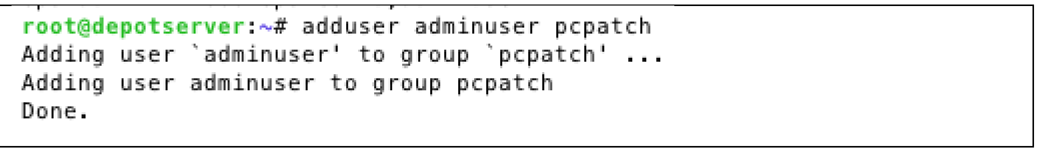


```
root@depotserver:~# getent group opsiadmin
opsiadmin:x:1000:opsiconfd,adminuser
```

The result displays that the group opsiadmin has two members: opsiconfd and adminuser – this is correct!

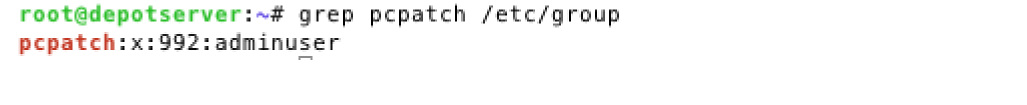
root is not member of opsiadmin and therefore not eligible to execute OPSI administration commands.

Additionally all users who should be eligible to pack products (opsi-makeproductfile), install products (opsi-package-manager) or manually configure files have to be in the group pcpatch:



```
root@depotserver:~# adduser adminuser pcpatch
Adding user `adminuser' to group `pcpatch' ...
Adding user adminuser to group pcpatch
Done.
```

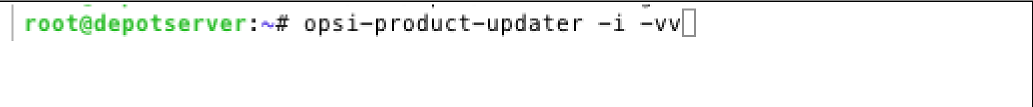
The check should look like this:



```
root@depotserver:~# grep pcpatch /etc/group
pcpatch:x:992:adminuser
```

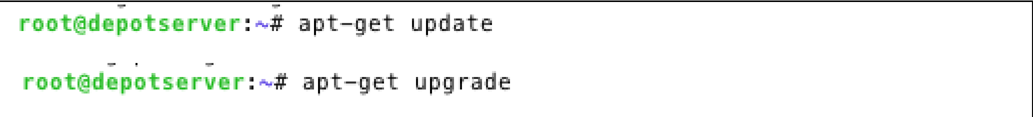
5.7 Import the minimal OPSI products

The OPSI products consist of installation procedures for Operating Systems and more. There is a tool called opsi-product-update which automatically downloads all necessary packages from the OPSI repository. Opsi-product-update installs the packages afterwards on the server. This is the command to execute the update process:



```
root@depotserver:~# opsi-product-updater -i -vv
```

Finally it is advisable to check for updates and upgrade the server in the end:



```
root@depotserver:~# apt-get update

root@depotserver:~# apt-get upgrade
```

5.7.1 OPSI Backup

Like every other rather complex system also OPSI should be backed up from time to time since OPSI is an important centralized tool for Windows Client and Windows Server Management. There are many methods to backup up OPSI. The usual method takes use of the integrated opsi-backup command.

Basically OPSI consists of 4 core elements. Although those core elements can vary on each system depending on their configuration.

OPSI Configuration:

ℹ	This is the most important element, since it holds the configuration. All configuration files are stored in this location: /etc/opsi.
⚠	This element is saved with opsi-backup command!

OPSI Backends

ℹ	This is the core element regarding of data storage. All clients, products, configuration, etc… are saved in the corresponding data backend. The available backends are: file-backend, mysql-backend, ldap, univention and dhcp. All information is stored in the /etc/opsi/backendManager/dispatch.conf file.
⚠	This element is saved with opsi-backup command!

OPSI Depotfiles

ℹ	This element hosts the actual distribution software data. Therefore depending on the amount of different operating systems used, this folder /opt/pcbin/install can be very huge in size.
⚠	This element is **NOT** saved with opsi-backup

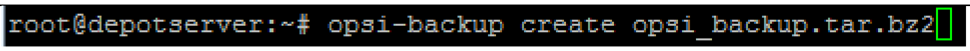
OPSI Repository

ℹ	This element is not important and can be left out in the backup strategy. The element is responsible to ease the synchronisation process of the opsi-product-updater. The folder is to be found under /var/lib/opsi/repository
⚠	This element is **NOT** saved with opsi-backup

OPSI Workbench

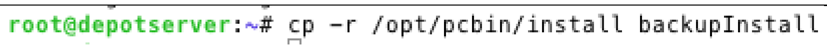
ℹ	This element is responsible for the own software packetizing. Therefore all self- integrated software is stored on the samba share /home/opsiproducts.
⚠	This element is **NOT** saved with opsi-backup

This command will automatically backup all backends plus configuration. The backup is created in the active directory with the name opsi_backup.tar.bz2.

>_	 `root@depotserver:~# opsi-backup create opsi_backup.tar.bz2`

Since not everything will be saved with that command it is recommended to just copy and paste at least the depotfiles. (Depending on the content; Repository and Workbench too.)

In my case it is advisable to save the /opt/pcbin/install directory since this directory contains the whole OPSI software including the packages. This command copies the whole directory and names it "backupInstall".

>_	 `root@depotserver:~# cp -r /opt/pcbin/install backupInstall`
⚠	This directory contains the complete software including Operating Systems therefore the backup could become huge.

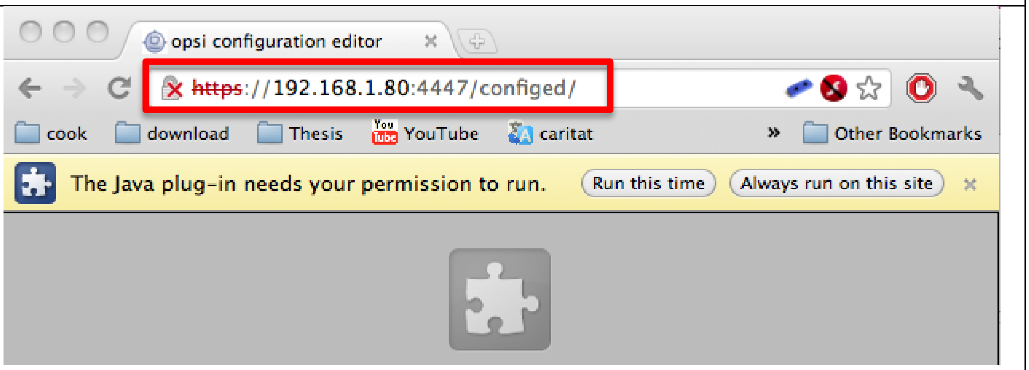
6 OPSI Management

After the installation and configuration of OPSI it is time to log into OPSI for the first time!

6.1 Management Interface

The opsi-configed is a convenient management interface to administrate OPSI and all its features. There are two possible ways to run it.

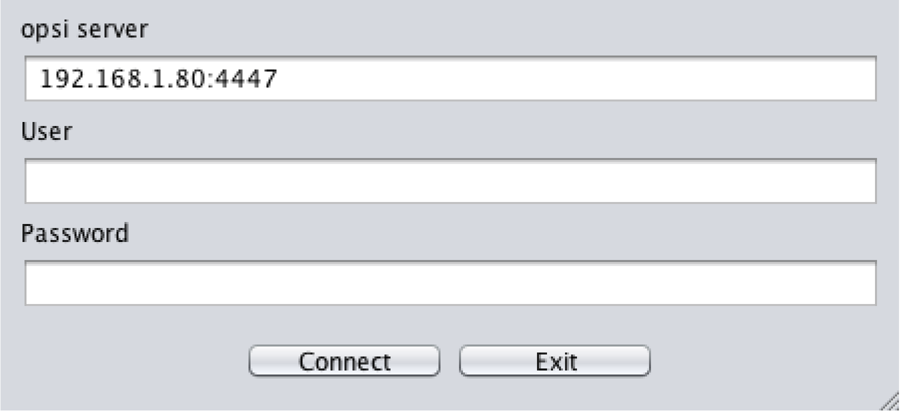
Option 1:

ℹ	Just enter the IP or Hostname in any browser of your choice. In Google Chrome I had to confirm some certificates and allow access to the computer.
	 opsi configuration editor https://192.168.1.80:4447/configed/ cook download Thesis YouTube caritat Other Bookmarks Run this time Always run on this site The Java plug-in needs your permission to run.
⚠	The applet requires java version bigger than 1.6.

Option 2:

ℹ	You need to be logged onto the OPSI depotserver. On the GUI you simply select and run the opsi-configed management interface. Optionally you switch to /usr/bin to execute following command to start opsi-configed
>_	 `root@depotserver:/usr/bin# opsi-configed`

Then I entered the user "adminuser" which is part of the admin-group and provided the password:



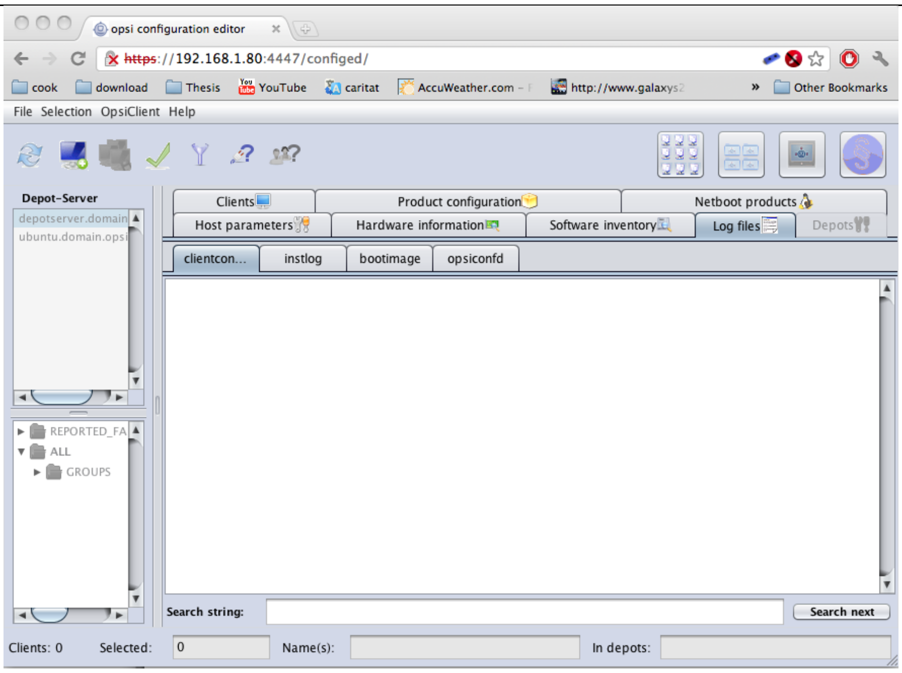
opsi server

192.168.1.80:4447

User

Password

Finally I was logged into OPSI. The management interface is pretty self-explanatory.

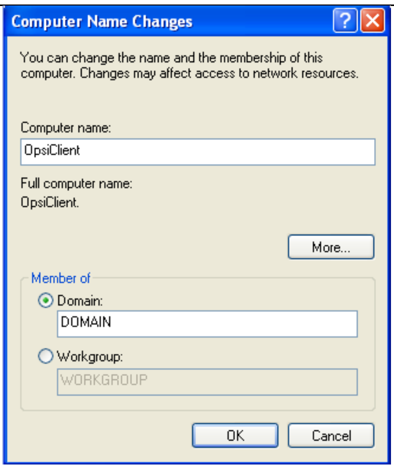


https://192.168.1.80:4447/configed/

Clients: 0 Selected: 0 Name(s): In depots:

6.2 Existing Windows Clients Integration

Now it is time to integrate already existing clients to OPSI. I will integrate my existing Windows XP test client 1 to OPSI. Henceforth here are the steps to achieve that. The client has to be in the domain. If not already it has to be joined to the domain. Afterwards it is recommendable to perform with PING a connection test to Google (internet connection), to the depotserver (OPSI connection) and win2003Server (DNS connection):



Computer Name Changes

You can change the name and the membership of this computer. Changes may affect access to network resources.

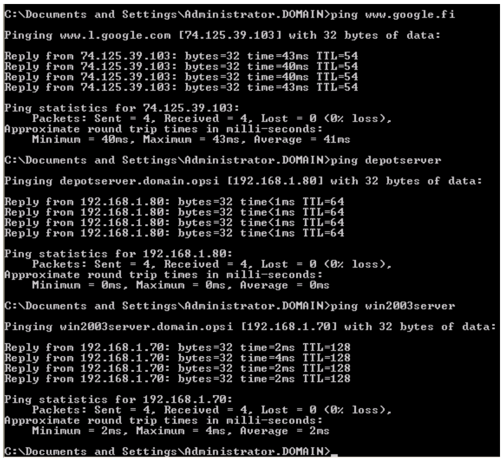
Computer name: OpsiClient

Full computer name: OpsiClient.

Member of

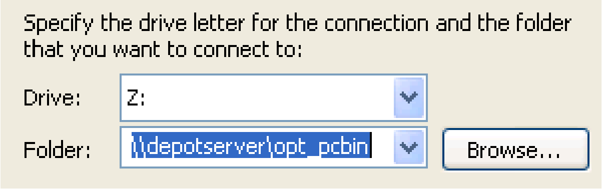
● Domain: DOMAIN

○ Workgroup: WORKGROUP



```
C:\Documents and Settings\Administrator.DOMAIN>ping www.google.fi

Pinging www.l.google.com [74.125.39.103] with 32 bytes of data:

Reply from 74.125.39.103: bytes=32 time=43ms TTL=54
Reply from 74.125.39.103: bytes=32 time=40ms TTL=54
Reply from 74.125.39.103: bytes=32 time=40ms TTL=54
Reply from 74.125.39.103: bytes=32 time=43ms TTL=54

Ping statistics for 74.125.39.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 40ms, Maximum = 43ms, Average = 41ms

C:\Documents and Settings\Administrator.DOMAIN>ping depotserver

Pinging depotserver.domain.opsi [192.168.1.80] with 32 bytes of data:

Reply from 192.168.1.80: bytes=32 time<1ms TTL=64
Reply from 192.168.1.80: bytes=32 time<1ms TTL=64
Reply from 192.168.1.80: bytes=32 time<1ms TTL=64
Reply from 192.168.1.80: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.80:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\Administrator.DOMAIN>ping win2003server

Pinging win2003server.domain.opsi [192.168.1.70] with 32 bytes of data:

Reply from 192.168.1.70: bytes=32 time=2ms TTL=128
Reply from 192.168.1.70: bytes=32 time=4ms TTL=128
Reply from 192.168.1.70: bytes=32 time=2ms TTL=128
Reply from 192.168.1.70: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.1.70:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 4ms, Average = 2ms

C:\Documents and Settings\Administrator.DOMAIN>_
```

Mount the share \\depotserver\opt-pcbin to a device drive on the client:

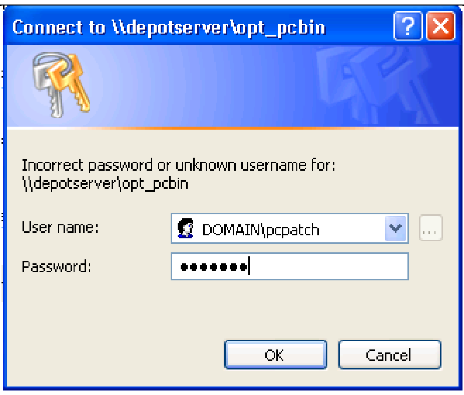


Specify the drive letter for the connection and the folder that you want to connect to:

Drive: Z:

Folder: \\depotserver\opt_pcbin

Furthermore I started the script service_setup.cmd which can be found in the "install\opsi-client-agent" folder. The script is contacting the OPSI depotserver and creates the client on the depotserver.

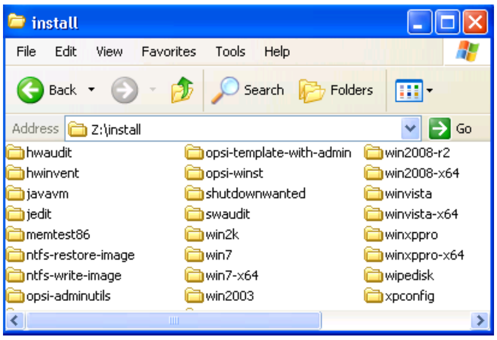


Connect to \\depotserver\opt_pcbin

Incorrect password or unknown username for:
\\depotserver\opt_pcbin

User name: DOMAIN\pcpatch

Password: •••••••

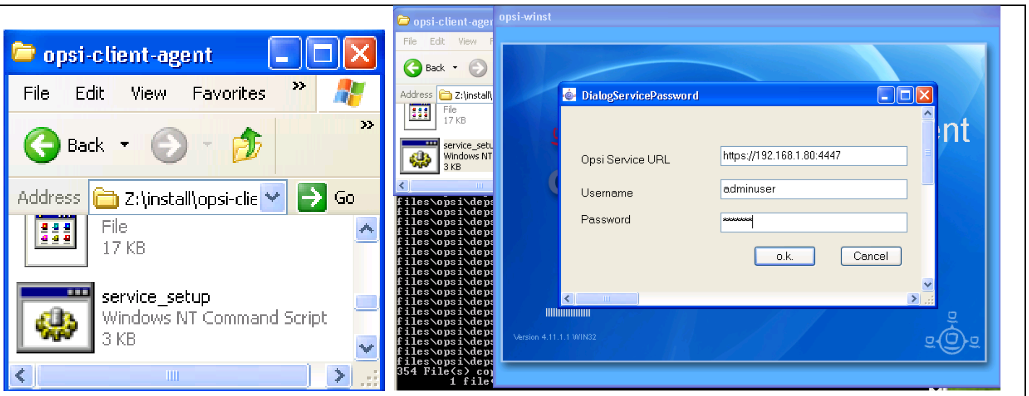


install

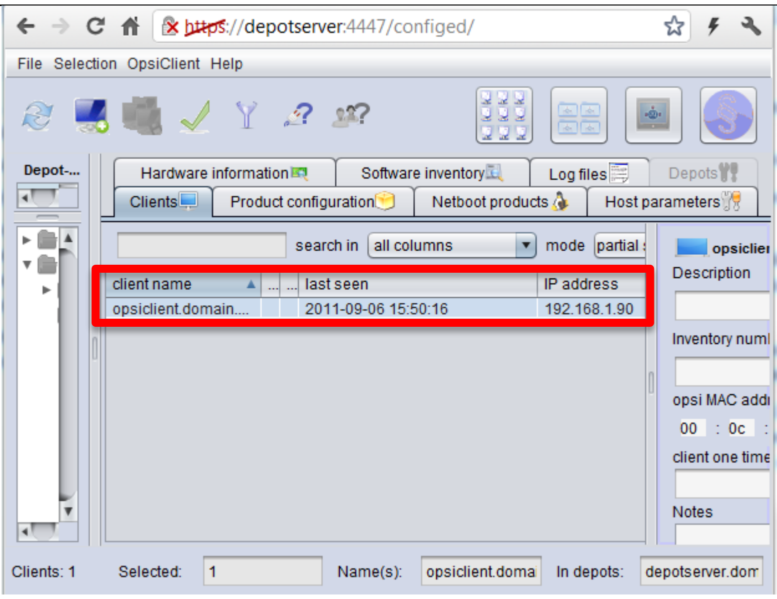
Address Z:\install

hwaudit
hwinvent
javavm
jedit
memtest86
ntfs-restore-image
ntfs-write-image
opsi-adminutils
opsi-template-with-admin
opsi-winst
shutdownwanted
swaudit
win2k
win7
win7-x64
win2003
win2008-r2
win2008-x64
winvista
winvista-x64
winxppro
winxppro-x64
wipedisk
xpconfig

During the installation the script asked me to provide the credentials for a user which is in the opsi-admin group. Therefore I put the login data for adminuser:

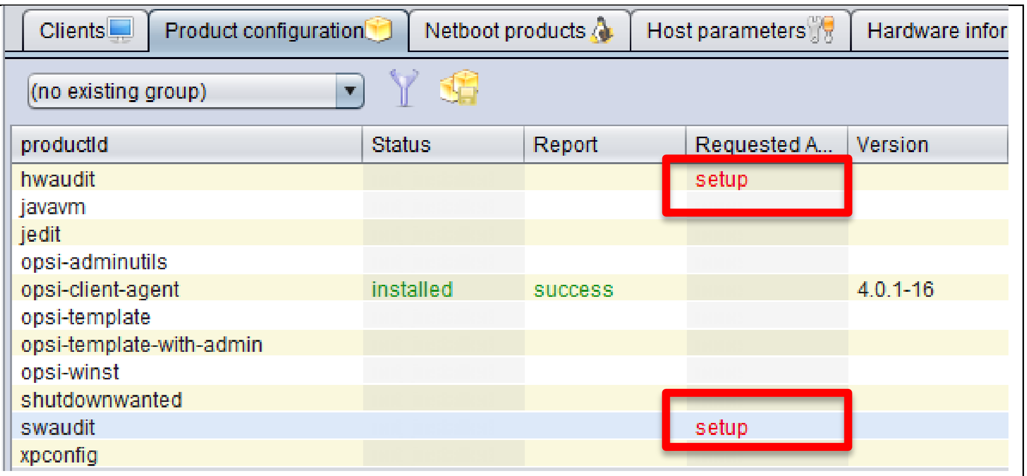
ℹ	 opsi-client-agent Address Z:\install\opsi-clie File 17 KB service_setup Windows NT Command Script 3 KB DialogServicePassword Opsi Service URL: https://192.168.1.80:4447 Username: adminuser Password: *******
⚠	The client eventually reboots after installation

After the obligatory reboot the client is integrated in OPSI. You can see it listed in the management interface:

ℹ	 https://depotserver:4447/configed/ client name | last seen | IP address opsiclient.domain.... | 2011-09-06 15:50:16 | 192.168.1.90 Clients: 1 Selected: 1 Name(s): opsiclient.doma In depots: depotserver.dom

6.3 Inventory

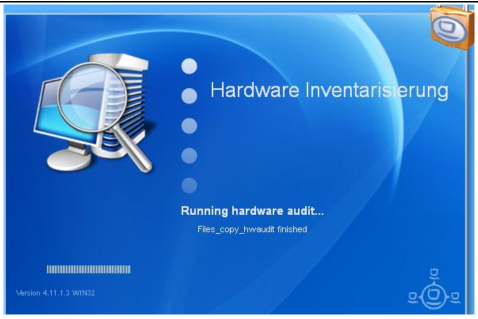
Since the test client 1 is now integrated into OPSI it is possible to execute a hardware and software inventory of that particular client. To start the process I had to initialise in the OPSI administrations interface the products hwaudit and swaudit. This happened when I changed the "Requested Action" from empty to "setup". Confirmed with pressing the red tick to save the changes:



(no existing group)

productId	Status	Report	Requested A...	Version
hwaudit			setup	
javavm				
jedit				
opsi-adminutils				
opsi-client-agent	installed	success		4.0.1-16
opsi-template				
opsi-template-with-admin				
opsi-winst				
shutdownwanted				
swaudit			setup	
xpconfig				

The client is going down for a reboot. While it is starting up, it establishes a connection to the OPSI and executes the hardware and software inventory. In the end it will transmit all gathered information to the OPSI depotserver. It took about 5 minutes:

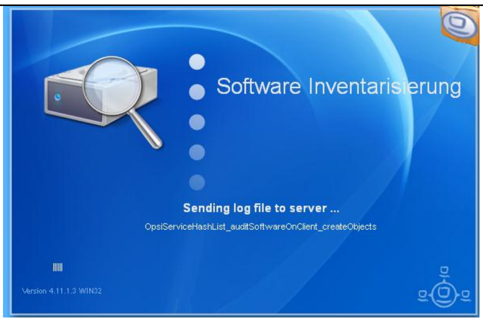


Hardware Inventarisierung

Running hardware audit...

Files_copy_hwaudit finished

Version 4.11.1.3 WIN32



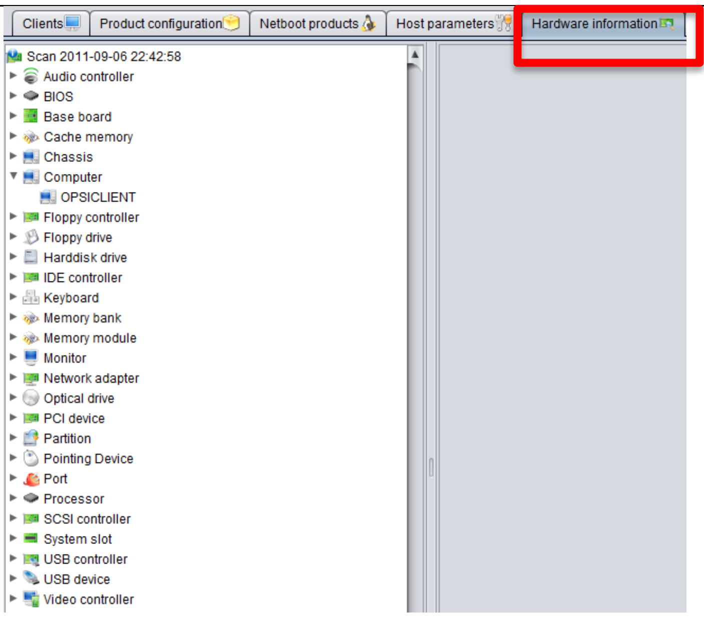
Software Inventarisierung

Sending log file to server ...

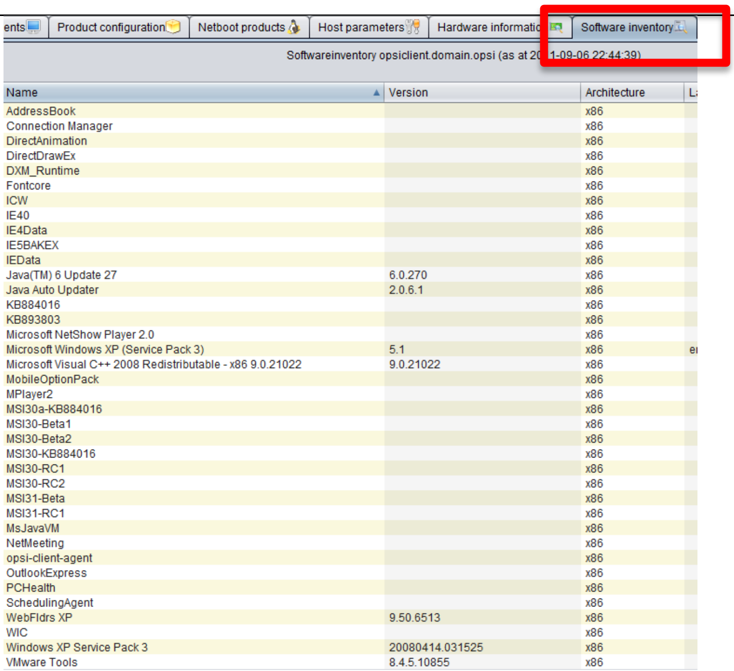
OpsiServiceHashList_auditSoftwareOnClient_createObjects

Version 4.11.1.3 WIN32

Finally I was able to see in the administration interface of OSPI the hardware and software specifications of my test client. The information can be found under the tabs “Hardware information” and “Software inventory”.



Scan 2011-09-06 22:42:58
- Audit controller
- BIOS
- Base board
- Cache memory
- Chassis
- Computer
  - OPSICLIENT
- Floppy controller
- Floppy drive
- Harddisk drive
- IDE controller
- Keyboard
- Memory bank
- Memory module
- Monitor
- Network adapter
- Optical drive
- PCI device
- Partition
- Pointing Device
- Port
- Processor
- SCSI controller
- System slot
- USB controller
- USB device
- Video controller

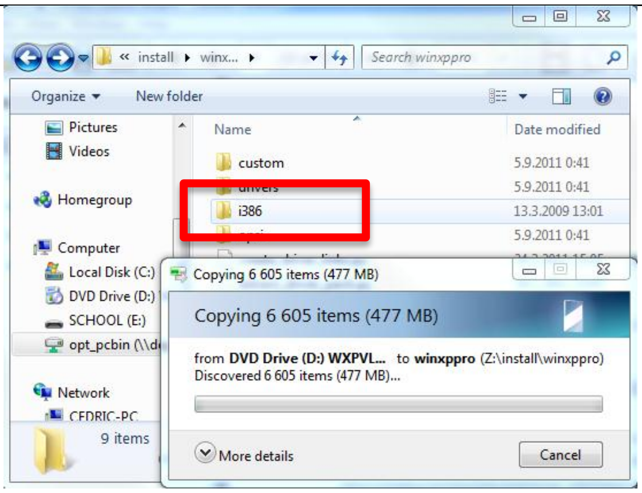


Softwareinventory opsiclient.domain.opsi (as at 20 1-09-06 22:44:39)

Name	Version	Architecture	L
AddressBook		x86	
Connection Manager		x86	
DirectAnimation		x86	
DirectDrawEx		x86	
DXM_Runtime		x86	
Fontcore		x86	
ICW		x86	
IE40		x86	
IE4Data		x86	
IE5BAKEX		x86	
IEData		x86	
Java(TM) 6 Update 27	6.0.270	x86	
Java Auto Updater	2.0.6.1	x86	
KB884016		x86	
KB893803		x86	
Microsoft NetShow Player 2.0		x86	
Microsoft Windows XP (Service Pack 3)	5.1	x86	e
Microsoft Visual C++ 2008 Redistributable - x86 9.0.21022	9.0.21022	x86	
MobileOptionPack		x86	
MPlayer2		x86	
MSI30a-KB884016		x86	
MSI30-Beta1		x86	
MSI30-Beta2		x86	
MSI30-KB884016		x86	
MSI30-RC1		x86	
MSI30-RC2		x86	
MSI31-Beta		x86	
MSI31-RC1		x86	
MsJavaVM		x86	
NetMeeting		x86	
opsi-client-agent		x86	
OutlookExpress		x86	
PCHealth		x86	
SchedulingAgent		x86	
WebFldrs XP	9.50.6513	x86	
WIC		x86	
Windows XP Service Pack 3	20080414.031525	x86	
VMware Tools	8.4.5.10855	x86	

6.4 Preparation Windows Client Installation (OS Installation)

Unfortunately OPSI is only providing the basic Windows OS packages. This means no installation files are included in this package, only the files which help to support the automatized OS installation. In my opinion this makes sense for many reasons. First of all it would blow up the "install" folder on the depotserver (\\depotserver\opt_pcbin) since the installation file sizes of all operating systems would be something over 50 GB. Second, OPSI is open source and probably not eligible to provide Microsoft Windows products via the depotserver. Because my target is to install the Windows test client 2 with the Microsoft Windows XP Pro operating system I had to consequently insert the official Windows XP Pro CD and copy the i386 folder from the CD into the winxppro folder on the depotserver:



Name	Date modified
custom	5.9.2011 0:41
drivers	5.9.2011 0:41
i386	13.3.2009 13:01
opsi	5.9.2011 0:41

Copying 6 605 items (477 MB)

Copying 6 605 items (477 MB)

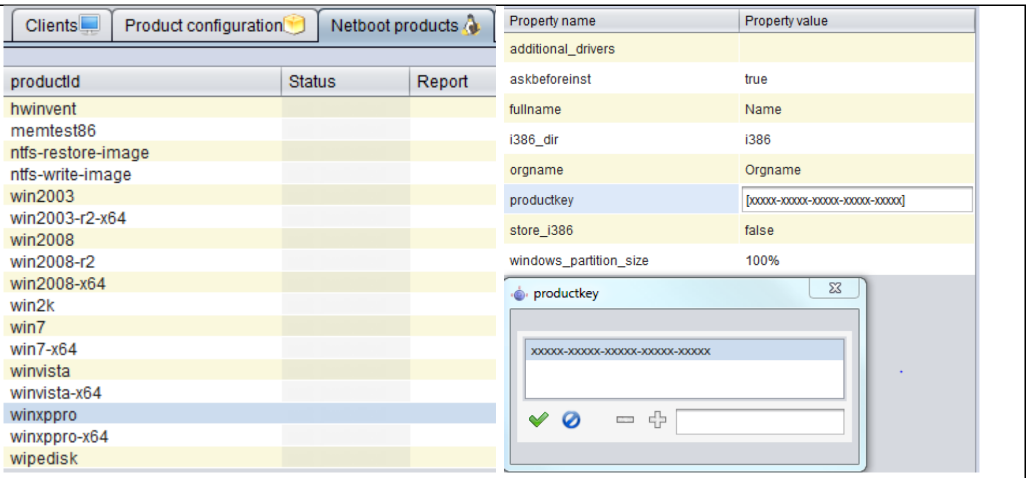
from **DVD Drive (D:) WXPVL...** to **winxppro** (Z:\install\winxppro)
Discovered 6 605 items (477 MB)...

⚠	The installation preparation process for Microsoft Windows Vista/2008/Win7 is a little bit more complex.

6.4.1 Set Windows activation key

The precondition for this step is to have the new OPSI client already captured and set up in the configuration editor. I have created the new Windows test client 2 in the administration interface and about to enter the Windows product key. I found the place to enter the product key like that: → chose client →tab netboot products → chose winxppro. Then enter the product key on the right panel. I confirmed by clicking the red tick.

ℹ



productId	Status	Report
hwinvent		
memtest86		
ntfs-restore-image		
ntfs-write-image		
win2003		
win2003-r2-x64		
win2008		
win2008-r2		
win2008-x64		
win2k		
win7		
win7-x64		
winvista		
winvista-x64		
winxppro		
winxppro-x64		
wipedisk		

Property name	Property value
additional_drivers	
askbeforeinst	true
fullname	Name
i386_dir	i386
orgname	Orgname
productkey	[xxxxx-xxxxx-xxxxx-xxxxx-xxxxx]
store_i386	false
windows_partition_size	100%

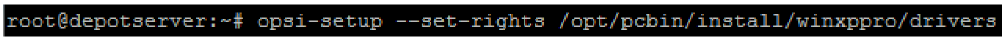
productkey

xxxxx-xxxxx-xxxxx-xxxxx-xxxxx

⚠	If you don't provide a Windows product key, you will be prompted during installation

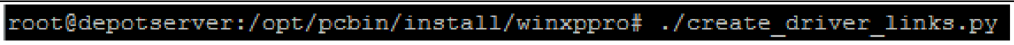
6.5 Integrate Missing Drivers

In this paragraph I want to demonstrate how to implement any driver into the automatized Windows installation routine. The big advantage of this method is to implement drivers for multiple clients. It is possible to implement drivers for a special group of clients. In my case I will include printer drivers for my Brother HL-1440 which is working on the local area network. All drivers have to be in a*.INF format. This additional driver has to be copied to the /opt/pcbin/install/winxppro/drivers /drivers/additional in a folder called "printer" which I created for this purpose. After each manipulation in the driver's folder it is advisable to execute following command:



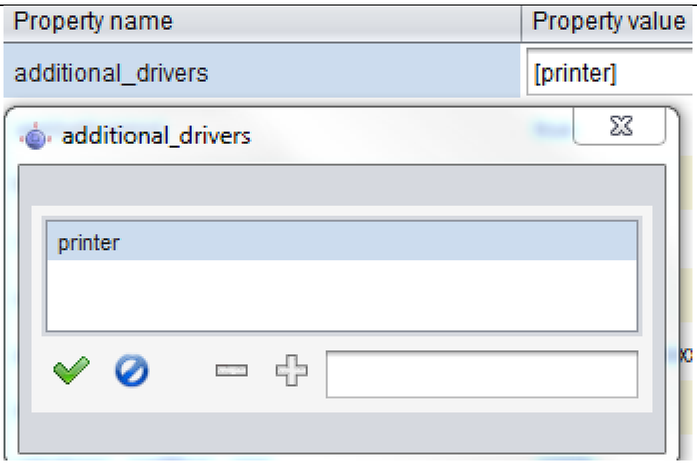
```
root@depotserver:~# opsi-setup --set-rights /opt/pcbin/install/winxppro/drivers
```

Afterwards I had to execute a script which crawls through all folders in the directory ../drivers/drivers and creates links from drivers to hardware:



```
root@depotserver:/opt/pcbin/install/winxppro# ./create_driver_links.py
```

Furthermore in the administration interface of OPSI I had to point the client to the correct folder which I just created. To do so I had to select the desired client (opsiclient 2), click on the operating system (winxppro) in the "netboot products" tab and just included the printer folder in the additional_drivers field:

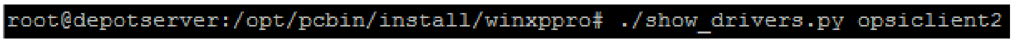


Property name	Property value
additional_drivers	[printer]

additional_drivers

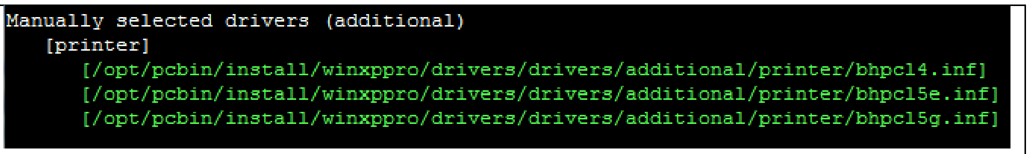
printer

Thereupon the drivers are now linked and included. To check import and links execute following command:



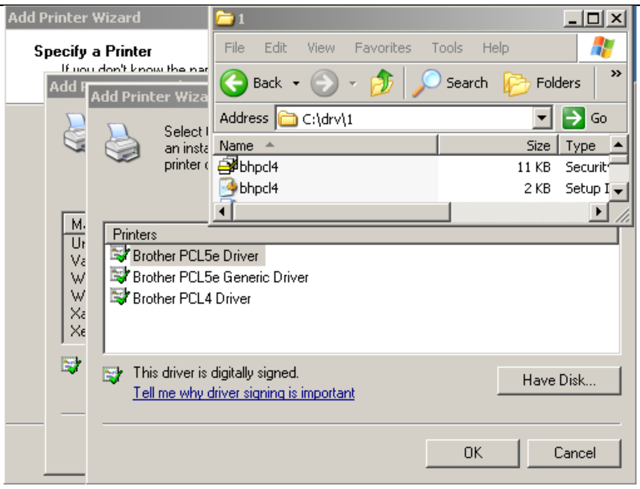
```
root@depotserver:/opt/pcbin/install/winxppro# ./show_drivers.py opsiclient2
```

The output of that file looks like that:



```
Manually selected drivers (additional)
    [printer]
        [/opt/pcbin/install/winxppro/drivers/drivers/additional/printer/bhpcl4.inf]
        [/opt/pcbin/install/winxppro/drivers/drivers/additional/printer/bhpcl5e.inf]
        [/opt/pcbin/install/winxppro/drivers/drivers/additional/printer/bhpcl5g.inf]
```

Finally, when the OS installation finished, I was able to choose on the opsiclient 2 the correct driver for my Brother printer by selecting the automatically implemented drivers:



Address: C:\drv\1

Name	Size	Type
bhpcl4	11 KB	Securit
bhpcl4	2 KB	Setup I

Printers:

- Brother PCL5e Driver
- Brother PCL5e Generic Driver
- Brother PCL4 Driver

This driver is digitally signed.
Tell me why driver signing is important

7 Testing

In this chapter I want to find out if the components and features of OPSI are working as intended. The goal is to test single OPSI features on their behaviour and compare the expected output to the effective output. For this purpose I have create a test plan which covers the basic and most important functions of OPSI. It includes installation and administrational test cases. The functions are tested by feeding them input and examining the output. To disregard is the internal program structure of OPSI.

7.1 Test Method

I have chosen the black-box testing method since I think this is the appropriate one for this case. Black-box testing describes a kind of system analysis whereas the testing procedure runs without knowledge of the inner functions (source code). This method is called black-box because the tester sees the system just as a black-box. The exact architecture and structure of the application is hidden from the tester. On the contrary the only thing tester is interested in is the visible behaviour of the system (interfaces, input forms, etc…). Usually this test is performed in the end before it's shipped to the customer.

SwtWiki, Blackbox-Test, 2007

7.2 Test Plan

The test plan can be viewed from the appendix. Attachment 7 contains all necessary information about the executed tests in detail. The results of the black-box tests can be found in the next chapter.

7.3 Test Result

The Automated Software Distribution, Automated OS Installation, Inventory and Administration test cases were all executed with a successful result. The reason for that is mainly the stable OPSI environment and well integration of the test clients. However the "Wake on Lan" test case couldn't be executed because there was no supported network interface card available. But I strongly believe that this test case would result successfully too.

8 Monitoring

Monitoring describes the systematical collection of relevant data with help of tools. Generally the data collection of certain kind of methods or processes includes observation, surveillance and log. It is recommended to record and monitor processes over a long period of time hence the information is more precise. The purpose of monitoring is to detect the bottle necks and take appropriate action to enhance the performance.

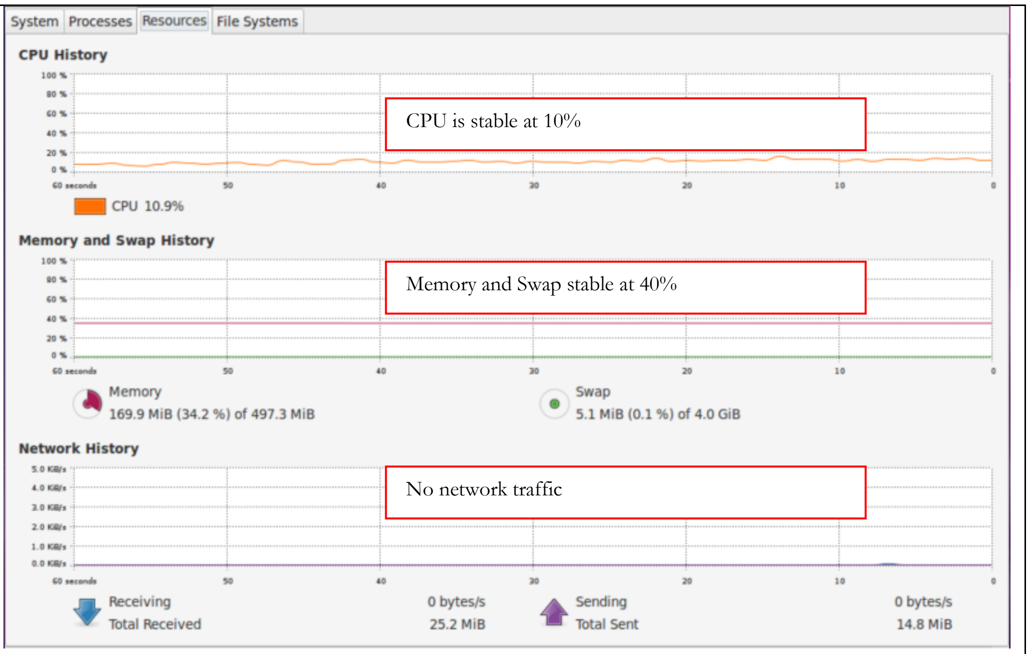
8.1 OPSI System Monitor

The best tool for administering the system load of your OPSI depotserver is the System Monitor. It is an integrated tool in Ubuntu which comes with every standard installation. The system monitor tool provides information about how the system hardware and software are working. What I am interested in are the resources. They show real-time information about CPU, memory usage and network traffic. The CPU History displays the running real-time CPU percentage utilization. Memory and Swap History displays two running real-time graphs, one for the amount of memory used by the user and one for the amount of swap space used by the user. Finally the Network History displays the amount of data sent and received from the network interfaces.

↗	Blum, 2009, Chapter 18

For this purpose I have analysed three events, which I would like to explain closer in the following paragraph. The first event is captured when there is no access whatsoever from or to the OPSI depotserver. The second event describes the remote start of the configuration manager editor. The last event shows the resource usage on the depotserver during a client software installation.

No access from or to the depotserver



CPU is stable at 10%

Memory and Swap stable at 40%

No network traffic

CPU 10.9%

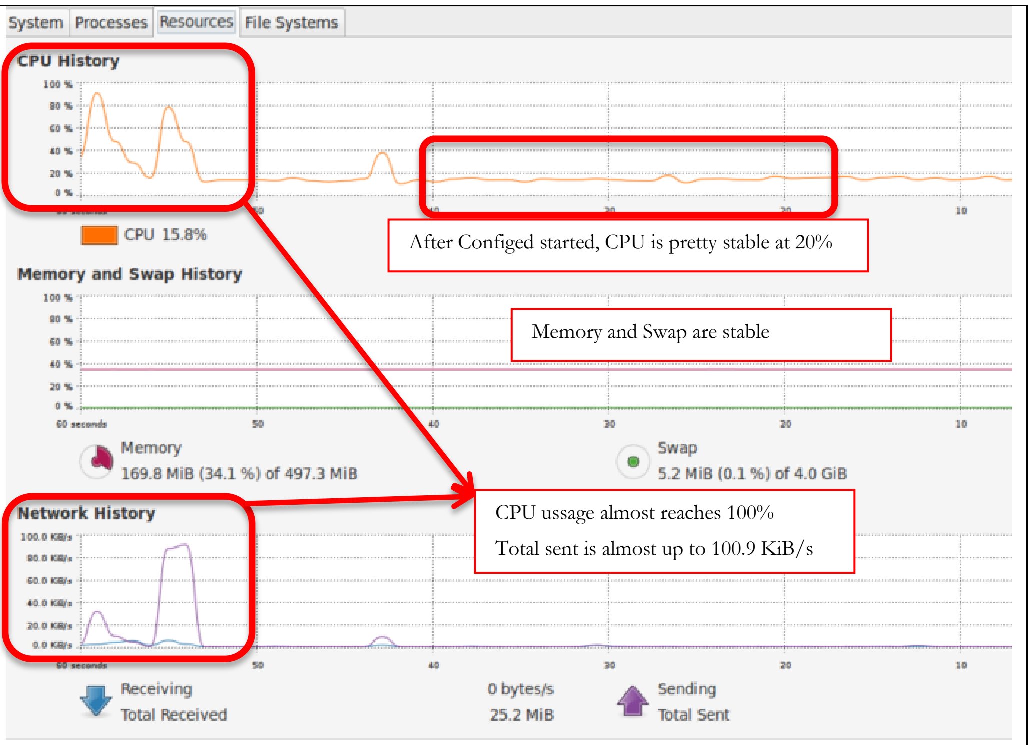
Memory
169.9 MiB (34.2 %) of 497.3 MiB

Swap
5.1 MiB (0.1 %) of 4.0 GiB

Receiving 0 bytes/s
Total Received 25.2 MiB

Sending 0 bytes/s
Total Sent 14.8 MiB

Configuration Editor Start



After Configed started, CPU is pretty stable at 20%

Memory and Swap are stable

CPU ussage almost reaches 100%
Total sent is almost up to 100.9 KiB/s

CPU 15.8%

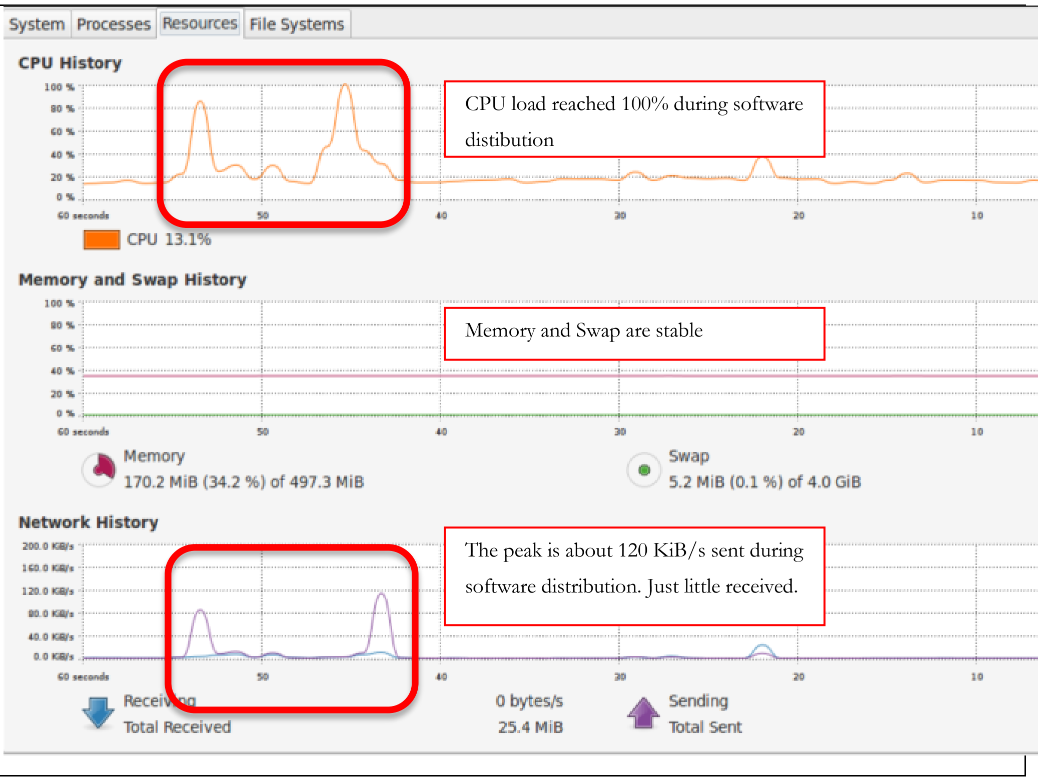
Memory
169.8 MiB (34.1 %) of 497.3 MiB

Swap
5.2 MiB (0.1 %) of 4.0 GiB

Receiving 0 bytes/s
Total Received 25.2 MiB

Sending
Total Sent

Software distribution/installation



CPU 13.1%

Memory
170.2 MiB (34.2 %) of 497.3 MiB

Swap
5.2 MiB (0.1 %) of 4.0 GiB

Receiving
Total Received

0 bytes/s
25.4 MiB

Sending
Total Sent

CPU load reached 100% during software distibution

Memory and Swap are stable

The peak is about 120 KiB/s sent during software distribution. Just little received.

8.1.1 Conclusion

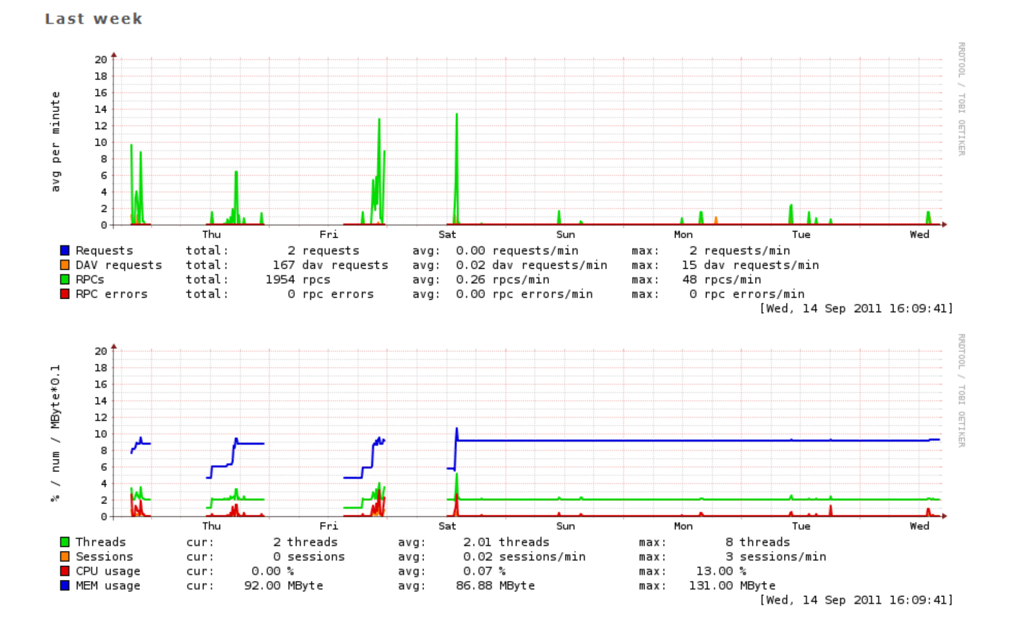
In the idle mode, OPSI depotserver doesn't require much system resources. However the start of the configuration editor lets the CPU jump higher. In my opinion this is not yet critically because every program start requires for a short time of period CPU usage. What concerns me more is the CPU usage during a software distribution. If many clients receive a new software distribution at the same time my OPSI depotserver wouldn't stand the load. Depending on the amount of your clients it is advisable to have the depotserver running on good hardware. Positively were the network load and the memory which was pretty stable and not critical at all.

8.2 OPSI Server Processes

The service opsiconfd, hosts the opsi-server-API (configuration editor) and is responsible for several other tasks. All traffic and communication between the clients and the server is handled by this service therefore it is the main OPSI service. The possibility to monitor this process and its load is an important matter. OPSI has a very informative tool included which lets you monitor the opsiconfd service. Enter following URL into your browser:

ℹ	https://depotserver:4447/info
⚠	Replace depotserver with your own OPSI-server name

OPSI info displays information about the opsiconfd load of the last hour /last day /last week /last month /last year in a chart. Furthermore it displays some additional information.

ℹ	

9 Summary

9.1 Result

Tools which are supporting the automatic software distribution and operating system installation become in larger networks more and more important nowadays. The server-based computing reduces the maintenance, support and administration of such networks. OPSI is one of those tools which eliminates the close-support. By close support I mean the filed-work – no one has to run from a client to another anymore since everything can be handled remotely. The biggest advantage is that OPSI is open-source therefore there are no immense licencing costs. Thanks to this thesis work I have a deep knowledge about the basic functions of OPSI. Therefore I will present my impression, experience and findings in this paragraph. First of all I was astonished of all the features and functions OPSI has to offer. OPSI is capable to administrate, manipulate, set-up and examine clients within seconds or minutes. It feels great to have so much power over multiple clients. However, also OPSI is not totally perfect. The installation process is a real challenge for IT beginners. There are many steps you have to take into consideration and an overall good unix/linux basic knowledge. Moreover it is not possible to integrate any other non-windows client at the moment. Even the integration of newer Windows operating systems like Vista or Win7 is trickier than the previous versions. I think the integration process will be simplified in next OPSI updates. Another important aspect is the performance. OPSI should run on a very strong hardware based server if there are many clients in the network. With only two clients, my depotserver already went to its maximum CPU capacity. In my opinion another minus of OPSI is the slow Java-based configuration editor. On the other hand this can be caused by the lack of ram/cpu my depotserver is running on. Finally the last negative aspect in my opinion is the fact that OPSI is open-source. Open-source is great, free of license fees and highly customizable but in general the whole application gives somehow an impression to me that it is still in beta state. It just feels and acts not that smoothly, especially the user interface could need a redesign. Additionally the cofinancing process which was invented by the developers to finance the open source development is not so convenient for testers since it forces you to pay certain functions.
The advantages of OPSI are obvious. The operating system installation time is massively shorter and safer because it is fully automated. Similarly the automated software

distribution time is decreased. Multiple clients are very easy to administer just with few clicks thanks to the configuration editor. Moreover the configuration editor displays well-arranged the hardware and software inventory of each client. Finally the result of this thesis includes an installation guide and a user guide. Those documents show how to install, administrate and maintain OPSI. With the installation guide it should be possible for every system administrator to install and perform OPSI on a Linux Server. With the user guide every beginner should be able to execute basic operations in OPSI. Readers who are interested in OPSI are welcome to use these documents to experience the first steps with OPSI.

9.1.1 Milestones

Phase	**Milestone**	**Date**	**Status**
1	Properly working configuration and environment.	1.8.2011	✓
2	The OPSI services are working properly and the documentation is finished	15.9.2011	✓
3	The user manual and testing is finished	15.10.2011	✓

9.1.2 Reached Goals

The target of this project was to create a working OPSI environment, which will be documented and tested properly. OPSI should adopt the inventory, installation and software distribution of the clients at its final stage. The goal was to install an OPSI server, which is able to accomplish those tasks. Following goals were defined by me:

- Properly working OPSI environment
- System documentation, which explains installation and management
- User documentation, which explains the common tasks
- Test plan, which covers the functionality

In retrospect I can ensure that all mentioned goals have been achieved. All practical work has been executed and the documentation is created. All goals were achieved in time.

9.2 Conclusion

The purpose of this thesis was to evaluate Open PC and Server Integration to demonstrate the capabilities and possibilities. Furthermore it should incite the use of OPSI to small or middle sized companies in Finland. The ability to have an automatic software distribution management, automatic OS installation and hardware and software inventories in one single tool is invaluable. The study revealed that OPSI has many advantages for a company. In conclusion I think following pros are worth mentioning. It is possible to downsize the client administration remarkably. The tasks are not solved locally anymore. Everything is centralised controlled by few people. Furthermore OPSI doesn't know boundaries therefore it will even work in a WAN. There is no version conflict of applications or patches since the client environment is unified by the depotserver.

Based on the thesis work and its result OPSI has also disadvantages even though it is free of charge. As a matter of fact the expenses of hardware equipment acquisition can be extremely high. Furthermore it usually takes some time to familiarize with the OPSI documentation and preparation and installation of the depotsever as well as the actual client rollout. Therefore it is recommendable to provide OPSI instruction courses for system administrators. Additionally, for each single application used in the company you have to create a package. Packetizing of each single application is rather time consuming. However despite of all this I can recommend OPSI to every small or middle sized company. The limit of simultaneously clients in OPSI 4.0.1 is somewhere around a few thousands. Therefore I wouldn't recommend big companies with more than 5000 employees (clients) to use OPSI since they are not able to administrate all clients and hence unable to take advantage of OPSI.

OPSI in the basic form may be used by anyone as it is. Being free and customisable offers a competitive advantage over commercial alternatives. However, you should not underestimate the initial setup effort or the maintenance overhead!

The study concludes that the benefits of OPSI have a remarkable impact on companies' client administration effort. The efficiency of administration, software distribution and installation of clients is tremendously higher with OPSI.

9.3 Reflection

When I was about to start the thesis I didn't know where to begin. OPSI as an application is quite challenging and has many functions and extensions. I decided to focus on the main part which was software distribution, inventory and OS installation. After creating a project plan I got more confident about the working process. The more I got involved in the topic the more I learned to appreciate it. Thanks to the milestones I forced myself to follow the time schedule. The biggest challenge was the installation of OPSI. Each installation step had to be perfectly executed and documented. Therefore the most time consuming work was the documentation. But I guess that's what it takes to deliver a serious IT documentation which can be followed by someone later on. All in all the thesis work was a good experience and I could learn a lot about project management, project execution and of course OPSI which is an amazing open-source tool.

In my opinion the importance of reflecting on what you are doing as part of the learning process is huge. Without reflective observation it is not possible to estimate if the work was in any sense successful. Therefore I want to review the thesis work process from beginning to end. I will mention the strengths and weaknesses I had during the process as well as areas of fear and improvement. Furthermore I point out my time management and how it can be improved. Finally I bring up the most important part of reflection which are the problems I was struggling with and how I solved them.

9.3.1 Strengths and Weaknesses

During my thesis work I encountered major and minor problems. Not only that but reflecting the whole process I can identify my personal strengths and weaknesses. Afterwards with the benefit of hindsight you are always wiser. Therefore I list my strengths and weaknesses during this project as followed:

Strengths	Motivation, had an interesting topic, had a detailed planning, had a clear structure, had a seamless execution, took screenshots while doing, had a good time management, captured on-going problems immediately.
Weaknesses	not updating work journal regularly, not documenting simultaneously

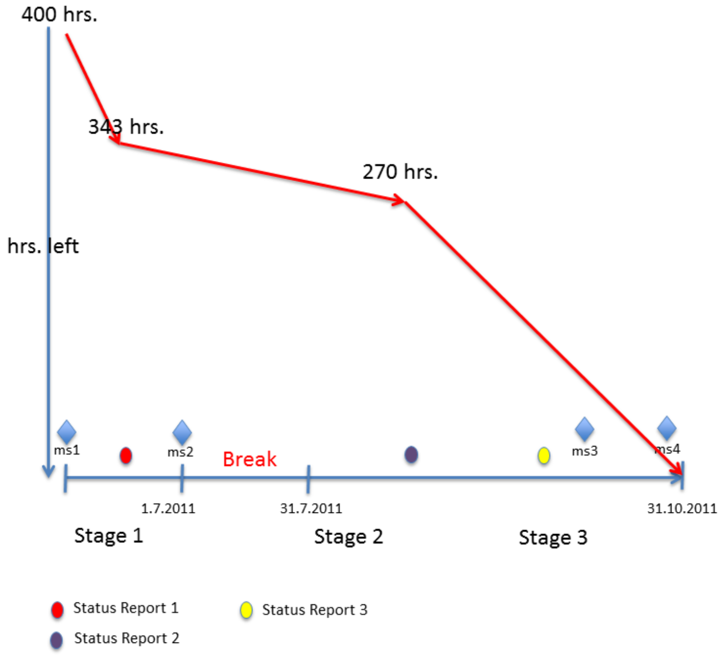
9.3.2 Fears and Improvements

Especially in the beginning I was worried if OPSI would be the appropriate topic for my thesis. By then, OPSI was still pretty much unknown for me. But more and more I learned the benefits this open source application. However I was concerned if I can follow the planned time schedule. What if I encounter a major problem? What if something goes wrong during installation, will I finish the project in time? Furthermore I wasn't sure if I meet the requirements and miss out any details in my thesis. But finally I think I have managed well and covered and described all basic functions of OPSI in my thesis; even in time.

What I would do differently next time is better multitasking. By that I mean carry out work at the same time. I handled the work more in a sequential process. Moreover the documentation of the practical work should have happened simultaneously. One major improvement for next time would be skipping the one month break. Instead of taking a good time off, it did more damage since it threw me out of the rhythm.

9.3.3 Time Management

Since I planned my time very carefully, there was no extra time nor any rush in the end. Thanks to the good planning and no major problems my time management was really good. The estimated time for the whole project was 410 hours. The actual hours used in this project were 417.5. This graph represents the burn down chart of my thesis work.



This table shows the hours used for each activity stage:

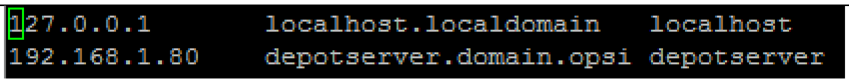
hrs/activity	Evaluation	Introduction	Planning	Preparation	Realisation	Testing	Monitoring	Summary
Planned	30	50	30	30	150	50	20	50
Actual	36	73.5	26	24	150.5	37	19	50
Total Planned	410 hours							
Total Actual	416,5 hours							

9.3.4 Problems and Solutions

13.8.2011, easy

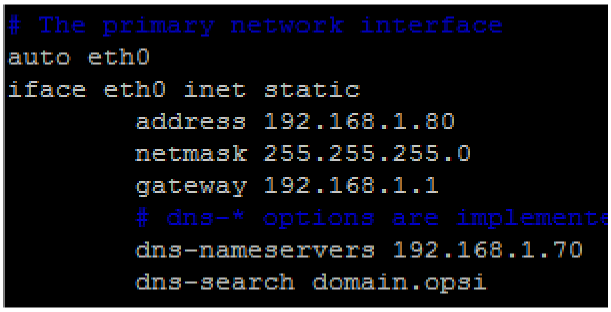
Problem	Content of Hosts and Interfaces was wrong.
Description	The problem occurred during the first installation steps of OPSI. The server had to be integrated in the local area network environment. Unfortunately the server didn't join the domain properly therefore I had to edit the hosts and interfaces file manually.
Solution	In the hosts file I had to provide the correct IP address and the FQDN for the depotserver:

<table>
<tr><td></td><td>



```
127.0.0.1         localhost.localdomain    localhost
192.168.1.80      depotserver.domain.opsi depotserver
```

I had to edit the interfaces file so it looks like this:



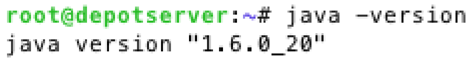
```
# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.1.80
        netmask 255.255.255.0
        gateway 192.168.1.1
        # dns-* options are implemente
        dns-nameservers 192.168.1.70
        dns-search domain.opsi
```

</td></tr>
</table>

16.8.2011, easy

<table>
<tr><td>Problem</td><td>How to check the installed Java version in Ubuntu.</td></tr>
<tr><td>Description</td><td>Since I was uncertain how to check the actual installed Java version on Ubuntu, I was stuck at that point. OPSI requires a Java version equal or bigger than 1.6.</td></tr>
<tr><td>Solution</td><td>

Thanks to Google I found the answer pretty fast:



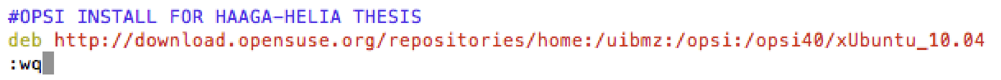
```
root@depotserver:~# java -version
java version "1.6.0_20"
```

</td></tr>
</table>

17.8.2011, moderate

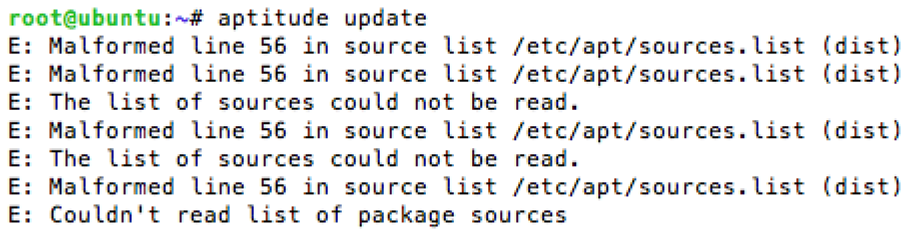
<table>
<tr><td>Problem</td><td>Spelling mistake in the sources list</td></tr>
<tr><td>Description</td><td>

To receive the OPSI repositories you have to add a download path to the sources.list. The wrong version looked like this:



```
#OPSI INSTALL FOR HAAGA-HELIA THESIS
deb http://download.opensuse.org/repositories/home:/uibmz:/opsi:/opsi40/xUbuntu_10.04
:wq
```

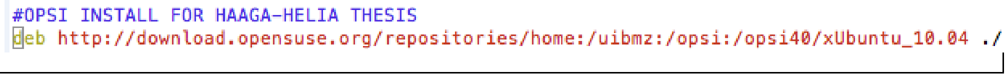
Therefore I got following errors after executing the update command:



```
root@ubuntu:~# aptitude update
E: Malformed line 56 in source list /etc/apt/sources.list (dist)
E: Malformed line 56 in source list /etc/apt/sources.list (dist)
E: The list of sources could not be read.
E: Malformed line 56 in source list /etc/apt/sources.list (dist)
E: The list of sources could not be read.
E: Malformed line 56 in source list /etc/apt/sources.list (dist)
E: Couldn't read list of package sources
```

</td></tr>
<tr><td>Solution</td><td>

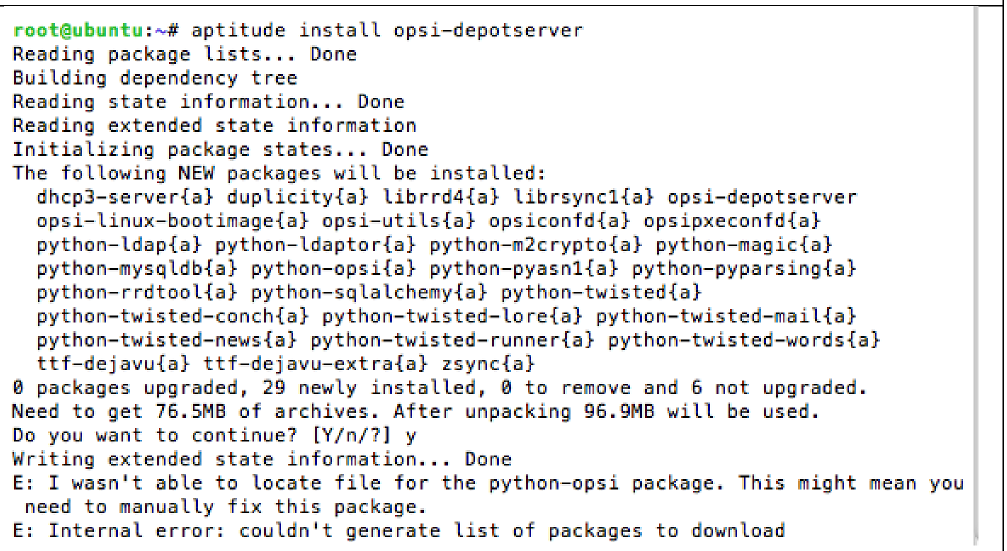
As tiny as it seems, the “./” was missing in the end of the path:



```
#OPSI INSTALL FOR HAAGA-HELIA THESIS
deb http://download.opensuse.org/repositories/home:/uibmz:/opsi:/opsi40/xUbuntu_10.04 ./
```

</td></tr>
</table>

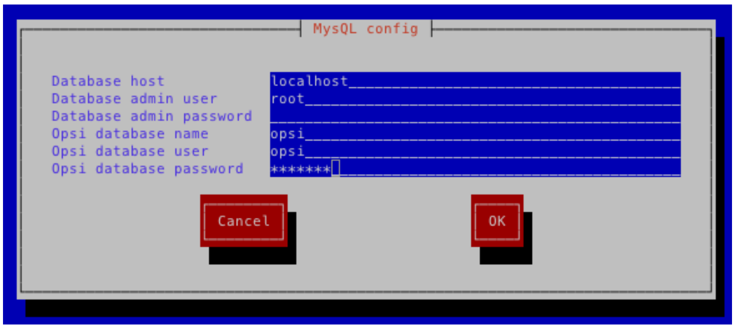
17.8.2011, easy

Problem	Internal Error occurred while executing the depotserver install command
Description	

```
root@ubuntu:~# aptitude install opsi-depotserver
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading extended state information
Initializing package states... Done
The following NEW packages will be installed:
  dhcp3-server{a} duplicity{a} librrd4{a} librsync1{a} opsi-depotserver
  opsi-linux-bootimage{a} opsi-utils{a} opsiconfd{a} opsipxeconfd{a}
  python-ldap{a} python-ldaptor{a} python-m2crypto{a} python-magic{a}
  python-mysqldb{a} python-opsi{a} python-pyasn1{a} python-pyparsing{a}
  python-rrdtool{a} python-sqlalchemy{a} python-twisted{a}
  python-twisted-conch{a} python-twisted-lore{a} python-twisted-mail{a}
  python-twisted-news{a} python-twisted-runner{a} python-twisted-words{a}
  ttf-dejavu{a} ttf-dejavu-extra{a} zsync{a}
0 packages upgraded, 29 newly installed, 0 to remove and 6 not upgraded.
Need to get 76.5MB of archives. After unpacking 96.9MB will be used.
Do you want to continue? [Y/n/?] y
Writing extended state information... Done
E: I wasn't able to locate file for the python-opsi package. This might mean you
 need to manually fix this package.
E: Internal error: couldn't generate list of packages to download
```

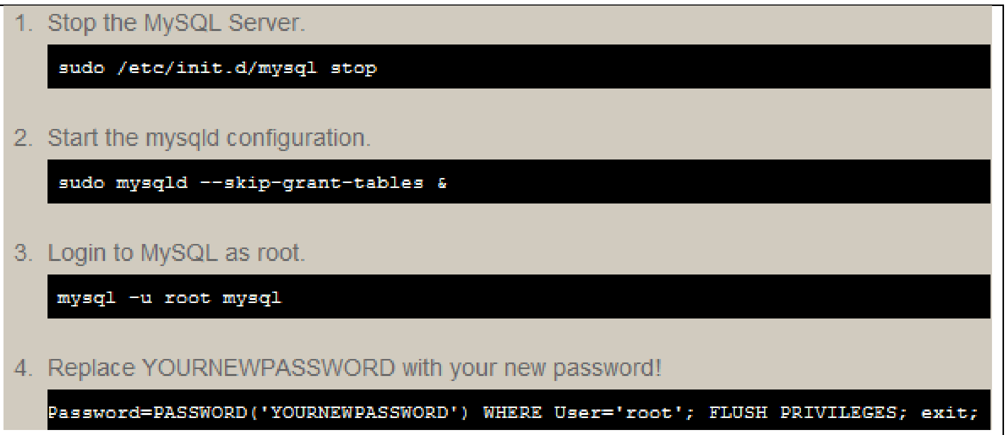
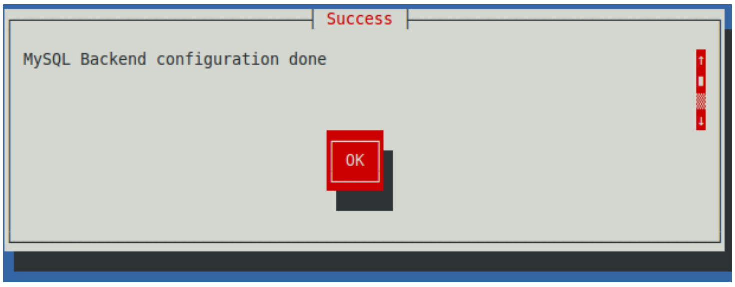
Solution	Just executed the “aptitude install opsi-depotserver” command again.

19.8.2011, difficult

Problem	MySQL backend configuration wasn’t possible because of wrong user rights.
Descrip-tion	After providing the CORRECT admin and database password, the MySQL config stated that the password for the user is wrong. 

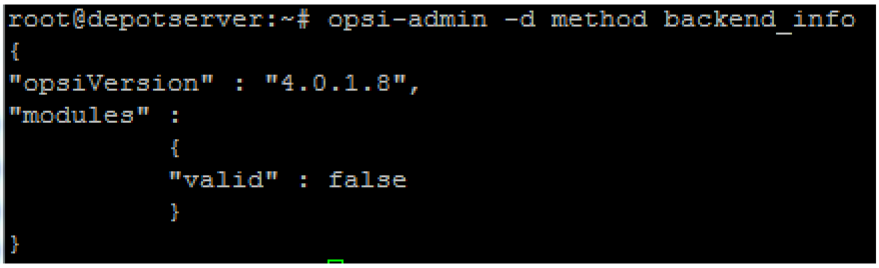
Database host localhost
Database admin user root
Database admin password
Opsi database name opsi
Opsi database user opsi
Opsi database password *******

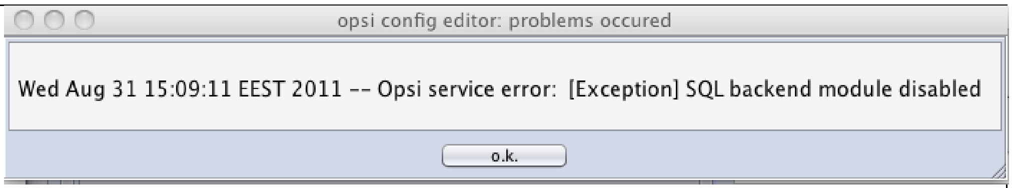
Solution	I had to manually change the MySQL root password on Ubuntu. I found the solution online. This is how I did it:

<table>
<tr><td></td><td>

1. Stop the MySQL Server.
<pre>sudo /etc/init.d/mysql stop</pre>
2. Start the mysqld configuration.
<pre>sudo mysqld --skip-grant-tables &</pre>
3. Login to MySQL as root.
<pre>mysql -u root mysql</pre>
4. Replace YOURNEWPASSWORD with your new password!
<pre>Password=PASSWORD('YOURNEWPASSWORD') WHERE User='root'; FLUSH PRIVILEGES; exit;</pre>
After that the OPSI MySQL config accepted the password:


<pre>MySQL Backend configuration done</pre>
</td></tr>
</table>

18.8.2011, moderate

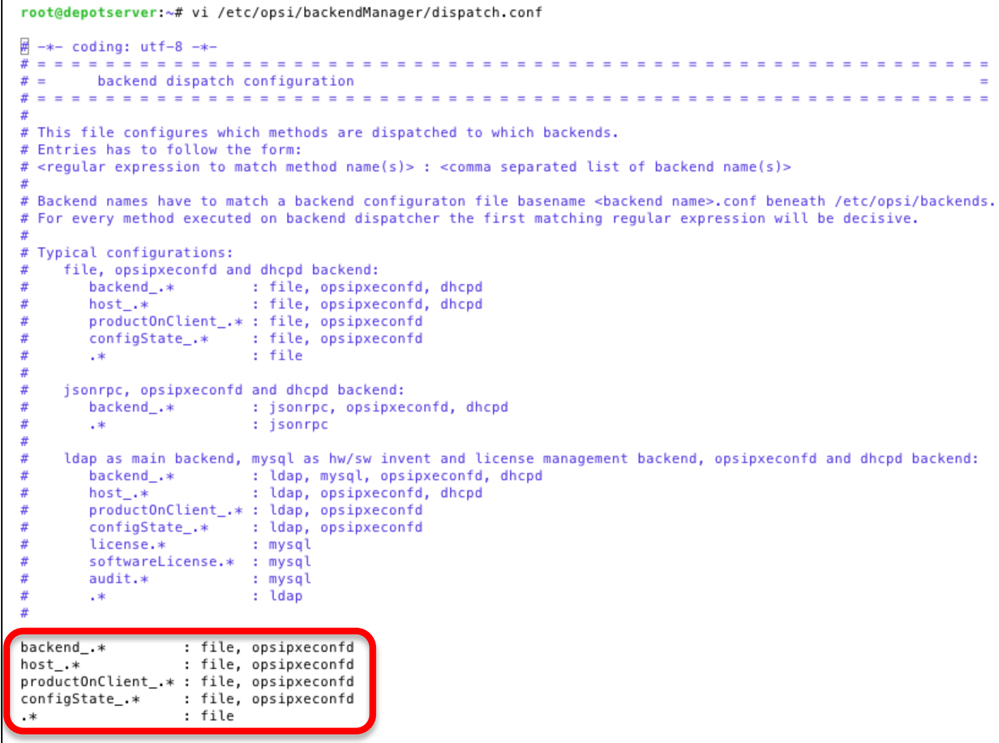
<table>
<tr><td>Problem</td><td>Use MySQL as backend is not working.</td></tr>
<tr><td>Description</td><td>This error is actually not a problem. In the first place I thought there must be something wrong with the MySQL as a backend module. After hours of trying out, editing and changing I found out that OPSI MySQL as backend is a cofounding project. That means it is not free at the moment until the founding state reached 100% (atm 60%). The implementation of this feature costs at the moment 2000 Euro. Therefore my modules are empty because I don’t have any paid ones.

<pre>root@depotserver:~# opsi-admin -d method backend_info
{
"opsiVersion" : "4.0.1.8",
"modules" :
          {
          "valid" : false
          }
}</pre>
After logging into the Configuration Editor it prompted with following error message:</td></tr>
</table>

<table>
<tr>
<td></td>
<td>


opsi config editor: problems occured

Wed Aug 31 15:09:11 EEST 2011 -- Opsi service error: [Exception] SQL backend module disabled

</td>
</tr>
<tr>
<td>Solution</td>
<td>
The only solution was to change dispatch.conf file in the backendManager to “file”. Hence, all information is stored in files instead of a DB now.



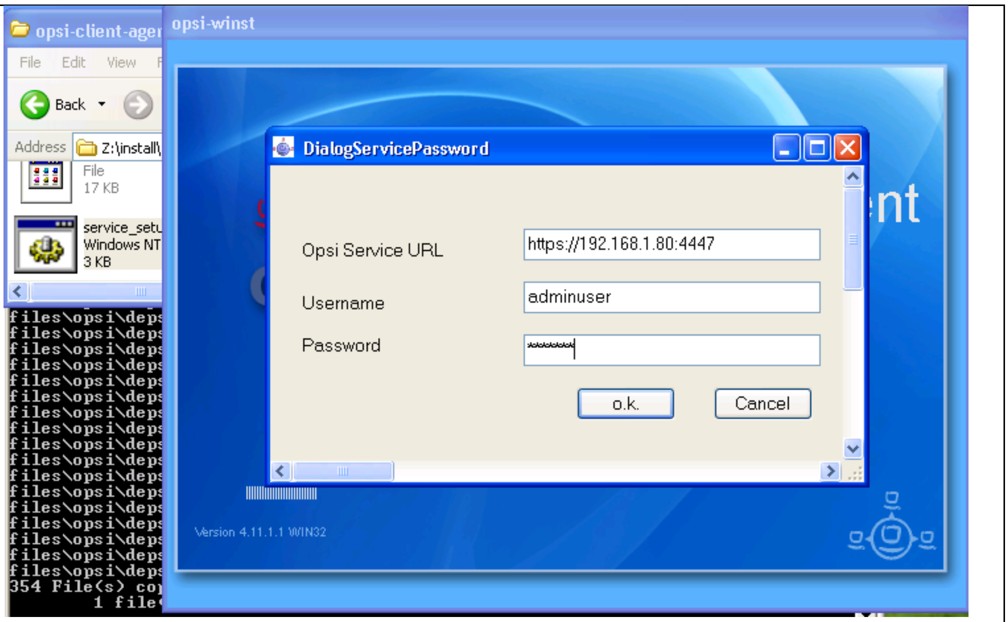
<pre>
root@depotserver:~# vi /etc/opsi/backendManager/dispatch.conf

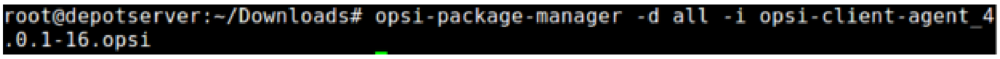
-*- coding: utf-8 -*-
=
=   backend dispatch configuration   =
=
#
This file configures which methods are dispatched to which backends.
Entries has to follow the form:
<regular expression to match method name(s)> : <comma separated list of backend name(s)>
#
Backend names have to match a backend configuraton file basename <backend name>.conf beneath /etc/opsi/backends.
For every method executed on backend dispatcher the first matching regular expression will be decisive.
#
Typical configurations:
file, opsipxeconfd and dhcpd backend:
backend_.*   : file, opsipxeconfd, dhcpd
host_.*   : file, opsipxeconfd, dhcpd
productOnClient_.* : file, opsipxeconfd
configState_.*   : file, opsipxeconfd
.*   : file
#
jsonrpc, opsipxeconfd and dhcpd backend:
backend_.*   : jsonrpc, opsipxeconfd, dhcpd
.*   : jsonrpc
#
ldap as main backend, mysql as hw/sw invent and license management backend, opsipxeconfd and dhcpd backend:
backend_.*   : ldap, mysql, opsipxeconfd, dhcpd
host_.*   : ldap, opsipxeconfd, dhcpd
productOnClient_.* : ldap, opsipxeconfd
configState_.*   : ldap, opsipxeconfd
license.*   : mysql
softwareLicense.* : mysql
audit.*   : mysql
.*   : ldap
#

backend_.*   : file, opsipxeconfd
host_.*   : file, opsipxeconfd
productOnClient_.* : file, opsipxeconfd
configState_.*   : file, opsipxeconfd
.*   : file
</pre>
</td>
</tr>
</table>

24.8.2011, difficult

Problem	Connection to depotserver while running the service_setup was not possible.
Description	To include an existing windows client to OPSI you have to run the service_setup script on the client. Unfortunately I was always stuck at one point when it tried to establish the connection to the depotserver because the username and password didn’t work. Any user in the group “opsiadmin” can establish a connection to the depotserver. My user adminuser is in this group; however I have spent days to reset/set/update the username and password. Nothing seemed to work.

	
	 opsi-client-agen
	File Edit View
	Back
	Address Z:\install\
	File 17 KB
	service_setu Windows NT 3 KB
	opsi-winst
	DialogServicePassword
	o.k.
	Cancel
	Version 4.11.1.1 WIN32
	Opsi Service URL: https://192.168.1.80:4447
	Username: adminuser
	Password: ********
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`files\opsi\deps`
	`354 File(s) co`
	`1 file`

Solution	Finally I came across a good hint in some forum I read. Obviously the "vcredist" tool was missing from the "minimal OPSI packages". I had to download this additional package from: http://download.uib.de/opsi4.0/experimental/opsi-client-agent_4.0.1-16.opsi
	After execution following command, the service_setup script was working flawless:
	
	`root@depotserver:~/Downloads# opsi-package-manager -d all -i opsi-client-agent_4`
	`.0.1-16.opsi`

9.4 Outlook

OPSI is already a widespread application in Europe. Unfortunately no Finnish company has discovered the advantages of OPSI yet. The potential of this application is already immense. In my opinion with a major breakthrough will occur when the possibility to integrate Linux clients into OPSI will happen. Let's think straight forward – Even a MAC OS X client implementation sounds feasible and realistic to me. Although these intentions aren't mentioned by the developers yet. Up to now it is possible to integrate several thousands of clients. I can imagine that the numbers of clients will rise in the upcoming updates of OPSI. I am sure OPSI will grow in the future and additional functions are already planned to be added. One of those functions is monitoring. At the moment OPSI is not a monitoring tool, therefore it is impossible to observe servers and clients. The target is to implement the monitoring feature with help of some additional products which can be easily integrated into OPSI. OPSI is a powerful tool, in the basic form, free of charge and yet highly customisable - for this reason I predict a bright future.

9.5 Enhancement

At the moment OPSI is available in version 4.0.1. Future enhancements are planned to be included in the next update. The biggest innovation is probably going to be the integration of Linux clients. Therefore it will be possible to install Linux operating system on clients as well as software distribution. Furthermore it is planned to create inventory extensions. To put it another way the inventory will be cross platform to support Linux clients and additional network components. Interoperability is a key word which describes OPSI´s openness to connect with other system tools. There are plans to implement monitoring as well as data exchange possibilities. Finally the last planned enhancement for the upcoming update will be load balancing. Load balancing will distribute the work load of the depotserver across multiple computer clusters which will result in better performance.

	uib gmbh, open source client management, 2011

Bibliography

OPSI.org. 2011. Features. URL: http://www.opsi.org/features/. Quoted: 15.6.2011.

Oertel, D. CeBit 2011. Extensions. URL: http://www.techcast.com/events/cebit11/di11/. Quoted: 2.8.2011.

UiB GMBH. 2011. opsi Getting Started. URL: http://download.uib.de/opsi_stable/doc/opsi-getting-started-stable-en.pdf. Quoted 1.6.2011

NIXCRAFT. 2006. Ubuntu Linux OpenSSH Server installation and configuration. URL: http://www.cyberciti.biz/faq/ubuntu-linux-openssh-server-installation-and-configuration/. Quoted: 13.8.2011

VIVEK. 2006. Linux Set or Change User Password. URL: http://www.cyberciti.biz/faq/linux-set-change-password-how-to/. Quoted: 13.08.2011

VIVEK. 2007. Ubuntu Linux restart network. URL: http://www.cyberciti.biz/faq/ubuntu-restart-start-stop-networking-service-howto/. Quoted: 13.08.2011

Wikipedia. 2011. Samba (software). URL: http://en.wikipedia.org/wiki/Samba_(software). Quoted: 15.08.2011

Wikipedia. 2011. Trivial File Transfer Portocol. URL: http://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol. Quoted: 17.08.2011

SwtWiki. 2007. Blackbox-Test. URL: http://www.imn.htwk-leipzig.de/~weicker/pmwiki/pmwiki.php/Main/Blackbox-Test. Quoted: 04.09.2011

Blum, R. 2009. Ubuntu Linux Secrets

UiB GMBH. 2011. Open Source Client Management. URL: http://www.netways.de/uploads/media/Erol_UEluekmen_opsi_Open_Source_Management.pdf. Quoted: 25.09.2011

HAAGA-HELIA
University of Applied Sciences

Open PC Server Integration 4.0.1

Cédric Zwimpfer

User documentation

BiTE

2011

Table of contents

10 Management Summary

10.1 Documentation structure

The documentation of my thesis is divided in following parts:

Document	Content
System documentation	Project Management, Evaluation, Planning, Preparation, Realisation, Monitoring, Summary
User documentation	Basic OPSI management operations
Work journal	Record daily workload
Test plan	Testing, execute different tests
Project plan	Project planning
Summary report	Sum up the thesis

10.2 Abbreviations

API	Application Programming Interface
CPU	Central Processing Unit
DB	Database
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
GB	Gigabyte
GHZ	Giga Hertz
GPL	General Public License
GUI	Graphical User Interface
HW	Hardware
IP	Internet Protocol
LAN	Local Area Network
MB	Megabyte
NIC	Network Interface Controller/Card
OPSI	Open PC Server Integration

OS	Operating System
PXE	Preboot Execution Environment
RAM	Random Access Memory
SSH	Secure Shell
SW	Software
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
URL	Uniform Resource Locator
VM	Virtual Machine
VPN	Virtual Private Network
WAN	Wide Area Network
wInst	Windows Installer
WLAN	Wireless Local Area Network
WMI	Windows Management Instrumentation

11 Introduction

11.1 Purpose of this document

The purpose of this document is to describe the basic functions of OPSI. The intention is to have a valuable guide for OPSI beginners. This guide consists of an overview about OPSI and the most used functions and methods. All steps have been practically tested by me.

11.2 Validity

The content of this document refers to the OPSI version 4.0.1 and my test home network environment. The user documentation is based on the official OPSI documentation

↗	uib gmbh, OPSI Getting Started, 2011

11.3 Used Symbols

⚠	Attention
⛔	Error
ℹ	Information
>_	Console commands
↗	References

12 OPSI Basics

12.1 How OPSI works

↙	OPSI.org, Features, 2011

12.1.1 Automated OS- Installation

During the boot process a PXE- Image is loaded on the windows client, which starts a hardware inventory. All necessary installation files and drivers will be transmitted from the OPSI depot server to the windows client. Afterwards an unattended windows installation will be executed.

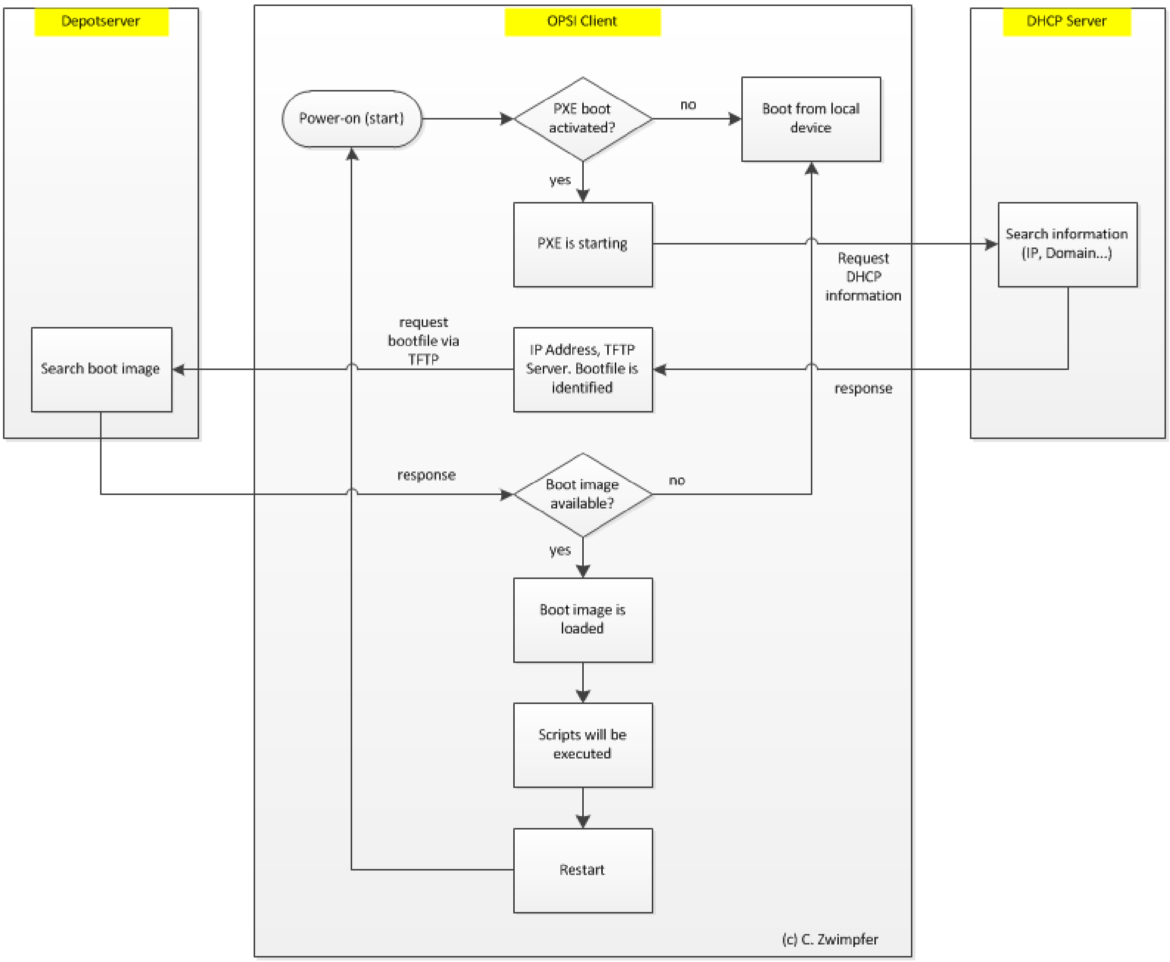
12.1.2 Automated Software Distribution

The OPSI-Client Agent is responsible for the automated software distribution on the clients. This service, which is installed on the client side, connects to the OPSI depotserver each system start and checks if there are any new packages available for the client. In case there are some packages ready to be installed, a small open source program called "Windows Installer" (wInst) will install those packages automatically. The graphical user interface (GUI) is locked to the user during the whole process.

12.1.3 Hard- and Software Inventory

The OPSI inventory is based on the Windows Management Instrumentation (WMI). All client data is gathered by special software packages and saved in MySQL database. All gathered information is accessible through the OPSI ad-ministration interface.

12.2 OPSI Network Boot Process



uib gmbh 2011, opsi-handbuch-stable-de.pdf, Chapter 7.2

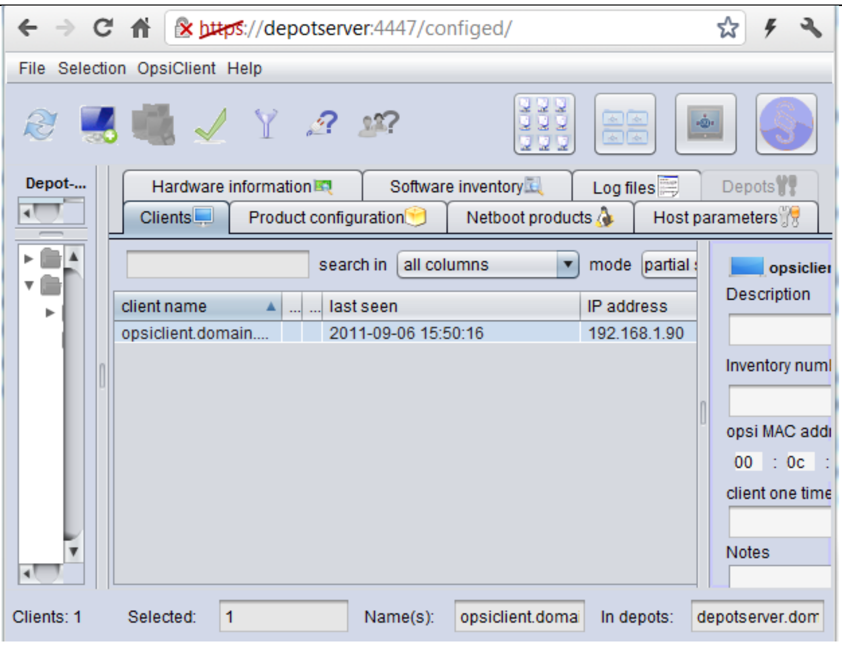
12.3 Boot from local device

Not always is the PXE Network boot a possibility to set up new clients. Moreover the boot image can be located on a CD/DVD. The client will load the provided boot image and starts to execute the scripts. This alternative way is good in such occasions when it is not possible for certain reasons to boot from PXE. If there is no boot image available (not from PXE, and neither from CD/DVD) then the client will boot from the local hard disk. During the boot process and before the user login, OPSI checks with help of the configuration data if there is any software to install and what kind of software there is to install. The installation program wInst (Windows installer) will be executed if there are any software packages to be installed. The scripts and installation data is stored on the OPSI depotserver. wInst will access that data during the software installation. During the installation the user interface will be locked.

↗	uib gmbh 2011, opsi-handbuch-stable-de.pdf, Chapter 7.2.6

13 OPSI Administration Interface

The OPSI administration interface is pretty much self-explanatory. You can reach it by entering following URL into your browser: https://depotserver:4447/configed



https://depotserver:4447/configed/

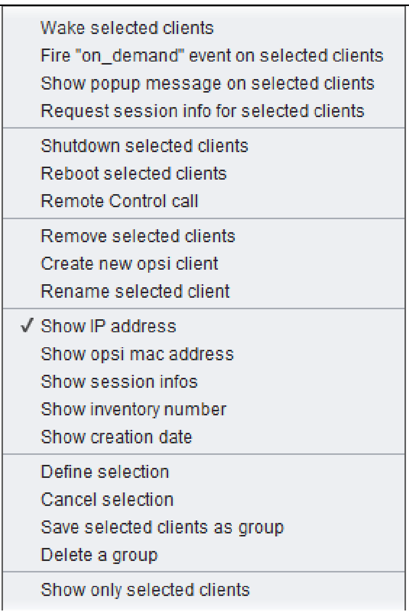
client name	...	...	last seen	IP address
opsiclient.domain....			2011-09-06 15:50:16	192.168.1.90

search in: all columns mode: partial

opsi MAC addr: 00 : 0c :

Clients: 1 Selected: 1 Name(s): opsiclient.doma In depots: depotserver.dom

Anyways I want to point out some basic features. In the client menu bar "OpsiClient" or the context-menu you can access and start some client specific operations:



13.1 Wake client

It is possible to send a WakeOnLan-command to selected clients. WOL makes it possible to start a powered-off computer via the network card. The client receives the signal from OPSI and thus starts the boot process.

13.2 Show pop-up message

You have the possibility to send messages to selected clients. Provide the content of your message in the given textbox and send it by pressing the red tick. The target client receives the message instantly.

13.3 Shutdown

You can send the signal to shut down the selected client(s). The client will shut down instantly without asking for a confirmation.

13.4 Reboot

You can send the signal to reboot the selected client(s). The client will reboot instantly without asking for a confirmation.

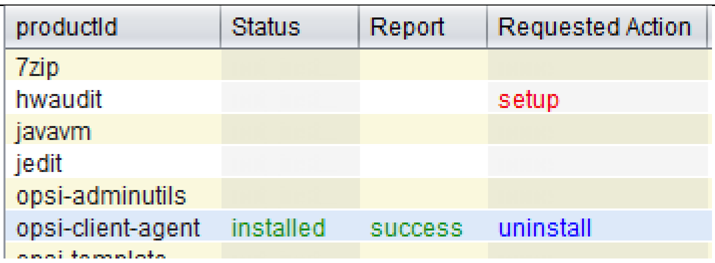
13.5 Remote control call

This is a very powerful tool to remote control call software. It can be used to execute any command that the target operating system offers. There is already an example integrated which can be used to send a ping to the selected client.

14 Install Products (Software)

The installation of new products on clients is very easy. In the administration interface you just have to selected the client and change to the "Product configuration" tab. From there, every single product (software) can be installed and uninstalled. Choose your product and change the "Requested Action" to either "setup" for installation or uninstall for uninstallation. The client will start to install the assigned software after the reboot.

ℹ



productId	Status	Report	Requested Action
7zip			
hwaudit			setup
javavm			
jedit			
opsi-adminutils			
opsi-client-agent	installed	success	uninstall
opsi-template			

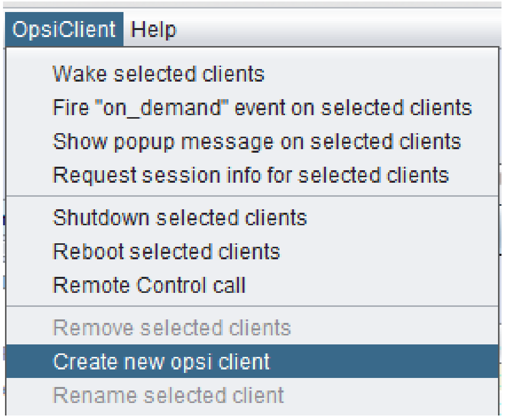
⚠ Dont forget to confirm the input by clicking the red tick

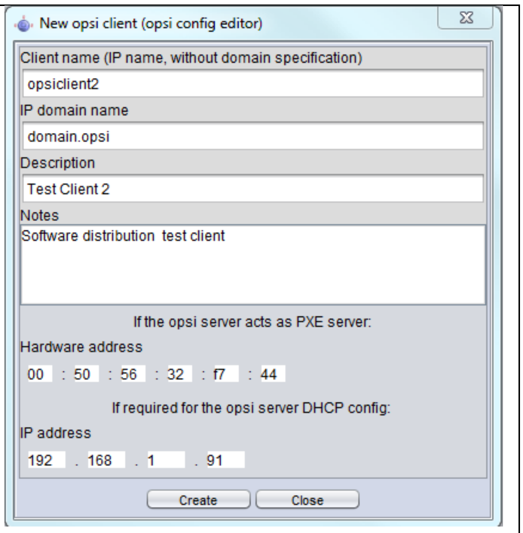
15 OPSI Configuration Editor

A new physical client in the network calls for the creation of a new OPSI client. Therefore every new client has to be included into OPSI to be able to administrate it. In the following chapter I explain the steps to include a new physical client into the OPSI system.

15.1 Add new client

On the depotserver administration interface, in the drop-down menu bar, you will find the command "Create new opsi client". Afterwards a mask will pop up to enter all required information to create a new OPSI client.


OpsiClient
Help
Wake selected clients
Fire "on_demand" event on selected clients
Show popup message on selected clients
Request session info for selected clients
Shutdown selected clients
Reboot selected clients
Remote Control call
Remove selected clients
Create new opsi client
Rename selected client


New opsi client (opsi config editor)

Client name (IP name, without domain specification)
opsiclient2

IP domain name
domain.opsi

Description
Test Client 2

Notes
Software distribution test client

If the opsi server acts as PXE server:

Hardware address
00 : 50 : 56 : 32 : f7 : 44

If required for the opsi server DHCP config:

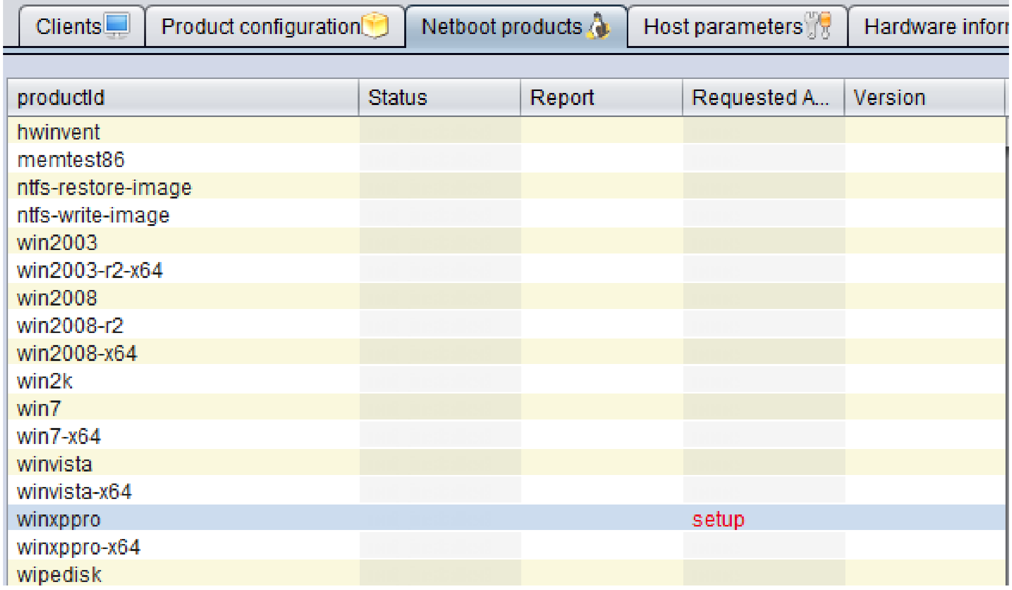
IP address
192 . 168 . 1 . 91

Provide all possible information to ensure a flawless integration of the OPSI client. Be sure to provide at least the client name, the IP domain name and the MAC-Address. Press "Create" to finish the process.

⚠	The MAC-Address is needed for the PXE boot. Be sure to put the correct MAC-Address otherwise the client wont boot from PXE. If you have already decided the IP-Address, you are free to provide that too, although it is not required!

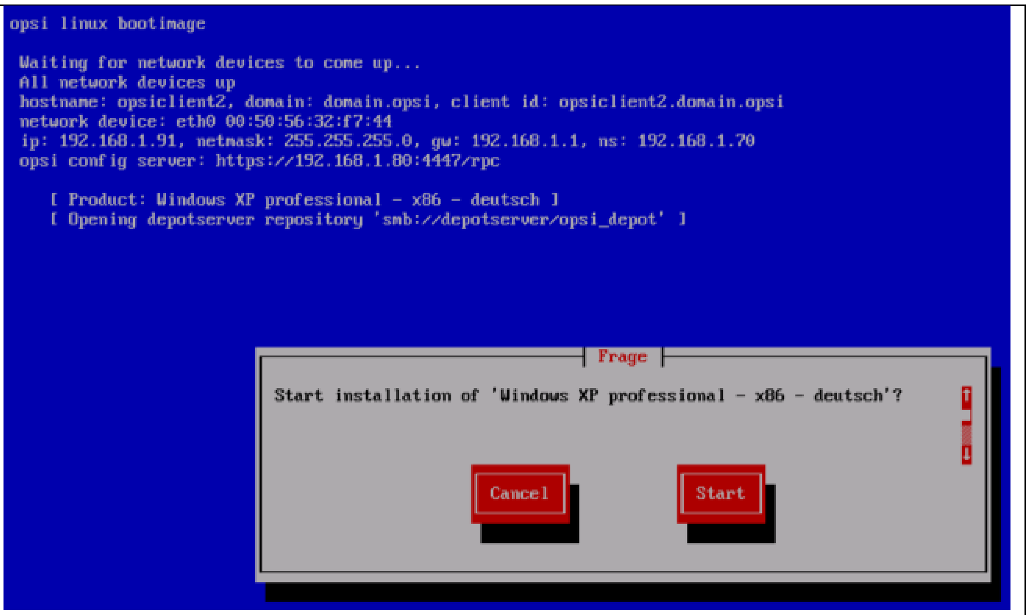
15.2 Start automated OS installation

Since now the new client is now integrated into OPSI it is about time to assign an operating system to it. Change to the Netboot products tab where you find all available operating systems. To initialize the installation, select your desired OS and change the "Requested Action" to setup:



productId	Status	Report	Requested A...	Version
hwinvent				
memtest86				
ntfs-restore-image				
ntfs-write-image				
win2003				
win2003-r2-x64				
win2008				
win2008-r2				
win2008-x64				
win2k				
win7				
win7-x64				
winvista				
winvista-x64				
winxppro			setup	
winxppro-x64				
wipedisk				

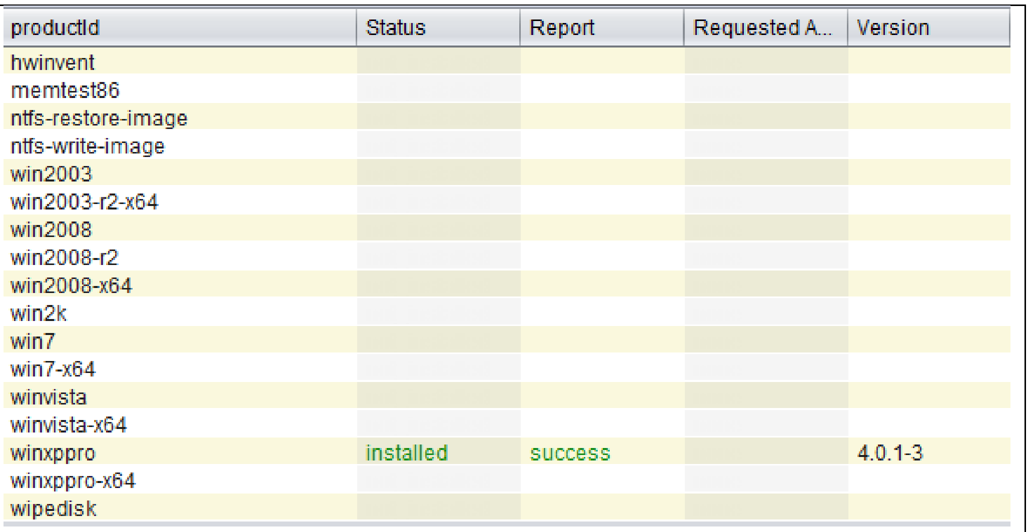
After confirming your action by clicking the red tick you reboot the client. The client will boot from PXE (Set BIOS boot priority). The client loads the assigned boot image from the depotserver. The unattended installation of Windows XP will begin after you pressed the Start button:



```
opsi linux bootimage

 Waiting for network devices to come up...
 All network devices up
 hostname: opsiclient2, domain: domain.opsi, client id: opsiclient2.domain.opsi
 network device: eth0 00:50:56:32:f7:44
 ip: 192.168.1.91, netmask: 255.255.255.0, gw: 192.168.1.1, ns: 192.168.1.70
 opsi config server: https://192.168.1.80:4447/rpc

    [ Product: Windows XP professional - x86 - deutsch ]
    [ Opening depotserver repository 'smb://depotserver/opsi_depot' ]
```

Start installation of 'Windows XP professional - x86 - deutsch'?

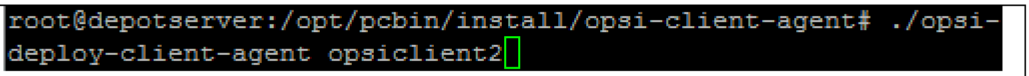
The whole installation is automated and takes about 20 minutes. After the installation is done, you will find the installation report in the administration interface:



productId	Status	Report	Requested A...	Version
hwinvent				
memtest86				
ntfs-restore-image				
ntfs-write-image				
win2003				
win2003-r2-x64				
win2008				
win2008-r2				
win2008-x64				
win2k				
win7				
win7-x64				
winvista				
winvista-x64				
winxppro	installed	success		4.0.1-3
winxppro-x64				
wipedisk				

15.3 Add existing client with opsi-deploy-client

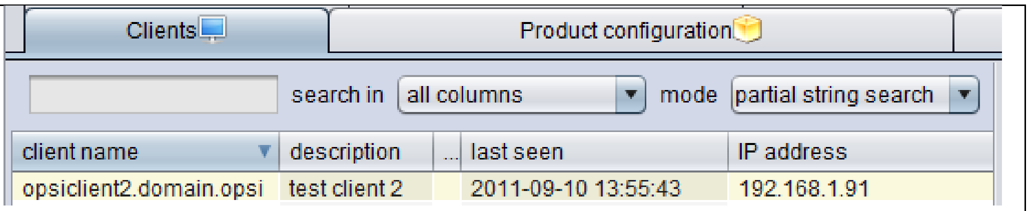
As in the system documentation written, there is a script based client integration method called “service_setup”. Fortunately this is not the only way how to integrate clients into OPSI; think about integrating 1000 clients. The script method would take ages to include all clients since you have to operate on the client side (for example mount share). I want to introduce an alternative way to include clients where Windows XP is already installed. The tool “opsi-deploy-agent” suits that purpose. Once executed, it will integrate the clients, which are defined in the list, into OPSI. To install “opsi-deploy-agent”, it is required to have a local user with admin rights, an open c$ share, an open admin$ share and you have to know the NetBIOS-Name of the client. To run the “opsi-deploy-agent” you have to change to directory “/opt/pcbin/install/opsi-client-agent” and execute following command:



```
root@depotserver:/opt/pcbin/install/opsi-client-agent# ./opsi-
deploy-client-agent opsiclient2
```

It is possible to provide a list of clients for deployment

After the deployment, the client(s) are integrated into OPSI:

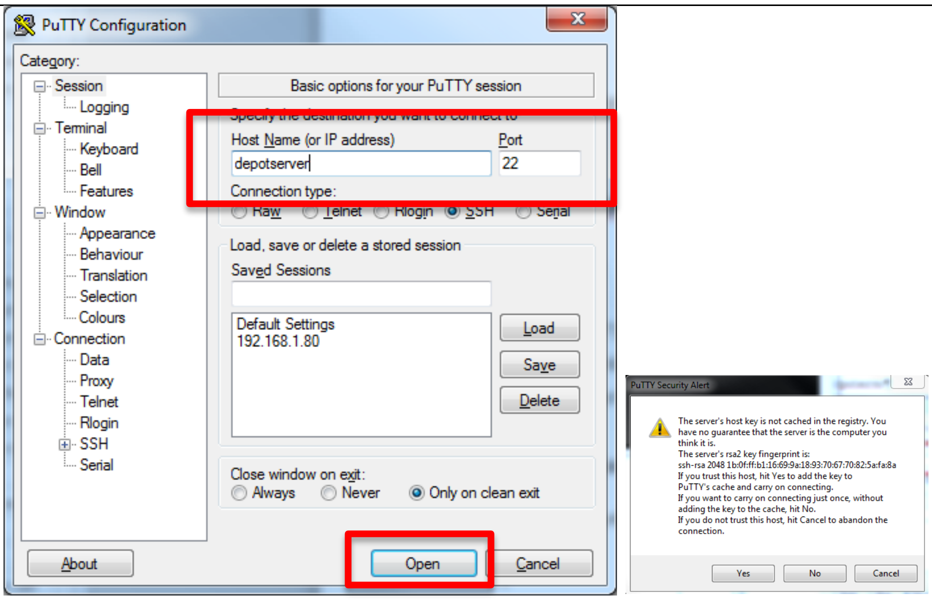


client name	description	...	last seen	IP address
opsiclient2.domain.opsi	test client 2		2011-09-10 13:55:43	192.168.1.91

16 OPSI Depotserver Administration

16.1 Putty

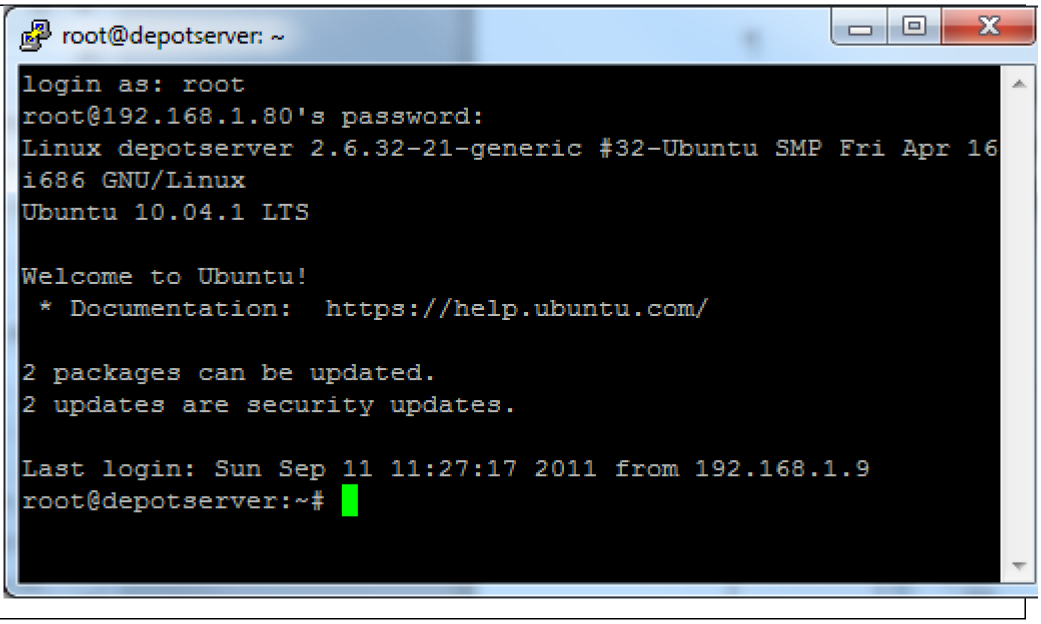
Putty is a very powerful and handy application to administrate the depotserver remotely. After downloading Putty from the internet you can start the program and create a session. Enter the OPSI hostname or IP Address into the “Host Name” field. Press “Open” to connect to the OPSI depotserver and confirm the warning with “YES”.



Host Name (or IP address): depotserver
Port: 22
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial
Saved Sessions:
Default Settings
192.168.1.80
Close window on exit: ○ Always ○ Never ● Only on clean exit

The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 2048 1b:0f:ff:b1:16:69:9a:18:93:70:67:70:82:5a:fa:8a
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

The Putty window is opened and asking for the user and password. After confirming the password a Linux-Shell will appear. The connection establishment to the depotserver is complete. The configuration of the OPSI depotserver can begin!

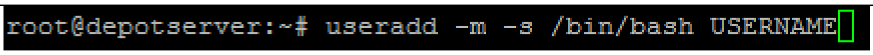


```
login as: root
root@192.168.1.80's password:
Linux depotserver 2.6.32-21-generic #32-Ubuntu SMP Fri Apr 16
i686 GNU/Linux
Ubuntu 10.04.1 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

2 packages can be updated.
2 updates are security updates.

Last login: Sun Sep 11 11:27:17 2011 from 192.168.1.9
root@depotserver:~#
```

16.2 Administrate group opsiadmin and pcpatch

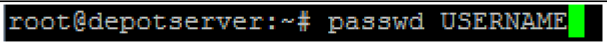
The administration of OPSI is only allowed to those users which are members of the UNIX group "opsiadmin". Users who are eligible to packetize products (makeproductfile command), install (opsiinst command) or manually edit configuration data have to be additionally in the UNIX group "pcpatch". Here are examples how to create users and add users to certain groups in Linux:

Create User

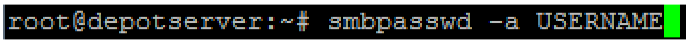


```
root@depotserver:~# useradd -m -s /bin/bash USERNAME
```

Set UNIX and SAMBA password

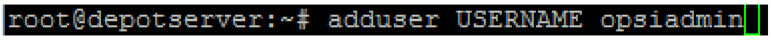


```
root@depotserver:~# passwd USERNAME
```



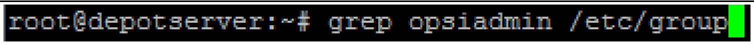
```
root@depotserver:~# smbpasswd -a USERNAME
```

Move user to group



```
root@depotserver:~# adduser USERNAME opsiadmin
```

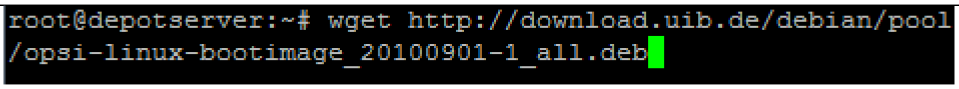
Display user in group



```
root@depotserver:~# grep opsiadmin /etc/group
```

16.3 Update OPSI boot image

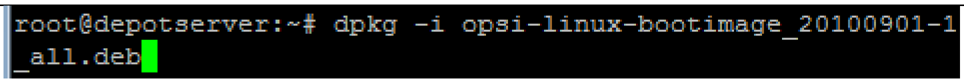
Especially with new clients it is possible that the boot image doesn't recognise their network cards. In that situation an update of the boot image version is most of the time helpful. This is how you download a new version of the boot image:



```
root@depotserver:~# wget http://download.uib.de/debian/pool
/opsi-linux-bootimage_20100901-1_all.deb
```

This is at the moment the latest version (01-Sep-2010 16:51):
opsi-linux-bootimage_20100901-1_all.deb

Install the boot image via the packet manager:



```
root@depotserver:~# dpkg -i opsi-linux-bootimage_20100901-1
_all.deb
```

17 Package Assembly with OPSI

17.1 Composition of OPSI packages

The installation data, the wInst-Scripts and the meta data is always one package in OPSI. This package, typically with the file name ending *.opsi, is installed on the OPSI depotserver and then ready to deployed to the clients. The significant advantages of this OPSI package format are:

- "newprod" is an easy and user friendly application to create products (software)
- All meta data is stored in an easy editable file
- Optional information editing of previously created products
- All product information (product version, package version, …) is stored in one file. All information is displayed in the OPSI config-editor.
- No root rights are necessary to create and to unpack products.

The package itself consists of a Gzip compressed archive. In this archive there are three subfolders.

- CLIENT_DATA : In this folder are the files which supposed to end up in the directory /opt/pcbin/install/<productid>
- SERVER_DATA: In this folder you can create directories which will be unpacked to "/" (For that you need root)
- OPSI In this folder are the meta data stored. (product dependencies)

17.2 Creation of an OPSI packages

17.2.1 Plan package

First of all a package has to be planned and some decisions have to be made. Simultaneously you have to set properties and dependencies of the future package.

Decision	Possible Answers
Which installation process?	- Unattended Setup - MSI-Package - Setup with automised answers

In general there are three different methods or processes to automatized software distribution with OPSI. Which method is the best depends heavily on the software and therefore has to be analysed and decided each time.

- **Unattended Setup**

  This method is based on the original setup routine (*exe). You can modify the start parameter to run in a silent or passive mode. All options can be viewed in a command shell. Just run the setup with parameter "/h" or "/?".

- **MSI Package**

  In my opinion the best and easiest solution is the Microsoft Installer Service installation method. The software is executed with a "quiet" option. Moreover this method is the safest and most used one. Furthermore it is possible to create own MSI packages extracted from Setup (*exe) files. If you are more interested, just Google for such tools online.

- **Setup with automatized answers**

  With this method you are forced to analyse the setup routine. You have to write down in a script which windows the installer setup shows and which questions and answers you have to provide. There are tools for recording the setup process. One of those tools is called "Autoit". During the OPSI software distribution process, the setup runs under the control of an automatized program (Autoit), which guides the setup program according to the script. As you already might imagine, this method is pretty complicated and not recommended. Many unexpected errors can occur during the process, which have impact on the setup.

Additionally there is also software which has own setup routines for the automatized installation. (MS Office, McAfee Virusscan,…). Additional setup instructions can be found mostly in the README file of the particular software.

Furthermore you have to plan the actions. What routine do you want to implement:

Decision	Possible Solutions
What action?	- setup - uninstall - update - always - once - custom - user login

For each possible action, you have to create later on a wInst script.

- **setup**

  This script is executed for the installation of the product. It consists normally of an operating system, disk space check and of course the required installation commands. Additionally it's possible to include an image or icon.

- **uninstall**

  This script is executed for the uninstallation of the product. It solely consists of uninstallation commands.

- **update**

  This script allows executing the update routine of software.

- **always**

  The commands in this script are executed on each system start. Can be useful for inventory software but in general it is used rarely.

- **once**

  This script is just executed once. After that it is set to "off" or "not_installed". This script type is highly rarely used and should be ignored.

- **custom**

This script type doesn't affect the installation status. After execution OPSI doesn't detect any changes. This script type is highly rarely used and should be ignored.

- **user login**

  This feature is not yet implemented.

The last step is to plan the dependencies and properties of a product:

Does the product have dependencies?	- No - Yes, which ones? (product id)
Does the product have special properties?	- No - Yes, which ones?

The properties are then important when a product has multiple installation possibilities, versions or other adjustments. These options can vary from client to client and therefore are saved for each computer individually.

17.2.2 Prepare package

This is the installation script which I have created for the 7zip application. The installation script is based on a MSI installer. Be aware, this is just an example. You have to customize your own script and adjust the links:
[setup.ins]

ℹ

```
[Initial]
;Installation Message
Message=7Zip installation
;Write error into log file, but dont cancel script
LogLevel=6
ExitOnError=false
;Show Syntax error in the script
ScriptErrorMessages=on
;Dont show protocol messages as pop-up window
```

```
TraceMode=off
;Dont show started programs
StayOnTop=false

[Actions]
;Init
DefVar $OS$
DefVar $MinorOS$



set $OS$ = GetOS
set $MinorOS$ = GetNTVersion

;Check Operating System
if ($OS$ = "Windows_NT" AND not ($MinorOS$="NT4"))

		;Show logo
		ShowBitmap /3 "%scriptpath%\7zip.png" "7Zip"

		;Show message
		Message "7Zip is installing"

		;Start 7Zip installation
		WinBatch_install

endif

;7Zip Installation
[WinBatch_install]
msiexec.exe /i %SCRIPTPATH%\7z920.msi REBOOT=ReallySuppress
/qb-!
```

[uninstall.ins]

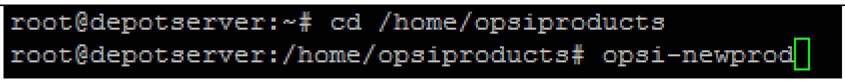
```
[Initial]
;Installation Message
Message=7Zip uninstallation
;Write error into log file, but dont cancel script
LogLevel=6
ExitOnError=false
;Show Syntax error in the script
ScriptErrorMessages=on
;Dont show protocol messages as pop-up window
TraceMode=off
;Dont show started programs
StayOnTop=false

[Actions]
;Init
DefVar $OS$
DefVar $MinorOS$



set $OS$ = GetOS
set $MinorOS$ = GetNTVersion

;Check Operating System
if ($OS$ = "Windows_NT" AND not ($MinorOS$="NT4"))

        ;Show logo
        ShowBitmap /3 "%scriptpath%\7zip.png" "7Zip"

        ;Show message
        Message "7Zip is uninstalling"

        ;Start 7Zip uninstallation
        WinBatch_install
```

	endif ;7Zip UnInstallation [WinBatch_install] msiexec.exe /x %SCRIPTPATH%\7z920.msi REBOOT=ReallySuppress /qb-!

➚	uib gmbh 2011, winstdoc.pdf

17.2.3 Create package

A new OPSI software package is built with the application “newprod”. In this paragraph you will learn how to execute the newprod command and create any desired OPSI package on the server. First of all you have to change in to the directory /home/opsiproducts (standard folder for new opsi products) on the depotserver and afterwards run the application by executing following command in the console:



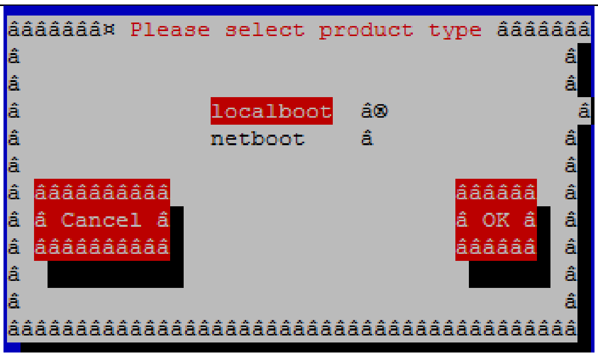
```
root@depotserver:~# cd /home/opsiproducts
root@depotserver:/home/opsiproducts# opsi-newprod
```

⚠	You need at least user rights of the pcpatch group.

As a result a graphical user interface will open. After that you are prompted to choose the product type. The product type defines the installation routine be used for the planned installation package. There are 2 different product types:

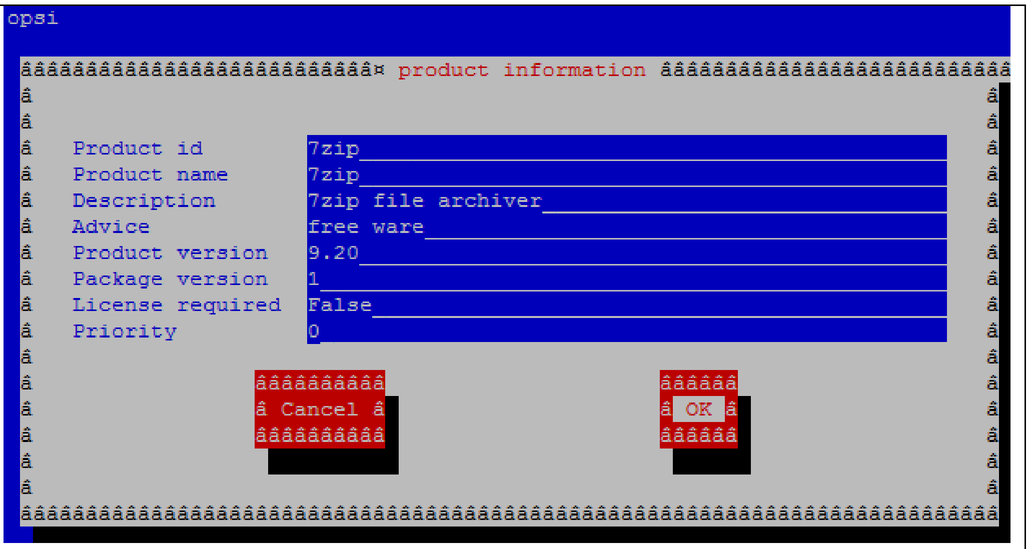
1. localboot: software products which are installed on the client by the OPSI preloginloader (before windows user login).
2. netboot: software products which trigger a boot image start. This is useful for OS installations or drive image installations).

This guide is focusing on the localboot routine. Generally this is also the most used product type. Choose “localboot” and press “OK”:



```
Please select product type

          localboot
          netboot

   Cancel                        OK
```

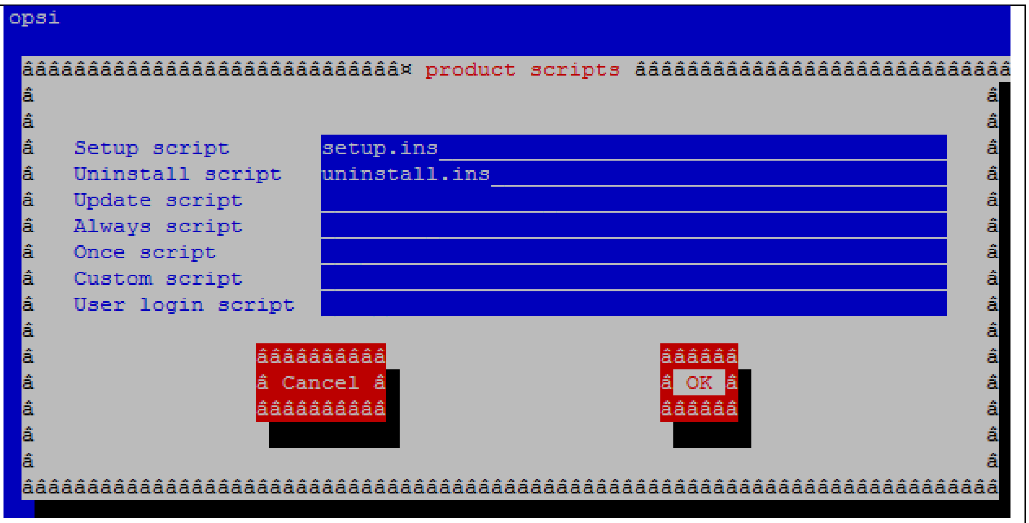
In the following screen you have to enter the product information:



```
opsi

product information

    Product id          7zip
    Product name        7zip
    Description         7zip file archiver
    Advice              free ware
    Product version     9.20
    Package version     1
    License required    False
    Priority            0

               Cancel                      OK
```

- Product id:	Unique description of the product
- Product name:	The displayed name of the product
- Description	Additional information of the product
- Advice	Additional information of the product
- Product version	Version of the product
- License required	Boolean for licensing
- Priority	Defines the installation order

After you have filled out the product information, press “OK” to reach the product scripts screen. Enter the installation and uninstallation script names. Usually the names are setup.ins and uninstall.ins. The rest can be left empty:

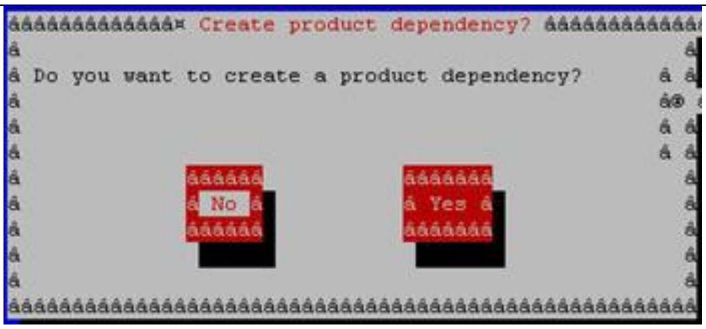


opsi

product scripts

Setup script	setup.ins
Uninstall script	uninstall.ins
Update script	
Always script	
Once script	
Custom script	
User login script	

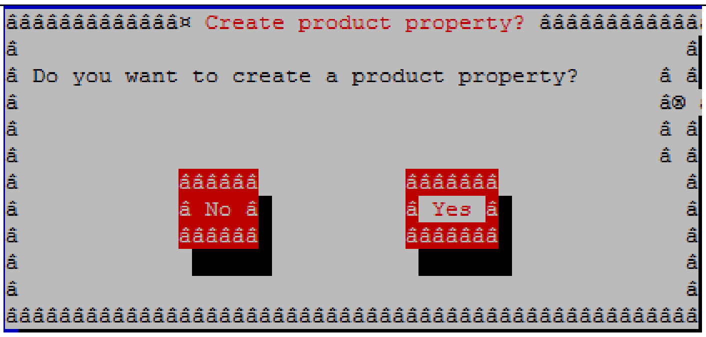
Press OK to get to the next screen. OPSI is asking if you want to create a product dependency. Since 7zip doesn't require any dependencies you should press "No". Otherwise, if your product does require any dependencies you should press "OK".



Create product dependency?

Do you want to create a product dependency?

The next screen is asking for a product property. Product properties are values, which are passed to the installation routine (wInst-Script). For example serials, usernames, direcotires and so on…) In my case I wanted to have the possibility to create a desktop shortcut of the 7zip program. Therefore I have chosen "Yes":

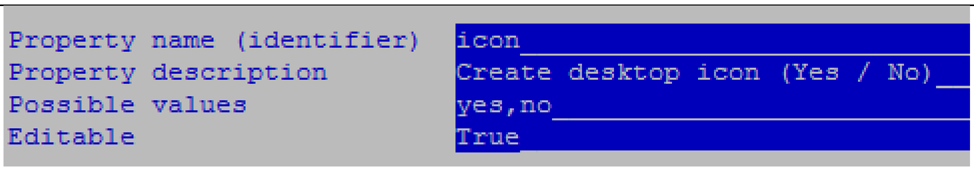


Create product property?

Do you want to create a product property?

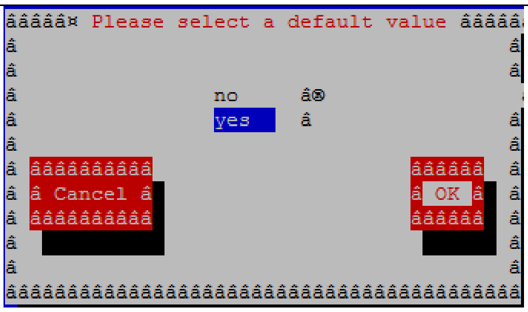
Product properties can be read later on from the installation script. They are saved for each client and contain following fields:

- Property name: Name of the property. With this particular name the wInst-Script retrieves the values of the property.
- Property description: Description of the property which will be displayed in the config editor.
- Possible values: All possible values of the property. They are separated by commas;
-



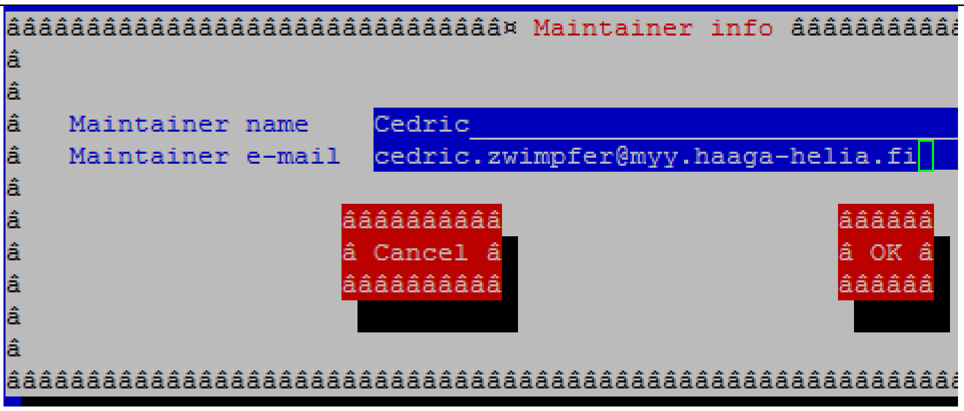
```
Property name (identifier)    icon
Property description          Create desktop icon (Yes / No)
Possible values               yes,no
Editable                      True
```

That followed you can define a default value for your property:



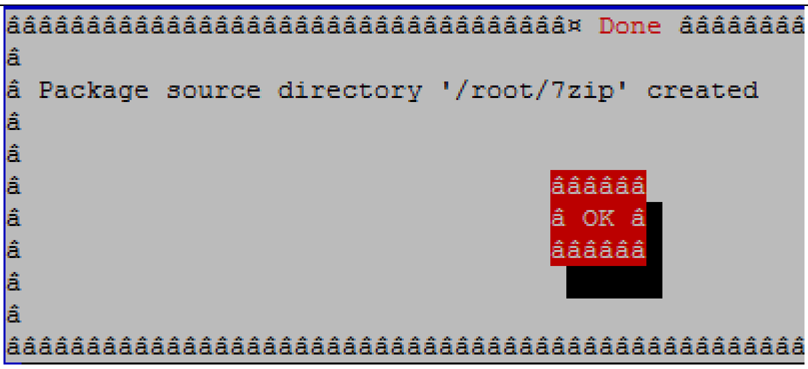
```
âââââ¤ Please select a default value âââââ
â                                       â
â                                       â
â                no      â®
â                yes     â              â
â                                       â
â ââââââââââ                   ââââââ   â
â â Cancel â                   â OK â   â
â ââââââââââ                   ââââââ   â
â                                       â
â                                       â
ââââââââââââââââââââââââââââââââââââââââ
```

Towards the end the "newprod" installation wizard is asking about the maintainer info. The maintainer is the responsible guy for the created product:



```
ââââââââââââââââââââââââââââââââ¤ Maintainer info ââââââââââ
â
â
â   Maintainer name     Cedric
â   Maintainer e-mail   cedric.zwimpfer@myy.haaga-helia.fi
â
â                     ââââââââââ                         ââââââ
â                     â Cancel â                         â OK â
â                     ââââââââââ                         ââââââ
â
â
ââââââââââââââââââââââââââââââââââââââââââââââââââââââââââââ
```

When you have created all dependencies you will reach the end of the newprod wizard. This screen will indicate the end of the "newprod" wizard:

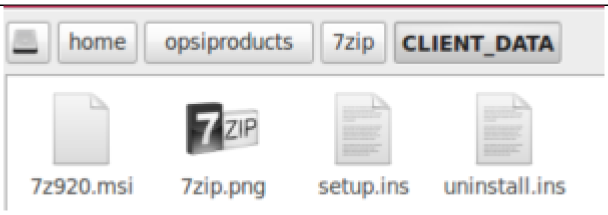


```
ââââââââââââââââââââââââââââââââââââ¤ Done ââââââââ
â
â Package sosurce directory '/root/7zip' created
â
â
â                                  ââââââ
â                                  â OK â
â                                  ââââââ
â
â
ââââââââââââââââââââââââââââââââââââââââââââââââââ
```

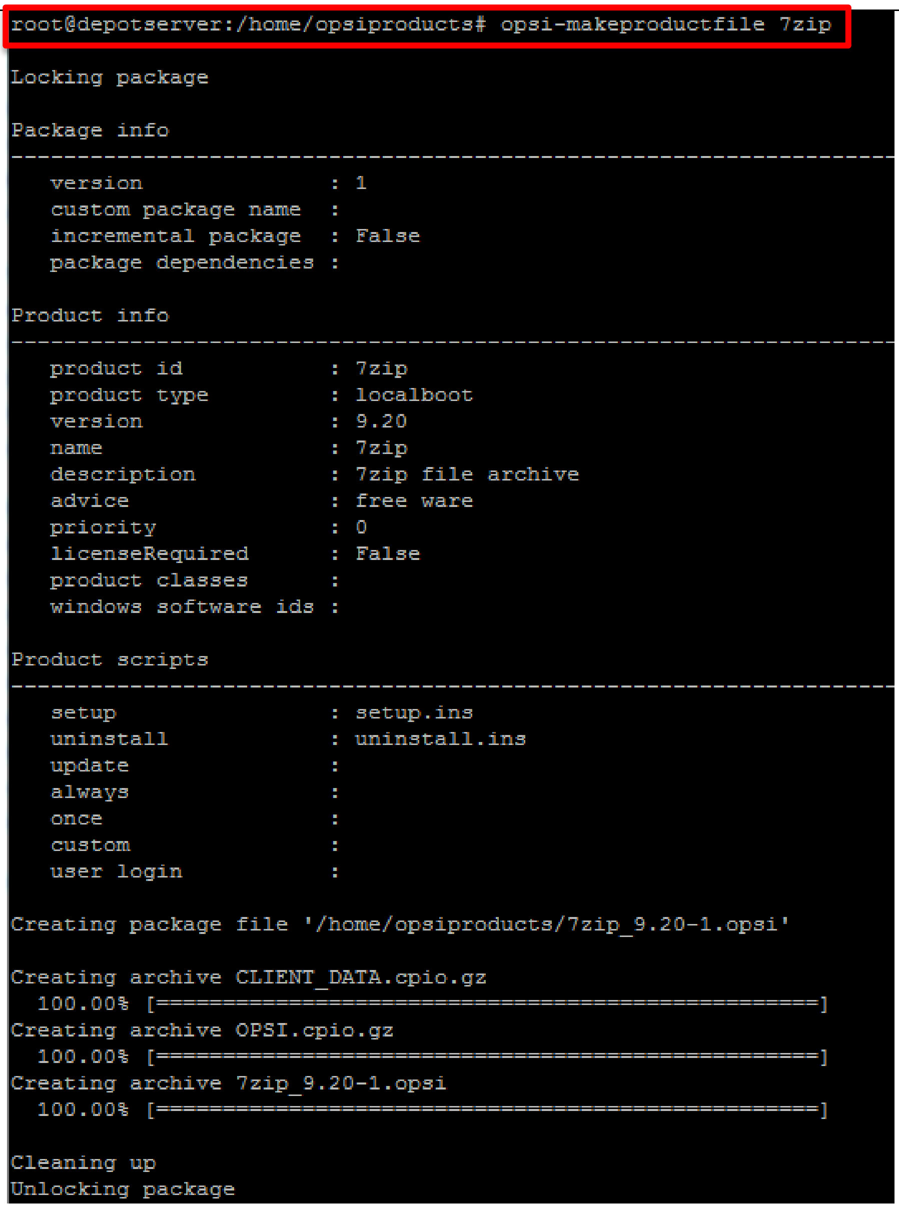
Finally, after all work done, there is a folder called <product id> (In my case "7zip") in the directory "/home/opsiproducts". This folder contains the three subfolders "CLIENT_DATA", SERVER_DATA and OPSI.

17.2.4 Install package

All files (setup.ins, uninstall.ins, PNG and MSI) have to be put into the subfolder "CLIENT_DATA".

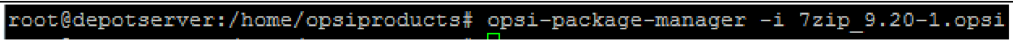


The files are now ready to be packetized. Execute following command:



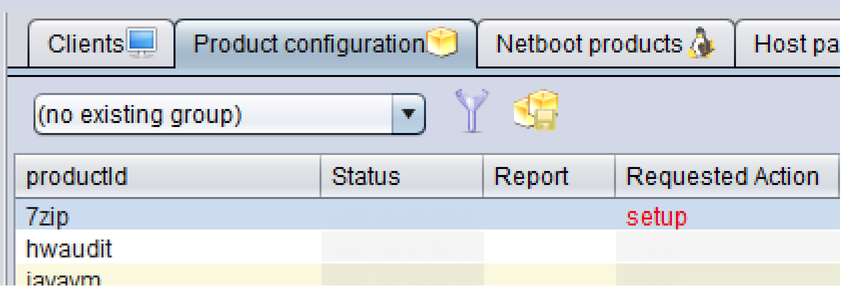
```
root@depotserver:/home/opsiproducts# opsi-makeproductfile 7zip

Locking package

Package info
------------------------------------------------------------------
   version                : 1
   custom package name    :
   incremental package    : False
   package dependencies   :

Product info
------------------------------------------------------------------
   product id             : 7zip
   product type           : localboot
   version                : 9.20
   name                   : 7zip
   description            : 7zip file archive
   advice                 : free ware
   priority               : 0
   licenseRequired        : False
   product classes        :
   windows software ids   :

Product scripts
------------------------------------------------------------------
   setup                  : setup.ins
   uninstall              : uninstall.ins
   update                 :
   always                 :
   once                   :
   custom                 :
   user login             :

Creating package file '/home/opsiproducts/7zip_9.20-1.opsi'

Creating archive CLIENT_DATA.cpio.gz
  100.00% [==================================================]
Creating archive OPSI.cpio.gz
  100.00% [==================================================]
Creating archive 7zip_9.20-1.opsi
  100.00% [==================================================]

Cleaning up
Unlocking package
```

Since the files are now in a package, it is ready to be installed. You have to execute the opsi package manager command:



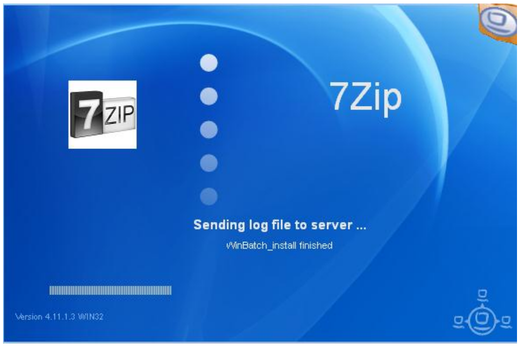
```
root@depotserver:/home/opsiproducts# opsi-package-manager -i 7zip_9.20-1.opsi
```

Finally the software package is installed on the OPSI depotserver and ready to be used. Select the destination client and switch the “Requested Action” on “setup” for 7zip.



productId	Status	Report	Requested Action
7zip			setup
hwaudit			
javavm			

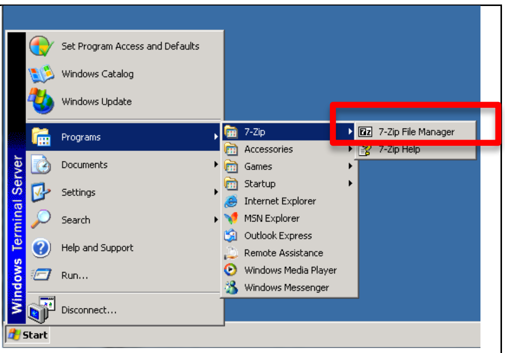
After confirming on the red tick, the client will install 7zip by the next time it reboots. Hence the OPSI-Preloginloader contacts the depotserver and starts to install 7zip. During the installation the keyboard and mouse are locked. After the installation the user is entitled to log into the system. At last 7zip is installed on the win XP client.



Sending log file to server ...

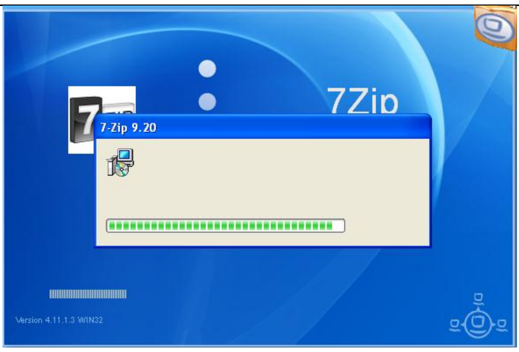
WinBatch_install finished

Version 4.11.1.3 WIN32

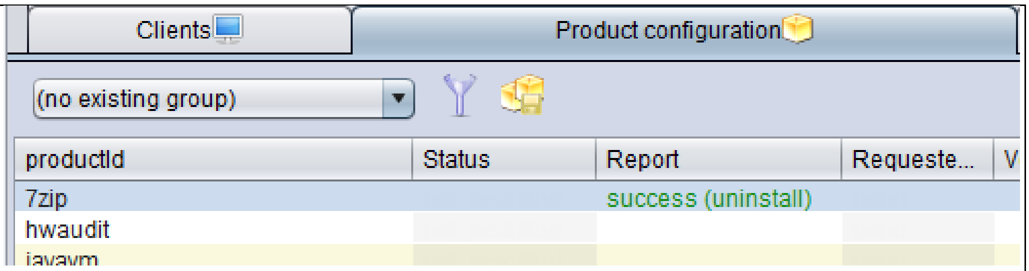


If something went wrong with the installation you have the possibilities to check the script log files.It will tell you the exit code which gives you the reason for the error. Here you can find the log for each client: “C:\tmp\instlog.txt

The uninstallation is the same process as the installation. Choose the designated client and put the “Requested Action” on “uninstall” for 7zip. The client will start to uninstall by the next time it reboots.



Version 4.11.1.3 WIN32

productId	Status	Report	Requeste...	V
7zip		success (uninstall)		
hwaudit				
javavm				

↗	uib gmbh 2011, opsi-getting-started-stable-de.pdf Chapter 4.2

17.3 MSI Packages

17.3.1 What are MSI packages

The Windows Installer (before Microsoft Installer) provides a runtime environment for Microsoft OS installation routines. The Windows Installer consists of a Windows System Service, files with ending on *.mst and*. msp and is able to interpret, unpack and execute those files. The Windows System Service files are packages in the MSI (Microsoft Software Installation) format. The *.mst files are transform files and *.msp are patch files. Due to the relational database structured MSI files, it is very easy to edit modify those files with different suitable programs. Another advantage of the Windows Installer Technology is the diagnosis and reparation of corrupted applications. The Windows Installer Service monitors if all associated components are available. If one of those components is missing, the installer tries to reinstall it. You can also trigger this process by rerunning the setup routine. You will be prompted if you like to repair or remove the application.

↗	Microsof Technet, Windows Installer overview, 2005

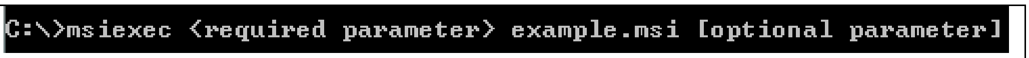
17.3.2 Msiexec

The program Msiexec allows you to install, modify or remove MSI packages in the windows command shell. This program is crucial for the software distribution via Windows Installer. Msiexec is usually used in the WinBacht Sections of the Winst-Scripts.

ℹ	Check my 7Zip script example on how to use Msiexec.
↗	Microsof Technet, Msiexec, 2005

17.3.3 Syntax

The basic Msiexec syntax is as followed:



```
C:\>msiexec <required parameter> example.msi [optional parameter]
```

The two most used <required parameter> are:

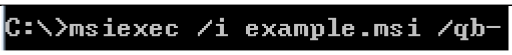
Parameter	Description
/i	Installs a product
/x	Uninstalls a product

Additionally to the required parameter there are also some optional <optional parameter> ones. These are helpful to define the behavior of the Windows Installer. The most important optional parameter is “/q”. It defines the user interface and its interoperability.

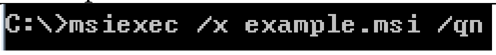
Parameter	Description
/qb-	Shows a progress bar but no cancel button. The user cannot cancel the installation.
/qn	Doesn’t show any user interface

17.3.4 Examples

The package example.msi should be installed with a progress bar but without user input.

>_	 `C:\>msiexec /i example.msi /qb-`

The package example.msi should be uninstalled in quiet mode.

>_	 `C:\>msiexec /x example.msi /qn`
⚠	Msiexec parameters are not case sensitive.

Bibliography

UiB GMBH. 2011. opsi Getting Started. URL: http://download.uib.de/opsi_stable/doc/opsi-getting-started-stable-en.pdf. Quoted 1.6.2011

OPSI.org. 2011. Features. URL: http://www.opsi.org/features/. Quoted: 15.6.2011.

UiB GMBH. 2011. opsi-handbuch-stable-de.pdf

UiB GMBH. 2011. opsi-winst Reference Manual

Microsof TechNet. 2005. Windows Installer Overview. URL: http://technet.microsoft.com/en-us/library/cc739268(WS.10).aspx. Quoted: 24.09.2011

Microsof TechNet. 2005. Msiexec. URL: http://technet.microsoft.com/en-us/library/cc759262.aspx. Quoted: 24.09.2011