Dokumentation

Stand: 15.06.2010

opsi
Getting Started

opsi-Version 3.4

Installationsanleitung opsi-Server

Inbetriebnahme

Erste Schritte

open pc server integration:

Stand: 15.06.2010

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

Inhaltsverzeichnis

:

:

1 Einführung

Diese Anleitung beschreibt detailliert die Installation und Inbetriebnahme eines opsi-Servers ausgehend von den zur Verfügung gestellten Installationspaketen bis zur Testinstallation eines Clients.

Die dargestellte Netzwerkkonfiguration ist exemplarisch und bezieht sich auf ein Netz u.a. ohne konkurrierende DHCP-Server (z.B. ein isoliertes Testnetz, in das der opsi-Server und seine Clients für die ersten Versuche gestellt werden können).

Wir empfehlen Ihnen dringend erste Versuche mit opsi in einem Testnetz zu machen, das getrennt von anderen DHCP-Servern ist, welches Sie aber zeitweise an Ihr Hauptnetz ankoppeln können, um z.B. Aktualisierungen und Pakete aus dem Internet laden zu können

Für die Einbindung in bestehende Netze können Sie ggfs. auf Beratungsleistungen z.B. durch uib zurückgreifen.

1.1 Die Schritte der Installation und Inbetriebnahme

Die Installation und Inbetriebnahme eines opsi-Servers erfolgt in drei Schritten:

(A) Grundinstallation des Servers
(B) Anpassung des Servers: Konfiguration des Netzwerks, Passwortvergabe, Aktualisierung des Servers,
(C) Download und Installation der notwendigen opsi-Produkte für die Clients
(D) Vervollständigen der Betriebssystem-Basispakete für Windows von den Original-CDs

Danach kann bereits ein Client automatisch installiert werden.

Für die Grundinstallation existieren mehre Varianten, die je nach Interessen und Vorliebe genutzt werden können.

Die Verfahrensweisen bei den Varianten der Grundinstallation sind in Abschnitt 2 der vorliegenden Anleitung geschildert.

1.2 Hardwarevoraussetzungen

Für den opsi-Server in realer Hardware wird benötigt:

- Intel-x86-kompatibler PC
- von Standard-Linuxkernel unterstützte Netzwerkkarte
- eine Festplatte mit mindestens 16 GB Kapazität
- ein bootfähiges CD-ROM-Laufwerk

Weder im Test- noch im Echtbetrieb sind die Anforderungen an die Leistungsfähigkeit der Maschine besonders hoch.

Bei der Verwendung der VMware-Maschine muss ein angemessener Wirtsrechner genutzt werden. Für eine Teststellung kann problemlos eine weitere VMware-Maschine im selben Wirtsrechner als Client dienen.

2 Installation

2.1 opsi-Server-Grundinstallation

In diesem Abschnitt werden verschiedene Varianten der Realisierung des opsi-Servers dargestellt. Wenn alle Schritte klappen, erhalten Sie ein Serversystem, das bereit für die endgültige Konfiguration und Inbetriebnahme ist. Sie haben also die Wahl und können die Abschnitte überspringen, die die nicht gewählten Wege zum opsi-Server beschreiben.

Wir empfehlen zur Evaluierung von opsi, den Weg über die vorinstallierte VMware-Maschine zu wählen.

Die Grundregel ist, wenn Sie der Reihe nach die Befehle in den

```
grauen Felder
```

ausführen (z.B. per cut&paste aus diesem Dokument holen), dann sollte die Installation auch klappen.

Bei Problemen wenden Sie sich bitte an: https://forum.opsi.org

2.1.1 Inbetriebnahme einer VMware-Maschine

Da die Anforderungen an die Rechengeschwindigkeit eher niedrig sind, lässt sich der opsi-Server auch problemlos als virtuelle Maschine installieren. Für VMware haben wir bereits eine entsprechende Maschine eingerichtet. Die Dateien stehen im Internet zur

Verfügung. Zum Betrieb genügt ein kostenfreier VMware-Player. Sie können diese Maschine aber auch in VMware-Server oder ESX betreiben. Bei ESX verwenden Sie den VMware-converter (eventuell müssen Sie bei ESX nach dem Import auf einen anderen SCSI Controller schalten)

2.1.1.1 Auspacken und erster Start

Sofern Sie bereits über einen Gastrechner verfügen, auf der die VMware-Vollversion oder ein VMware-Player installiert ist, erledigen Sie die Grundinstallation des opsi-Servers mit wenigen Mausklicken:

- Laden Sie die Datei opsi3.4-servervm.zip aus dem Internet.
- Entpacken Sie den Zip-File, das Verzeichnis opsiserver wird erzeugt
- Starten Sie den VMware-Player. Suchen Sie in dessen Dateiauswahldialog das Verzeichnis opsiserver und darin die Datei opsiserver.vmx. Nach dem Aufruf dieser Datei erscheinen eventuell Meldungen, die Sie darauf hinweisen, dass das CDROM- und Diskettenlaufwerk bei Ihnen eine andere Adresse haben als vorinstalliert – das können Sie ignorieren. Der virtuelle Rechner bootet.

Den VMware-Player können Sie für alle gängigen Betriebssysteme kostenfrei bei vmware.com beziehen. Er lässt sich in der Regel problemlos installieren, sofern die Ausstattung des Wirtsrechners insbesondere mit Speicher den Bedarf mehrerer parallel laufender Betriebssysteme abdeckt.

Die von uib bereitgestellte virtuelle Maschine ist unter Linux erstellt. Bestimmte Eigenschaften der von uns verwendeten Host-Systems finden sich in der Konfigurationsdatei opsiserver.vmx. Wenn Sie das opsiserver-Image unter Windows ausführen bzw. auf Ihrem Linux-System Geräte andere Adressen haben, so müssen Sie möglicherweise diese Datei anpassen.

2.1.1.2 Sprachauswahl

Nach dem Start des Systems müssen Sie die gewünschte Sprache auswählen:

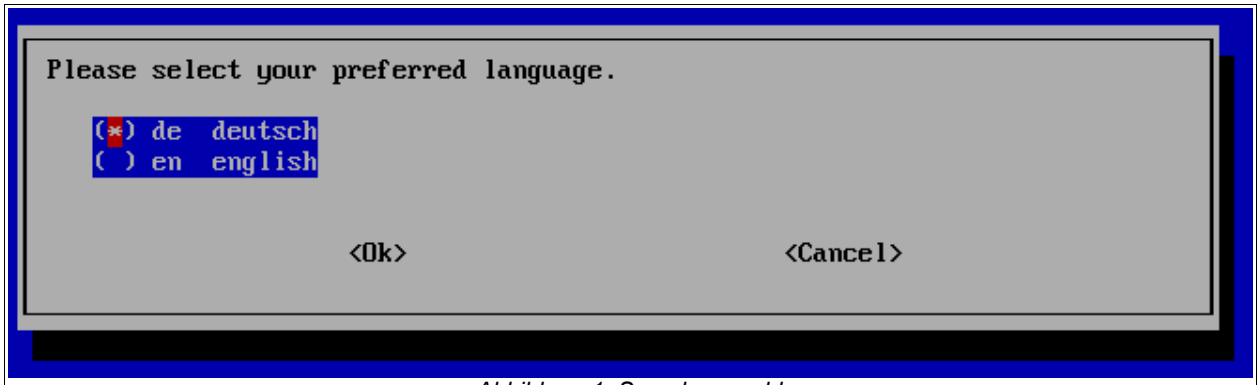

Abbildung 1: Sprachauswahl

2.1.1.3 „1stboot“

Zur Arbeit mit dem opsi-Server ist es von sehr großem Vorteil, wenn dieser direkt mit dem Internet verbunden ist. Zur Netzwerkkonfiguration wird beim ersten Start der VMware-Appliance automatisch das Skript `1stboot.py` aufgerufen.

Wenn Sie das System anders aufgesetzt haben oder einen neuen Anlauf nehmen wollen, können sie `1stboot.py` bzw. `/usr/local/bin/1stboot.py` auch auf der Kommandozeile aufrufen.

Abbildung 2: 1stboot.py Startmaske

Sie werden dann zur Eingabe von Informationen zur Konfiguration des Netzwerkes aufgefordert. Beantworten Sie die Fragen.

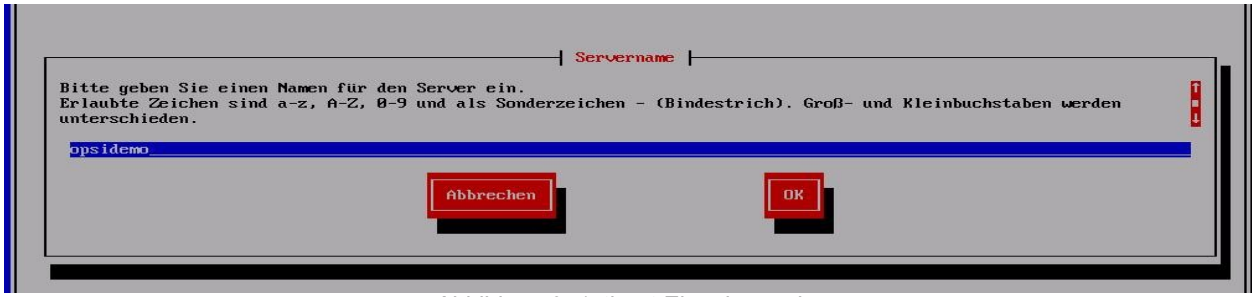

Abbildung 3: 1stboot Eingabemaske

Im Folgenden werden Sie gefragt nach:

Servername:	Name diese Servers (ohne Domain) z.B. **opsidepot**
Domain	DNS-Domain (nicht Windows-Domain, muss einen Punkt enthalten) z.B. opsi.local oder meinefirma.local
IP-Adresse	Adresse dieses Servers z.B. 192.168.1.50
Netzmaske	Netzmaske dieses Servers z.B. 255.255.255.0
Länderkennung	Für die Erstellung des SSL-Zertifikats: Kennung der Nation in 2 Großbuchstaben z.B. DE
Bundeslandkennung:	Für die Erstellung des SSL-Zertifikats: Kennung des Bundeslands z.B. RPL
Stadt	Für die Erstellung des SSL-Zertifikats: Stadt z.B. Mainz
Firma	Für die Erstellung des SSL-Zertifikats: Firma z.B. uib gmbh
Abteilung	Für die Erstellung des SSL-Zertifikats (Optional)
Mail Adresse	Für die Erstellung des SSL-Zertifikats(Optional): Mailadresse
Gateway	IP-Adresse des Internetgateways z.B. 192.168.1.1
Proxy	Soweit zum Internetzugriff benötigt, die Angaben zum Proxy: z.B. http://myuser:mypass@192.168.1.5:8080
DNS-Server	IP-Adresse des Nameservers z.B. 192.168.1.1
Mailrelay	IP-Adresse des Mailservers z.B. 192.168.1.1
Tftpserver:	Als 'TFTP server' geben Sie in der Regel die IP-Nummer des Servers (='IP-Adresse') ein.
Passwort für root	

Nach Abschluss des Programms 1stboot.py wird die VMware-Maschine, sofern Sie automatisch gestartet war, auch automatisch neu gebootet.

Noch ein technischer Hinweis zum Programm 1stboot.py:
Das Programm verwendet Templates um die Konfigurationsdateien zu verändern. Sollten Sie das Programm wiederholt verwenden wollen und Konfigurationsdateien auch von Hand editieren, so finden Sie die templates unter /var/lib/1stboot/templates/.

2.1.1.4 Zweiter Start

Nach dem Neustart bzw. nach Fertigstellen der Netzwerkkonfiguration loggen Sie sich als root mit dem von Ihnen vergebenen Passwort ein.

Sie befinden sich direkt auf der graphischen Oberfläche des opsi-Servers (für sie wird ein Ressourcen schonender Windowmanager verwendet). Zur Begrüßung erscheint ein

„Iceweasel“-Browser-Fenster mit dem Verweis auf das vorliegende Handbuch und weiteren Hinweisen.

Wenn die Meldung erscheint, dass keine Netzwerkverbindung verfügbar ist, kann dies mit der speziellen Start-Konfiguration der VMware-Appliance zusammenhängen. Vor einer weiteren Fehlersuche sollten Sie zunächst probieren, den Server nochmals zu rebooten (z.B. mit dem Ausschaltknopf in der Bedienleiste unten auf der graphischen Oberfläche).

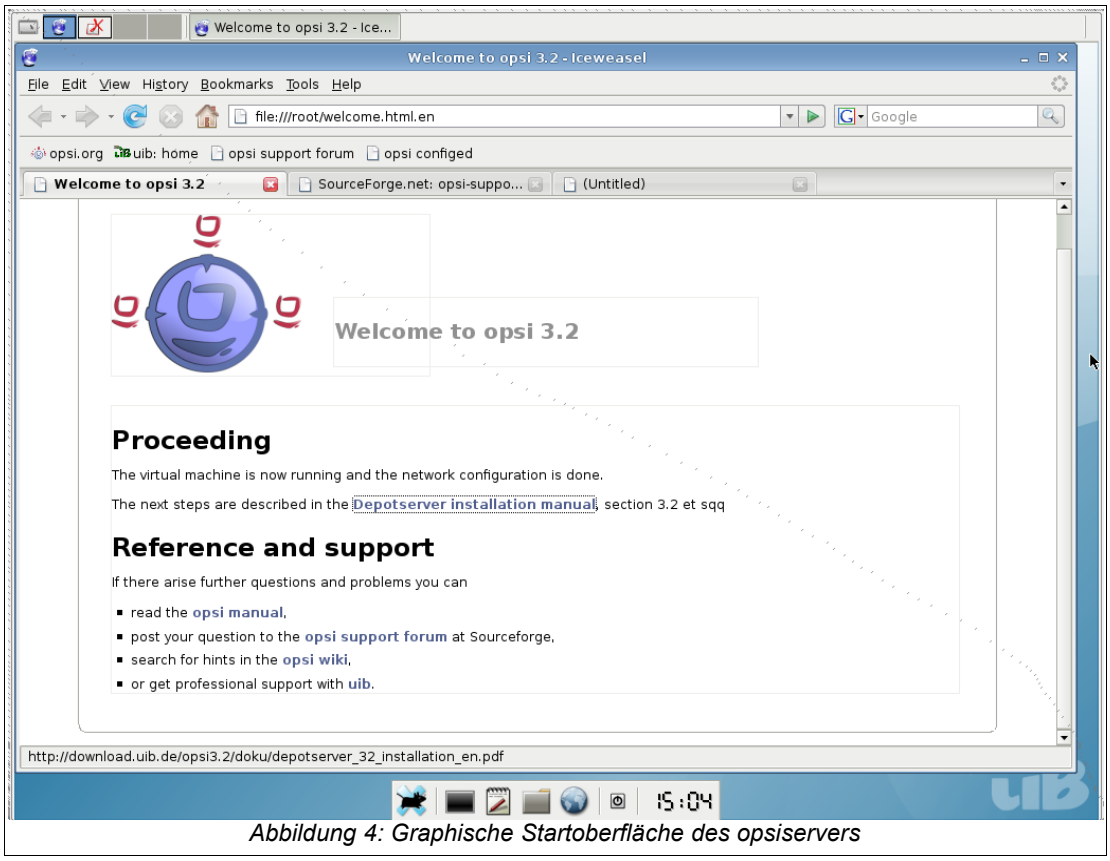

Abbildung 4: Graphische Startoberfläche des opsiservers

Sobald die Netzwerkkonfiguration funktioniert, können Sie auch remote auf den opsi-Server zugreifen:

- Per ssh (in Linux-System stets vorhanden, unter Windows mit putty, s. http://www.chiark.greenend.org.uk/~sgtatham/putty/) kommen Sie auf die Kommandozeile des Servers. Als Benutzernamen verwenden Sie root, Sie authentifizieren sich mit dem Root-Passwort.

- Per vnc (unter Linux z.B. mit in der Regel vncviewer oder krdc, unter Windows z.B. mit ultravnc, http://www.uvnc.com/) können Sie remote eine graphische Oberfläche nutzen. Die vnc-Adresse wird aus der IP-Adresse (oder dem Servernamen bei bereits funktionierender Namensauflösung) und angehängtem „:1“ gebildet. Das Passwort für den vnc-Zugriff lautet „linux123“. Dies sollten Sie baldmöglichst ändern – wie, steht in Abschnitt 2.1.1.6.

2.1.1.5 Terminalfenster

Im Folgenden müssen einige Befehle auf der Kommandozeile eingegeben werden. Bei anderen ist die Eingabe der Befehle auf der Kommandozeile der schnellere Weg zum gewünschten Ergebnis.

Ein Fenster zur Text-Eingabe von Befehlen, d.h. ein Terminalfenster, erhält man auf verschiedenen Wegen:

- Remotezugriff per ssh auf den opsi-Server (s. vorherigen Abschnitt)
- Öffnen eines Terminalfensters in der graphischen Oberfläche (direkt auf dem opsi-Server oder per vnc) durch Klicken auf das Terminal-Icon in der Icon-Leiste der graphischen Oberfläche.
- Öffnen eines Terminalfensters in der graphischen Oberfläche (direkt auf dem opsi-Server oder per vnc) durch Rechtsklick in der Fläche und Auswahl von „Terminal“. Dazu hilfreich: die graphische Oberfläche hat mehrere Arbeitsflächen, erreichbar durch die Auswahl-Schaltflächen in der linken oberen Bildschirmecke.

Besonders vorteilhaft ist es, Befehle aus den Anleitungen, z.B. diesem Handbuch, direkt per Kopieren & Einfügen in ein Terminalfenster zu übertragen, soweit die entsprechende Anwendungsumgebung dies unterstützt.

2.1.1.6 Ändern des VNC-Passwortes

Das Default-Passwort für die vnc-Verbindung lautet „linux123“ - dies müssen Sie auf jeden Fall ändern, am besten sofort:

Öffnen Sie jetzt ein neues Terminalfenster und schreiben Sie:

```
vncpasswd
```

Auf die folgende Frage geben Sie das neue Passwort ein. Es sollte mindestens 8 Zeichen umfassen.

In Produktivsystemen sollte aus Security-Gründen erwogen werden, den vnc-Zugang generell zu sperren oder nur über SSH getunnelt zuzulassen. Wenn vnc für das interne Netz zugelassen ist, muss die Firewall den vnc-Port zum Internet blockieren.

2.1.1.7 Überprüfen und ggfs. korrigieren der Netzwerkanbindung

Wenn die Netzwerkkonfiguration korrekt ist und der Rechner Anbindung an das Internet hat, können Sie mit dem Browser im Startfenster bereits auf eine beliebige Adresse im Internet zugreifen.

Sofern nicht alles funktioniert, öffnen Sie am besten ein Terminalfenster (möglicherweise geht es dann noch nicht remote, sondern nur auf der Server-Oberfläche), und prüfen die Netzwerkanbindung mit den üblichen, hier nicht zu erklärenden Checks.

Sie können im Terminalfenster auch das Kommando

```
1stboot.py
```

aufrufen und die Netzwerkkonfiguration neu eingeben.

Ein Systemneustart wird dann durch den Befehl

```
reboot
```

erzwungen.

Wenn die Netzwerkanbindung funktioniert, können jetzt die opsi-Pakete eingespielt bzw. aktualisiert (Kapitel 2.2) und schließlich die Umgebung für den ersten Installationstest hergestellt werden.

2.1.2 Installation auf einem Debian (Lenny) System per aptitude

In diesem Abschnitt wird davon ausgegangen, dass Sie mit dem Debian-Paketsystem vertraut sind (Informationen zu diesem Thema finden Sie in den einschlägigen Büchern, in den man-pages oder unter http://www.debiananwenderhandbuch.de/).

Bitte beachten Sie, das ein opsi-Server in den Verzeichnissen /opt/pcbin und /var/lib/opsi einen empfohlenen freien Speicher von mindestens 8 GB benötigt.

Wir empfehlen zunächst folgende Installationen:

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

Weiterhin muss Samba installiert sein. Entweder von Debian:

```
aptitude install samba samba-common smbclient smbfs samba-doc
```

oder Sie installieren Samba von den Repositories der Firma Sernet:
Machen Sie folgenden Eintrag in die Datei /etc/apt/sources.list:

```
deb http://ftp.sernet.de/pub/samba/tested/debian lenny main
```

und installieren Samba mit:

```
aptitude update
aptitude install sernet-cifs-mount sernet-samba sernet-samba-doc sernet-smbclient sernet-smbfs
```

Wenn Sie MySQL als Backend für die Inventarisierungsdaten und Lizenzmanagement verwenden möchten benötigen Sie noch mysql-server:

```
aptitude install mysql-server
```

Auf einem Ubuntu Server benötigen Sie noch das Paket: openbsd-inetd.

Prüfen Sie den Eintrag für den opsi-Server in der Datei `/etc/hosts` oder aber die Ausgabe von

```
getent hosts `hostname -f`
```

Das Ergebnis sollte beispielsweise so aussehen:

```
192.168.1.1 server.domain.tld server
```

Sieht das Ergebnis nicht so aus (enthält z.B. 127.0.0.1 oder localhost) dann müssen Sie Ihre /etc/hosts oder Namensauflösung zunächst korrigieren.

Um nun opsi zu installieren tragen Sie in die Datei /etc/apt/sources.list ein:

```
deb http://download.uib.de/debian lenny opsi3.4
```

Führen Sie nun folgende Befehle aus:

```
aptitude update
aptitude remove tftpd
update-inetd --remove tftpd
aptitude install opsi-atftpd
aptitude install opsi-depotserver
aptitude install opsi-configed
```

Bei der Installation des Tftpd werden Sie evtl nach dem Tftp-Basisverzeichnis gefragt. Beantworten Sie diese Frage mit '/tftpboot'. Die Fragen nach Multicast Support können Sie mit 'Nein' beantworten.

Bei der Installation des Paketes opsiconfd werden Sie nach Angaben zur Erstellung eines SSL-Zertifikates gefragt.

Bei der Installation des opsi-Servers werden Sie gefragt, ob die dhcpd.conf und die smb.conf gepatcht werden darf. Beantworten Sie die Fragen mit 'Ja'. Weiterhin werden

Sie nach einem Passwort für den User pcpatch gefragt. Vergeben Sie ein Passwort (und beachten Sie den folgenden Abschnitt zum Ändern der Passwörter).

Da Sie opsi auf einer existierenden Maschine eingespielt haben gehen wir davon aus das Ihre Netzwerkkonfiguration korrekt ist. Machen Sie daher mit dem Punkt 'Aktualisieren und Konfigurieren des opsi-Servers / Überprüfung der Java-Konfiguration' weiter.

2.1.3 Installation auf einem Univention Corporate Server (UCS 2.x)

Getestet und freigegeben für UCS <= 2.2

Die Installation von opsi4ucs ist möglich auf den Rollen Master, Backup und Slave. Memberserver sind nicht geeignet.

Das Paket opsi4ucs-ldap-schema muss auf dem Master eingespielt werden.

Die folgende Dokumentation geht von einer Installation auf dem Master aus:

Tragen Sie in die Datei /etc/apt/sources.list folgendes ein:

```
deb http://apt.univention.de/2.0/unmaintained/ 2.0-0/all/
deb http://apt.univention.de/2.0/unmaintained/ 2.0-0/i386/
deb http://apt.univention.de/2.0/unmaintained/ 2.0-0/amd64/
deb http://apt.univention.de/2.0/unmaintained/ 2.0-0/extern/
deb-src http://apt.univention.de/2.0/unmaintained/ 2.0-0/source/
```

Für UCS 2.1 zusätzlich:

```
deb http://apt.univention.de/2.1/unmaintained/ 2.1-0/all/
deb http://apt.univention.de/2.1/unmaintained/ 2.1-0/i386/
deb http://apt.univention.de/2.1/unmaintained/ 2.1-0/amd64/
deb http://apt.univention.de/2.1/unmaintained/ 2.1-0/extern/
deb-src http://apt.univention.de/2.1/unmaintained/ 2.1-0/source/
```

Für UCS 2.2 zusätzlich:

```
deb http://apt.univention.de/2.2/unmaintained/ 2.2-0/all/
deb http://apt.univention.de/2.2/unmaintained/ 2.2-0/i386/
deb http://apt.univention.de/2.2/unmaintained/ 2.2-0/amd64/
deb http://apt.univention.de/2.2/unmaintained/ 2.2-0/extern/
deb-src http://apt.univention.de/2.2/unmaintained/ 2.2-0/source/
```

Und weiterhin:

```
deb http://download.uib.de/debian ucs2.0 opsi3.4
```

Führen Sie nun folgende Befehle aus:

```
aptitude update
aptitude install opsi4ucs-ldap-schema
aptitude install opsi4ucs
```

Bei der Installation des tftpd werden Sie u.U. nach dem Tftp-Basisverzeichnis gefragt. Beantworten Sie diese Frage mit '/var/lib/univention-client-boot/'. Die Fragen nach Multicast Support können Sie mit 'Nein' beantworten.

Bei der Installation werden u.U. einige Fragen gestellt, siehe 2.1.2 Installation auf einem Debian (Lenny) System per aptitude

Ist die Rolle des UCS eine andere als Master muss nun noch das opsi4ucs Join-Skript ausgeführt werden:

```
univention-join [options]
```

Nun kann, optional, der Opsi-Konfigurations-Editor als Applet auf dem UCS-Server installiert werden.

Führen Sie dafür die folgende Befehle aus:

```
aptitude install opsi-configed
/etc/init.d/opsiconfd restart
```

Das Applet kann nun über die URL https://<servername>:4447/configed aufgerufen werden.

Um den Opsi-Konfigurations-Editor verwenden zu können, muss ein Benutzer Mitglied der Gruppe opsiadmin sein. Die Gruppenzugehörigkeit eines Users kann über Univention-Admin bearbeitet werden.

Bitte Beachten:
Die in den opsi-Handbüchern des öfteren erwähnte Datei
/etc/opsi/backendManager.d/30_vars.conf
wird bei einer opsi4ucs Installation überschrieben durch die Einstellungen aus der Datei
/etc/opsi/backendManager.d/40_opsi4ucs.conf

Da Sie opsi auf einer existierenden Maschine eingespielt haben gehen wir davon aus das Ihre Netzwerkkonfiguration korrekt ist. Machen Sie daher mit dem Punkt 'Aktualisieren und Konfigurieren des opsi-Servers / Überprüfung der Java-Konfiguration' weiter.

2.1.4 Installation unter openSUSE 10.3/11.0

Zunächst ein paar allgemeine Hinweise:

- Diese Pakete sind getestet unter Open-Suse 10.3/11.0. Open Suse 11.1 wird nicht unterstützt (wegen python 2.6)

- Zu Evaluierung von opsi empfiehlt die uib gmbh weiterhin die Verwendung der opsi-VM oder eines Debian oder Ubuntu Systems, da hier die Paketabhängigkeiten viel bequemer aufgelöst werden.
- Support für Suse basierte Systeme leistet die uib gmbh im Rahmen eines Professional Support Vertrages.

Notwendige Vorbereitungen:

- Der Befehl “hostname -f” muss einen fully qualified domainname zurück liefern, in dem mindestens zwei Punkte vorkommen, z.B. opsiserver.uib.local
- Der Befehl “getent hosts `hostname -f`” muss die IP-Adresse der Netzwerkschnittstelle zurück liefern, zu der sich die Clients verbinden sollen. Gibt der Befehl die Adresse 127.0.0.2 aus muss die Datei /etc/hosts korrigiert werden.
- Samba muss konfiguriert sein.
- Soll die Maschine auch als DHCP-Server eingesetzt werden, muss der Daemon dhcpd konfiguriert und am Laufen sein.

Hinzufügen des opsi-SUSE-Repositories per zypper:

```
zypper ar 'http://download.uib.de/suse/opsi3.4' opsi3.4
```

Je nach verwendeter openSUSE-Version sind nun noch einige Paketabhängigkeiten aufzulösen. Die folgenden Pakete sind unter Umständen nicht direkt in der Distribution enthalten:

- python-crypto
- python-mysql
- python-twisted-web2
- python-newt
- duplicity

Die folgenden Community-Repositories enthalten zum heutigen Datum die benötigten Pakete:

openSUSE 10.3

http://download.opensuse.org/repositories/devel:/languages:/python/openSUSE_10.3/
http://download.opensuse.org/distribution/10.3/repo/oss/suse
http://download.opensuse.org/distribution/10.3/repo/non-oss/suse
http://download.opensuse.org/repositories/home:/lrupp:/Factory/openSUSE_10.3
http://download.opensuse.org/repositories/home:/Saviq/openSUSE_10.3
http://download.opensuse.org/repositories/home:/jimfunk/openSUSE_10.3/

openSUSE 11.0

http://download.opensuse.org/repositories/devel:/languages:/python/openSUSE_11.0/
http://download.opensuse.org/repositories/home:/dsbhayangkara/openSUSE_11.0_Update/
http://download.opensuse.org/repositories/home:/Saviq/openSUSE_11.0/

Nach dem Hinzufügen der Repositories kann die Installation gestartet werden:

```
zypper install opsi-depotserver
zypper install opsi-configed
rcopsiconfd restart
rcopsipxeconfd restart
```

Da Sie opsi auf einer existierenden Maschine eingespielt haben gehen wir davon aus das Ihre Netzwerkkonfiguration korrekt ist. Machen Sie daher mit dem Punkt 'Aktualisieren und Konfigurieren des opsi-Servers / Überprüfung der Java-Konfiguration' weiter. Achtung: Die verwendeten Unix-Befehle in den folgenden Kapiteln beziehen sich auf Debian-Systeme, sie müssen Sie durch die entsprechenden Suse-Befehle ersetzen.

2.2 Aktualisieren und Konfigurieren des opsi-Servers

2.2.1 Proxy-Eintrag in apt-Konfiguration

Sofern für Ihren Internet-Zugang erforderlich, passen Sie die Datei

```
/etc/apt/apt.conf
```

an Ihre Netzwerkgegebenheiten an (richtigen Proxy eintragen oder Zeile auskommentieren / löschen). Eine Datei können Sie editieren z. B. mithilfe des Programms „midnight commander“:

```
mcedit /etc/apt/apt.conf
```

2.2.2 Aktualisierung des opsi-Servers

Bringen Sie den opsi-Server auf den letzten Stand, in dem Sie nacheinander in einem Terminalfenster die folgenden Kommandos aufrufen:

```
aptitude update
aptitude safe-upgrade
```

Sofern die Version des opsi-Servers sich nicht geändert hat, sollte die Aktualisierung unproblematisch sein. Wie opsi-Server von früheren Versionen nachgezogen werden können, ist im Anhang beschrieben.

2.2.3 Ändern von Passwörtern

Auf dem System ist ein Pseudo-User 'pcpatch' eingerichtet. Die PCs melden sich zwecks Installation von Softwarepaketen als dieser User an und haben dann Zugriff auf die Installationsdateien auf den hierfür vorgesehenen Shares. Der User 'pcpatch' muss mit korrektem Passwort – gleichzeitig als System-User, als Samba-User und als opsi-User – eingerichtet werden.

Rufen Sie in einem Terminalfenster das Programm opsi-admin mit der Option zum Setzen des pcpatch-Passwortes (in einem für opsi, Unix und Samba).

```
opsi-admin -d task setPcpatchPassword
```

Nach „Abschicken“ des Befehls werden Sie zur Passworteingabe aufgefordert.

Sofern Sie das Passwort für den Zugang zum vnc-Dienst noch nicht eingerichtet bzw. geändert haben, sollten Sie dies aus Sicherheitsgründen jetzt tun (s. Abschnitt 2.1.1.6).

2.2.4 Überprüfung der Java-Konfiguration

Zur Verwaltung des opsi-Servers und der angeschlossenen Clients dient das Programm opsi-configed. Dieses Programm ist in Java geschrieben und benötigt mindestens Java Version 6 bzw. entsprechend der alten Zählung Version 1.6.

Kontrollieren Sie, ob Java in der benötigten Version installiert ist, indem Sie in einem Terminal

```
java -version
```

aufrufen.

Sollte hier nicht mindestens java version „1.6.0“ angezeigt werden, so müssen Sie dies in einem Terminalfenster mit **`update-alternatives`** anpassen:

```
update-alternatives --config java

There are 3 alternatives which provide `java'.
  Selection    Alternative
-----------------------------------------------
 +    1        /usr/lib/j2se/1.4/bin/java
*     2        /usr/lib/j2sdk1.5-sun/bin/java
      3        /usr/lib/j2re1.6-sun/bin/java

Press enter to keep the default[*], or type selection number: 3
Using `/usr/lib/j2re1.6-sun/bin/java' to provide `java'.
```

2.2.5 User einrichten und Gruppen opsiadmin und pcpatch pflegen

Die Administration von opsi ist nur Benutzern gestattet, die Mitglied der Unix-Gruppe opsiadmin sind.

In der vorkonfigurierten VMware-Maschine ist nur root Mitglied dieser Gruppe.

Im folgenden wird als Beispiel der neue Benutzer 'adminuser' so eingerichtet, wie Sie sich einrichten sollten.

Zunächst wird der user erstellt:

```
useradd -m -s /bin/bash adminuser
```

Wir vergeben nun Passwörter für unix:

```
passwd adminuser
```

und für samba

```
smbpasswd -a adminuser
```

Nun wird die Gruppenmitgliedschaft eingerichtet und getestet mit der Befehlsfolge:

```
adduser adminuser opsiadmin
grep opsiadmin /etc/group
```

Der grep-Befehl sollte dann so etwas ausgeben wie:

```
opsiadmin:x:993:root,adminuser
```

Wenn root nicht Mitglied von opsiadmin, kann er auch nicht die opsi-Administrationskommandos ausführen!

Alle User, die Produkte packen (opsi-makeproductfile), installieren (opsi-package-manager) oder Konfigurationsdateien manuell bearbeiten wollen, müssen zusätzlich in der Gruppe pcpatch sein:

```
adduser adminuser pcpatch
```

Der Test

```
grep pcpatch /etc/group
```

ergibt

```
pcpatch:x:992:adminuser
```

root darf dies alles ohnehin und muss daher nicht explizit in die Gruppe aufgenommen werden.

2.3 DHCP-Konfiguration

2.3.1 Wichtig

Eine korrekt funktionierende Namensauflösung und DHCP ist für das Funktionieren von opsi essentiell. Um die Installation zu vereinfachen ist die von uib bereitgestellte VM schon mit einem DHCP-Server ausgestattet. Auf der anderen Seite ist im Produktivbetrieb in der Regel ein DHCP-Server schon vorhanden, der weiter genutzt werden soll. Daher werden im folgenden beide Alternativen beschrieben.

2.3.2 Alternative: DHCP auf dem opsi-Server

Der DHCP-Server auf der opsi-Server VM ist so konfiguriert, das er keine freien leases hat, also keine IP-Nummern an unbekannte Rechner vergibt. Wenn Sie im opsi-configed einen Client erzeugen, müssen Sie daher IP-Nummer und MAC-Adresse angeben, da diese in die **`/etc/dhcp3/dhcpd.conf`** eingetragen und der dhcpd neu gestartet werden.

2.3.3 Alternative:externer DHCP-Server

Da der DHCP-Server keine IP-Nummern an unbekannte Rechner vergibt, ist er nicht direkt störend. Trotzdem empfiehlt es sich, ihn zu deaktivieren. Dazu führen Sie folgende Befehle aus:

```
/etc/init.d/dhcp3-server stop
update-rc.d -f dhcp3-server remove
update-rc.d dhcp3-server stop 20 2 3 4 5 .
```

Nun müssen Sie den externen DHCP-Server so konfigurieren, dass er ein PXE-Boot über den opsi-Server ermöglicht. Wenn Ihr DHCP-Server auf einem Linux läuft, sind folgende Einträge in der **`/etc/dhcp3/dhcpd.conf`** für die Clients notwendig:

```
next-server <ip of opsi-server>;
filename "linux/pxelinux.0";
```

Wobei <ip of opsi-server> durch die IP-Nummer des opsi-Servers zu ersetzten ist.

Bei einem Windows-Server sind die entsprechenden Einträge **`Startserver`** (Option 66) und **`Startfile`** (Option 67).

Wenn Sie im opsi-configed einen Client erzeugen müssen Sie die MAC-Adresse angeben aber keine IP-Nummer.

2.3.4 Überprüfung/Anpassung Backendkonfiguration für DHCP-Nutzung

Je nachdem ob der interne oder ein externer DHCP-Server verwendet wird, muss die Konfiguration von opsi angepasst werden.

In der Datei **/etc/opsi/backendManager.d/30_vars.conf** ist festgelegt, welche Backendmanager von opsi zum Einsatz kommen (**BACKEND_FILE31, BACKEND_FILE, BACKEND_LDAP**).

Im Eintrag **clientManagingBackend** wird u.a. gesteuert, ob der opsi-Server auch die lokale DHCP-Konfiguration – also die Zuweisung von Internet-Adressen zu den Hardware-Adressen der Netzwerkkarten – mit übernimmt. Dies muss so eingerichtet sein, wenn für die opsi-Clients die DHCP-Einträge durch die opsi-Konfigurationsaufrufe erzeugt werden sollen. Der entsprechende Eintrag muss dann lauten:

```
self.clientManagingBackend = [ BACKEND_DHCPD, BACKEND_FILE31 ]
```

Wenn der opsi-Server den DHCP-Dienst nicht bereitstellen soll (weil ein anderer Server im lokalen Netz diese Aufgabe übernimmt und auch für die opsi-Clients gepflegt wird), so wird **BACKEND_DHCPD** nicht benötigt:

```
self.clientManagingBackend = BACKEND_FILE31
```

Nach Anpassung der Backendkonfiguration muss der opsiconfd neu gestartet werden:

```
/etc/init.d/opsiconfd restart
```

2.4 Einspielen / Überprüfen der Freischaltdatei

Auch wenn opsi Opensource ist, so gibt es einige Zusatz-Komponenten die im Rahmen eines Cofinanzierungsprojektes erstellt wurden und evtl. noch nicht Opensource bzw. noch nicht kostenlos sind. Sobald die Entwicklungskosten eingenommen sind, werden auch diese Module Opensource bzw. kostenlos sein. Um bis dahin die Verwendung dieser Module den zahlenden Kunden und zu Evaluierungszwecken zu gestatten, gibt es die Freischaltdatei **/etc/opsi/modules**, welche durch eine elektronische Signatur vor unautorisierter Veränderung geschützt ist. Ist diese Datei nicht vorhanden, so funktionieren nur die 'freien' Module von opsi.

Um zu Evaluierungszwecken eine zeitlich befristet gültige Freischaltdatei zu erhalten, wenden Sie sich an info@uib.de. Im Rahmen einer Beteiligung an den entsprechenden Cofinanzierungsprojekten erhalten Sie eine unbefristet gültige Freischaltdatei. Für die abgeschlossenen Kofinanzierungsprojekte finden Sie eine gültige Freischaltdatei unter http://download.uib.de/opsi3.4/modules. Diese können Sie mit root-Rechten nach /etc/opsi kopieren.

Kontrollieren Sie die Freischaltung mit einer der folgenden Methoden:

Im opsi-configed können Sie über den Menüpunkt Hilfe/opsi-Module sich den Status Ihrer Freischaltung anzeigen lassen.

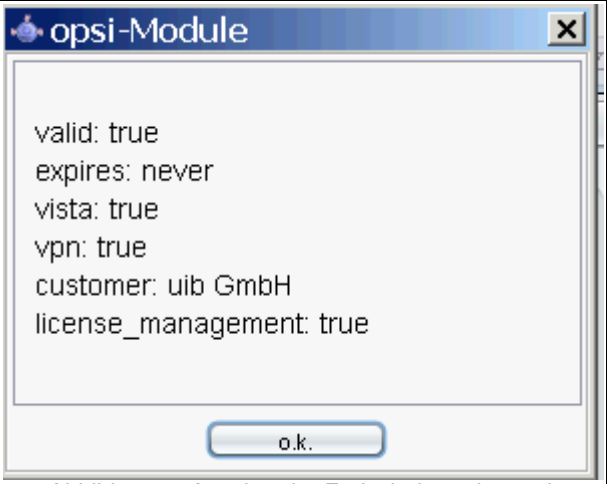

Abbildung 5: Anzeige der Freischaltung im opsi-configed

Mit der Methode **`getOpsiInformation_hash`** können Sie mit opsi-admin überprüfen, welche Module freigeschaltet sind. (Hinweis: Geben Sie die weder die Datei noch die Ausgabe dieses Befehls öffentlich weiter, zumindest nicht ohne die Signatur zu löschen).

```
opsi-admin -d method getOpsiInformation_hash
{
"opsiVersion" : "3.4.0.0",
"modules" :
          {
          "customer" : "uib GmbH",
          "vista" : false,
          "license_management" : true,
          "expires" : "never",
          "valid" : true,
          "signature" : "DIES-IST-KEINE-ECHTE-SIGNATUR",
          "vpn" : true
          }
}
```

2.5 Einspielen der minimalen opsi-Produkte

Holen Sie sich die aktuellen notwendigen opsi-Pakete im opsi-Paketformat.

Die Pakete finden Sie unter http://download.uib.de/opsi3.4/produkte/essential/

Nach dem Download müssen Sie die Pakete auf dem Server mit dem Befehl **`opsi-package-manager -i <paketname>.opsi`** installiert werden

Sie können das interaktiv für jedes einzelne Paket tun. Wir empfehlen Ihnen dies automatisiert zu tun, wie im folgenden beschrieben.

```
cd /home/opsiproducts
wget -r -l1 -nc -nd -A '*.opsi' http://download.uib.de/opsi3.4/produkte/essential
```

Sollte der wget Befehl scheitern, so muss evtl. die Umgebungsvariable http_proxy auf den korrekten Proxystring gesetzt werden (z.B. http_proxy=http://192.168.1.5:8080/)

Die heruntergeladenen Pakete müssen nun auf dem Server installiert werden, damit sie für die Clients bereit stehen. Die interaktive Installation eines opsi-Pakets erfolgt mithilfe des Befehls: **opsi-package-manager -i <paketname>.opsi**

Der folgende (empfohlene) Befehl installiert alle heruntergeladenen Pakete nacheinander und ohne Interaktion:

```
opsi-package-manager -i *.opsi
```

Bitte beachten Sie, dass OS-Installationsprodukte wie winxppro und win2k nach der Installation nicht sofort einsatzbereit sind. Die Installation muss noch durch die Installationsdateien des entsprechenden Installationsmediums ergänzt werden (siehe: Kapitel 3.2.4, Betriebssysteminstallation: Vervollständigen der Basispakete für Windows).

Wenn Sie wollen, können Sie noch weitere opsi-Produkte von download.uib.de herunterladen und auf die gleiche Weise auf Ihrem opsi-Server installieren.

2.6 Start der Management-Oberfläche (opsi-configed)

Opsi bietet mit dem opsi-configed ein komfortables Management Interface.

Sie können es auf mehrere Weisen starten:

- Wenn Sie in einem Browser (irgendwo im Netz) die Adresse **https://<opsidepotserver>:4447/configed** eingeben, erscheint eine Webseite mit als Applet eingebettetem opsi-configed. Damit es funktioniert, müssen Sie auf dem aufrufenden Rechner eine Java-Version >= 1.6 installiert haben.
- Alternativ können Sie auf der graphischen Oberfläche Ihres opsi-Servers (auf dem Server selbst oder im vnc) mit Klick auf die rechte Taste das Kontextmenü öffnen und „opsi config editor“ auswählen.
- Der Konfigurationseditor ist auch Bestandteil der opsi-adminutils, die im Rahmen der opsi-Anwendung lokal auf Clients installiert werden können.

Loggen Sie sich als ein User ein, der Mitglied der Gruppe opsiadmin ist (in der VMware-Appliance als root, solange Sie keinen anderen User eingerichtet haben).

Die Bedienung des Management-Interfaces ist weitgehend selbsterklärend. Daher hier nur ein Hinweis: Änderungen im opsi-Management Interface müssen gespeichert

werden, bevor Sie wirksam werden und Veränderungen der Daten müssen vom Server über 'Daten neu laden' geholt werden.

Sie finden eine ausführliche Anleitung im opsi-Handbuch.

3 Erste Schritte

Nachdem nun der Server installiert und konfiguriert ist folgt die Einbindung von Clients. Hier haben Sie zwei Möglichkeiten:

- Integration vorhandener Windows-Clients in opsi
- Installation eines neuen Windows Rechner über opsi

Beide Wege werden in den folgenden Kapiteln beschrieben. Sie können diese Wege in beliebiger Reihenfolge ausprobieren

3.1 Softwareverteilung

3.1.1 Integration vorhandener Windows-Clients in opsi.

Um vorhandene Windows-Clients in opsi aufzunehmen, muss auf diesen der opsi-preloginloader installiert werden. Dies kann auf mehrere Arten durchgeführt werden. Nach dem Sie wie im Folgenden beschrieben den opsi-preloginloader installiert haben, erscheint der Client auch in der Client-Liste des opsi-configed.

3.1.1.1 Verwendung von service_setup.cmd

Diese Methode dient zur Installation auf einzelnen Rechnern. Für ein Massen Rollout siehe weiter unten.

1. Loggen Sie sich mit administrativen Rechten auf dem Client ein.
2. Mounten Sie den share \\<opsiserver>\opt_pcbin auf einen Laufwerksbuchstaben.
3. Starten Sie das Script install\preloginloader\ service_setup.cmd

4. Das Skript nimmt per opsi-Webservice Kontakt zum Server auf um serverseitig den Client zu erzeugen und den pckey zu erfahren. Dies erfolgt zunächst mit der in der config.ini eingetragenen user/password Kombination. Schlägt dies fehl, so erscheint eine Art Login-Fenster mit Service-URL user und password. Dort kann die Operation mit dem Accountdaten eines Mitglieds der Gruppe opsiadmin authorisiert werden.

Achtung: Der Client rebootet nach der Installation.

3.1.1.2 Verwendung von opsi-deploy-preloginloader

Das opsi-deploy-preloginloader Skript verteilt den opsi-preloginloader direkt vom opsi-server auf die Clients. Voraussetzung hierfür sind bei den Clients:

- ein offener c$ share
- ein offener admin$ share
- ein administrativer account

Das Skript erzeugt serverseitig den Client, kopiert die Installations-Dateien und Konfigurationsinformationen wie den pckey auf den Client und startet dort die Installation.

Mit dem opsi-deploy-preloginloader Skript kann auch eine ganze List von Clients bearbeitet werden. Das Skript findet sich unter /opt/pcbin/install/preloginloader. Führen Sie das Script mit root Rechten aus.

```
bonifax:/home/uib/oertel# cd /opt/pcbin/install/preloginloader
bonifax:/opt/pcbin/install/preloginloader# ./opsi-deploy-preloginloader -h
Usage: opsi-deploy-preloginloader [options] [host]...
Deploy opsi preloginloader to the specified clients.
The c$ and admin$ must be accessable on every client.
Simple File Sharing (Folder Options) should be disabled on the Windows
machine.
Options:
    -h        show this help text
    -V        show version information
    -v        increase verbosity (can be used multiple times)
    -u        username for authentication (default: Administrator)
    -p        password for authentication
    -c        use fqdn instead of hostname for smb/cifs connection
    -r        reboot computer after installation
    -s        shutdown computer after installation
    -f        file containing list of clients (one hostname per line)
```

3.1.2 Erste Produkt-Tests

3.1.2.1 Hardware-Inventarisierung mit dem localboot-Produkten hwaudit und swaudit

Wählen Sie im opsi-configed, Modus Client-Konfiguration, unter dem Reiter Client-Auswahl den betreffenden Client aus. Wenn noch nicht geschehen, aktualisieren Sie den Datenbestand des opsi-configeds mittels Datei/Daten neu laden bzw. Anklicken des entsprechenden Icons. Wechseln Sie zum Reiter Produktkonfiguration, gehen Sie in das Feld "Anstehende Aktion" des Produkts "hwaudit" und wählen Sie in der dort angebotenen Liste die Aktion "setup". Wiederholen Sie das für das Produkt swaudit. Beachten Sie , das auch das Produkt python sich auf setup gestellt hat. Der Haken in der Icon-Menüleiste sollte seine Farbe auf Rot wechseln. Wenn Sie ihn anklicken, werden die neuen Einstellungen zum opsi-Server übermittelt, im Anschluss ist seine Farbe wieder grün.

Booten Sie dann den Client. Er sollte jetzt den opsi-preloginloader starten, die Produkte python, hwaudit und swaudit 'installieren'. Bei hwaudit und swaudit wird werden Hard- bzw Softwareinformationen erhoben und zum opsi-Server übermittelt. Die Informationen sind unter den Tabs "Hardware-Informationen" bzw. "Software-Inventur"zu besichtigen.

3.1.2.2 Hardware-Inventarisierung mit dem netboot-Produkt hwinvent

Sofern Sie bereits einen Client eingerichtet haben und das Produkt hwinvent installiert ist, können Sie bereits etwas Nützliches mit opsi tun:

Wählen Sie im opsi-configed, Modus Client-Konfiguration, unter dem Reiter Client-Auswahl den betreffenden Client aus. Wenn noch nicht geschehen, aktualisieren Sie den Datenbestand des opsi-configeds mittels Datei/Daten neu laden bzw. Anklicken des entsprechenden Icons. Wechseln Sie zum Reiter Netboot-Produkte, gehen Sie in das Feld "Anstehende Aktion" des Produkts "hwinvent" und wählen Sie in der dort angebotenen Liste die Aktion "setup". Der Haken in der Icon-Menüleiste sollte seine Farbe auf Rot wechseln. Wenn Sie ihn anklicken, werden die neuen Einstellungen zum opsi-Server übermittelt, im Anschluss ist seine Farbe wieder grün.

Booten Sie dann den Client. Er sollte jetzt per PXE über das Netz ein Linux-Image ziehen, das die Hardware des PCs scannt und dann den Rechner rebootet (wenn der Rechner nicht ansonsten schon eingerichtet war, kommt im Anschluss korrekterweise die Meldung, dass auf der Platte kein Betriebssystem installiert ist).

Das Ergebnis des Hardware-Scans hat der PC zum opsi-Server übermittelt. Es ist unter dem Reiter "Hardware-Informationen" zu besichtigen.

3.2 Installation eines neuen Windows Rechner über opsi (OS-Installation)

3.2.1 Anlegen eines neuen opsi-Clients über die Management Oberfläche

Als Client-Rechner eignen sich reale oder virtuelle Rechner mit mindestens 512 MB RAM, die über eine Netzwerkkarte mit Netzwerkboot-Unterstützung verfügen: D.h., sie unterstützen das PXE-Protokoll zum Laden von Boot-Systemen via Netzwerk. Der Netzwerkboot ist ggfs. im Bios-Menü zu aktivieren bzw. an die erste Stelle der Bootoptionen zu rücken.

Für einen ersten Tests emfehlen wir eine VMware-Appliance, die einen „nicht installierten Rechner“ abbildet und im VMware-Player laufen kann, Diese könne Sie z.B. bei download.uib.de herunterladen (http://download.uib.de/vmware_pxeclient.zip).

Diese virtuelle Hardware wird auf eher von den Standardtreibern von Windows unterstützt, wenn Sie später eine Testinstallation von Windows durchführen. Zur Installation von Windows auf neueren realen Rechnern müssen Sie sehr wahrscheinlich vorab zusätzliche Treiber integrieren.

Den Client können Sie jetzt mit dem opsi-configed beim opsi-Server registrieren.

Wählen Sie den Menü-Punkt 'OpsiClient/Neuen opsi-Client erstellen' und geben Sie ein:

- IP-Namen,
- (Internet-) Domain (falls abweichend von der Vorgabe),
- Beschreibung (fakultativ);
- IP-Nummer (zwingend, sofern der opsi-Server DHCP-Server ist)
- MAC-Adresse der Netzwerkkarte des Clients (zwingend, sofern der opsi-Server DHCP-Server ist; andernfalls dringend empfohlen)

Nach Eingabeabschluss wird der Client dem opsi-Server bekanntgemacht und gleichzeitig in der DHCP-Konfiguration als PXE-Client angelegt.

Ein Client kann auch auf der Kommandozeile erzeugt werden:

Kommandozeile opsi-admin:

```
opsi-admin -d method createClient <clientname> <domain>\ <description>
<notes>  <ipAddress>  <hardwareAddress>
```

z.B.:

```
opsi-admin -d method createClient pxevm uib.local "Testclient" \  ""
192.168.0.5 00:0c:29:12:34:56
```

Die Liste der eingerichteten opsi-Clients kann jederzeit im opsi-configed, Modus „Client-Konfiguration“, unter dem Reiter Client-Auswahl eingesehen werden.

3.2.2 Hardware-Inventarisierung mit dem netboot-Produkt hwinvent

Sofern Sie bereits einen Client eingerichtet haben und das Produkt hwinvent installiert ist, können Sie bereits etwas Nützliches mit opsi tun:

Wählen Sie im opsi-configed, Modus Client-Konfiguration, unter dem Reiter Client-Auswahl den betreffenden Client aus. Wenn noch nicht geschehen, aktualisieren Sie den Datenbestand des opsi-configeds mittels Datei/Daten neu laden bzw. Anklicken des entsprechenden Icons. Wechseln Sie zum Reiter Netboot-Produkte, gehen Sie in das Feld "Anstehende Aktion" des Produkts "hwinvent" und wählen Sie in der dort angebotenen Liste die Aktion "setup". Der Haken in der Icon-Menüleiste sollte seine Farbe auf Rot wechseln. Wenn Sie ihn anklicken, werden die neuen Einstellungen zum opsi-Server übermittelt, im Anschluss ist seine Farbe wieder grün.

Booten Sie dann den Client. Er sollte jetzt per PXE über das Netz ein Linux-Image ziehen, das die Hardware des PCs scannt und dann den Rechner rebootet (wenn der Rechner nicht ansonsten schon eingerichtet war, kommt im Anschluss korrekterweise die Meldung, dass auf der Platte kein Betriebssystem installiert ist).

Das Ergebnis des Hardware-Scans hat der PC zum opsi-Server übermittelt. Es ist unter dem Reiter "Hardware-Informationen" zu besichtigen.

3.2.3 Anlegen eines neuen opsi-Clients mit Hilfe der opsi-client-bootcd

Auf der Downloadseite von uib finden Sie unter http://download.uib.de/opsi3.4/ verschieden iso-Images der opsi-client-boot-cd. Laden Sie sich das neueste herunter und brennen es auf eine CD. Booten Sie den Rechner von der CD. Sie sollten dann folgendes Bild sehen:

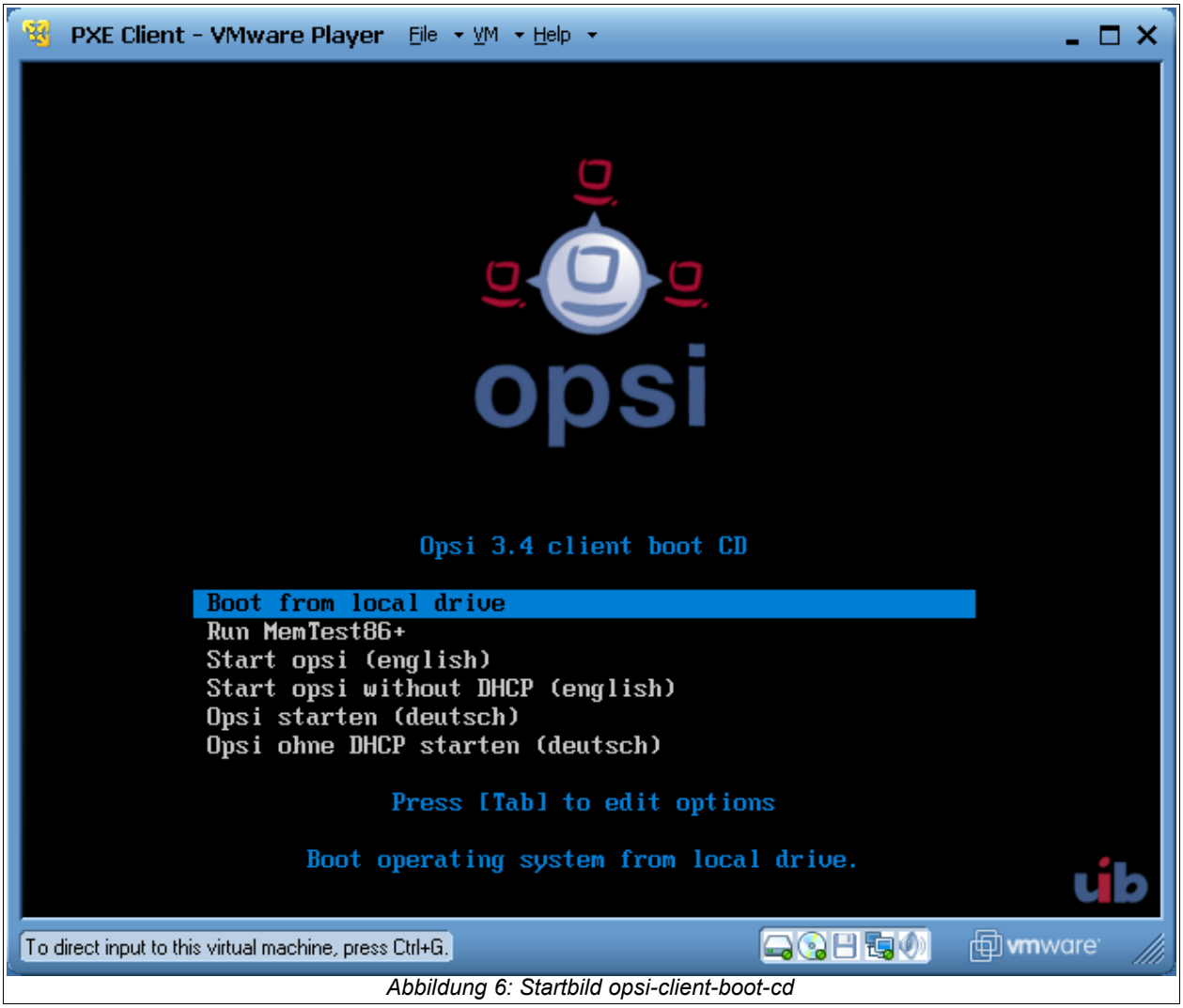

Abbildung 6: Startbild opsi-client-boot-cd

Wählen Sie 'Opsi starten'. Nach einer weile wird folgender Bildschirm erscheinen. Wenn Ihr DHCP-Server IP-Nummern an unbekannte Rechner vergibt ist die Maske schon weitgehend ausgefüllt. Ansonsten müssen Sie es von Hand tun. In der Regel müssen Sie mindestens den Hostnamen vergeben.

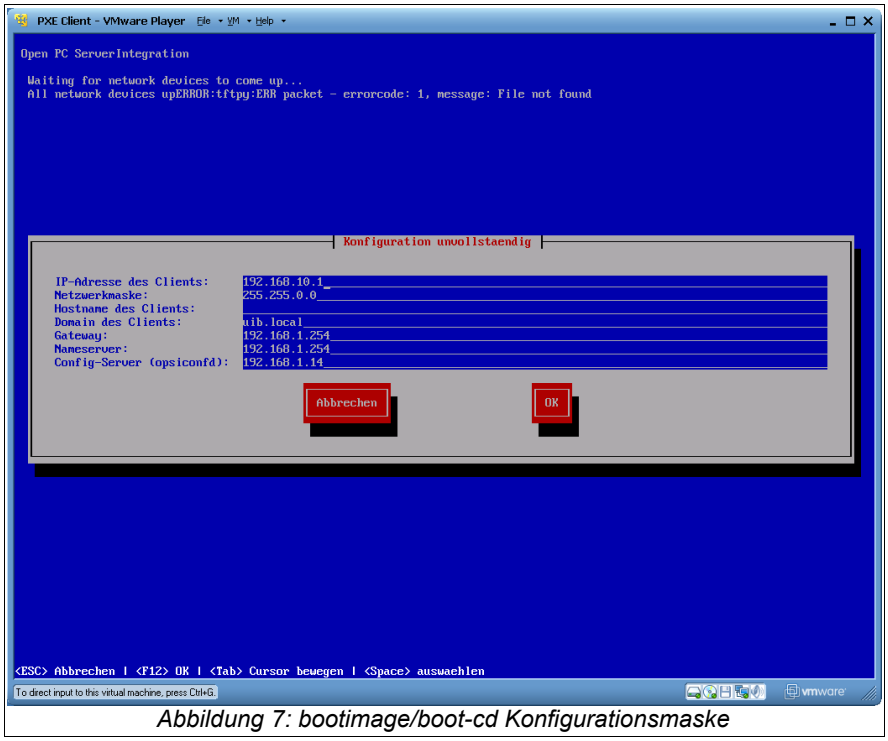

Abbildung 7: bootimage/boot-cd Konfigurationsmaske

Wählen Sie dann 'ok'.

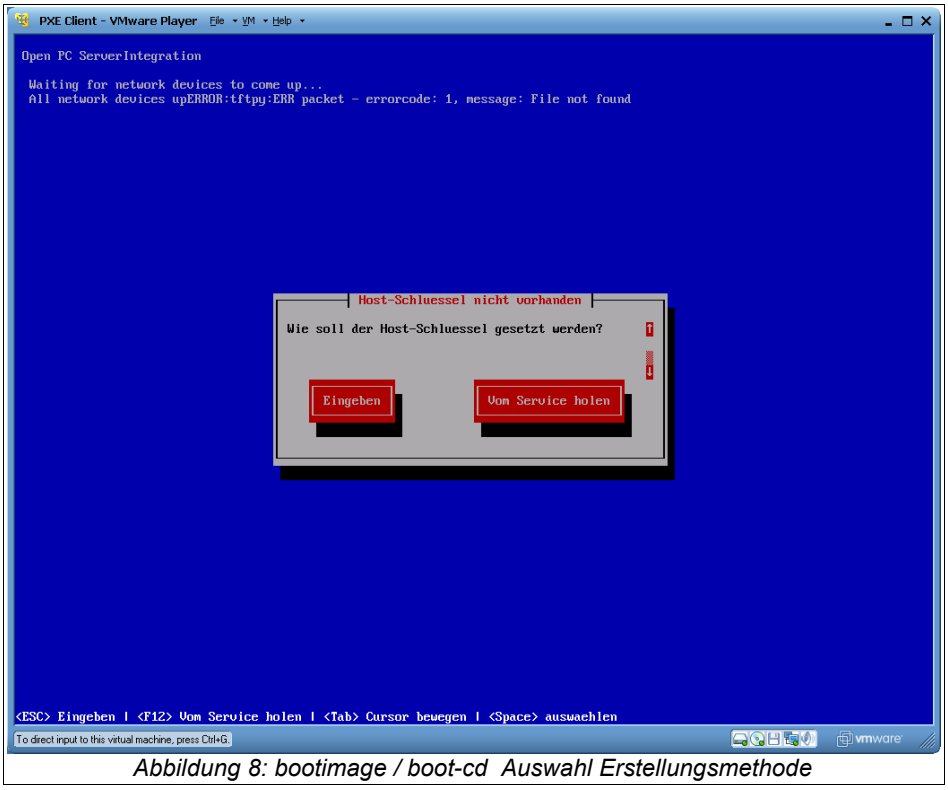

Abbildung 8: bootimage / boot-cd Auswahl Erstellungsmethode

Wählen Sie dann 'Vom Service holen'. Sie erklären damit, dass der Client sich selbst beim opsiserver anmelden und erstellen soll. Dieser Vorgang muss natürlich autorisiert werden.

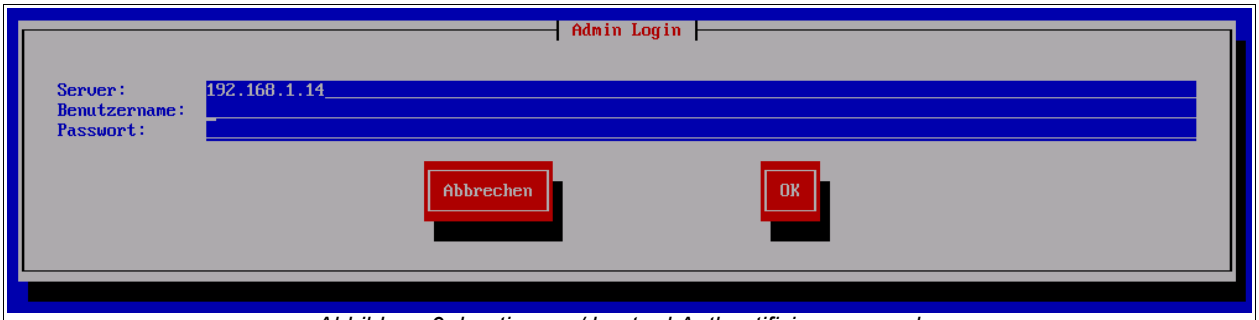

Abbildung 9: bootimage / boot-cd Authentifizierungsmaske

Sie erhalten daher eine Loginmaske, bei der Sie sich als ein Mitglied der Gruppe opsiadmin authentifizieren müssen. Wenn dies Erfolgreich war so teilt der Client dem Server seine Daten mit und der Client wird auf der Serverseite automatisch erstellt. Als nächstes fragt der Client die Liste der verfügbaren netboot Produkte ab und stellt Sie Ihnen zur Auswahl zur Verfügung.

Abbildung 10: bootimage / boot-cd: netboot Produktauswahl

Sie können jetzt direkt das zu installierende Betriebssystem (oder z.B. hwinvent) auswählen.

3.2.4 Betriebssysteminstallation: Vervollständigen der Basispakete für Windows

Zu den zum Download empfohlenen Paketen gehören Pakete wie winxppro, win2003 und win2k zur Installation der Windows-Betriebssysteme. Diese stellen wir nur als Basispakete zur Verfügung, die Dateien zur Automatisierung der Betriebssystem-installation enthalten, jedoch nicht die Dateien des Betriebssystems.

Falls Sie die automatische Windows-Betriebssysteminstallation testen oder verwenden wollen, müssen Sie Ihre eigenen Original-Windows-Installationsdateien kopieren und den Windows-Lizenzschlüssel auf dem Server ablegen.

3.2.5 Windows 2000, XP, 2003

3.2.5.1 Füllen des i368-Verzeichnisses

Kopieren Sie das i386-Verzeichnis einer Installations-CD für Microsoft Win2k/Win2003/WinXP Professional in das Verzeichnis

/opt/pcbin/install/win2k

bzw.

/opt/pcbin/install/win2003

bzw.

/opt/pcbin/install/winxppro

auf dem opsi-Server. Sorgen Sie anschließend noch dafür, dass das Verzeichnis i386 die richtigen Rechte für den Zugriff durch den Installationsprozess hat: Wechseln Sie in das Verzeichnis win2k bzw. winxppro und geben Sie ein

```
chown -R opsiconfd:pcpatch i386/
chmod -R ug+rw i386/
```

Statt vom opsi-Server aus unter Linux können Sie die Dateien auch über das Windows-Netzwerk kopieren. Dafür müssen Sie sich mit der Freigabe “opt_pcbin“ auf dem opsi-Server als Benutzer “pcpatch“ verbinden. Das entsprechende Verzeichnis befindet sich auf der Freigabe in **install\winxppro** bzw. **install\win2k**. bzw. **install\win2003.**

3.2.6 Windows Vista / 2008 / Windows 7

Die Vorbereitung der Installation dieser Systeme ist etwas aufwendiger.

Dazu ist die Erstellung eines eigenen PE-Images als „DOS-Ersatz“ notwendig. Dabei muss man einmalig für 32-Bit Betriebssysteminstallationen ein 32-Bit PE, bzw. für 64-Bit 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 Windows Automated Installation Kit (Windows AIK) und ein in der Liste für unterstützte Betriebssysteme aufgelistetes Windows (die Architektur ist hierbei anscheinend nicht wichtig, beide PEs wurden unter einem 64-Bit Windows 7 erfolgreich erstellt):

http://www.microsoft.com/downloads/details.aspx?familyid=696DD665-9F76-4177-A811-39C26D3B3B34&displaylang=de

Version vom 06.08.2009 in deutscher Sprache.

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

3.2.6.1 Erstellen eines PE

Die Befehle für 32- und 64-Bit sind nahezu identisch, nur müssen die Einträge <ARCH> mit entweder **x86** , **amd64** oder **ia64** ersetzt werden.

1. Erstellen einer WinPE Umgebung:

   Als Administrator die Eingabeaufforderung starten (Start → Programme → Zubehör → mit rechter Maustaste auf "Eingabeaufforderung" → Ausführen als … → Administrator) und folgenden Befehl eingeben:

```
"%ProgramFiles%\Windows AIK\Tools\PETools\copype.cmd" <ARCH> C:\winpe
```

2. Image bearbeiten:

   Als Administrator die Eingabeaufforderung starten und folgenden Befehle eingeben:

```
"%ProgramFiles%\Windows AIK\Tools\<ARCH>\imagex.exe" /mountrw
"C:\winpe\winpe.wim" 1 "C:\winpe\mount"
```

```
echo c:\opsi\startnet.cmd >
"C:\winpe\mount\Windows\System32\startnet.cmd"
```

(Hinweis: **c:\opsi\startnet.cmd** wird vom opsi-linuxbootimage in der **setup.py** erstellt nebst dem oben entfernten **wpeinit**-Aufruf).

```
"%ProgramFiles%\Windows AIK\Tools\<ARCH>\imagex.exe" /commit /unmount
"C:\winpe\mount"
```

```
move "C:\winpe\winpe.wim" "C:\winpe\ISO\sources\boot.wim"
```

3. Das Verzeichnis **C:\winpe\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

3.2.6.2 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**

3.2.6.3 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.

3.2.6.4 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
```

3.2.6.5 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)

3.2.7 Windows-Produktschlüssel

Wenn Sie über das Modul opsi-Lizenzmanagement verfügen, können sie die Windows Lizenzschlüssel über das Lizenzmanagement Modul verwalten. Lesen Sie dazu das Lizenzmanagement Handbuch bzw. das entsprechende Kapitel im opsi-Handbuch.

Haben Sie kein Lizenzmanagement oder wollen dies nicht verwenden, gehen Sie wie folgt vor.

Wenn Sie bereits opsi-Clients eingerichtet haben, können Sie im opsi-Konfigurationseditor einen Windows-Produktschlüssel per Client eintragen:

- einen Client auswählen
- zum Tab Netboot-Produkte wechseln
- dort das Produkt winxppro auswählen
- rechts in der Schalter-Liste in die Property-Zeile productkey gehen
- in das Value-Feld den Schlüssel eintragen
- das Feld verlassen, die Änderungen speichern.

Oder Sie arbeiten auf der Kommandozeile. Ohne Vorgabe eines bestimmten Clients werden gemeinsame Werte für alle Clients abgefragt/gesetzt:

Die vorgegebenen Werte der Produkt-Properties erfahren Sie mit

```
opsi-admin -d method getProductProperties_hash winxppro
```

Korrigieren Sie bei Bedarf, indem Sie den richtigen Productkey einsetzen (alles in eine Zeile!):

```
opsi-admin -d method setProductProperty winxppro "productkey" "ABCDE-FGHIJ-KLMNO-QRTUV-WXYZ1"
```

3.2.8 Start der Windows-Installation

Zum Starten einer Windows-Installation wählen Sie nun im opsi-configed den betreffenden Client aus, setzen unter dem Karteireiter 'Netboot-Produkte' für die gewünschten Betriebssystem (z.B. winxppro) die Aktion auf 'setup' und klicken auf den roten Haken (der wieder grün wird).

Der Client sollte jetzt beim Booten ein Linux Bootimage übers Netz ziehen, in dem Sie nochmal die PC-Neu-Installation bestätigen müssen. Dann sollte alles automatisch weiter laufen, bis schließlich die Logon-Aufforderung des installierten Windows auf dem Bildschirm steht.

3.2.9 Aufbau der Produkte zur unattended Installation

Diese Kapitel betrifft die folgenden Produkte

- win2k
- winxppro
- winvista
- win2003
- win2008

- winvista-x64
- win2008-x64
- win7
- win7-x64
- win2008r2

3.2.9.1 Übersicht des Verzeichnisbaums

```
<productid>┐
           ├─i386/                          NT5 only: Installations files
           ├─installfiles/                  NT6/7 only: Installations files
           ├─winpe/                         NT6/7 only
           ├─opsi/                          Scripte und Templates by opsi.org
           │  ├─$oem$/                      $oem$ gemaess MS
           │  ├─posinst.d/                  Scripte nach OS-Install by opsi.org
           │  └─unattend.txt.template       Template by opsi.org
           ├─custom/                        Scripte und Templates by customer
           │  ├─$oem$/                      $oem$ gemaess MS by customer
           │  ├─posinst.d/                  Scripte nach OS-Install by customer
           │  └─unattend.txt                unattend.txt by customer
           ├─drivers/                       drivers Verzeichnis
           │  ├─drivers/                    drivers Verzeichnis
           │  ├─pciids/                     Symlinkbaum zu Treibern
           │  ├─vendors/                    Symlinkbaum zu Treibern
           │  ├─classes/                    Symlinkbaum zu Treibern
           │  ├─usbids/                     Symlinkbaum zu Treibern
           │  ├─hdaudioids/                 Symlinkbaum zu Treibern
           │  ├─pci.ids                     PCI-IDs DB
           │  └─usb.ids                     USB-IDs DB
           ├─setup.py                       Installationsscript
           ├─<productid>_<version>.control      Meta Daten (nur zur Info)
           ├─<productid>.files              Dateiliste (automatisch erstellt)
           ├─create_driver_links.py         Script zur Treiberverwaltung
           ├─show_drivers.py                Script zur Treiberverwaltung
           ├─download_driver_pack.py        Script zur Treiberverwaltung
           └─extract_driver_pack.py         Script zur Treiberverwaltung
```

3.2.9.2 Die Dateien

•setup.py
Dies ist das Installationsscript, welches vom Bootimage ausgeführt wird.

•<productid>_<version>.control
enthält die Metadaten des Produkts, so wie sie vom Paketierer bereitgestellt wurden. Die Datei liegt hier nur zu Informationszwecken, d.h. Änderungen an dieser Datei haben keinerlei Auswirkungen auf das System.

•<productid>.files
Diese Datei wird automatisch erzeugt und sollte nicht verändert werden.

•create_driver_links.py
show_drivers.py
download_driver_pack.py

extract_driver_pack.py
Dies sind Scripte zur Treiberintegration die im Kapitel Fehler: Referenz nicht gefunden Fehler: Referenz nicht gefunden auf Seite Fehler: Referenz nicht gefunden näher erläutert werden.

3.2.9.3 Verzeichnis i386 / installfiles / winpe

Beachten Sie zu diesen Verzeichnissen die Informationen aus dem Installationshanduch.

•i386
Dieses Verzeichnis enthält den i386 Baum der Windows Installations-CD bei Windows Versionen 2000 bis XP (NT 5.x).
Es kann mehrere i386 Verzeichnisse geben (i386 , i386_en , i386_xxx). Welcher bei der Installation zur Verwendung kommt, wird über das Produktproperty 'i386_dir' gesteuert.

•installfiles
Enthält bei Windows Vista und größer (NT 6.x / 7) den Inhalt der Installations-CD.

•winpe
Enthält bei Windows Vista und größer (NT 6.x / 7) ein bootbares winpe Image.

3.2.9.4 Verzeichnis opsi / custom

Diese beiden Verzeichnisse enthalten Scripte und Konfigurationsdateien zur Steuerung der Betriebssysteminstallation. Während der Installation wirken diese Verzeichnisse zusammen, indem die Dateien aus custom Vorrang haben.

Das Verzeichnis opsi enthält Dateien, die mit Updates jederzeit überspielt werden können. Hier sollten also keine Änderungen vorgenommen werden. Für Anpassungen können Sie Änderungen im Verzeichnis custom vornehmen, welches bei Updates nicht überspielt wird.

Das Unterverzeichnis postinst.d enthält Scripte, welche nach der eigentlichen Installation des Betriebssystems über die posinst.cmd gestartet werden um z.B. den opsi-preloginloader zu installieren. Die Scripte werden dabei in alphabetischer Reihenfolge abgearbeitet. Um die Reihenfolge zu verdeutlichen, fangen die Dateinamen mit einer zweistelligen Nummer an (10_dhcp.cmd). Wollen Sie hier Erweiterungen vornehmen, so können Sie im Verzeichnis custom/posinst.d Scripte mit

Nummern zwischen den vollen 10ern ablegen (13_myscript.cmd). Die vollen 10er sind für die Pflege durch opsi.org/uib reserviert. Das Script 99_cleanup.cmd ist das letzte und endet mit einem Reboot.

3.2.9.5 Verzeichnis drivers

Dieses Verzeichnis dient der Treiberintegration und ist im folgenden Kapitel beschrieben.

3.2.10 Vereinfachte Treiberintegration in die automatische Windowsinstallation

Administriert man einen Pool von PCs, die Geräte besitzen, deren Treiber nicht in der Windows-Standardinstallation enthalten sind, so ist es meist sinnvoll, diese Treiber direkt in die Installation zu integrieren. Bei Netzwerkgeräten kann dies teilweise sogar unumgänglich sein, denn ein startendes Windows ohne Netzwerkkarte ist für den Administrator nicht ohne weiteres erreichbar.

opsi unterstützt Sie durch eine Automatisierung der Treibereinbindung und vereinfacht so die Bereitstellung der Treiber. Dabei müssen die Treiber nur in dem korrekten Verzeichnis abgelegt werden. Durch den Aufruf eines Scriptes werden dann die Treiberverzeichnisse durchsucht und ein Katalog erstellt, anhand dessen das Bootimage automatisch die richtigen Treiber erkennen und einbinden kann. Dabei können sowohl Standard-Treiber , USB-Treiber, HD-Audio-Treiber wie auch Treiber für Festplattencontroller (Textmode Treiber) abgelegt und automatisch eingebunden werden.

Damit die Treiber sofort bei der Windowsinstallation mit installiert werden, müssen Sie in einer bestimmten Form auf dem Server hinterlegt werden. Hierzu sind Treiberverzeichnisse geeignet die eine *.inf-Datei enthalten, die den Treiber für das Windows-Setupprogramm beschreibt. Irgendwelche in setup.exe, *.zip oder anders verpackten Treiber sind hier unbrauchbar. Mit dem Programm 'double driver' (http://www.boozet.org/dd.htm) können Sie von einem installierten Rechner die Treiber im geeigneten Format extrahieren.

Haben sie nur wenige unterschiedliche Hardware zu unterstützen, so können Sie die Treiber bei den Herstellern suchen.

Haben Sie sehr viel unterschiedliche Hardware, so können Sie komplette Packs für XP-Treiber bei http://driverpacks.net/ finden.

Die Driverpacks (!!! ca 2,5 GB !!!) von http://driverpacks.net/DriverPacks/overview.php herunterladen und auf dem Server abspeichern. Mit dem Befehl:
`/opt/pcbin/install/winxppro/extract_driver_pack.py` <pfad zu den komprimierten driverpacks>
werden die Treiber entpackt und in das Verzeichnis winxppro/drivers/drivers/D abgelegt.

Alternativ kann mit dem Befehl:
`/opt/pcbin/install/winxppro/download_driver_pack.py` versucht werden, die Treiber von einem Mirrorserver von driverpacks.net herunter zu laden und zu entpacken.

Struktur des Treiber Verzeichnisses und Ablage der Treiber:

```
/opt/
  └─pcbin/
     └─install/
        └─winxppro/
           └─drivers
              ├─classes/            (Links auf Treiber über Geräteklassen)
              ├─hdaudioids/         (Links auf HD-Audio Treiber)
              ├─pciids/             (Links auf Treiber über PCI-Kennung)
              ├─pci.ids             (PCI Datenbank)
              ├─usbids/             (Links auf Treiber über USB-Kennung)
              ├─usb.ids             (USB Datenbank)
              ├─vendors/            (Links auf Treiber über Hersteller)
              └─drivers             (Platz für allg. Treiber Packs)
                 ├─additional/      (Für Treiber die nicht erkannt werden)
                 ├─buildin/         (Daten aus dem i386 Baum)
                 ├─preferred/       (geprüfte Treiber)
                 └─mydriverpacks/   (Beispiel Treiber Pack)
```

Zusätzliche bzw. geprüfte Treiber gehören in jeweils eigene Verzeichnisse (Name und Tiefe der Verzeichnisstruktur egal) unterhalb des Verzeichnisses **`winxppro/drivers/drivers/preferred`** .

Danach bzw. nach jeder Änderung im **`drivers/drivers`** Verzeichnis rufen Sie im winxppro Verzeichnis das Script **`create_driver_links.py`** auf. Dieses durchsucht die Verzeichnisse unterhalb von **`drivers/drivers`** und erzeugt eine Reihe von Links anhand deren die Zuordnung der Treiber zu bestimmter Hardware (PCI-IDs, USB-IDs, HD-Audio-IDs) zu erkennen ist. Die Treiber aus dem **`preferred`** Verzeichnis werden von dem Script bevorzugt verwendet. Das winxxpro.py Script des Bootimages untersucht die Hardware des zu installierenden Computers und identifiziert die notwendigen Treiber. Diese werden dann auf die Platte kopiert und die unattend.txt entsprechend gepatcht. Das Script **`create_driver_links.py`** durchsucht auch

einmalig den i386 Baum und extrahiert die Inf-Dateien der von Windows mitgelieferten Treiber nach windows_builtin. Sollten Sie am i386-Baum eine Änderung vornehmen (z.B. durch das Einspielen eines Servicepacks) so löschen Sie dieses Verzeichnis und führen **`create_driver_links.py`** erneut aus.

Liegt zu einem Client eine Hardware-Inventarisierung vor, so kann über den Befehl:
`winxppro/show_drivers.py <clientname>`
ausgegeben werden, welche Treiber das Bootimage via PCI-IDs und USB-IDs zur Installation auswählen würde und zu welcher Hardware noch kein Treiber bereit steht.

Zusätzliche Treiber, die unabhängig von ihrer Zuordnung bzw. Erkennung über die PCI- oder USB-IDs installiert werden sollen, gehören in jeweils eigene Verzeichnisse (Name und Tiefe der Verzeichnisstruktur egal) unterhalb des Verzeichnisses **`winxppro/drivers/drivers/additional`** . Über das Produkt-Property 'additional' von winxppro können sie einen oder mehrere Pfade von Treiberverzeichnissen innerhalb von additional einem Client zu ordnen. Im Produkt-Property 'additional-drivers' angegebene Verzeichnisse werden rekursiv durchsucht und alle enthaltenen Treiber eingebunden. Dabei wird auch symbolischen Links gefolgt. Dies können Sie nutzen, um für bestimmte Rechner-Typen ein Verzeichnis zu erstellen (z.B. dell-optiplex-815).

Beispiel einer show_drivers.py Ausgabe:

```
./show_drivers.py pcdummy

PCI-Devices
   [(StandardsystemgerÃ¤te), PCI Standard-PCI-zu-PCI-BrÃ¼cke]
      No driver - device directory
'/opt/pcbin/install/winxppro/drivers/pciids/1022/9602' not found
   [ATI Technologies Inc., Rage Fury Pro (Microsoft Corporation)]
      Using build-in windows driver
   [(Standard-IDE-ATA/ATAPI-Controller), Standard-Zweikanal-PCI-IDE-
Controller]
      /opt/pcbin/install/winxppro/drivers/drivers/D/M/N/123
   [Realtek Semiconductor Corp., Realtek RTL8168C(P)/8111C(P) PCI-E Gigabit
Ethernet NIC]
      /opt/pcbin/install/winxppro/drivers/drivers/preferred/realtek_gigabit_ne
t_8111_8168b
   [IEEE 1394 OHCI-konformer Hostcontroller-Hersteller, OHCI-konformer IEEE
1394-Hostcontroller]
      No driver - device directory
'/opt/pcbin/install/winxppro/drivers/pciids/197B/2380' not found
   [Advanced Micro Devices, Inc., AMD AHCI Compatible RAID Controller]
      /opt/pcbin/install/winxppro/drivers/drivers/preferred/ati_raid_sb7xx
   [(Standard-USB-Hostcontroller), Standard OpenHCD USB-Hostcontroller]
      No driver - device directory
'/opt/pcbin/install/winxppro/drivers/pciids/1002/4397' not found
   [ATI Technologies Inc, ATI SMBus]
      /opt/pcbin/install/winxppro/drivers/drivers/preferred/ati_smbus
```

```
USB-Devices
   [(Standard-USB-Hostcontroller), USB-VerbundgerÃ¤t]
      /opt/pcbin/install/winxppro/drivers/drivers/preferred/brother_844x_pGerb
   [Microsoft, USB-DruckerunterstÃ¼tzung]
      /opt/pcbin/install/winxppro/drivers/drivers/preferred/brother_844x_pGerb

Additional drivers
   [ati_hdaudio_azalia]
     /opt/pcbin/install/winxppro/drivers/drivers/additional/ati_hdaudio_azalia
```

4 Einbindung eigener Software in die Softwareverteilung von opsi

Die Installation von Software erfolgt bei opsi durch den opsi-Clientagenten und insbesondere durch das skript gesteuerte Setup Programm opsi-winst. Daher muss zu jedem opsi-Produkt ein Winst-Script erstellt werden. Danach werden dieses Script, die Installationsdateien und die Metadaten zu einem opsi-Produkt gepackt, welches sich schließlich auf dem opsi-Server installieren lässt.

4.1 Ein kleines Tutorial zur Erstellung eines opsi-Winst Skriptes

4.1.1 Einführung

Diese Tutorial kann keine Schulung oder das Studium der Handbücher ersetzen. Es dient nur dazu eine Einführung zu bekommen. Daher als erstes der Verweis auf weiterführende Quellen:

Schulungen:

Durch uib in Schulungszentren siehe:

http://uib.de/www/opsi/service_support/opsi-kurse/index.html

Durch uib als Inhouse Schulung siehe:

http://uib.de/www/opsi/service_support/support/index.html

Handbücher:

http://download.uib.de/doku

http://download.uib.de/opsi_stable/doku

Besonders wichtig:

Winst-Reference-Card und Winst-Handbuch

Wiki (Scripte, Tips, Links):

http://www.opsi.org/opsi_wiki/OpsiWikiPage

Support Forum:

siehe http://forum.opsi.org

4.1.2 Methoden der nicht interaktiven Softwareinstallation

Prinzipiell gibt es drei Verfahren der Einbindung eines Softwarepaketes in die automatische Softwareverteilung für Windows-Betriebssysteme, zuzüglich einer Variante, die sich auf die Pakete für den Microsoft Installer Service bezieht.

1. **Unattended / Silent Setup**:
   Das Original-Setupprogramm wird verwendet und über Kommandozeilenargumente in einen nicht-interaktiven Modus versetzt. Der wichtigste Spezialfall davon ist der

   „stille“ Aufruf eines MSI-Pakets:
   Ein Paket für den Microsoft Installer Service ist vorhanden und wird mit einer „quiet“-Option aufgerufen.

2. **Interaktives Setup mit automatisierten Antworten**:
   Zur Vorbereitung wird bei einem Lauf des Original-Setupprogramms festgestellt, welche Fenstertitel das Programm zeigt und welche Fragen und Antworten beim Setup anfallen. Dies wird in einem Skript niedergeschrieben. Im Prozess der Softwareverteilung läuft das Setupprogramm dann unter Kontrolle eines Automatisierungs-Programms wie z.B. AutoIt oder Autohotkey, welches das Setupprogramm gemäß dem Skript steuert.

3. **Analysieren und Neu-Paketieren**:
   Es wird (teil-automatisiert) untersucht, welche Komponenten auf einem Testpc,

auf dem nur das Betriebssystem bzw . allgemeine Basissoftware verfügbar ist, installiert werden müssen, damit die Software wie gewünscht läuft. Diese Analyse dient als Basis, um ein neues Verteilungspaket zu bauen. Das Paket kann dabei direkt mit Winst-Mitteln erstellt werden. Es kann aber auch als MSI-Paket ausgeführt werden, das dann in einen beliebigen Verteilungsmechanismus eingebunden werden kann.

opsi unterstützt alle drei Varianten. In der Praxis werden sie häufig ergänzend verwendet.

4.1.3 Struktur eines Winst-Skripts

Zunächst ein Beispiel für ein Script:

```
[Aktionen]
winbatch_tightvnc_silent_install

[winbatch_tightvnc_silent_install]
"%SCRIPTPATH%\tightvnc-1.3.9-setup.exe" /silent
```

Ein Winst-Skript besteht aus primären und sekundären Sektionen. Die primäre Sektion ist hier [Aktionen] die sekundäre ist [Winbatch....].

Die primären Sektionen sind das Hauptprogramm in dem der Ablauf des Scriptes gesteuert wird. Hierzu gibt es:

- Variablen: Strings und Stringlisten
- if else endif Anweisungen
- for Schleifen über Stringlisten
- Funktionen

Die eigentlichen Arbeiten zur Installation werden in themenspezifischen Sekundärsektionen definiert, die von den Primärsektionen aufgerufen werden und jeweils über eine spezielle Syntax verfügen.

4.1.4 Globale Konstanten

Globale Konstanten sind Textplatzhalter, die in primären und sekundären Sektionen eingesetzt werden können und zur Laufzeit textuell durch ihre Werte ersetzt werden.

Beispiele:

- %ProgramFilesDir% c:\programme
- %Systemroot% c:\windows
- %System% c:\winnt\system32
- %Systemdrive% c:\
- %Scriptpath% <Pfad zu dem gerade ausgeführten Script>

4.1.5 Primäre Sektionen:

Initial

Dient dazu Parameter zum Laufzeitverhalten des opsi-Winst zu setzen.

Kann entfallen.

Aktionen / Actions

Die [Actions] Sektion ist das eigentliche Hauptprogramm.

Programmabschnitte, die wiederholt benötigt werden, können in Unterprogramme ('sub sektionen') ausgelagert werden.

Sub-Sektionen

Primäre Sektionen die wiederholt aufgerufen werden oder in externen Dateien liegen.

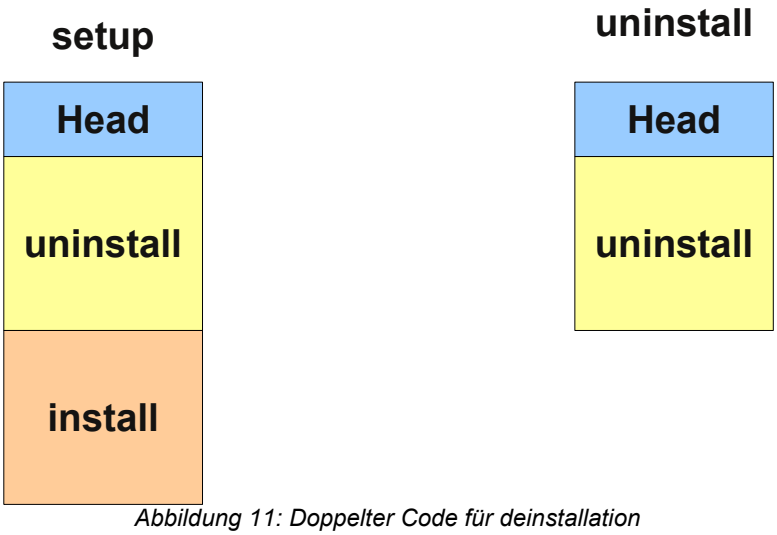

Abbildung 11: Doppelter Code für deinstallation

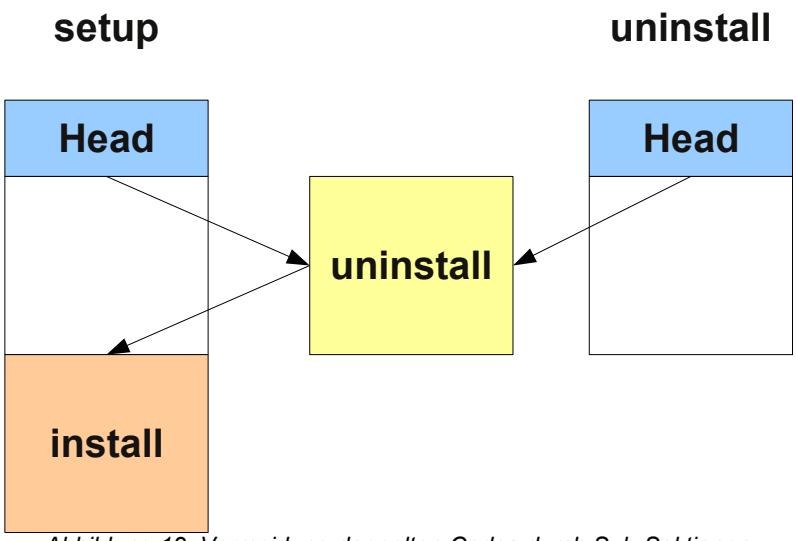

Abbildung 12: Vermeidung doppelten Codes durch Sub-Sektionen

4.1.6 Wichtige sekundäre Sektionen:

Files

Dateioperationen wie

- kopieren (mit Versionskontrolle, rekursiv,....)
- löschen
- Verzeichnisse anlegen
- .....

WinBatch

Dient zum Aufrufen von Programmen über die Windows API. Z. B. Aufrufe von Setupprogrammen im silent mode werden in diesen Sektionen durch geführt.

DosBatch/DosInAnIcon

Der Inhalt dieser Sektionen wird der cmd.exe zur Ausführung übergeben. Hier können also normale Batch-Scripte abgelegt werden.

Eine Variante von DosBatch ist DosInAnIcon, bei dem die cmd.exe mit minimiertem Fenster aufgerufen wird.

ExecWith

Der Inhalt dieser Sektionen wird einem anzugebenden Programm zur Ausführung übergeben. (z.B. Autoit)

Registry

Die Registry-Sektionen dienen zur Manipulation der Registry.

Linkfolder: Startmenü und Desktop

Linkfolder-Sektionen dienen zur Manipulation von Startmenüeinträgen.

4.1.7 Zweites Beispiel: tightvnc

Zur Erläuterung nun ein einfaches Script zur Installation von tightvnc. Eigentlich würde dieses Script mit dem Aufruf der Silentinstallation in der Winbatch-Sektion auskommen. Bei einer wiederholten Installation erscheint hier (wegen der Installation eines laufenden Services trotzdem ein Rückfrage Fenster. Dieses wird (so es auftaucht) mit Hilfe von autoit geschlossen.

tightvnc.ins

```
[Aktionen]
Message "installiere tightvnc 1.3.9 ......"
ExecWith_autoit_confirm "%SCRIPTPATH%\autoit3.exe" WINST /letThemGo
winbatch_tightvnc_silent_install
killtask "autoit3.exe"

[winbatch_tightvnc_silent_install]
"%SCRIPTPATH%\tightvnc-1.3.9-setup.exe" /silent

[ExecWith_autoit_confirm]
; wait for the confirm dialog
; which only appears if tightvnc was installed before as service

; waiting for the window
WinWait("Confirm")
; activating
WinActivate("Confirm")
; say no
```

```
Send("N")
Sleep(500)
;and good bye
Exit
```

4.1.8 Elementare Befehle für primäre Sektionen

String-Variable

Variablen-Deklaration: **DefVar <variable name>**

Variablen-Zuweisung: **Set <variable name> = <value>**

Beispiel:

```
DefVar $ProductId$

set $ProductId$ = "softprod"
```

Wichtig:
Stringvariablen werden in primären und sekundären Sektionen unterschiedlich behandelt. In primären Sektionen sind Stringvariablen eigenständige Objekte. Nur hier können sie deklariert und ihnen Werte zugewiesen werden. Entsprechend ist die Verbindung von Variablen und Strings zu einem Stringausdruck mit einem Operator '+' durchzuführen. Beispiel: **"Installing "+$ProductId$+" ..."**
In sekundären Sektionen werden Stringvariablen vor der Ausführung der Sektion durch den Inhalt der Variable ersetzt.
Beispiel: **"Installing $ProductId$ ..."**
Dies ist zu beachten wenn entsprechende Stringausdrücke per cut+paste im Skript kopiert werden.
Der Vorteil dieser Konstruktion ist, dass in Sektionen die ausserhalb des opsi-winst ausgeführt werden (DosBatch / Execwith) problemlos mit Winst-Variablen gearbeitet werden kann.

Message / showbitmap

Zur Textausgabe während der Installation:

Message <string>

Beispiel:

```
Message "Installing "+$ProductId$+" ..."
```

Zur Ausgabe von Grafik während der Installation:

```
ShowBitMap [<file name>] [<sub titel>]
```

Beispiel:

```
ShowBitmap "%scriptpath%\python.png" "Python"
```

if else endif

Syntax:

if <condition>

;statement(s)

[**else**

;statement(s)]

endif

Funktionen

- **HasMinimumSpace**
  Prüft auf freien Platz auf der Festplatte
- **FileExists**
  prüft auf Existenz einer Datei oder eines Verzeichnisses

Fehler, Logging und Kommentare

- **Kommentarzeichen ';'**
  Zeilen die mit ';' beginnen werden nicht interpretiert
- **comment**
  schreibt Kommentarmeldung in die Logdatei
- **LogError**
  schreibt Fehlermeldung in die Logdatei

- **isFatalError**
  bricht Scriptausführung ab und meldet die Installation als gescheitert

Bedingung zur Ausführung:

- **requiredWinstVersion**
  gibt die (mindestens) benötigte winst Version an

4.1.9 Drittes Beispiel: Standardtemplate 'opsi-template'

Verwenden Sie dieses Template (besser: aktualisierte Versionen von download.uib.de) zur Erstellung Ihrer eigenen Scripte. Das Templatepaket können Sie auf Ihrem Server mit **`opsi-package-manager`** installieren (-i) oder auspacken (-x) um an die Scripte zu gelangen.

Umrandete Befehle gehören in eine (!) Zeile.

setup.ins: Installationsscript

```
; Copyright (c) uib gmbh (www.uib.de)
; This sourcecode is owned by uib
; and published under the Terms of the General Public License.
; credits: http://www.opsi.org/credits/

[Actions]
requiredWinstVersion >= "4.10.5"

DefVar $MsiId$
DefVar $UninstallProgram$
DefVar $LogDir$
DefVar $ProductId$
DefVar $MinimumSpace$
DefVar $InstallDir$
DefVar $ExitCode$
DefVar $LicenseRequired$
DefVar $LicenseKey$
DefVar $LicensePool$

Set $LogDir$ = "%SystemDrive%\tmp"

; ----------------------------------------------------------------
; - Please edit the following values                             -
; ----------------------------------------------------------------
Set $ProductId$       = "opsi-template"
Set $MinimumSpace$    = "1 MB"
; the path were we find the product after the installation
Set $InstallDir$      = "%ProgramFilesDir%\path to the product"
Set $LicenseRequired$ = "false"
Set $LicensePool$     = "p_" + $ProductId$
; ----------------------------------------------------------------
```

```
if not(HasMinimumSpace ("%SystemDrive%", $MinimumSpace$))
      LogError "Not enough space on %SystemDrive%, " + $MinimumSpace$ + " on
drive %SystemDrive% needed for " + $ProductId$
      isFatalError
      ; Stop process and set installation status to failed
else
      comment "Show product picture"
      ShowBitmap "%ScriptPath%\" + $ProductId$ + ".png" $ProductId$

      if FileExists("%ScriptPath%\delsub.ins")
            comment "Start uninstall sub section"
            Sub "%ScriptPath%\delsub.ins"
      endif

      Message "Installing " + $ProductId$ + " ..."

      if $LicenseRequired$ = "true"
            comment "Licensing required, reserve license and get license key"
            Sub_get_licensekey
      endif

      comment "Start setup program"
      Winbatch_install
      Sub_check_exitcode

      comment "Copy files"
      Files_install

      comment "Patch Registry"
      Registry_install

      comment "Create shortcuts"
      LinkFolder_install

      comment "Test for installation success"
      ; Test if software marked as installed in registry
      ; if (GetRegistryStringValue(
"[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\
{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}] DisplayName") = "")
      ;     logError "Fatal: After Installation
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\
{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}] not found"
      ;     isFatalError
      ; else
      ;     comment "Successful Installation"
      ; endif

endif

[Winbatch_install]
; Choose one of the following examples as basis for your installation
; You can use $LicenseKey$ var to pass a license key to the installer
;
; === Nullsoft Scriptable Install System ================================
; "%ScriptPath%\Setup.exe" /S
;
; === MSI package =======================================================
```

```
; You may use the parameter PIDKEY=$Licensekey$
; msiexec /i "%ScriptPath%\some.msi" /l* "$LogDir$\
$ProductId$.install_log.txt" /qb! ALLUSERS=2 REBOOT=ReallySuppress
;
; === InstallShield + MSI=============================================
; Attention: The path to the log file should not contain any whitespaces
; "%ScriptPath%\setup.exe" /s /v" /l* $LogDir$\$ProductId$.install_log.txt
/qb! ALLUSERS=2 REBOOT=ReallySuppress"
; "%ScriptPath%\setup.exe" /s /v" /qb! ALLUSERS=2 REBOOT=ReallySuppress"
;
; === InstallShield ==================================================
; Create setup.iss answer file by running: setup.exe /r /f1"c:\setup.iss"
; "%ScriptPath%\setup.exe" /s /sms /f1"%ScriptPath%\setup.iss" /f2"$LogDir$\
$ProductId$.install_log.txt"
;
; === Inno Setup =====================================================
; http://unattended.sourceforge.net/InnoSetup_Switches_ExitCodes.html
; You may create setup answer file by: setup.exe /SAVEINF="filename"
; You may use an answer file by the parameter /LOADINF="filename"
; "%ScriptPath%\setup.exe" /sp- /silent /norestart

[Files_install]
; Example of recursively copying some files into the installation directory:
;
; copy -s "%ScriptPath%\files\*.*" "$InstallDir$"

[Registry_install]
; Example of setting some values of an registry key:
;
; openkey [HKEY_LOCAL_MACHINE\Software\$ProductId$]
; set "name1" = "some string value"
; set "name2" = REG_DWORD:0001
; set "name3" = REG_BINARY:00 af 99 cd

[LinkFolder_install]
; Example of deleting a folder from AllUsers startmenu:
;
; set_basefolder common_programs
; delete_subfolder $ProductId$
;
; Example of creating an shortcut to the installed exe in AllUsers startmenu:
;
; set_basefolder common_programs
; set_subfolder $ProductId$
;
; set_link
;     name: $ProductId$
;     target: $NewExe$
;     parameters:
;     working_dir: $InstallDir$
;     icon_file:
;     icon_index:
; end_link
;
; Example of creating an shortcut to the installed exe on AllUsers desktop:
;
; set_basefolder common_desktopdirectory
```

```
; set_subfolder ""
;
; set_link
;     name: $ProductId$
;     target: $NewExe$
;     parameters: /some_param
;     working_dir: $InstallDir$
;     icon_file: $NewExe$
;     icon_index: 2
; end_link

[Sub_get_licensekey]
if opsiLicenseManagementEnabled
 comment "License management is enabled and will be used"

 comment "Trying to get a license key"
 Set $LicenseKey$ = demandLicenseKey ($LicensePool$)
 ; If there is an assignment of exactly one licensepool
 ; to the product the following call is possible:
 ; Set $LicenseKey$ = demandLicenseKey ("", $ProductId$)
 ;
 ; If there is an assignment of a license pool
 ; to a windows software id, it is possible to use:
 ; DefVar $WindowsSoftwareId$
 ; $WindowsSoftwareId$ = "..."
 ; Set $LicenseKey$ = demandLicenseKey ("", "", $WindowsSoftwareId$)

 DefVar $ServiceErrorClass$
 set $ServiceErrorClass$ = getLastServiceErrorClass
 comment "Error class: " + $ServiceErrorClass$

 if $ServiceErrorClass$ = "None"
  comment "Everything fine, we got the license key '" + $LicenseKey$ + "'"
 else
  if $ServiceErrorClass$ = "LicenseConfigurationError"
   LogError "Fatal: license configuration must be corrected"
   LogError getLastServiceErrorMessage
   isFatalError
  else
   if $ServiceErrorClass$ = "LicenseMissingError"
    LogError "Fatal: required license is not supplied"
    isFatalError
   endif
  endif
 endif
else
 LogError "Fatal: license required, but license management not enabled"
 isFatalError
endif


[Sub_check_exitcode]
comment "Test for installation success via exit code"
set $ExitCode$ = getLastExitCode
; informations to exit codes see
; http://msdn.microsoft.com/en-us/library/aa372835(VS.85).aspx
; http://msdn.microsoft.com/en-us/library/aa368542.aspx
if ($ExitCode$ = "0")
```

```
 comment "Looks good: setup program gives exitcode zero"
else
 comment "Setup program gives a exitcode unequal zero: " + $ExitCode$
 if ($ExitCode$ = "1605")
  comment "ERROR_UNKNOWN_PRODUCT 1605 This action is only valid for products"
  comment "that are currently installed."
  comment "Uninstall of a not installed product failed - no problem"
 else
  if ($ExitCode$ = "1641")
   comment "looks good: setup program gives exitcode 1641"
   comment "ERROR_SUCCESS_REBOOT_INITIATED 1641 The installer has initiated a
restart. This message is indicative of a success."
  else
   if ($ExitCode$ = "3010")
    comment "looks good: setup program gives exitcode 3010"
    comment "ERROR_SUCCESS_REBOOT_REQUIRED 3010 A restart is required to"
    comment "complete the install. This message is indicative of a success."
   else
    logError "Fatal: Setup program gives an unknown exitcode unequal zero: "
    logError $ExitCode$
    isFatalError
   endif
  endif
 endif
endif
```

delsub.ins: ausgelagerte deinstallations Sektion

```
; Copyright (c) uib gmbh (www.uib.de)
; This sourcecode is owned by uib gmbh
; and published under the Terms of the General Public License.
; credits: http://www.opsi.org/credits/


Set $MsiId$ = '{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}'
Set $UninstallProgram$ = $InstallDir$ + "\uninstall.exe"

Message "Uninstalling " + $ProductId$ + " ..."

if FileExists($UninstallProgram$)
      comment "Uninstall program found, starting uninstall"
      Winbatch_uninstall
      sub_check_exitcode
endif
if not (GetRegistryStringValue(
"[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\" +
$MsiId$ + "] DisplayName") = "")
 comment "MSI id " + $MsiId$ + " found in registry, starting msiexec to
uninstall"
 Winbatch_uninstall_msi
 sub_check_exitcode
endif

comment "Delete files"
Files_uninstall

comment "Cleanup registry"
```

```
Registry_uninstall

comment "Delete program shortcuts"
LinkFolder_uninstall

[Winbatch_uninstall]
; Choose one of the following examples as basis for program uninstall
;
; === Nullsoft Scriptable Install System ================================
; "$UninstallProgram$" /S
;
; === Inno Setup ========================================================
; "$UninstallProgram$" /silent /norestart


[Winbatch_uninstall_msi]
msiexec /x $MsiId$ /qb! REBOOT=ReallySuppress

[Files_uninstall]
; Example for recursively deleting the installation directory
; (don't forget the trailing backslash):
;
; delete -sf "$InstallDir$\"

[Registry_uninstall]
; Example of deleting a registry key:
;
; deletekey [HKEY_LOCAL_MACHINE\Software\$ProductId$]

[LinkFolder_uninstall]
; Example of deleting a folder from AllUsers startmenu:
;
; set_basefolder common_programs
; delete_subfolder $ProductId$
;
; Example of deleting a shortcut from AllUsers desktop:
;
; set_basefolder common_desktopdirectory
; set_subfolder ""
; delete_element $ProductId$


[Sub_check_exitcode]
(.... siehe oben .....)
```

uninstall.ins: Deinstallationsscript

```
; Copyright (c) uib gmbh (www.uib.de)
; This sourcecode is owned by uib gmbh
; and published under the Terms of the General Public License.
; credits: http://www.opsi.org/credits/

[Actions]
requiredWinstVersion >= "4.10.5"

DefVar $MsiId$
DefVar $UninstallProgram$
DefVar $LogDir$
```

```
DefVar $ExitCode$
DefVar $ProductId$
DefVar $InstallDir$
DefVar $LicenseRequired$
DefVar $LicensePool$

Set $LogDir$ = "%SystemDrive%\tmp"

; ----------------------------------------------------------------
; - Please edit the following values                             -
; ----------------------------------------------------------------
Set $ProductId$       = "opsi-template"
Set $InstallDir$      = "%ProgramFilesDir%\path to the product"
Set $LicenseRequired$ = "false"
Set $LicensePool$     = "p_" + $ProductId$
; ----------------------------------------------------------------


comment "Show product picture"
ShowBitmap "%ScriptPath%\" + $ProductId$ + ".png" $ProductId$

Message "Uninstalling " + $ProductId$ + " ..."

if FileExists("%ScriptPath%\delsub.ins")
      comment "Start uninstall sub section"
      Sub "%ScriptPath%\delsub.ins"
endif

if $LicenseRequired$ = "true"
      comment "Licensing required, free license used"
      Sub_free_license
endif

[Sub_free_license]
if opsiLicenseManagementEnabled
      comment "License management is enabled and will be used"

      comment "Trying to free license used for the product"
      DefVar $result$
      Set $result$ = FreeLicense($LicensePool$)
      ; If there is an assignment of a license pool to
      ; the product, it is possible to use
      ; Set $result$ = FreeLicense("", $ProductId$)
      ;
      ; If there is an assignment of a license pool to
      ; a windows software id, it is possible to use
      ; DefVar $WindowsSoftwareId$
      ; $WindowsSoftwareId$ = "..."
      ; set $result$ = FreeLicense("", "", $WindowsSoftwareId$)
else
      LogError "Error: licensing required, but license management not enabled"
      isFatalError
endif
```

4.1.10 Interaktives Erstellen und Testen eines opsi-Winst Scriptes

Sie können ein Script interaktiv anpassen und testen.

Erstellen Sie sich dazu ein Verzeichnis (z.B. c:\test) und kopieren Sie die Scripte des opsi-template (setup.ins, delsub.ins und uninstall.ins) in dieses Verzeichnis.

Starten Sie den Winst per doppelklick. Wenn der opsi-Clientagent bereits auf Ihrem Rechner installiert ist, finden Sie unter C:\Programme\opsi.org\preloginloader\opsi-winst. Wenn nicht kopieren Sie Sich das Verzeichnis opsi-winst vom share \\<opsiserver\opt_pcbin aus dem Verzeichnis install\opsi-winst\files.

Sie sehen dann folgendes Fenster:

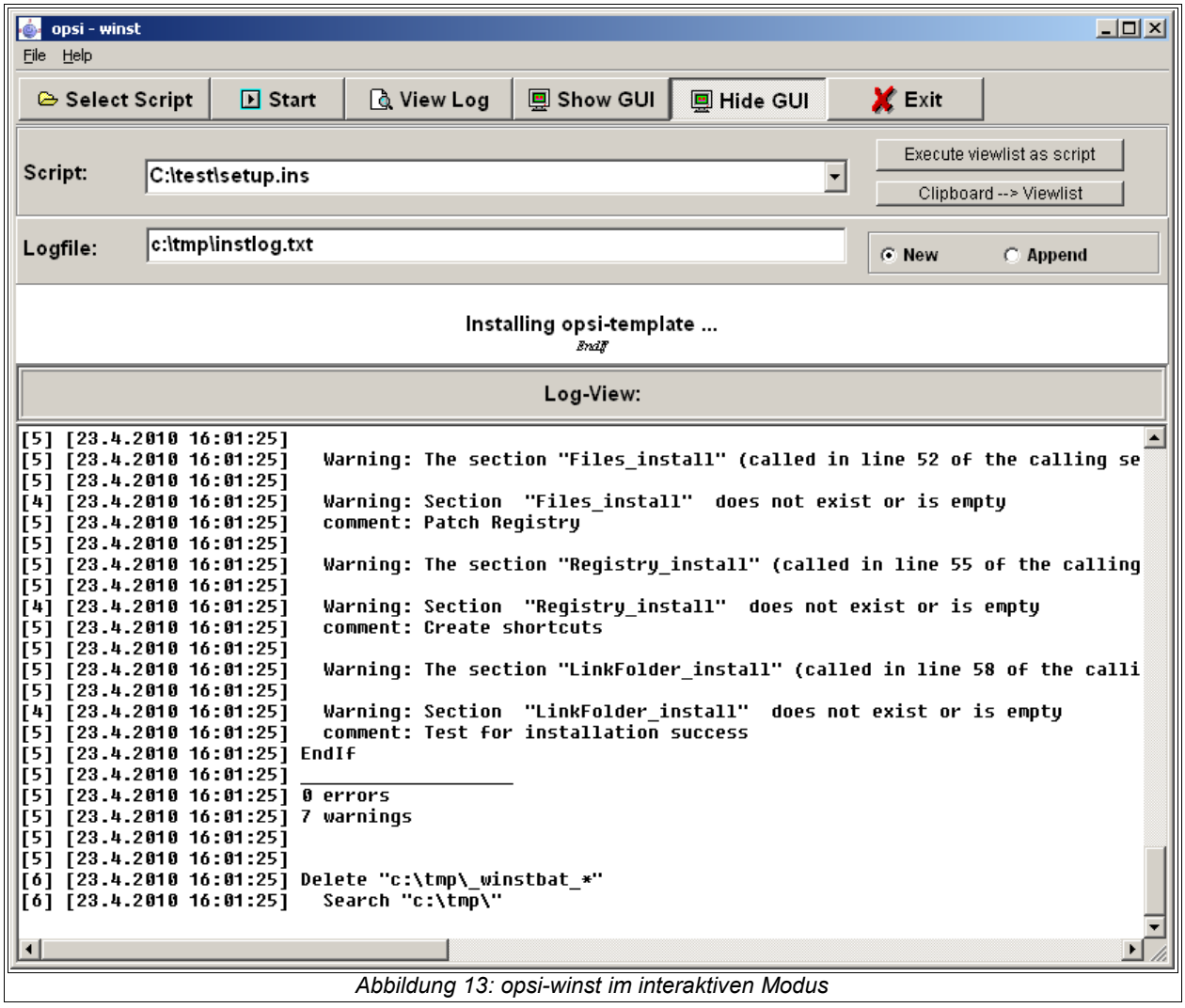

Abbildung 13: opsi-winst im interaktiven Modus

Über 'Select Script' können Sie das Script auswählen, das Sie ausführen möchten. Mit 'Start' können Sie das Script starten. Dabei wird das Script auf diesem Rechner

ausgeführt. Über 'View Log' können Sie sich die Logdatei der Scriptauführung anschauen.

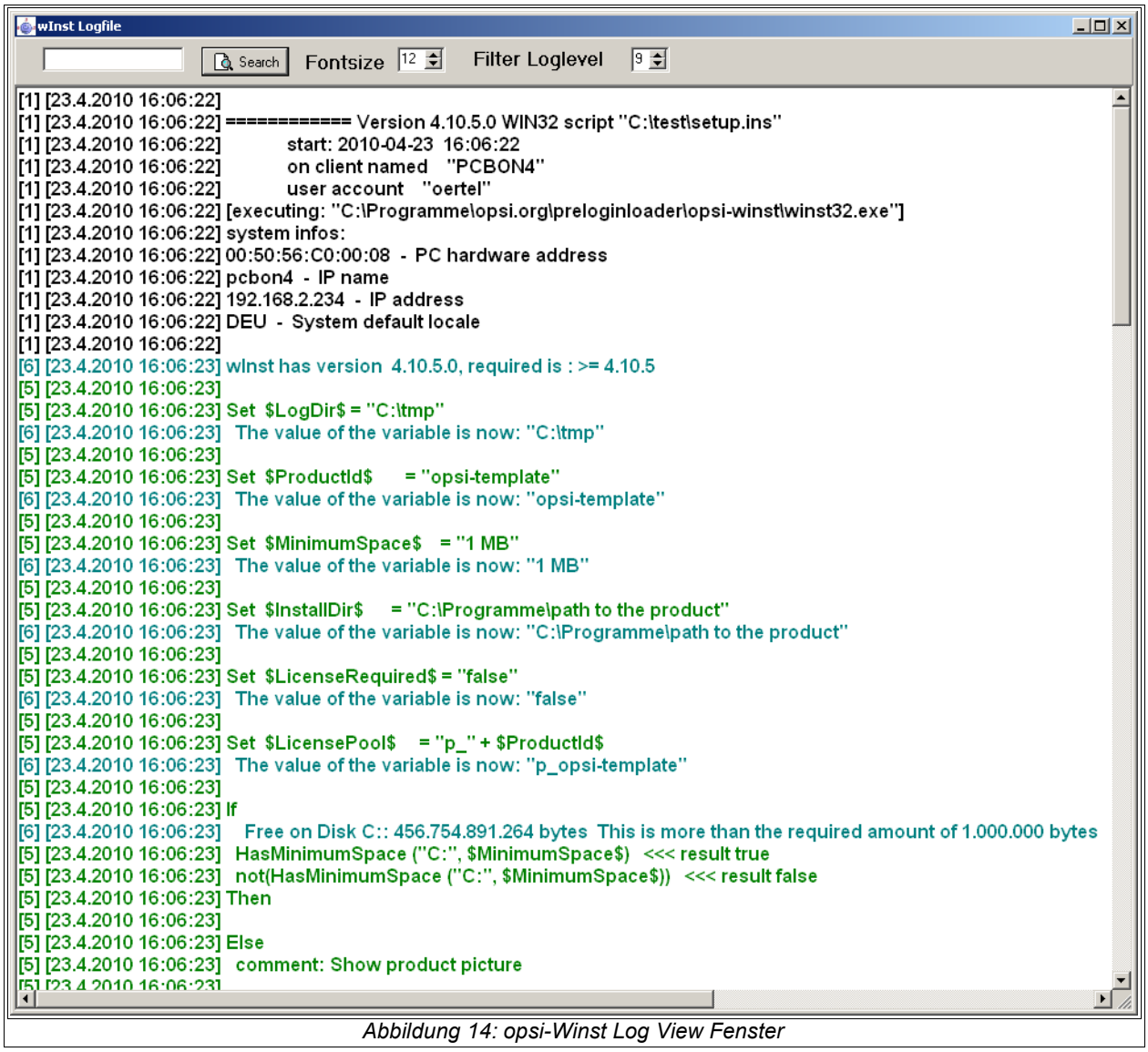

Abbildung 14: opsi-Winst Log View Fenster

Schauen Sie sich anhand der Logdatei an wie der opsi-winst das Script interpretiert.

Kopiern sie die Setup.exe welche Sie installieren wollen in das Verzeichnis in dem die Scripte liegen (z.B. c:\test).

Öffenen Sie das Script setup.ins in einem Editor. Im Prinzip können Sie jeden beliebigen Editor verwenden. Wir empfehlen den jEdit mit opsi-winst Syntax-Highlightning wie Sie ihn in der Grundausstattung der opsi-produkte finden.

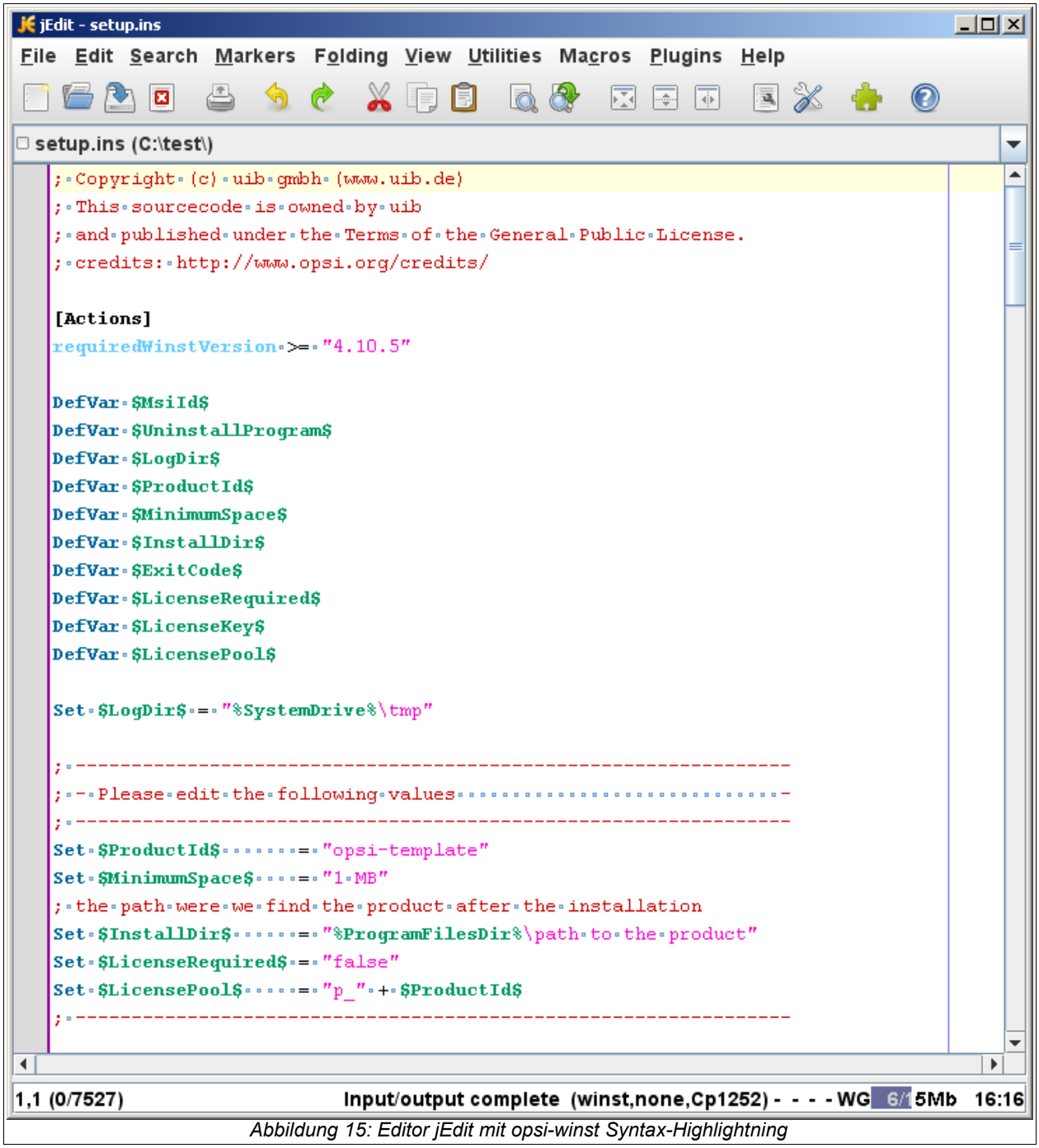

Abbildung 15: Editor jEdit mit opsi-winst Syntax-Highlightning

Sie können nun das Script im Editor anpassen und speichern (Sie können den Editor geöffnete lassen). Wechseln Sie zum Winst-Fenster und starten Sie das Script erneut über den Knopf 'Start' (das Script muss nicht neu ausgewählt werden). Schauen Sie sich das auf Basis Ihrer Änderungen im Script veränderte Log über 'View Log' an.

Auf diese Art und Weise, also über die wiederholung der Punkte:

- Anpassung des Scriptes und speichern
- Script ausführen
- Log überprüfen

können Sie nach und nach Ihre Scripte so anpassen, das das tun was Sie wünschen.

Hinweise zur Lösung von Detailproblemen finden Sie im nächsten Kapitel.

Im übernächsten Kapitel wird erklärt wie Sie aus den so erstellten Scripten ein opsi-Produkt machen, das Sie auf dem Server installieren können.

4.1.11 Hinweise zu den Teilaufgaben im opsi-template

4.1.11.1 Silent oder Unattended Schalter finden

Beim „unattended“ oder „silent setup“ wird das Original-Setupprogramm über Kommandozeilen Argumente in einen nicht interaktiven Modus gestellt.

Das Problem dieser Installationsmethode ist, die geeigneten Kommandozeilenargumente zu finden.

Suche in Schaltersammlungen im Internet

Bevor man sich in Forschungen stürzt, ist dringend zu empfehlen, bei opsi.org zu schauen, ob jemand das Problem bereits gelöst hat:

Fertige Winst-Scripte aus der Community gibt es unter:

http://www.opsi.org/opsi_wiki/WinstScripts

Eine Sammlung von Weblinks zu Schaltersammlungen im Internet findet sich unter:

http://www.opsi.org/opsi_wiki/SoftwareIntegrationWebLinks

Suche beim Hersteller des Programms

Da das Problem einer automatischen Installation inzwischen vielen Herstellern bewusst ist, finden sich häufig in der Produktdokumentation oder auf der Homepage des Herstellers Anleitungen oder Hinweise hierzu.

Suche beim Hersteller des Setup-Programms

Setupprogramme werden in der Regel von den Herstellern der diversen Softwareprodukte nicht selbst geschrieben. In den meisten Fällen bedienen sich vielmehr die Produkthersteller selbst spezieller Softwareprodukte, mit denen Setupprogramme relativ einfach erstellt werden können. Die verwendbaren Kommandozeilenargumente sind daher zumeist typisch für das verwendete Produkt zur Erstellung von Setupprogrammen.

Unter opsi.org

http://www.opsi.org/opsi_wiki/SoftwareIntegrationWebLinks

finden sich unter 'Installer specific switches / Allg. Schalter der Setup Programme' weiter Weblinks zu Seiten die Erläutern wie man Setupprogramme erkennt und wie Ihre typischen Schalter sind.

Auf der Homepage des Herstellers des Setupprogramms wird man zumeist mit der Suche nach Stichworten wie 'silent' , 'silent Install' oder 'unattended' fündig.

4.1.11.2 Weitere wichtige opsi-winst Befehle

Einen Überblick über die opsi-winst Befehle gibt die Referencecard:
http://download.uib.de/opsi3.4/doku/opsi-winst-reference-card.pdf

Der Syntax im Detail ist im opsi-winst Handbuch beschrieben:
http://download.uib.de/opsi3.4/doku/winstdoc-de.pdf

Hier noch ein paar Hinweise auf besonders wichtige Elemente:

Stringlisten

- Stringlisten sind sehr mächtig, insbesondere zur Auswertung von Ausgaben anderer Programme. Lesen Sie dazu die Winst-Dokus.

ExitWindows

- ExitWindows /Reboot
  Reboot nach Abschluß dieses Scripts

- ExitWindows /ImmediateReboot
  Sofortiger Reboot

- ExitWindows /ImmediateLogout
  Sofortige Beendigung der Scriptbearbeitung und Beendung des Winst

Product Properties

Für manche Produkte ist es erforderlich, Optionen zur Verfügung zu stellen. Diese werden zur Laufzeit clientspezifisch ausgewertet. Wie solche Properties erstellt werden ist weiter unten im Kapitel 4.2 Erstellen eines opsi-Produkt-Pakets beschrieben.

Das Auswerten der Werte der Properties geschieht durch die Funktion GetProductProperty

```
if GetProductProperty ("myproperty","off") = "on"
      Files_copy_extra_files
endif
```

4.1.11.3 Installation mit eingelogten user

Vereinzelt taucht das Problem auf, das sich Installationen nur mit eingelogten user durchführen lassen. Ein Hinweis auf diese Problematik ist es, wenn Sie ein Winst-Skript mit einer unattended Installation haben, das von einem Administrativen user aus aufgerufen funktioniert aber im Rahmen der automatischen Installation beim Boot scheitert.

Eine mögliche Ursache ist dann, daß dieses Setupprogramm einen eingeloggten user bzw. den Zugriff auf ein user-Profil benötigt.

Handelt es sich um eine MSI-Installation hilft evtl. die Option ALLUSERS=2.
Beispiel:

```
[Aktionen]
DefVar $LOG_LOCATION$
Set $LOG_LOCATION$ = "c:\tmp\myproduct.log"
winbatch_install_myproduct

[winbatch_install_myproduct]
msiexec /qb ALLUSERS=2 /l* $LOG_LOCATION$ /i %SCRIPTPATH%\files\myproduct.msi
```

Eine weitere Möglichkeit ist, dass das Installationsprogramm nicht ordentlich terminiert, z.B. weil ein ein Subprozess gestartet wird. Dann kann man den Winbatch-Aufruf

mittels /WaitSeconds [AnzahlSekunden]
oder aber mit /WaitForProcessEnding "program.exe" /TimeOutSeconds "seconds" probieren.

Eine andere aufwendigere Möglichkeit dieses Problem zu lösen, die Möglichkeit einen administrativen user temporär anzulegen, und diesen zur Installation des Programms zu Verwenden. Dies können Sie auf Basis des Templates 'opsi-template-with-admin' durchführen.

4.1.11.4 Arbeiten mit MSI-Paketen

Microsoft hat mit Windows 2000 ein eigenes Installationskonzept vorgestellt, das auf dem Microsoft Installer Service, kurz „MSI“ beruht. Inzwischen sind viele Setup-Programme MSI-konform.

MSI-Konformität bedeutet, dass die eigentliche Installation darin besteht, dass an den MSI ein Paket von Installations-Anweisungen übergeben wird (im Prinzip eine Datei mit einen Namen der Form „produkt.msi“) und der MSI dieses Paket dann ausführt.

In der Praxis sieht dies meist so aus, dass die zu einem Produkt gehörige „setup.exe“ eine Datei „produkt.msi“ und ein zusätzliches Steuerprogramm für die Installation enthält. Das Steuerprogramm packt „produkt.msi“ aus und fragt, ob eine Installation starten soll. Wird dies bestätigt, prüft das Steuerprogramm, ob der MSI schon eingerichtet ist und übergibt bei positivem Ergebnis der Prüfung diesem die „produkt.msi“. Ist der MSI nicht eingerichtet und wird insbesondere das Programm „msiexec.exe“ nicht gefunden, so startet das Steuerprogramm zuerst eine Installationsprogramm für den MSI.

Klickt man bei der Frage, ob die Installation starten soll, nicht auf „weiter“, sondern ruft den Explorer auf, so findet sich das ausgepackte MSI-Paket meist in einem temporären Verzeichnis.

Dieses Paket kann nun dazu verwendet werden, eine Installation „unattended“ - also „unbewacht“, d.h. ohne dass ein Benutzereingriff“ erforderlich ist - auszuführen. Dazu ist bei vorhandener msiexec.exe aufzurufen:

```
msiexec /i "%ScriptPath%\Product.msi" /qb-! ALLUSERS=2 REBOOT=ReallySuppress
```

4.1.11.5 Customizing nach einer silent/unattended Installation

Häufig will man nach einer erfolgreichen silent Installation noch Anpassungen an der Installation vornehmen. Hierzu bietet der opsi-Winst ein mächtiges Werkzeug. Doch bevor diese Eingesetzt werden kann muss oft ermitteltet werden, welche in der graphischen Oberfläche getätigten Änderungen zu welchen Veränderungen in Dateien und der Registry geführt haben.

Hierzu kann man die unter 'Analyse und Neu-Paketieren' vorgestellten Werkzeuge einsetzen. Häufig führen aber auch kleinere Werkzeuge schneller zum Erfolg.
Hier einige Links zu entsprechenden Werkzeugen:

http://www.sysinternals.com/

http://www.german-nlite.de/files/guides/regshot/regshot.html

4.1.11.6 Einbindung mit automatisierten Reaktionen des Setup-Programms

Eine weitere schnelle Möglichkeit zur Einbindung in die automatische Softwareverteilung ist das 'Setup mit automatisierten Antworten'. Hierzu wird eine Steuerungssoftware verwendet, die z.B. auf das Erscheinen eines bestimmten Fensters warten kann und dann in dieses Fenster skriptgesteuert eine Antwort gibt. Wir empfehlen hier den Einsatz der Software autohotkey (http://de.autohotkey.com/) oder AutoIt (http://www.hiddensoft.com/autoit3/).

AutoIt bietet eine ganze Reihe zusätzlicher Möglichkeiten, den Setupprozess zu steuern. Auch eventuelle Fehlerzustände können (so vorher bekannt) mit dem Einsatz von [ADLIB] Sektionen im Skript abgefangen werden.

Trotzdem bleibt ein prinzipielles Problem bestehen: Nicht vorhergesehene (und im Skript berücksichtigte) Fehlerfenster können das Skript zum Stoppen bringen. Außerdem kann der Anwender mit Maus und Tastatur 'dazwischen funken', wenn diese nicht gesperrt sind. Von daher ist aus unserer Sicht ein unattended oder silent setup die bessere Lösung.

Sehr gut kann auch eine Kombination aus beidem funktionieren: Das Silent-Setup übernimmt die eigentliche Installation und das AutoIt-Skript fängt bekannte Sonderbedingungen ab.

4.1.11.7 Analyse und Neu-Paketieren

Wenn der Entwickler einer Anwendung ein Paket zur Auslieferung der Anwendung schnürt, kennt er die benötigten Komponenten. Im nach hinein, wenn schon ein Paket existiert, das mittels eines Setup-Programm zu installieren ist, kann die Kenntnis, welche Komponenten installiert werden müssen, damit eine Anwendung wie gewünscht auf einem Arbeitsplatzrechner lauffähig ist, aus der Studie der Effekte bei der Ausführung des vorhandenen Setup-Programms gewonnen werden.

Als Werkzeuge zum Analysieren von Setupscripten kommen eine Reihe in Frage. So z.B.:

WinINSTALL LE ist (wieder) als Freeware verfügbar.

http://www.ondemandsoftware.com

4.1.11.8 Verfahren zur Deinstallation von Produkten

Um ein Produkt auch wieder von einem Rechner löschen zu können, muss ein Deinstallationsskript existieren. Grundsätzlich besteht bei einer Deinstallation die Schwierigkeit, dass nicht immer klar ist, wie das Produkt auf dem Rechner vorliegt und was alles entfernt werden muss. Es können neue Dateien oder neue Registry-Einträge zum Produkt nach der Installation hinzugekommen sein. Weiterhin muss darauf geachtet werden, nicht zu viel zu entfernen um nicht die Systemstabilität zu gefährden. Meist weiß nur der Hersteller genau, wie mit seinem Produkt bei der Deinstallation umzugehen ist. Ähnlich wie bei der Installation existieren zu diesem Zweck Deinstallationsroutinen die dem Produkt beiliegen. Wenn es die Möglichkeit gibt, diese ohne Benutzerinteraktion auszuführen, kann dies schon ein entscheidender Schritt sein. Ist eine solche Routine nicht vorhanden oder muss diese erweitert werden, so kennt der Winst Befehle, die zur Deinstallation nützlich sein können. Im Folgenden soll nun ein Überblick über Möglichkeiten zur Deinstallation gegeben werden, die durch Beispiele verdeutlicht werden.

Verwenden einer Deinstallationsroutine

Liefert der Hersteller des Produkts ein Programm (oder ein MSI-Paket) zur Deinstallation, so muss zunächst geprüft werden, ob dies auch ohne

Benutzerinteraktion ausgeführt werden kann (silent-mode). Sollte dies nicht von Hause aus Möglich sein, kann der Einsatz eines autoit-Skriptes zusammen mit der Deinstallationsroutine hilfreich sein. Der Aufruf der ausführbaren Datei kann im Winst-Skript dann in einer Winbatch-Sektion geschehen, z.B.:

```
[Winbatch_start_ThunderbirdUninstall]
%SYSTEMROOT%\UninstallThunderbird.exe /ma
```

Trotz dieser Unterstützung des Herstellers sollte man sich jedoch nicht auf die korrekte Beseitigung des Produkts verlassen und prüfen ob das System nach der Deinstallation auf einem Testsystem weiter stabil läuft oder welche Dateien/Einträge zurückgeblieben sind.

Falls man das Produkt zuvor mittels MSI installiert hat, ist es meist möglich an diesem Paket auch die Deinstallation aufzurufen. Dazu übergibt man das MSI-Paket mit dem Schalter **`/x`** an die **`msiexec.exe`**. Um die Benutzerfragen zu deaktivieren (das Skript läuft dann ohne Benutzerinteraktion durch) existiert der Schalter **`/qb-!`**. Dies ergibt nun folgende Anweisung:

```
msiexec.exe /x some.msi /qb-! REBOOT=ReallySuppress
```

Statt den Namen des Pakets zu übergeben, gibt es auch die Möglichkeit die GUID an **`msiexec.exe`** zu übergeben. Diese Nummer identifiziert das Produkt im System – sie ist als produktspezifisch. Sie findet sich zum Beispiel im Zweig

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall
```

der Registry. Ein Aufruf mit dieser GUID sieht dann folgendermaßen aus:

```
msiexec.exe /x {003C5074-EB37-4A75-AC4B-F5394E08B4DD} /qb-!
```

Sollten diese Methoden nicht oder nicht vollständig funktionieren, so muss mit einem Winst-Skript nachgeholfen werden, wie es der nächste Abschnitt beschreibt.

Nützliche Winst-Befehle zur Deinstallation

Wurde ein Produkt mit den Winst-Funktionen installiert oder gibt es keine Deinstallation vom Hersteller, so muss ein eigenes Winst-Skript zu Deinstallation geschrieben werden. Um den Programmierer bei dieser Arbeit zu unterstützen kennt der Winst einige Funktionen, die speziell bei der Deinstallation notwendig oder hilfreich sind. Es soll an dieser Stelle ein Überblick gegeben werden, eine genaue Beschreibung der Befehle und deren Parameter findet sich im Winst-Handbuch.

Der einfachste Fall ist das Löschen einer oder mehrerer Dateien vom System. Dies geschieht in einer Files-Sektion mit dem Befehl

```
delete -f Dateiname
```

oder für ein Verzeichnis mit Unterverzeichnissen

```
delete -sf Verzeichnisname\
```

Der Parameter f steht dabei für force um die Datei wirklich zu löschen, auch wenn diese schreibgeschützt ist, der Paramtere s für subdirectories (Unterverzeichnisse). Soll eine Datei oder ein Verzeichnis aus allen User-Profilen gelöscht werden so kann diese Files-Sektion mit dem Parameter /AllNTUserProfiles aufgerufen werden. (siehe Winst-Handbuch)

Möchte man einen Verzeichnisbaum löschen in dem sich auch Dateien mit dem Attribut „versteckt“ oder „systemdatei“ befinden, muss im Moment der Umweg über eine DosInAnIcon-Sektion gegengen werden in der nun der Dos-Befehl ausgeführt werden kann:

```
[DosInAnIcon_deleteDir]
rmdir /S /Q "<Verzeichnisname>"
```

Muss vor dem Löschen evtl. ein laufender Prozess beendet werden, so kann dies mit dem Namen des Prozesses (zu sehen im Task-Manager) und dem Winst-Befehl killtask geschehen

```
killtask "thunderbird.exe"
```

Sollte das Produkt – oder Teile davon – als Service laufen, so muss dieser vor der Deinstallation beendet werden. Man kann dazu den Service in der Registry auf “inaktiv“ schalten und den Rechner neustarten oder aber man benutzt den Befehl “net stop“ um den Service sofort zu stoppen um anschließend – ohne Neustart – die zugehörigen Dateien zu löschen.

```
net stop
```

Besondere Vorsicht ist beim Löschen von .dll-Dateien geboten, die noch von anderen Produkten verwendet werden könnten. Sie müssen individuell behandelt werden, weshalb hier leider kein allgemein gültiges Rezept gegeben werden kann.

Um einzelne Einträge aus der Registry mit dem Winst zu löschen kommt der Befehl **`DeleteVar`** zum Einsatz der nur in einer Registry-Sektion eines Winst-Skripts verwendet werden kann. Er löscht Einträge aus dem momentan geöffneten Key:

```
      DeleteVar <VarName>
```

Möchte man einen Registry-Key samt seiner Unterschlüssel und Registry-Variablen löschen, so geschieht dies mit dem Winst-Befehl DeleteKey, z.B.:

```
DeleteKey [HKLM\Software\Macromedia]
```

4.1.11.9 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. Neuere Winst Versionen kennen spezielle Befehle für 64 Bit SystemeBeachten Sie beim Arbeiten auf 64 Bit Systemen das entsprechende Kapitel im Winst-Handbuch.

4.2 Erstellen eines opsi-Produkt-Pakets

In opsi werden die Installationsdateien, das Winst-Skript zur Installation auf den Client und die Metadaten zu einem Paket zusammengefasst, welches zur Installation dieses Softwareproduktes auf einem opsi-server ident.

Die wesentlichen Vorteile dieses Paketformates sind:

- Vereinfachte menügeführte Erstellung mit dem Programm opsi-newprod.
- Ablage aller relevanten Metadaten in einer einfach zu editierenden Datei.
- Optional menügeführtes Auspacken des Paketes mit der Möglichkeit Vorgaben zu ändern.
- Informationen über die im Paket enthaltene Produktversion, Paketversion und eventueller Kundenspezifischer Erweiterungen werden abgespeichert und sind am Paketnamen erkennbar, werden im Installationsverzeichnis abgelegt und im opsi-Configeditor angezeigt. Auf diese Weise wird der Überblick über unterschiedliche Versionen erleichtert (Productlifecycle Management).
- Zur Erstellung und zum Auspacken von Produkten sind keine root-Rechte erforderlich. Es langen hierzu die Rechte der Gruppe 'pcpatch'.

Das Paket selber besteht aus einem per Gzip komprimierten cpio Archiv. In diesem Archiv befinden sich drei Verzeichnisse:

CLIENT_DATA

Hier liegen die Dateien die im Produktverzeichnis (z.B. /opt/pcbin/install/<productid> landen sollen.

SERVER_DATA

In diesem Zusammenhang nicht relevant.

OPSI

Hier liegen in der Datei 'control' die Metadaten des Produkts wie z.B. Produktabhängigkeiten. Weiterhin finden sich hier die Dateien preinst und postinst die vor bzw. nach der Installation ausgeführt werden. Hier können Sie soweit benötigt entsprechende Erweiterungen unterbringen.

Weiterhin finden Sie hier die Datei changelog.txt in der Sie Änderungen am Produkt festhalten sollten.

4.2.1 Erstellen, Packen und Auspacken eines neuen Produktes

Zur Erstellung eines Produktes müssen Sie sich auf dem Server einloggen (z.B. von Windows aus per putty.exe (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html).

Die wesentlichen Befehle zum Erstellen und Installieren eines opsi-produktes sind:

- **`opsi-newprod`**
- **`opsi-makeproductfile`**
- **`opsi-package-manager -i <opsi-product-file>`**

Zur Erstellung eines neuen Produktes benötigt man mindestens die Rechte der Gruppe pcpatch.

Sie sollten die Produkte in dem Verzeichnis /home/opsiproducts erstellen welches der Gruppe pcpatch gehört und die Rechte 2770 hat (Setgroupid Bit für Gruppe pcpatch gesetzt) sowie als share opsi_workbench freigegeben ist.

Achtung: Im folgenden sollten bei Eingaben keine Umlaute verwendet werden, da die Umsetzung zwischen den verschiedenen Zeichensätzen noch nicht sauber funktioniert.

Zum Erstellen wechselt man in dieses Verzeichnis und ruft 'opsi-newprod' auf. Das Programm fragt daraufhin nach dem Typ des zu Erstellenden Paketes. Dies ist üblicherweise der Typ 'localboot' für Produkte die über den Preloginloader/Winst installiert werden. Der Typ 'netboot' steht für Produkte die einen bootimage Start

auslösen (wie Hardware-Inventarisierung) und der Typ 'server' für Produkte die nur Dinge auf dem Server installieren.

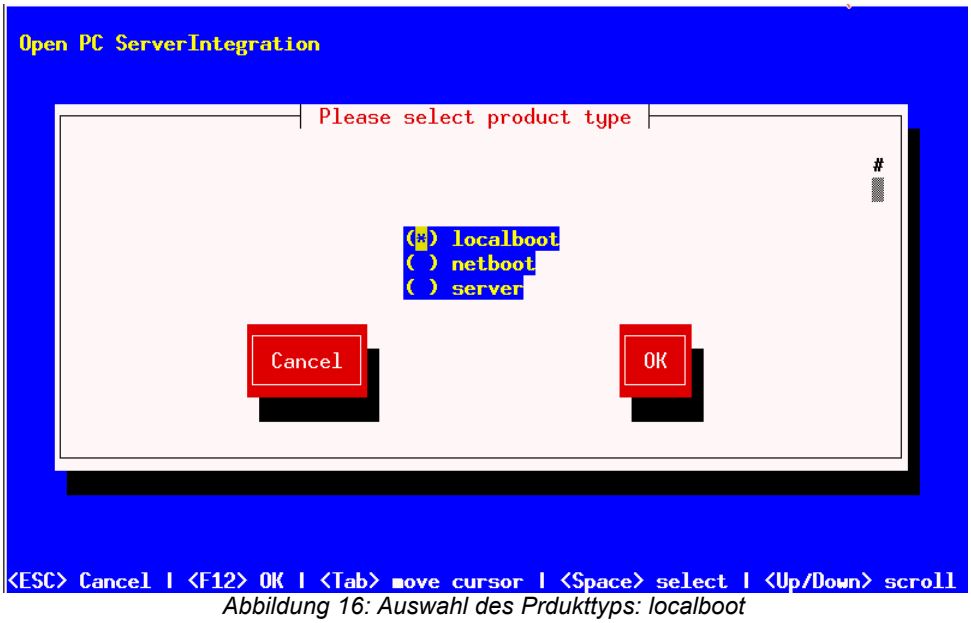

Abbildung 16: Auswahl des Prdukttyps: localboot

Wählen Sie nun mit Tab OK (oder bestätigen mit F12). Nun müssen Sie die wesentlichen Produktdaten eingeben. Am oberen Rand ist hierzu eine Hilfe die Erläutert was die Felder bedeuten.

- 'Product Id' ist ein eindeutiger Bezeichner für das Produkt in der Regel unabhängig von der Version
  Bitte nur Kleinbuchstaben verwenden, keine Umlaute, keine Leerzeichen, keine Sonderzeichen, '-' ist als Trenner erlaubt.

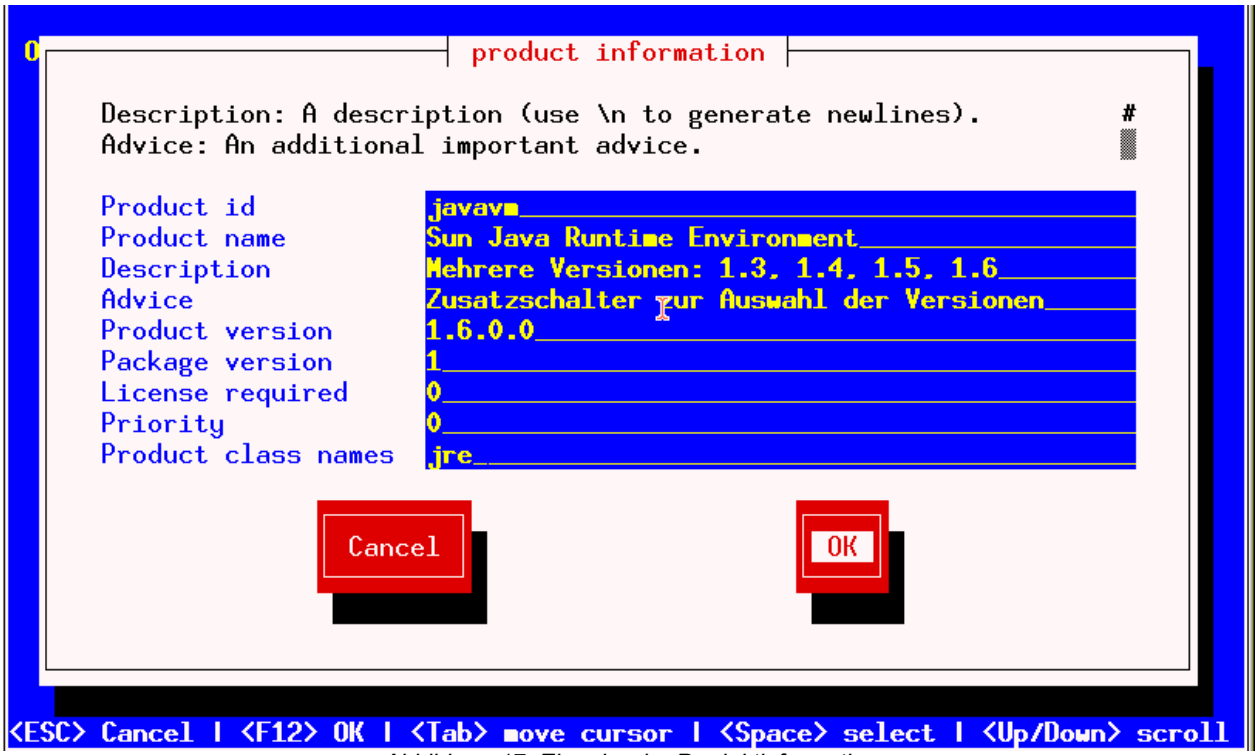

Abbildung 17: Eingabe der Produktinformationen

- 'Product name' ist der Klartextname des Produkts
- 'Desciription' ist eine ergänzende Beschreibung zum Produkt die z.B. im opsi-Configeditor unter 'Beschreibung' angezeigt wird.
- 'Advice' ist eine ergänzende Beschreibung in der Regel zum Umgang mit dem Produkt die zu Beachten ist und im im opsi-Configeditor unter 'Notiz' angezeigt wird.
- 'Product version' ist die Version der eingepackten Software (max. 32 Zeichen)
- 'Package Version' ist die Version des Paketes für die Produktversion. Die dient dazu um Pakete mit gleicher Produktversion aber z. B. korrigiertem Winst-Skript zu unterscheiden.
- 'Priority' wird zur Zeit noch nicht verwendet. Soll neben Produktabhängikeiten die Installationsreihenfolge beeinflussen.
- 'Product class' wird zur Zeit noch nicht verwendet (und auch nicht angezeigt).

Nach Eingabe der Produktinformationen werden Sie aufgefordert die Skripte anzugeben die Sie für unterschiedliche mögliche Aktionen bereit stellen werden.

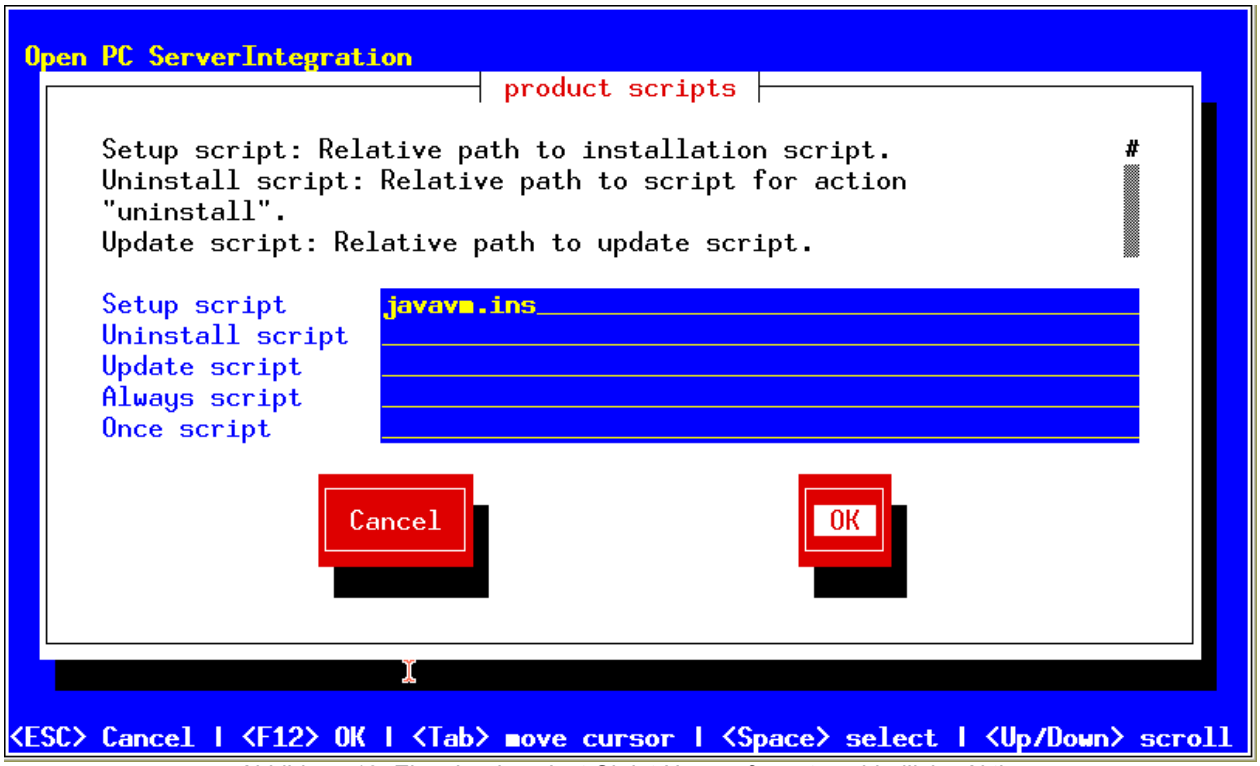

Abbildung 18: Eingabe der wInst-Skript Namen für unterschiedliche Aktionen

Üblicherweise heißt das 'Setup script' gleich setup.ins.
Üblicherweise heißt das 'Uninstall script' gleich uninstall.ins.

Ein 'Update-Script' dient zur geringfügigen Veränderung einer existierenden grossen Installation. Wird das Produkt auf setup gestellt, so wird nach dem abarbeiten des Setup-Skriptes automatisch auch das Update-Skript ausgeführt.

Ein 'Always-Script' wird bei jedem aktiv werden des opsi-Clientagenten ausgeführt (z.B. bei jedem Boot)

Ein 'Once-Script' hat den Folgestatus 'not_installed'.

Typ	*Folgestatus*	*Folgeaktion*
setup	installed	none
uninstall	not_installed	none
update	installed	none
always	installed	always
once	not_installed	none

Nachdem nun das Produkt selber beschrieben ist, können Sie eine oder mehrere Produktabhängigkeiten definieren. Wollen Sie keine Produktabhängigkeit definieren so geben Sie 'No' ein.

Abbildung 19: Eine (weitere) Produktabhängigkeit definieren: Ja / Nein

Zur Erstellung einer Produktabhängigkeit geben Sie die folgenden Daten an. Beachten Sie auch die Hilfe im oberen Teil des Fensters:

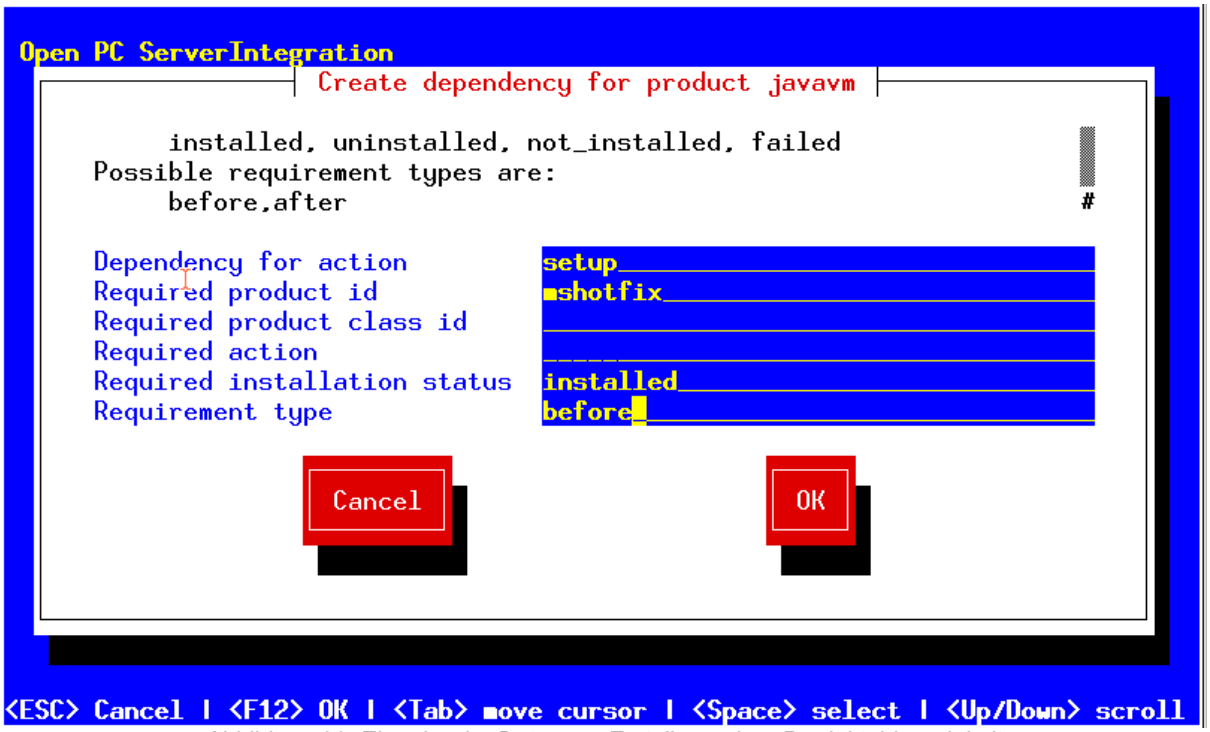

Abbildung 20: Eingabe der Daten zur Erstellung einer Produktabhängigkeit

- 'Dependency for Action' : Für welche Aktion des Produktes welches Sie gerade erstellen soll die Abhängigkeit gelten (setup, deinstall,...)
- 'Required product id': Productid (Bezeichner) des Produkts zu dem eine Abhängigkeit besteht.
- 'Required product class id': wird zur Zeit noch nicht verwendet. Leer lassen ! Bezeichner der Productklasse zu der eine Abhängigkeit besteht.
- 'Required action': Sie können entweder eine Aktion anfordern oder (siehe unten) einen Status. Aktionen können z.B. sein : setup, uninstall, update,...
- 'Required installation status': Status den das Produkt zu dem eine Abhängigkeit besteht haben soll. Typischerweise 'installed', liegt ein anderer Status vor so wird das Produkt auf setup gestellt.
- 'Requirement type': Installationsreihenfolge. Wenn das Produkt zu dem eine Abhängigkeit besteht installiert sein muss bevor mit der Installation des aktuellen Produkts begonnen werden kann dann ist dies 'before'. Muss es nach dem aktuellen Produkt installiert werden so ist dies 'after'. Ist die Reihenfolge egal so muss hier nichts eingetragen werden.

Hinweis:

Leider gibt es derzeit nicht wirklich einen generischen Mechanismus für Deinstallations-Produktabhängigkeiten. Zuverlässig ist der ProductDependency-Mechanismus nur für action: setup und die hierbei zu triggernden (before- oder after-) setup Aktionen und installed Status. Ein requiredAction: uninstall führt leider definitiv zu Fehlern.

Nach dem eine Produktabhängigkeit definiert ist, werden Sie wieder gefragt ob Sie eine (weitere) Produktabhängigkeit definieren wollen. Wenn ja wiederholt sich der Vorgang; wenn nein, so werden Sie gefragt ob sie eine Produkteigenschaft (Zusatzschalter) definieren wollen mit dem Sie die Installation des Produktes modifizieren können.

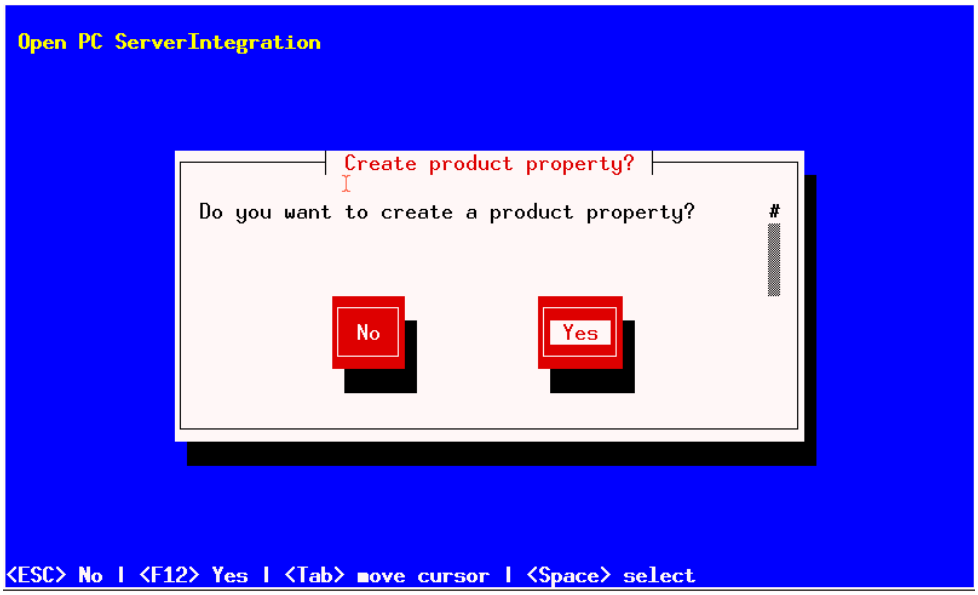

Abbildung 21: Eine (weitere) Produkteigenschaft definieren ?

Antworten Sie ja, so müssen Sie die Produkteigenschaft beschreiben:

Die Produkteigenschaft wird clientspezifisch gespeichert und besteht aus einem Namen (key) der verschiedene Werte (Values) zugeordnet bekommen kann und die dann vom Winst-Skript aus abgefragt werden können. Weiterhin wird eine Beschreibung (Product description) benötigt die beim Auspacken des Produkts und im opsi-Configedit als Hilfe angezeigt wird. Weiterhin müssen Sie, durch Komma getrennt, alle Werte angeben, die der Key annehmen darf. Wird hier nichts angegeben so kann später im opsi-Configeditor ein beliebiger Wert eingegeben werden.

Im Folgefenster müssen Sie festlegen was der Defaultwert dieser Produkteigenschaft ist.

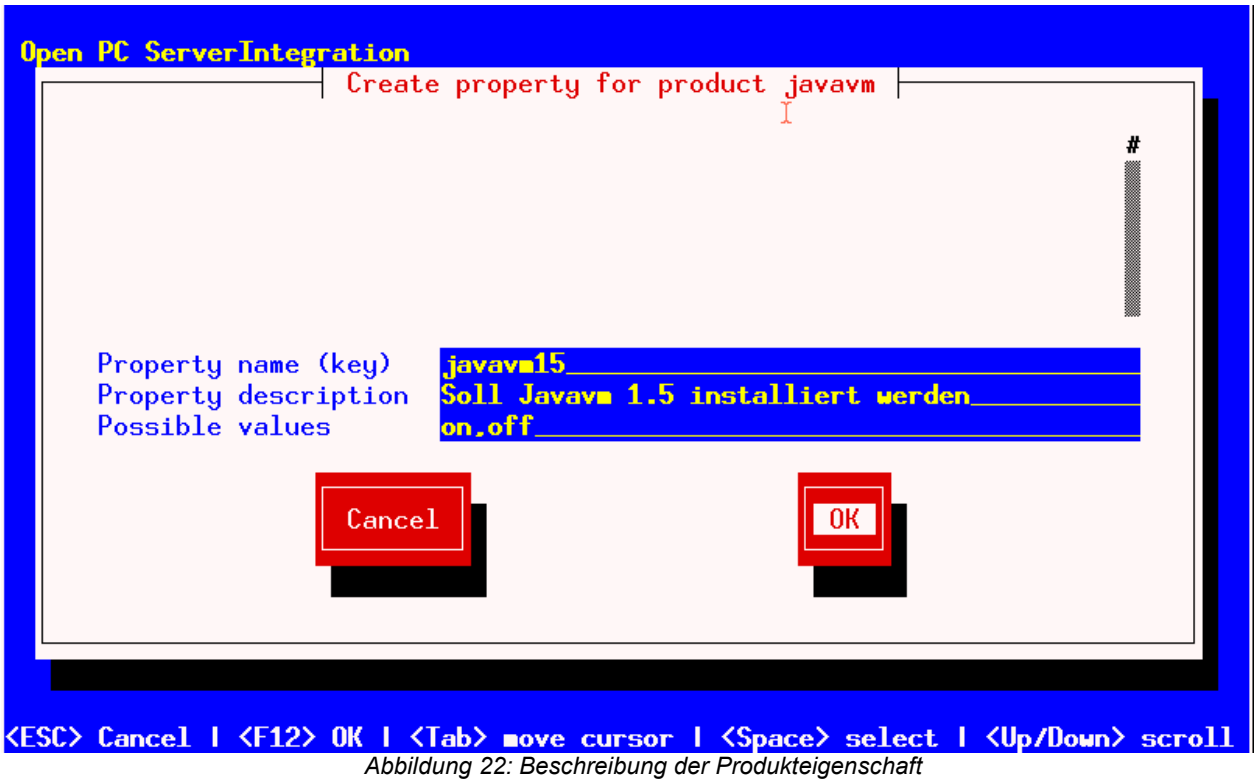

Abbildung 22: Beschreibung der Produkteigenschaft

Im folgenden Fenster entscheiden Sie welches der default Wert ist.

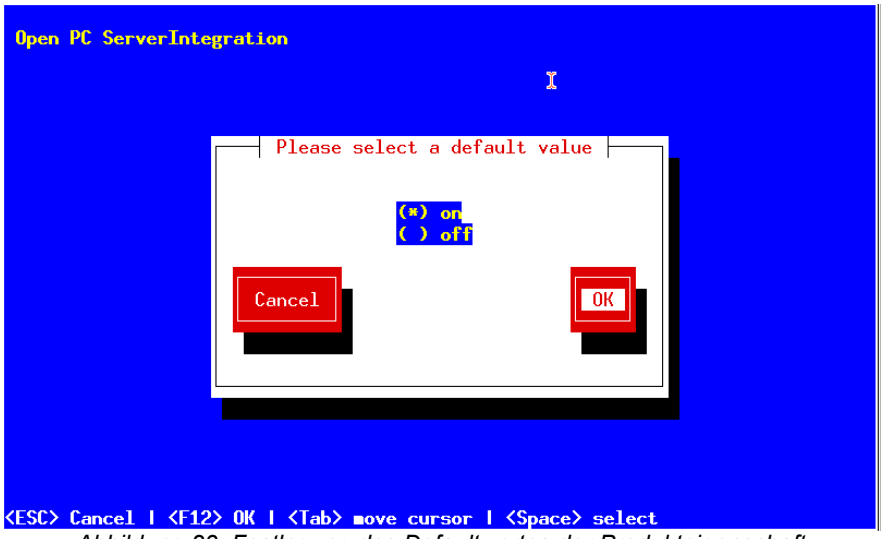

Abbildung 23: Festlegung des Defaultwertes der Produkteigenschaft

Nach dem eine Produkteigenschaft definiert ist, werden Sie wieder gefragt ob Sie eine (weitere) Produkteigenschaft definieren wollen. Wenn ja wiederholt sich der Vorgang; wenn nein, so ist das Grundgerüst des Produktes fertig gestellt.

Mit Hilfe des ls-Befehls finden Sie die oben beschriebene Verzeichnis Struktur. Wechseln Sie in den OPSI-Ordner und setzen Sie erneut den ls-Befehl ab. Hier

befindet sich unter anderem die 'control'-Datei, welche die eben eingegebenen Daten enthält und Ihnen auch die Möglichkeit bietet, diese im Editor zu kontrollieren oder zu modifizieren.

Beispiel einer 'control' Datei:

```
[Product]
type: localboot
id: javavm
name: Sun Java Runtime Environment
description:
Mehrere Versionen: 1.3, 1.4, 1.5, 1.6
advice: Zusatzschalter zur Auswahl der Versionen
version: 1.6.0.0
packageVersion: 1
priority: 0
licenseRequired: True
productClasses: jre
setupScript: javavm.ins
uninstallScript:
updateScript:
alwaysScript:
onceScript:

[ProductDependency]
action: setup
requiredProduct: mshotfix
requiredStatus: installed
requirementType: before


[ProductProperty]
name: default13
description: on=Version 1.3 wird default JRE; off=Die aktuellste installierte
Javavm wird default JRE
values: on, off
default: off

[ProductProperty]
name: javavm15
description: Soll Javavm 1.5 installiert werden
values: on, off
default: on

[ProductProperty]
name: javavm16
description: Soll Javavm 1.6 installiert werden
values: on, off
default: off
```

Als nächstes müssen Sie Ihr für das Produkt erstellte Winst-Skript und die entsprechenden Dateien nach CLIENT_DATA kopieren.

Wenn Sie z.B. Ihr erstelltes Script unter c:\test liegen haben, so mounten Sie \\<opsiserver\opsi_workbench z.B. nach w: und kopieren den Inhalt von c:\test in das Verzeichnis CLIENT_DATA.

Danach können Sie das Produkt packen. Gehen Sie dazu in das Stammverzeichnis des Produkts und rufen Sie 'opsi-makeproductfile' auf. Es wird nun das Produkt gepackt.

opsi-makeproductfile kennt einige Optionen die sein Verhalten modifizieren:

```
# opsi-makeproductfile --help

Usage: opsi-makeproductfile [-h] [-v|-s] [-f] [-F format] [-l log-level] [-i|-
c custom name] [-I required version] [-t temp dir] [source directory]
Provides an opsi package from a package source directory.
If no source directory is supplied, the current directory will be used.
Options:
   -v          verbose
   -s          silent
   -l          log-level 0..6
   -f          fast, no topicality test
   -n          do not compress
   -F          archive format [tar|cpio], default: cpio
   -h          follow symlinks
   -I          incremental package
   -i          custom name (add custom files)
   -c          custom name (custom only)
   -t          temp dir
```

Das gepackte Paket können Sie mit 'opsi-packet-manager -i <paketname>' auf dem Server installieren.

Weitere Informationen zum opsi-packet-manager siehe opsi-Handbuch.

5 Weitere Informationen

Das opsi-Handbuch (http://download.uib.de/opsi3.4/doku/opsi-handbuch-v34-de.pdf) enthält eine Fülle von weiteren Informationen die für den produktiven Betrieb wichtig sind.

Wenn Sie dort nicht fündig werden oder Hilfe benötigen, wenden Sie sich an https://forum.opsi.org

Für produktive Installationen empfehlen wir professionelle Unterstützung durch uib im Rahmen eines Pflege- und Supportvertrages:
http://uib.de/www/opsi/service_support/support/index.html