Documentation

as of: **03/25/2009**

opsi
Vista / 2008
Support

installation manual

uib gmbh

uib gmbh
Bonifaziusplatz 1B,
55118 Mainz
Tel.: +49 6131-275610
www.uib.de
info@uib.de

Table of Contents

1. Introduction

Since September 2008 there is support for Vista / 2008 clients available with opsi.

According to the big changes made by Microsoft it was a lot of work for us to do:

•the OS-Installation had to be modified.

•the preloginloader and opsi-loginblocker had to be rewritten completely.

After all the cost of development for Vista support was about 30 000 Euros.

All our software is open source.
New major features and enhancements (as the Vista support is) become open source once the development is financed.

Most of the opsi enhancements are funded by customers and then become available to the open source community. 30 000 Euros for financing the Vista support is too much for a single customer, but funded by 15 customers, it may be a valuable and convenient investment of 2 000 Euros each.

So we decided to sell the development cost in chunks of min. 2 000 Euros
until the funding is complete. Up to this point, the opsi Vista support is available only to the 'cooperative funders'.

Once the development is funded completely, this software will be released as open source.

So you are not allowed to give these opsi extensions to a third party until it is released as open source.

2. Download and installation

For installation perform the following steps:

- Upgrade your opsi server with the commands:
  (for Debian / Ubuntu based opsi servers)

apt-get update
apt-get upgrade

- Download the essential opsi packages (preloginloader, winvista, win2008) from:
  http://download.uib.de/abo/vista
  Therefor you need a account (user name/password) which you will get from the uib gmbh, if you participate at funding of development.

- Install this packages with:
  opsi-package-manager -i *.opsi

- The packages for the OS installation must be completed by the installation files of Microsoft like it is described below.

3. completion of the OS-package WinVista

Because the Vista installation only starts from a Win32/Win64 environment we must build a PE-Image which is used to startup the installation.

"To install a 64-bit version of Windows you must use a 64-bit version of Windows PE. Likewise, to install a 32-bit version of Windows, you must use a 32-bit version of Windows PE."
http://technet.microsoft.com/en-us/library/cc766093.aspx

Therfore you need the Windows Automated Installation Kit (Windows AIK):

http://www.microsoft.com/downloads/details.aspx?FamilyId=94BB6E34-D890-4932-81A5-5B50C657DE08&displaylang=en

What you get is a ISO file which may be burned to CD or mounted by a virtual machine. The content of this CD must be installed in Vista (32 or 64 bit).

3.1. WinVista / Windows 2008 32-Bit version

3.1.1. Creating a 32-Bit Vista-PE

1. Installation of the Windows AIK at Vista 32-Bit or Windows 2003 with Servicepack 2 32-Bit

2. Create a WinPE

   Open as Administrator the cmd.exe

   cd „\Program Files\Windows AIK\Tools\PETools"

   copype.cmd x86 c:\winpe_x86

3. mount the image with:

   "C:\Program Files\WindowPhilips AIK\Tools\x86\imagex.exe" /mountrw C:\winpe_x86\ISO\sources\boot.wim 1 c:\winpe_x86\mount

   Edit with **Notepad.exe**
   C:\WinPE_x86\mount\Windows\System32\startnet.cmd
   Delete the entry wpeinit and add the following line:

   c:\opsi\startnet.cmd

   (Remark: The file **startnet.cmd** will be created by the opsi linux boot image by executing the script **winvista.py.** The startnetcmd contains the call to wpeinit.)

   unmount the image

   "C:\Program Files\Windows AIK\Tools\x86\imagex.exe" /unmount /commit C:\winpe_x86\mount

   Copy the directory **C:\WinPE_x86\ISO** with the target name **winpe** to **/opt/pcbin/install/winvista/** respectively **/opt/pcbin/install/win2008**

   Adjust the file access rights:

   chown -R opsiconfd:pcpatch /opt/pcbin/install/winvista/winpe

3.1.2. unattend.xml

The control file for the unattended Installation is the **unattend.xml** which you will find below **/opt/pcbin/install/winvista/opsi.** If you make any modifikations at

this file, it should be backuped in a different directory, because the opsi directory is subject of updates in future versions of the opsi winvista package.

The unattend.xml delivered wit the opsi winvista package contains the activating of the Administrator account with the password 'nt123'.

Documentation to the unattend.xml you will find (after the installation of the WAIK) in the directory
\Program Files\Windows\Waik\docs\chms

3.1.3. Driver integration

The integration of drivers works at the usual in the opsi manal described way: Place your driver directories in **/opt/pcbin/install/winvista/drivers/drivers** and then call the **create_driver_links.py** script.

Please keep in mind that Vista only accept signed drivers. Therefor it is no good idea to use driver packs which contain none vista drivers like the driver packs from driverpacks.net.

3.2. WinVista / Windows 2008 64-Bit version

3.2.1. Creating a 64-Bit Vista-PE

1. Installation of the Windows AIK at Vista 64-Bit or Windows 2003 Server 64-Bit

2. Create a WinPE

   Copy the file **boot.wim** from the Winvista-64-Bit-CD to the directory
   C:\winpe_amd64\ISO\sources

   Remove the write protection from **boot.wim**

3. mount the image with:

   ```
   "C:\Program Files\Windows AIK\Tools\amd64\imagex.exe"
   /mountrw C:\winpe_amd64\ISO\sources\boot.wim 1
   c:\winpe_amd64\mount
   ```

Edit with **Notepad.exe**
C:\WinPE_amd64\mount\Windows\System32\startnet.cmd
Delete the entry wpeinit and add the following line:

c:\opsi\startnet.cmd

(Remark: The file **startnet.cmd** will be created by the opsi Linux boot image by executing the script **winvista.py.** The startnetcmd contains the call to wpeinit.)

unmount the image

"C:\Program Files\Windows AIK\Tools\amd64\imagex.exe" /unmount /commit C:\winpe_amd64\mount

Please note: At 64-Bit Vista also the archive **2** must be mounted and edited:

"C:\Program Files\Windows AIK\Tools\amd64\imagex.exe" /mountrw C:\winpe_amd64\ISO\sources\boot.wim 2 c:\winpe_amd64\mount

Edit with **Notepad.exe**
C:\WinPE_amd64\mount\Windows\System32\startnet.cmd
Delete the entry wpeinit and add the following line:

c:\opsi\startnet.cmd

Additional remove the files **setup.exe** and **sources\setup.exe**

Image unmounten

"C:\Program Files\Windows AIK\Tools\amd64\imagex.exe" /unmount /commit C:\winpe_amd64\mount

Copy the directory **C:\WinPE_amd64\ISO** with the target name **winpe** to **/opt/pcbin/install/winvista64/** respectively **/opt/pcbin/install/win2008_64/**

Adjust the file access rights:

chown -R opsiconfd:pcpatch /opt/pcbin/install/winvista64/winpe

3.3. Providing the installation files

Copy the complete installation DVD to

/opt/pcbin/install/winvista/vistasrc

Adjust the file access rights:

```
chown -R opsiconfd:pcpatch /opt/pcbin/install/winvista/vistasrc
```

3.4. Log files of the installation

- **c:\Windows\Panther\setupact.log**:
  Log until the end of setup phase 4 (running under WinPE)
- **c:\Windows\Panther\setupact.err**:
  Error log until the end of setup phase 4 (running under WinPE)
- **c:\Windows\Panther\UnattendGC\setupact.log**:
  Log since specialize phase
- **c:\Windows\Panther\UnattendGC\setupact.err**:
  Error log since specialize phase
- **c:\Windows\System32\Winevt\Logs***
- **c:\Windows\ntbtlog.txt** (only with activated startup protocol)

4. opsi-preloginloader 3.4 (also) for Vista/Windows-2008

4.1. Overview

Due to the major changes between Windows XP and Vista, the opsi preloginloader (with the exception opsi-winst) has been completely new implemented.

This new implementation has been done in the Python language which is also used for the server parts of opsi. For the installation are files are used, which are compiled by the py2exe program. This makes the installation independent from any existing python installation at the client.

The important enhancements are:

•Event based control:
The activity of the opsi client agent (opsiclientd) may be triggered by different events in the client system. According to this fact the start of the installation is not fixed at the system startup any more.

•Control via web service:
This interface is used for maintenance purpose at the moment. It will be used for central (time) controlled installations in future.

•Remote configuration:
The configuration data for the clients may be changed (globally or client specific) at the server by editing the 'general config' parameters

The opsi-preloginloader 3.4 consists of multiple components :

•opsiclientd: the central service

•notifier: information and communication window

•opsi-loginblocker: block the login until the installation has finished

The legacy prelogin.exe based service may be installed as well, but it is deprecated and should be replaced by opsiclientd based installations.

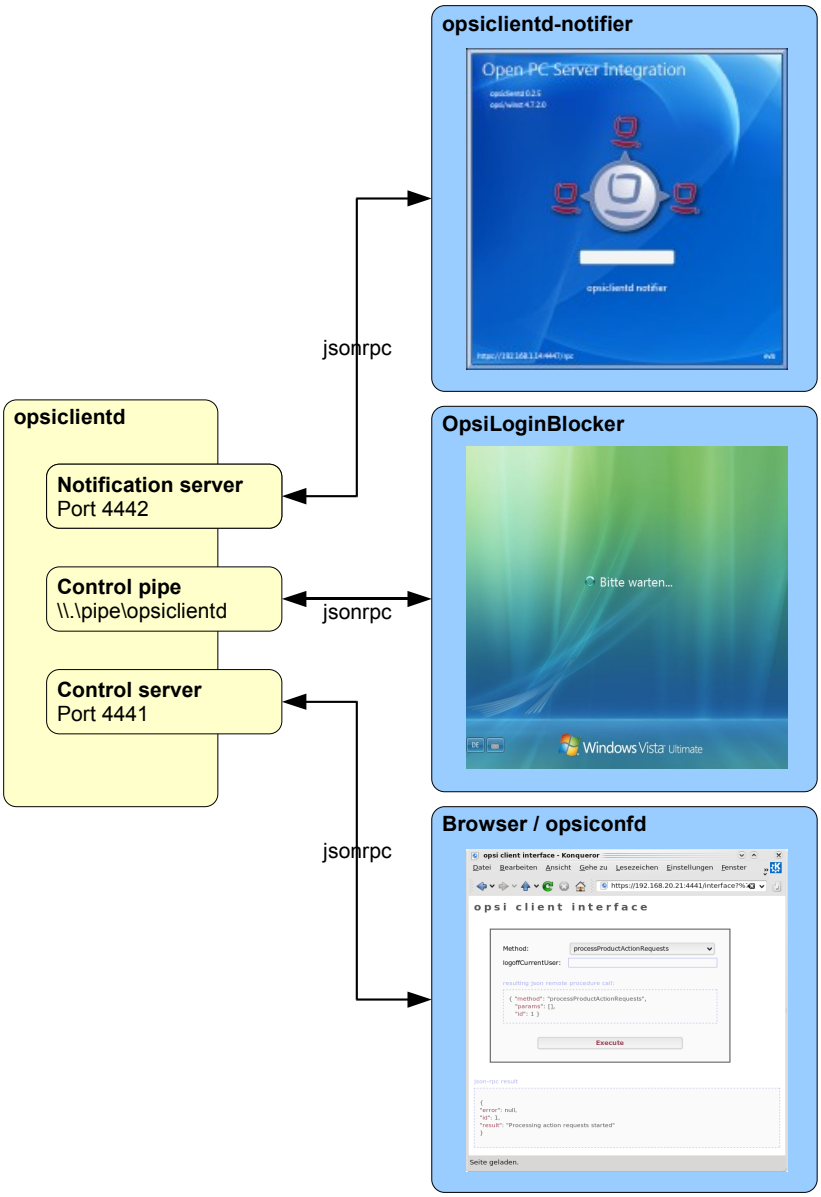

Figure 1: Scheme of the opsiclientd components

4.2. Installation

In case of automatic OS-Installation with opsi (not image based), the opsi preloginloader will be installed automatically.

For a subsequent installation on a existing Windows system or for repair purposes there two possibilities that described more detailed in the opsi manual:

•Login with a administrative account, mount the opt_pcbin share of the opsi server and call the script **`install\preloginloader\service_setup.cmd`**

•Use the server side script opsi-deploy-preloginloader

At clients which are still integrated in opsi, the new preloginloader can be install with the standard opsi process by switching the required action to setup. At Vista clients with the former preloginvista installed, this product will be switched to not_installed. The product preloginvista may (and should) deleted from the server. The OS-Installation packages (winvista, winxppro, ...) must be updated too in order to work with the new preloginloader.

The preloginloader got a new product property 'client_servicetype'. The default is 'opsiclientd' which is the new one. In special cases you may switch to the legacy 'prelogin'. But this one contains no new extensions and is deprecated.

For deinstallation of the preloginloader the action request may be switched to 'uninstall'

4.3. opsiclientd

Core component of the preloginloader is the service opsiclientd. This service starts at the boot time.

The opsiclientd has the following tasks:

- Getting active if the configurated event takes place. The default event is 'gui_startup' which will fire (like the legacy version) at boot time and before login.
- Via web service (JSON-RPC) the opsiclientd contacts the opsi server and ask for configuration data and required actions.
- Creates a named pipe which is used by the opsi login blocker to ask via JSON-RPC the opsiclientd when to unblock the login.
- Starting the opsiclientd_notifier as thread for information and interaction with the user.
- If needed, mounting the depot share and update and start of the opsi winst to process the action requests (installations).

4.4. opsiclientd_notifier

The opsiclientd_notifier implements the interaction with the user. They displays status messages and may give the possibility to interact with the process.

4.4.1. opsiclientd_event_notifier

The *event_notifier* gets active if a event fires and for this event is configured that **`warning_time`** is > 0 (default = 0). In this case the user will see **`warning_time`** seconds a message windows with the in **`message`** configured text and a '*Start now*' button. Is **`user_cancelable`** = true, so will be also a '*Abort*' button enabled. When the **`warning_time`** is expired or if the user choosed '*Start now*' the actions and the *action_notifier* will be started.

At the default event **`gui_startup`**, the *event_notifier* is disabled by default. It is enabled and more important at events like **`vpn_startup`**.

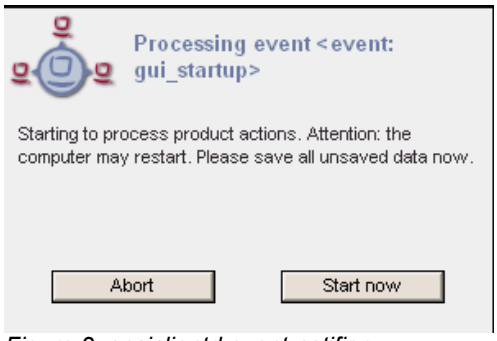

Figure 2: opsiclientd event notifier

4.4.2. opsiclientd_action_notifier

The *action_notifier* presents the action progress and gives (if so configured) the possibility to cancel the process.

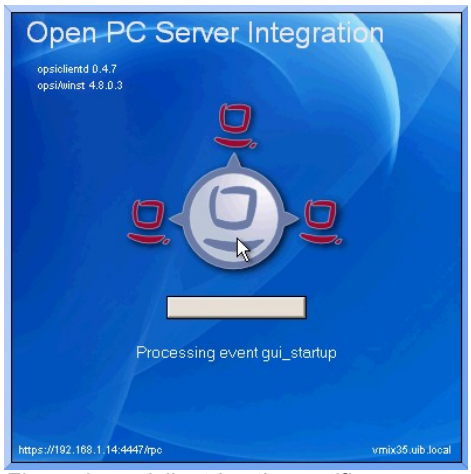

Figure 3: opsiclientd action notifier

4.5. opsi-loginblocker

The opsi login blocker at Vista is implemented as '**`credential provider filter`**'. It blocks all 'credential providers' until the release by the opsiclientd or the timeout.

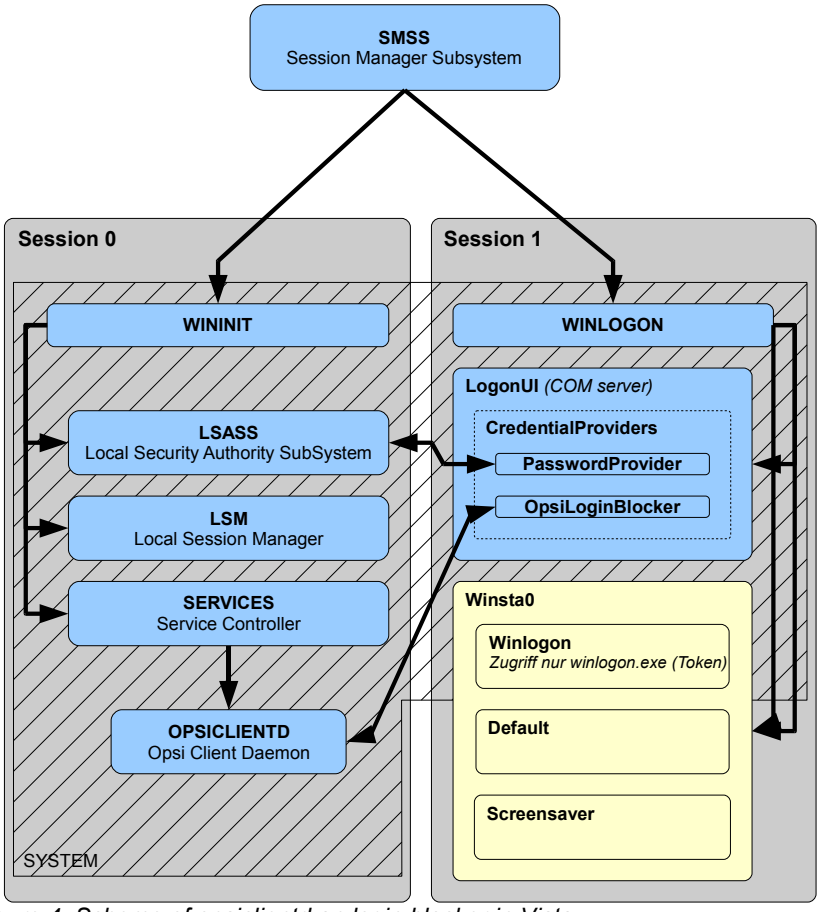

Figure 4: Scheme of opsiclientd an login blocker in Vista

The opsi login blocker at Win2K / Winxp is implemented as '**`pgina`**'. implementiert. It blocks the **`msgina.dll`** until the release by the opsiclientd or the timeout.

4.6. Configuration

4.6.1. Configuration via configuration file

The configuration file is:

c:\program files\opsi.org\preloginloader\opsiclientd\opsiclient.conf

The configuration written in this file may be changed by different configuration data, which comes via web service after a successful connection to the opsi-server.

A sample opsiclientd.conf:

```
; = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
; =     configuration file for opsiclientd                                    =
; = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =


; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; -     global settings                                                       -
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[global]

# Location of the log file.
log_file = c:\\tmp\\opsiclientd.log

# Set the log (verbosity) level
# (0 <= log level <= 9)
# 0: nothing, 1: critical, 2: errors, 3: warnings, 4: notices
# 5: infos, 6: debug messages, 7: more debug messages, 9: passwords
log_level = 4

# Opsi host key.
opsi_host_key =

# On every daemon startup the user login gets blocked
# If the gui starts up and no events are being processed the login gets
unblocked
# If no gui startup is noticed after <wait_for_gui_timeout> the login gets
unblocked
# Set to 0 to wait forever
wait_for_gui_timeout = 120

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; -     config service settings                                               -
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[config_service]

# Service url.
# http(s)://<opsi config server address>:<port>/rpc
url = https://opsi.uib.local:4447/rpc

# Conection timeout.
connection_timeout = 10

# The time in seconds after which the user can cancel the connection
establishment
user_cancellable_after = 0

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; -     control server settings                                               -
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[control_server]

# The network interfaces to bind to.
# This must be the IP address of an network interface.
# Use 0.0.0.0 to listen to all interfaces
interface = 0.0.0.0

# The port where opsiclientd will listen for HTTPS rpc requests.
port = 4441
```

```
# The location of the server certificate.
ssl_server_cert_file = %system.program_files_dir
%\\opsi.org\\preloginloader\\opsiclientd\\opsiclientd.pem

# The location of the server private key
ssl_server_key_file = %system.program_files_dir
%\\opsi.org\\preloginloader\\opsiclientd\\opsiclientd.pem

# The location of the static files
static_dir = %system.program_files_dir
%\\opsi.org\\preloginloader\\opsiclientd\\static_html

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; -     notification server settings                                     -
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[notification_server]

# The network interfaces to bind to.
# This must be the IP address of an network interface.
# Use 0.0.0.0 to listen to all interfaces
interface = 127.0.0.1

# The port where opsiclientd will listen for notification clients.
port = 4442

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; -     opsiclientd notifier settings                                    -
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[opsiclientd_notifier]

# Notifier application command
command = %system.program_files_dir%\\opsi.org\\preloginloader\\notifier.exe -
p %notification_server.port%

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; -     opsiclientd rpc tool settings                                    -
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[opsiclientd_rpc]

# RPC tool command
command = %system.program_files_dir
%\\opsi.org\\preloginloader\\opsiclientd_rpc.exe "%global.host_id%"
"%global.opsi_host_key%" "%control_server.port%"

; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; -     action processor settings                                        -
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[action_processor]
# Locations of action processor
local_dir = %system.program_files_dir%\\opsi.org\\preloginloader\\opsi-winst
remote_dir = \\install\\opsi-winst\\files\\opsi-winst
filename = winst32.exe
# Action processor command
command = "%action_processor.local_dir%\\%action_processor.filename%"
/opsiservice "https://%config_service.host%:%config_service.port%" /clientid
%global.host_id% /username %global.host_id% /password %global.opsi_host_key%
```

```
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; -     events                                                            -
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[event_daemon_startup]
type = daemon startup
active = false

[event_daemon_shutdown]
type = daemon shutdown
active = false

[event_gui_startup]
type = gui startup
message = Starting to process product actions. Attention: the computer may
restart. Please save all unsaved data now.
user_cancelable = false
block_login = true
lock_workstation = false
logoff_current_user = false
get_config_from_service = true
update_config_file = true
write_log_to_service = true
update_action_processor = true
event_notifier_command = %opsiclientd_notifier.command% -s notifier\\event.ini
event_notifier_desktop = current
action_notifier_command = %opsiclientd_notifier.command% -s
notifier\\action.ini
action_notifier_desktop = current
action_processor_command = %action_processor.command%
action_processor_desktop = current

[event_vpn_startup]
type = custom
active = false
wql = SELECT * FROM __InstanceModificationEvent WITHIN 2 WHERE TargetInstance
ISA 'Win32_NetworkAdapter' AND TargetInstance.Name = "TAP-Win32 Adapter V9"
AND TargetInstance.NetConnectionStatus = 2
message = Opsi will start software and hardware inventory on this computer.
You can continue your work in the meantime.
get_config_from_service = true
update_config_file = true
write_log_to_service = true
warning_time = 20
service_options = { "actionProcessingFilter": { "productIds": ["hwaudit",
"swaudit"] } }
event_notifier_command = %opsiclientd_notifier.command% -s notifier\\event.ini
event_notifier_desktop = current
action_notifier_command = %opsiclientd_notifier.command% -s
notifier\\action.ini
action_notifier_desktop = current
action_processor_command = %action_processor.command% /service_options
"%event_vpn_startup.service_options%"
action_processor_desktop = current
```

The above mentioned timeouts have the following relations:

1. If a event fires, the *event_notifier* shows **`warning_time`** seconds a message and according to the value of **`user_cancelable`** a 'Abort' button. Is the **`warning_time`** = 0 (default) the *event_notifier* don't starts.

2. After the **`warning_time`** the action starts, which means normally that the opsiclientd try to reach the opsi server using the **`url`** address.

3. If after **`user_cancellable_after`** seconds still no connection established, so the action_notifier will enable a 'Abort' button. Once the connection is established, there is no more possibility to abort.

4. If there no connection could be established in **`connection_timeout`** seconds, the opsiclientd abort the actions. To avoid a user from aborting, set **`user_cancellable_after`** = **`connection_timeout`** .

4.6.2. Configuration via web service (general config)

The opsiclientd configuration can be changed by the 'general config' at the server.

The entries in the general config have to been according to the following patterns:
opsiclientd.<name of the section>.<name of the key>

Example:
opsiclientd.global.log_level = 4
set in the configuration file opsiclientd.conf in the section [global] the value of log_level to the value 4.

The following Figure shows how to change the server wide general config via opsi configed

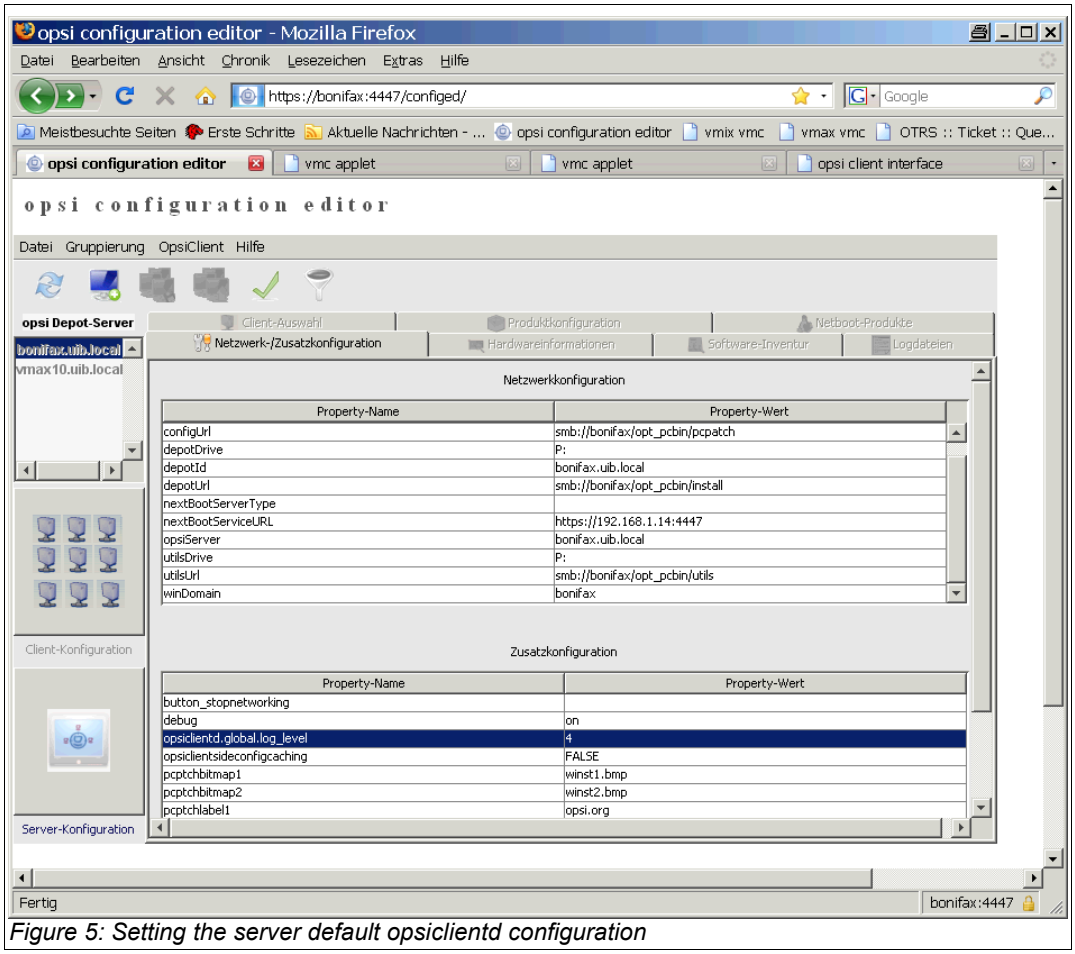

Figure 5: Setting the server default opsiclientd configuration

At the moment it isn't possible to manipulate these entries client specific via opsi configed (we working on this). So any client specific change at the general config must be done manually direct in the backend.

Here a example for the File31-Backend:

Excerpt from a <pcname>.ini file:

```
[generalconfig]
opsiclientd.global.log_level=6
```

Here a example for the LDAP-Backend (with JXplorer as LDAP-Browser):

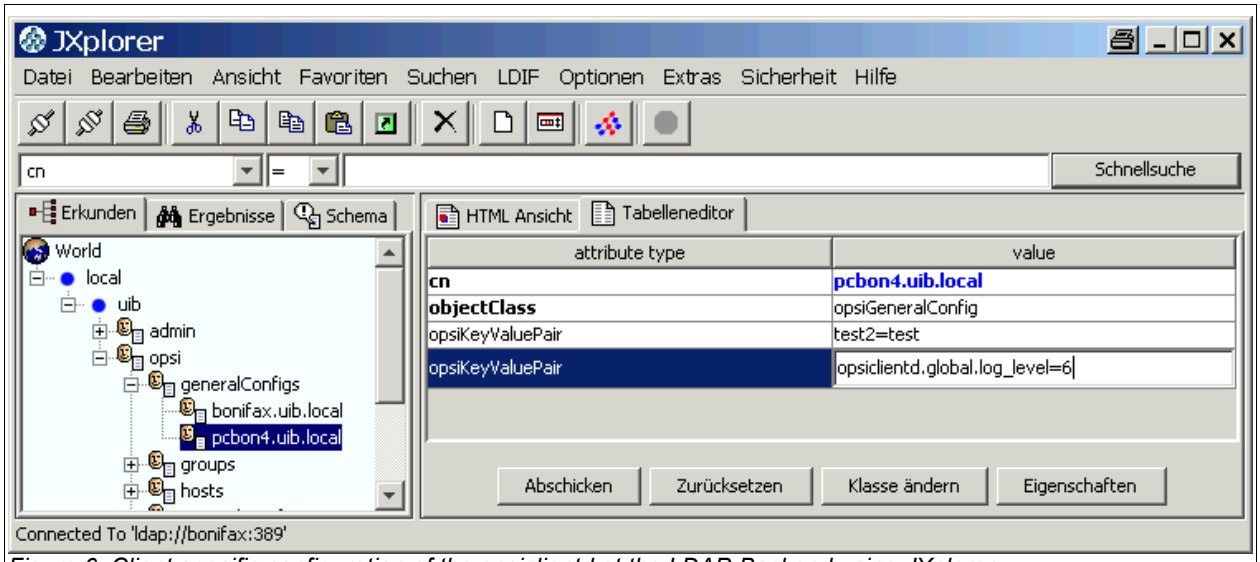

Figure 6: Client specific configuration of the opsiclientd at the LDAP-Backend using JXplorer

4.6.3. Configuration of different events

The following events are predefined:

•event_gui_startup
Default event at client boot - before login

•event_vpn_startup
Example of a 'custom event' where the fire condition is defined via a WMI-WQL query. In this case the condition is the activating of the VPN network interface:
`wql = SELECT * FROM __InstanceModificationEvent WITHIN 2 WHERE TargetInstance ISA 'Win32_NetworkAdapter' AND TargetInstance.Name = "TAP-Win32 Adapter V9" AND TargetInstance.NetConnectionStatus = 2`
where **`"TAP-Win32 Adapter V9"`** is the of the VPN network adapter which is specific for the used VPN software.

•event_daemon_startup
not implemented yet

•event_daemon_shutdown
not implemented yet

Setting the entry 'active = false' disables the event.

4.7. Logging

The opsiclientd logs to:

c:\tmp\opsicliend.log

All log informations will be transfered to the opsi server via web service. At the server you find these log infos at /var/log/opsi/clientconnect/<pcname>.log. They are presented in the opsi configed at the tab 'logfiles / client connect'.

Every line at the log has the pattern:
[<log level>] [<time stamp>] [message source] message.

There are the following log levels:

```
# Set the log (verbosity) level
# (0 <= log level <= 9)
# 0: nothing, 1: critical, 2: errors, 3: warnings, 4: notices
# 5: infos, 6: debug messages, 7: more debug messages, 9: passwords
```

Example:

```
[4] [Feb 02 17:30:11] [opsiclientd]  Config read  (opsiclientd.pyo|1602)
[0] [Feb 02 17:30:11] [opsiclientd]  Opsiclientd version: 0.4.4.4  (opsiclientd.pyo|1816)
[0] [Feb 02 17:30:11] [opsiclientd]  Commandline:
C:\Programme\opsi.org\preloginloader\opsiclientd.exe  (opsiclientd.pyo|1817)
[0] [Feb 02 17:30:11] [opsiclientd]  Working directory: C:\WINDOWS\system32  (opsiclientd.pyo|
1818)
[4] [Feb 02 17:30:11] [opsiclientd]  Using host id 'vmix35.uib.local'  (opsiclientd.pyo|1819)
[4] [Feb 02 17:30:11] [opsiclientd]  Starting control pipe  (opsiclientd.pyo|1825)
[4] [Feb 02 17:30:11] [opsiclientd]  Control pipe started  (opsiclientd.pyo|1829)
[4] [Feb 02 17:30:11] [opsiclientd]  Starting control server  (opsiclientd.pyo|1834)
[4] [Feb 02 17:30:11] [opsiclientd]  Control server started  (opsiclientd.pyo|1843)
[4] [Feb 02 17:30:11] [opsiclientd]  Starting notification server  (opsiclientd.pyo|1848)
[4] [Feb 02 17:30:11] [opsiclientd]  Notification server started  (opsiclientd.pyo|1863)
[4] [Feb 02 17:30:11] [opsiclientd]  Event 'daemon_shutdown' is deactivated  (opsiclientd.pyo|
1770)
[4] [Feb 02 17:30:11] [opsiclientd]  Event 'net_startup' is deactivated  (opsiclientd.pyo|1770)
[4] [Feb 02 17:30:11] [opsiclientd]  Event 'daemon_startup' is deactivated  (opsiclientd.pyo|
1770)
[4] [Feb 02 17:30:12] [control server]  Control server is accepting HTTPS requests on port 4441
(opsiclientd.pyo|1164)
[4] [Feb 02 17:30:12] [control server]  Control server exiting  (opsiclientd.pyo|1170)
[4] [Feb 02 17:30:12] [opsiclientd]  gui startup event 'gui_startup' created  (opsiclientd.pyo|
1784)
[4] [Feb 02 17:30:12] [opsiclientd]  Waiting for gui startup (timeout: 120 seconds)
(opsiclientd.pyo|1872)
[4] [Feb 02 17:30:13] [opsiclientd]  rpc getBlockLogin: blockLogin is 'True'  (opsiclientd.pyo|
2065)
[4] [Feb 02 17:30:15] [opsiclientd]  rpc getBlockLogin: blockLogin is 'True'  (opsiclientd.pyo|
2065)
[4] [Feb 02 17:30:17] [opsiclientd]  rpc getBlockLogin: blockLogin is 'True'  (opsiclientd.pyo|
2065)
[4] [Feb 02 17:30:19] [event gui_startup]  Firing event '<event: gui_startup>'  (opsiclientd.pyo|
258)
[4] [Feb 02 17:30:19] [opsiclientd]  Processing event <event: gui_startup>  (opsiclientd.pyo|
1936)
[4] [Feb 02 17:30:19] [event wait_for_gui]  Firing event '<event: wait_for_gui>'
(opsiclientd.pyo|258)
```

```
[4] [Feb 02 17:30:19] [opsiclientd]  Executing:
C:\Programme\\opsi.org\\preloginloader\\opsiclientd_rpc.exe "vmix35.uib.local" "*** confidential
***" "4441" "setCurrentActiveDesktopName(System.getActiveDesktopName())"  (Windows.pyo|628)
[4] [Feb 02 17:30:19] [opsiclientd]  Gui started  (opsiclientd.pyo|1874)
[4] [Feb 02 17:30:19] [opsiclientd]  rpc getBlockLogin: blockLogin is 'True'  (opsiclientd.pyo|
2065)
[4] [Feb 02 17:30:21] [opsiclientd]  rpc getBlockLogin: blockLogin is 'True'  (opsiclientd.pyo|
2065)
[4] [Feb 02 17:30:21] [control server]  Authorization request from vmix35.uib.local@127.0.0.1
(opsiclientd.pyo|888)
[4] [Feb 02 17:30:21] [control server]  Authorization request from vmix35.uib.local@127.0.0.1
(opsiclientd.pyo|888)
[4] [Feb 02 17:30:21] [opsiclientd]  rpc setCurrentActiveDesktopName: current active desktop name
set to 'Winlogon'  (opsiclientd.pyo|2152)
[4] [Feb 02 17:30:22] [opsiclientd]  Process ended: 1736  (Windows.pyo|636)
[4] [Feb 02 17:30:22] [event processing]  Starting notifier application in session '0' on desktop
'Winlogon'  (opsiclientd.pyo|1295)
[4] [Feb 02 17:30:22] [event processing]  Executing:
C:\Programme\\opsi.org\\preloginloader\\notifier.exe -p 4442 -s notifier\\action.ini
(Windows.pyo|628)
[4] [Feb 02 17:30:23] [opsiclientd]  rpc getBlockLogin: blockLogin is 'True'  (opsiclientd.pyo|
2065)
[4] [Feb 02 17:30:25] [opsiclientd]  Getting config from service  (opsiclientd.pyo|1647)
[4] [Feb 02 17:30:25] [service connection]  Connecting to config server
'https://192.168.1.14:4447/rpc' #1  (opsiclientd.pyo|1235)
[4] [Feb 02 17:30:25] [opsiclientd]  rpc getBlockLogin: blockLogin is 'True'  (opsiclientd.pyo|
2065)
[4] [Feb 02 17:30:26] [service connection]  Connected to config server
'https://192.168.1.14:4447/rpc'  (opsiclientd.pyo|1247)
[4] [Feb 02 17:30:27] [opsiclientd]  Got config from service  (opsiclientd.pyo|1664)
[4] [Feb 02 17:30:27] [opsiclientd]  Trying to write config to file:
'C:\Programme\opsi.org\preloginloader\opsiclientd\opsiclientd.conf'  (opsiclientd.pyo|1607)
[4] [Feb 02 17:30:27] [opsiclientd]  No need to write config file
'C:\Programme\opsi.org\preloginloader\opsiclientd\opsiclientd.conf', config file is up to date
(opsiclientd.pyo|1637)
[4] [Feb 02 17:30:27] [opsiclientd]  rpc getBlockLogin: blockLogin is 'True'  (opsiclientd.pyo|
2065)
[4] [Feb 02 17:30:28] [opsiclientd]  Got product action requests from configservice
(opsiclientd.pyo|2294)
[4] [Feb 02 17:30:28] [opsiclientd]  No product action requests set  (opsiclientd.pyo|2302)
[4] [Feb 02 17:30:29] [opsiclientd]  rpc getBlockLogin: blockLogin is 'True'  (opsiclientd.pyo|
2065)
[4] [Feb 02 17:30:31] [opsiclientd]  Writing log to service  (opsiclientd.pyo|1675)
```

The opsi login blocker logging to to the Windows event log. If the log level is 8 and up there is also a log file: c:\tmp\opsi_loginblocker.log.

4.8. control server

The control server port may be used for a remote control of the opsiclientd. For security reasons we need an authentication. This authentication may be the local Administrator with password (empty passwords are not allowed) or the full qualified client name withe client key (etc/opsi/pckeys) as password. At the moment the control server is used for maintenance purposes only.

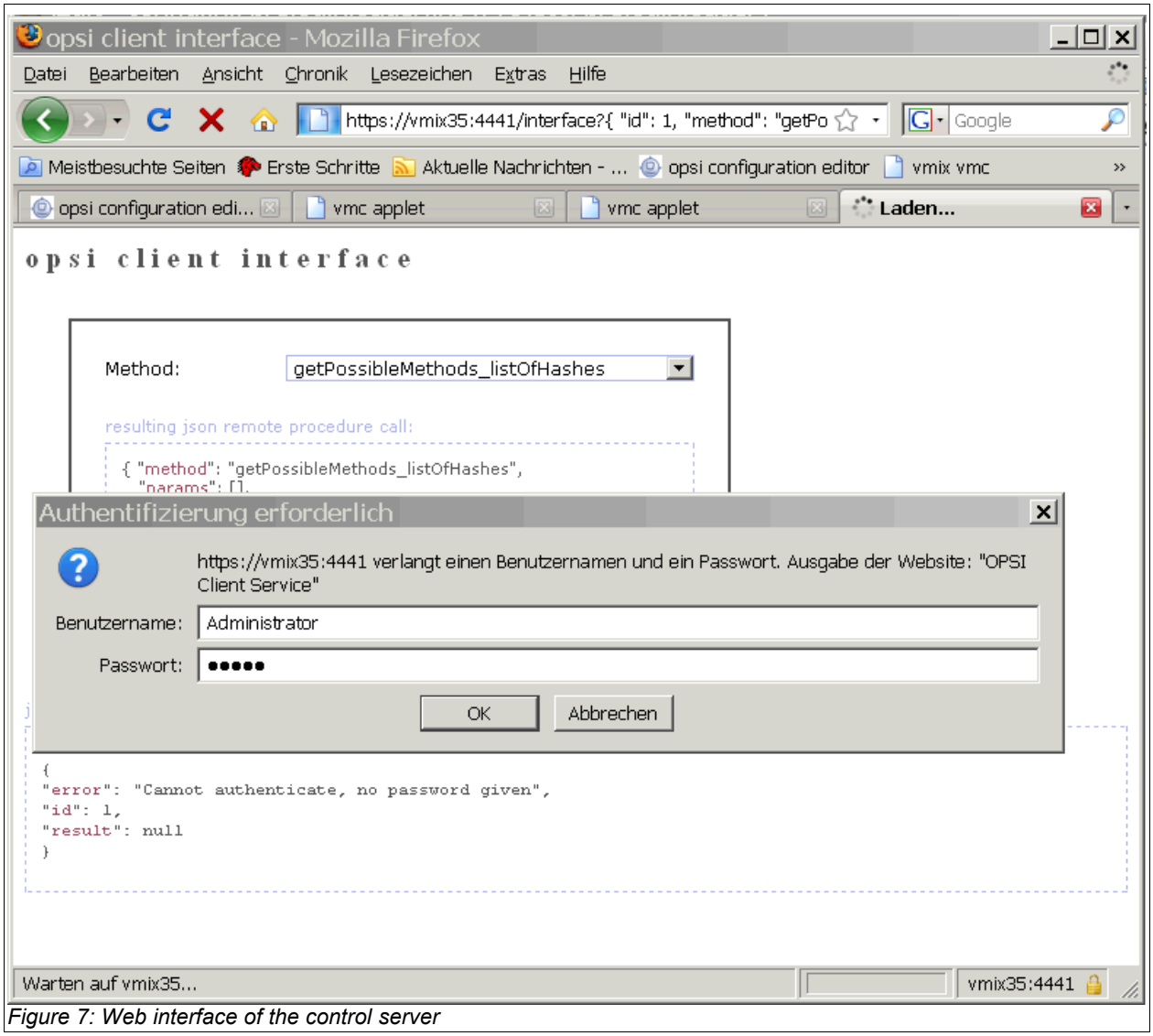

Figure 7: Web interface of the control server

5. opsi-winst with Vista / Windows-2008 32 Bit-Version

There are no known issues.

6. Known issues at the 64 Bit support

The opsi installer winst is a 32 bit program. There is no known problem installing 32 bit software on a 64 bit systim using opsi winst. For the installation of 64 bit software some constants like %ProgramFilesDir% give wrong values, registry settings going to the 32 bit part of the registry and not to the 64 bit part.

A s a workaround to perform 64-bit operations the c:\windows\system32\cmd.exe is copied as cmd64.exe to c:\windows. Calling this cmd64.exe via a ExecWith section makes it possible to copy files into 64-Bit parts of the file system (e.g. c:\windows\system32) and to manipulate the 64 bit part of the registry using the reg.exe command.