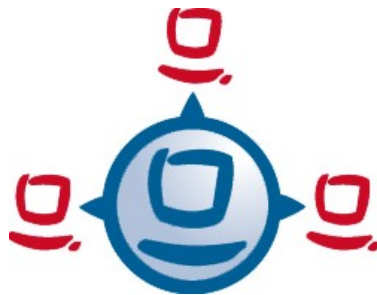

Dokumentation

Stand: 25.03.2009

opsi Vista / 2008 Unterstützung

Installationsanleitung



uib
uib gmbh

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

Inhaltsverzeichnis

1. Einführung.....	3
2. Herunterladen und Installation der Pakete.....	3
3. Vervollständigung der OS-Pakete WinVista.....	4
3.1. WinVista / Windows 2008 32-Bit Version.....	4
3.1.1. Erstellen eines 32-Bit Vista-PE.....	4
3.1.2. unattend.xml.....	5
3.1.3. Treiber-Integration.....	5
3.2. WinVista / Windows 2008 64-Bit Version	6
3.2.1. Erstellen eines 64-Bit Vista-PE.....	6
3.3. Bereitstellung der Installationsmedien.....	7
3.4. Log-Dateien der unattended-Installation.....	7
4. opsi-preloginloader 3.4 (auch) für Vista/Windows-2008.....	8
4.1. Übersicht.....	8
4.2. Installation.....	9
4.3. opsiclientd.....	10
4.4. opsiclientd_notifier.....	11
4.4.1. opsiclientd_event_notifier.....	11
4.4.2. opsiclientd_action_notifier.....	11
4.5. opsi loginblocker.....	12
4.6. Konfiguration.....	13
4.6.1. Konfiguration über die Konfigurationsdatei.....	13
4.6.2. Konfiguration über den Webservice (General config).....	16
4.6.3. Konfiguration unterschiedlicher Events.....	18
4.7. Logging.....	19
4.8. Control Server.....	20
5. opsi-winst unter Vista / Windows-2008 32 Bit-Version.....	21
6. Bekannte Besonderheiten der 64 Bit-Unterstützung.....	22

1. Einführung

Mit Vista / Windows-2008 hat Microsoft erhebliche Änderungen gegenüber der Vorversion durchgeführt. Dies hat für opsi zur Folge, dass erhebliche Teile des opsi-Preloginloaders und der OS-Installation neu konzipiert und implementiert werden mussten.

Zur Deckung der der damit verbundenen erheblichen Entwicklungskosten hat die uib gmbh ein kollaboratives Finanzierungskonzept vorgestellt. Dabei wird die Vista Unterstützung zunächst nur denen zur Verfügung gestellt, welche sich auch an der Finanzierung beteiligen. Bis zur vollständigen Deckung der Entwicklungskosten sind die neu entwickelten Komponenten kein Opensource-Code, stehen nicht öffentlich zur Verfügung und dürfen nicht weitergegeben werden.

Natürlich ist es unser Ziel die Deckung der Entwicklungskosten möglichst bald zu erreichen und den Code als Opensouce-Code freizugeben.

2. Herunterladen und Installation der Pakete

Zur Installation der Vista / Windows-2008 Unterstützung für opsi gehen Sie wie folgt vor:

- Aktualisieren Sie Ihren opsi-Server mittels
apt-get update
apt-get upgrade
- Laden Sie sich die notwendigen opsi-Pakete (preloginloader, winvista, win2008) von:
`http://download.uib.de/abo/vista`
herunter. Hierzu benötigen Sie einen Zugang (username/password) den Sie von der uib gmbh erhalten, so Sie sich an der Finanzierung beteiligen.
- Installieren Sie die opsi-Pakete mit:
opsi-package-manager -i *.opsi
- Die Pakete zur OS-Installation müssen wie weiter unten beschrieben um die eigentlichen Installationsdateien von Microsoft ergänzt werden.

3. Vervollständigung der OS-Pakete WinVista

Zur Anpassung von Vista ist die Erstellung eines eigenen PE-Images als „DOS-Ersatz“ notwendig. Dabei muss man einmalig ein eigenes 32-Bit PE und für 64-Bit Systeme ein 64-Bit PE erstellen

"Um eine 64-Bit-Version von Windows zu installieren, müssen Sie eine 64-Bit-Version von Windows PE verwenden. Um eine 32-Bit-Version von Windows zu installieren, müssen Sie eine 32-Bit-Version von Windows PE verwenden."

<http://technet.microsoft.com/de-de/library/cc766093.aspx>

Hierzu benötigen Sie das aktuelle Windows Automated Installation Kit (Windows AIK):

`http://www.microsoft.com/downloads/details.aspx?
displaylang=de&FamilyID=94bb6e34-d890-4932-81a5-5b50c657de08`

Dieses müssen Sie unter Vista (32-Bit oder 64-Bit) installieren.

Sie können das ISO brennen oder z.B. unter VMWARE mounten

3.1. WinVista / Windows 2008 32-Bit Version

3.1.1. Erstellen eines 32-Bit Vista-PE

1. Installation des Windows AIK unter Vista 32-Bit oder Windows 2003 mit Servicepack 2 32-Bit
2. Erstellen einer WinPE Umgebung

Als Administrator Eingabeaufforderung öffnen (start->programme->zubehör->mit rechter Maustaste "Eingabeaufforderung" als Administrator

```
cd „\Program Files\Windows AIK\Tools\PETools“
```

```
copy c:\winpe_x86
```

3. Image mounten mit

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

Mit `Notepad.exe`

`C:\WinPE_x86\mount\Windows\System32\startnet.cmd` editieren: Den Eintrag `wpeinit` entfernen und folgende Zeile einfügen:

```
c:\opsi\startnet.cmd
```

(Hinweis: Die `startnet.cmd` wird vom `opsi-linuxbootimage` in der `winvista.py` erstellt nebst dem `wpeinit`-Aufruf)

Image unmounten

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

Das Verzeichnis `C:\WinPE_x86\ISO` als Verzeichnis `winpe` nach `/opt/pcbin/install/winvista/` bzw. `/opt/pcbin/install/win2008` kopieren, Rechte anpassen:

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

3.1.2. unattend.xml

Die Steuerdatei für die unattended Installation ist die `unattend.xml`, welche unter `/opt/pcbin/install/winvista/opsi` zu finden ist. Mögliche Modifikationen an dieser Datei sollten Sie in ein anders Verzeichnis sichern, da das `opsi` Verzeichnis bei einem Update des `opsi winvista` Paketes überschrieben wird.

Die von uns mitgelieferte `unattend.xml` enthält die Aktivierung des Administrator Accounts mit dem Passwort 'nt123'.

Dokumente zur `Unattend.xml` finden sich nach Installation des WAIK in `\Program Files\Windows\Waik\docs\chms`

3.1.3. Treiber-Integration

Die Treiber-Integration verläuft analog zu dem bisherigen Verfahren: Die Treiber werden unter `/opt/pcbin/install/winvista/drivers/drivers` abgelegt. Danach wird das Script `create_driver_links.py` aufgerufen.

Zu beachten ist, dass nur signierte Treiber verwendet werden können. Die Verwendung von Treiber-Paketen wie z.B. von `driverpacks.net` ist nicht zu empfehlen, da hier vorwiegend Treiber zu finden sind, welche unter Vista nicht funktionieren.

3.2. WinVista / Windows 2008 64-Bit Version

3.2.1. Erstellen eines 64-Bit Vista-PE

1. Installation des Windows AIK unter Vista 64-Bit oder Windows 2003 Server 64-Bit
2. Erstellen einer WinPE Umgebung

Datei `boot.wim` von Winvista-64-Bit-CD auf Festplatte kopieren nach
`C:\winpe_amd64\ISO\sources\`

Schreibschutz von `boot.wim` entfernen

1. Image in `boot.wim` mounten mit

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

`C:\WinPE_amd64\mount\Windows\System32\startnet.cmd` editieren:
Den Eintrag `wpeinit` entfernen und folgende Zeile einfügen:

```
c:\opsi\startnet.cmd
```

(Hinweis: Die `startnet.cmd` wird vom opsi-linuxbootimage in der `winvista.py` erstellt nebst dem `wpeinit`-Aufruf)

Image unmounten

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

! Zusätzlich: Beim 64-Bit Vista muss man auch das Archiv **2** editieren und mounten (evtl. reicht sogar das zweite) mit

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

`C:\WinPE_amd64\mount\Windows\System32\startnet.cmd` editieren:
Den Eintrag `wpeinit` entfernen und folgende Zeile einfügen:

```
c:\opsi\startnet.cmd
```

! Zusätzlich Dateien `setup.exe` und `sources\setup.exe` löschen

Image unmounten

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

Das Verzeichnis `C:\WinPE_amd64\ISO` als `winpe` kopieren nach
`/opt/pcbin/install/winvista-x64/` bzw.
`/opt/pcbin/install/win2008-x64/`

Rechte anpassen:

```
chown -R opsiconfd:pcpatch /opt/pcbin/install/winvista-x64/winpe
```

3.3. Bereitstellung der Installationsmedien

Kopieren der Installations-DVD nach

```
/opt/pcbin/install/winvista/installfiles
```

und Rechte/Eigentümer anpassen:

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

3.4. Log-Dateien der unattended-Installation

- `c:\Windows\Panther\setupact.log`:
Log bis Ende Setup-Phase 4 (läuft unter WinPE)
- `c:\Windows\Panther\setupact.err`:
Fehler-Log bis Ende Setup-Phase 4 (läuft unter WinPE)
- `c:\Windows\Panther\UnattendGC\setupact.log`:
Log ab Specialize-Phase
- `c:\Windows\Panther\UnattendGC\setupact.err`:
Fehler-Log ab Specialize-Phase
- `c:\Windows\System32\Winevt\Logs*`
- `c:\Windows\ntbtlog.txt` (nur nach aktivierter Startprotokollierung)

4. opsi-preloginloader 3.4 (auch) für Vista/Windows-2008

4.1. Übersicht

Aufgrund der Veränderungen von XP zu Vista wurde der opsi-preloginloader (mit Ausnahme des opsi-winst) neu implementiert.

Diese Neuimplementierung wurde in der Sprache Python vorgenommen, in der auch die Serverseite von opsi implementiert ist. Die Installation erfolgt als per py2exe vorkompilierte exe mit den benötigten Bibliotheken. Damit ist die Installation unabhängig von Python-Installationen auf dem Client.

Wesentliche funktionale Neuerungen sind:

- Eventbasierte Steuerung: Es kann auf unterschiedliche Events im System reagiert werden. Dadurch ist ein Start der Installation nicht mehr ausschließlich an ein Reboot bzw. Start des Service gebunden.
- Steuerung über Webservice: Diese Schnittstelle dient zur Zeit zu Wartungszwecken, bietet aber auch in Zukunft die Möglichkeit für z.B. zentral zeitgesteuerte Installationen.
- Remote Konfiguration: Alle wesentlichen Konfigurationsdaten lassen sich zentral über Parameter der 'General config' global oder Client spezifisch steuern.

Der opsi-preloginloader 3.4 besteht aus mehreren Komponenten:

- opsiclientd: Der zentrale Service des Preloginloaders.
- notifier: Fenster zur Information / Kommunikation mit dem Anwender
- opsi-loginblocker: Sperrt Login bis die Installationen abgeschlossen sind

Der alte prelogin.exe basierte Service kann auch noch installiert werden, ist aber abgekündigt und soll daher durch den opsiclientd ersetzt werden.

4. : opsi-preloginloader 3.4 (auch) für Vista/Windows-2008

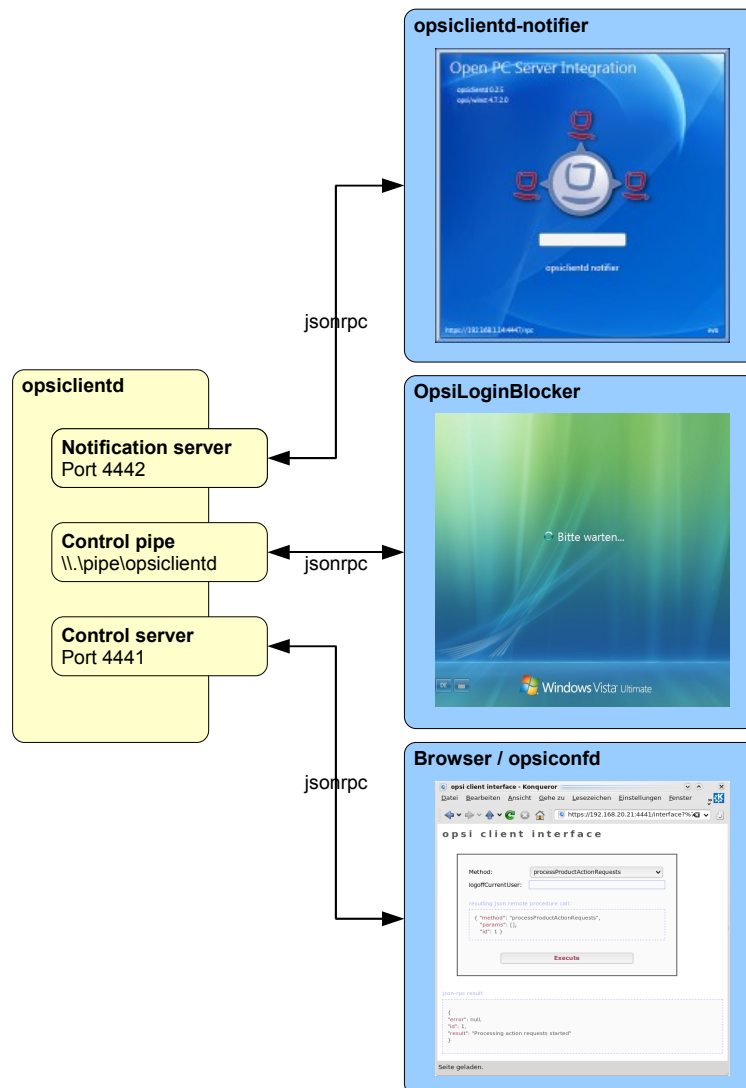


Abbildung 1: Schema der opsi-clientd Komponenten

4.2. Installation

Im Rahmen einer Neuinstallation eines Betriebssystems per unattended Setup mit opsi wird der opsi-preloginloader automatisch korrekt mit installiert.

Zur nachträglichen Installation oder zu Reparaturzwecken kann an dem Zielrechner von einem administrativen Account aus der opt_pcbin share des opsi-Depotservers gemountet und das Script install\preloginloader\service_setup.cmd ausgeführt werden. Da in diesem Fall der UAC nicht an die automatische Installation angepasst ist, muss hier einmalig der Start der service_setup.cmd bestätigt werden.

Die nachträgliche Installation per opsi-deploy-preloginloader funktioniert ebenfalls soweit der Share erreichbar ist.

Auf bereits in opsi integrierten Rechnern kann der preloginloader über den normalen opsi-Prozess eingespielt werden (Action Request=setup). Auf Vista-Rechnern wird dabei gegebenenfalls das Produkt 'preloginvista' ersetzt und auf not_installed gesetzt. Das Produkt 'preloginvista' kann vom opsi-Server gelöscht werden sobald auch die Produkte zur Betriebssysteminstallation (winxpro, winvista,...) auf die aktuellen Versionen upgedated sind.

Der aktuelle preloginloader installiert per default als 'client_servicetype' den 'opsiclientd'. Alternativ kann auch noch der bisherige und abgekündigte 'client_servicetype'='prelogin' verwendet werden. Dieser enthält aber keine aktuellen Erweiterungen und sollte soweit möglich nicht mehr verwendet werden.

Zur Deinstallation kann der preloginloader auf 'uninstall' gesetzt werden.

4.3. opsiclientd

Kernkomponente des preloginloader Paketes ist der Service opsiclientd. Dieser startet als Service beim Boot des Rechners.

Er übernimmt folgende Aufgaben:

- Er baut beim Auftreten der konfigurierten Events Kontakt zum opsi-config-server auf. Konfigurationen und anstehende Action-Requests werden per JSON-RPC abgefragt. Das Default-Event ist dabei 'gui_startup' , welches (wie gewohnt) beim Start des Rechners und vor dem Login aktiv wird.
- Stellt eine Named-Pipe bereit, über diese der opsi-Login-Blocker Kontakt zu ihm aufnehmen kann, um den Zeitpunkt der Freigabe des Logins zu erfragen. Auch diese Kommunikation erfolgt per JSON-RPC.
- Startet den opsiclientd_notifier als Thread zur Interaktion und Kommunikation mit dem Anwender.
- Bei Bedarf Mount des Depotshares, Update und Start des opsi-Winst zum Abarbeiten der anstehenden Action-Requests (Installationen).

4.4. opsiclientd_notifier

Die opsiclientd_notifier realisieren die Interaktion mit dem Anwender. Hier werden sowohl Statusmeldungen des opsiclientd ausgegeben als auch Fragen und Antwortmöglichkeiten, die zur Steuerung des opsiclientd dienen.

4.4.1. opsiclientd_event_notifier

Der *event_notifier* wird aktiv wenn ein Event eintritt und für dieses Event eine **warning_time** > 0 (default = 0) angegeben ist. In diesem Fall wird dem Anwender **warning_time** Sekunden lang ein Hinweis angezeigt mit dem in **message** festgelegten Text und einem Knopf 'Start now'. Steht **user_cancelable** auf true, so wird auch ein 'Abort' Knopf aktiviert. Nach Ablauf der **warning_time** bzw. wenn 'Start now' gewählt wurde, werden die Aktionen und der *action_notifier* gestartet.

Der *event_notifier* ist beim normalen **gui_startup** Event nicht aktiv, ist aber für Events wie **vpn_startup** wichtig.

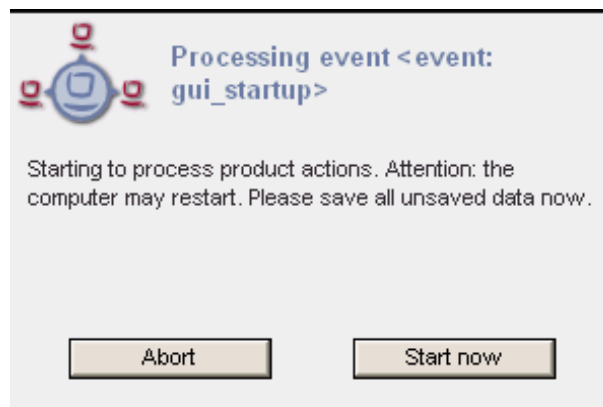


Abbildung 2: opsiclientd event notifier

4.4.2. opsiclientd_action_notifier

Der *action_notifier* informiert über den Fortschritt der Aktionen und gibt (so konfiguriert) die Möglichkeit zum Abbruch.

4. : opsi-preloginloader 3.4 (auch) für Vista/Windows-2008



Abbildung 3: opsiclientd action notifier

4.5. opsi loginblocker

Der opsi loginblocker ist unter Vista als **'credential provider filter'** implementiert. Er blockt alle **'credential provider'** bis zur Freigabe durch den opsiclientd oder dem timeout.

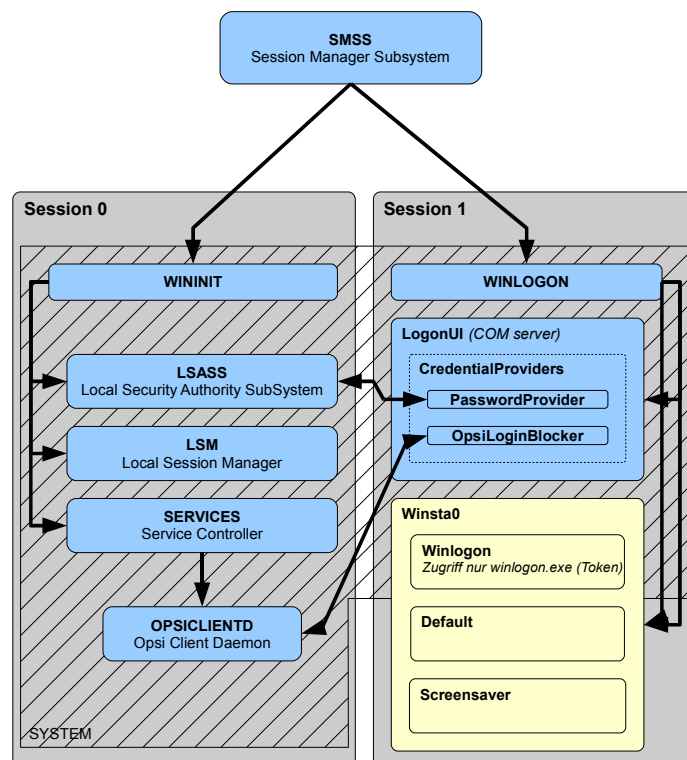


Abbildung 4: Schema opsiclientd und login blocker unter Vista

Der opsi loginblocker ist unter Win2K / Winxp als **'pgina'** implementiert. Er blockt die **msgina.dll** bis zur Freigabe durch den opsiclientd oder dem Timeout.

4.6. Konfiguration

4.6.1. Konfiguration über die Konfigurationsdatei

Die Konfigurationsdatei ist:

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

Die hier festgelegte Konfiguration kann nach erfolgreicher Verbindung zum opsi-configserver durch die dort festgelegte General-Konfiguration überschrieben werden.

Beispiel 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

# Connection timeout.
connection_timeout = 10
```

4. : opsi-preloginloader 3.4 (auch) für Vista/Windows-2008

```
# 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
```

4. : opsi-preloginloader 3.4 (auch) für Vista/Windows-2008

```
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
```

Die oben genannten Timeouts wirken wie folgt zusammen:

1. Tritt ein Event ein, dann zeigt der *event_notifier* **warning_time** Sekunden einen Hinweis und abhängig von **user_cancelable** einen 'Abort' Knopf. Ist die **warning_time = 0** (default) wird kein *event_notifier* angezeigt.
2. Danach werden die definierten Actions gestartet, d. h. in der Regel wird versucht den opsi-Server über die **url** zu erreichen.
3. Ist **user_cancellable_after** Sekunden nach dem Beginn des Versuchs, die Verbindung zum opsi-Server herzustellen, noch keine Verbindung erreicht, so wird ein Abbrechen-Knopf angezeigt. Sobald die Verbindung zum opsi-Server hergestellt ist, ist ein Abbrechen nicht mehr möglich.
4. Kann innerhalb von **connection_timeout** keine Verbindung zum opsi-Server hergestellt werden, so wird abgebrochen. Soll der User keine Möglichkeit zum Abbrechen haben, so muss **user_cancellable_after = connection_timeout** gesetzt werden.

4.6.2. Konfiguration über den Webservice (General config)

Die Konfiguration kann auch zentral gesteuert werden. Hierzu dienen Einträge in der 'general config' des opsi-Servers.

Diese Einträge müssen dem Muster folgen:

```
opsiclientd.<name der section>.<name des keys>
```

Ein Beispiel:

```
opsiclientd.global.log_level = 4
```

setzt in der Konfigurationsdatei opsi-clientd.conf in der Sektion [global] den Wert von log_level auf 4.

4. : opsi-preloginloader 3.4 (auch) für Vista/Windows-2008

Die folgende Abbildung zeigt wie diese Werte als Defaults für alle Clients über den opsi-configed gesetzt werden können.

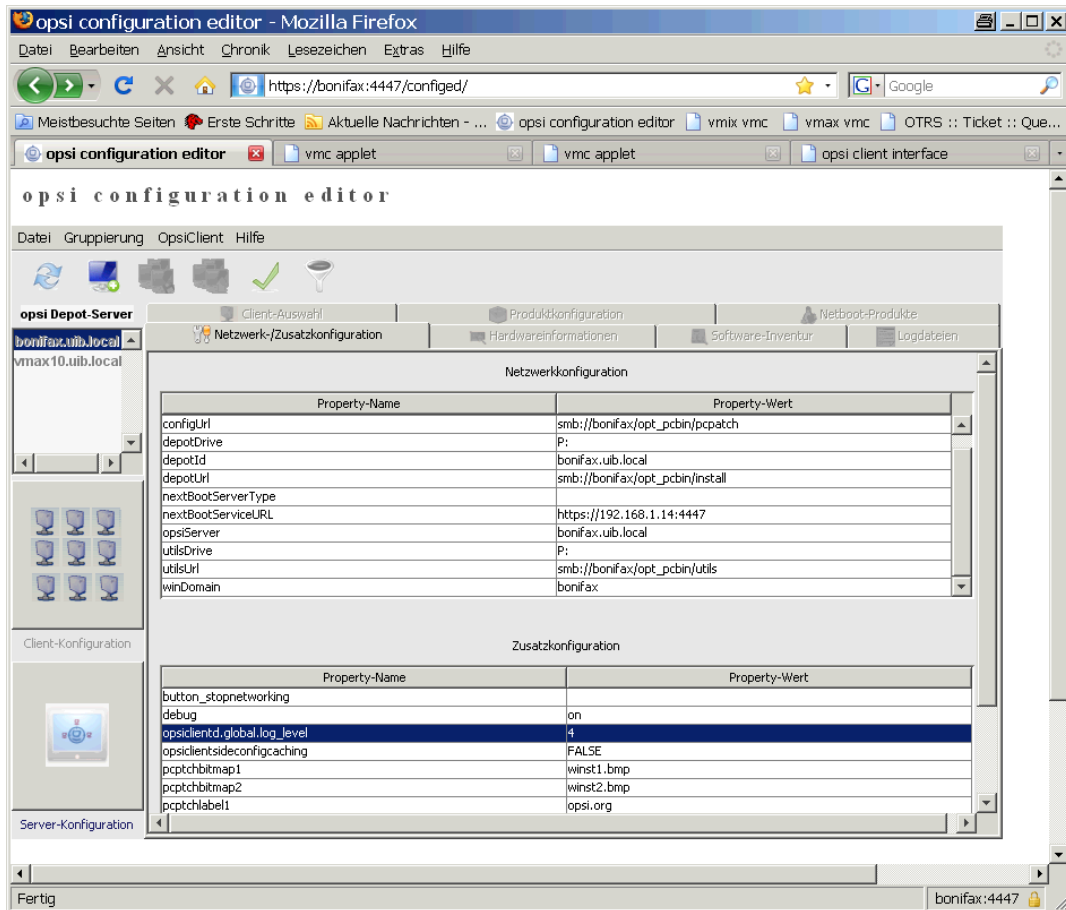


Abbildung 5: Serverweite Konfiguration des opsiclientd über den opsi-configed

Eine Client spezifische Änderung über den opsi-configed ist Moment noch nicht möglich, ist aber in Arbeit. Bis dahin muss eine Client spezifische Änderung direkt im Backend erfolgen.

Hier ein Beispiel für das File31-Backend:

Auszug aus einer <pcname>.ini Datei:

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

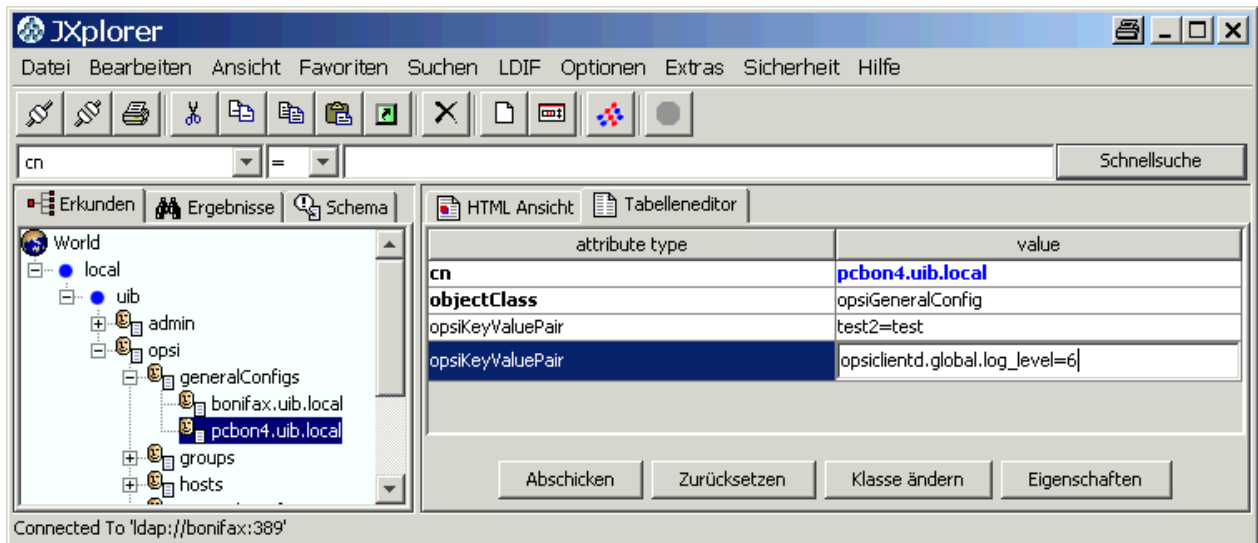


Abbildung 6: Client spezifische Konfiguration des opsiclientd im LDAP-Backend mit JXplorer

Hier ein Beispiel für das LDAP-Backend (mit JXplorer als LDAP-Browser):

4.6.3. Konfiguration unterschiedlicher Events

Folgende Events sind vorgesehen:

- event_gui_startup

Default Event beim Start des Clients - vor dem Login

- event_vpn_startup

Beispiel eines 'custom events' welches über eine per WMI-WQL frei definierbares Event gestartet wird. Hier als Beispiel die Aktivierung eines VPN-Netzwerkinterfaces:

```
wql = SELECT * FROM __InstanceModificationEvent WITHIN 2 WHERE
TargetInstance ISA 'Win32_NetworkAdapter' AND TargetInstance.Name
= "TAP-Win32 Adapter v9" AND TargetInstance.NetConnectionStatus =
2
```

wobei "TAP-Win32 Adapter v9" hier der Name des zu verwendenden VPN-Adapters ist, welcher spezifisch für die verwendete VPN-Software ist.

- event_daemon_startup

Noch nicht implementiert

- event_daemon_shutdown

Noch nicht implementiert

Über den Eintrag 'active = false' kann die Auswertung eines Events unterbunden werden.

4.7. Logging

Der opsiclientd logt nach:

c:\tmp\opsicliend.log

Die Log-Informationen werden auch an den opsi-config-server übertragen. Dort liegen sie unter /var/log/opsi/clientconnect/<pcname>.log. Sie sind im opsi-configed über Logdateien/clientconnect abrufbar.

Jede Zeile in der Logdatei folgt dem Muster:

[<log level>] [<datum zeit>] [Quelle der Meldung] Meldung.

Dabei gibt es die folgenden Loglevel:

```
# 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
```

Beispiel:

```
[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. : opsi-preloginloader 3.4 (auch) für Vista/Windows-2008

```
[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)
```

Sowohl der Vista opsi-Login-Blocker als auch der Win2k/WinXP opsi-Login-Blocker loggen in das Windows Eventlog. Ab einem Log-Level von 8 loggen sie zusätzlich in die Datei c:\tmp\opsi_loginblocker.log.

4.8. Control Server

Über den Control Server Port ist es möglich, steuernd auf den opsiclientd einzuwirken. Dazu muss man sich an diesem Webservice authentifizieren. Dies geschieht entweder mit dem lokalen Administrator Passwort (ein leeres Passwort ist unzulässig) oder mit dem vollständigen Client-Namen (incl. DNS-Domain) und dem Client-Key als Passwort.

4. : opsi-preloginloader 3.4 (auch) für Vista/Windows-2008

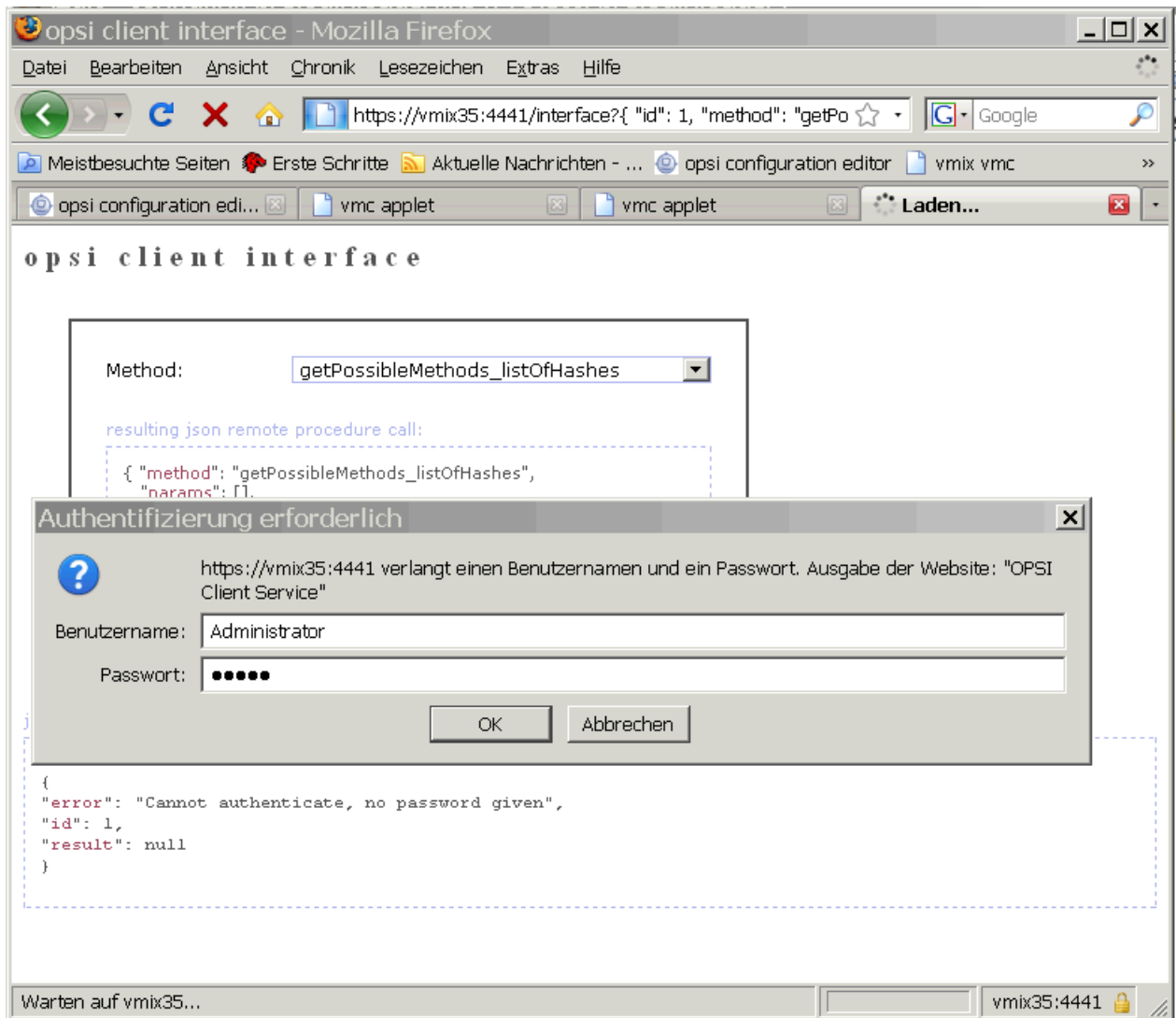


Abbildung 7: Webinterface des Control Servers

Im Moment dient der Control Server nur zu Wartungszwecken.

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

Bislang sind keine Einschränkungen bekannt.

6. Bekannte Besonderheiten der 64 Bit-Unterstützung

opsi-Winst installiert als 32 Bit Programm Scripte, die unter XP laufen, auch in 64Bit Systemen korrekt. Für die Installation von 64 Bit Programmen liefern einige Konstanten wie %ProgramFilesDir% die für 64 Bit Programme falschen Werte. Auch schreiben Registry-Sektionen in den 32-Bit Bereich der Registry.

Um trotzdem vom opsi-winst aus 64-Bit Operationen ausführen zu können, wird bei der Installation des preloginloaders die Datei c:\windows\system32\cmd.exe nach c:\windows\cmd64.exe kopiert. Durch den Aufruf von Scripten mit dieser cmd64.exe im Rahmen von ExecWith Sektionen können 64-Bit Operationen ausgeführt werden. Z.B, lassen sich hierdurch unter Verwendung von des Befehls reg.exe die 64-Bit Zweige der Registry manipulieren.