

# **MSI-Paketerstellung & Softwareverteilung**

*in Windows 2000/2003-Domänen  
mit kostenfreien Werkzeugen*

Hochschulrechenzentrum  
Dipl.-Inform. Carsten Vollrath

Mai 2005

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Erstellen von MSI-Paketen</b>	<b>4</b>
2.1	Extrahieren aus EXE-Installern.....	4
2.1.1	Adobe Acrobat Reader 7.....	5
2.1.2	Sun J2SE Runtime Environment 5.0.....	5
2.2	Erstellen aus proprietären Installern.....	7
2.2.1	GSview 4.7.....	7
<b>3</b>	<b>Bearbeiten von MSI-Paketen</b>	<b>8</b>
3.1	Bearbeiten mit Orca.....	8
3.2	Bearbeiten mit WinINSTALL LE 2003.....	9
<b>4</b>	<b>Verteilen von MSI-Paketen</b>	<b>10</b>
4.1	Einrichten einer Freigabe für die Softwareverteilung.....	10
4.2	Einrichten von Gruppen.....	11
4.3	Einrichten der Gruppenrichtlinien.....	11
<b>5</b>	<b>Zusammenfassung</b>	<b>13</b>

# 1 Einleitung

Microsoft-Installer-Pakete, kurz MSI-Pakete, erlauben eine komfortable Installation unter Microsoft Betriebssystemen wie *Windows 2000*, *Windows XP* oder *Windows Server 2003*. Sie können interaktiv installiert werden, aber auch über die in den Windows-Server-Systemen integrierten Softwareinstallationsmechanismen automatisch auf Clientcomputern aufgespielt werden. Für die zweite Variante, die in Windows-Domänen elegant über Gruppenrichtlinien konfiguriert werden kann, ist es oft notwendig, eigene MSI-Pakete zu erstellen, da nicht jeder Softwarehersteller für die Installation seiner Produkte MSI-Pakete anbietet. Aber auch wenn bereits MSI-Pakete vorhanden sind, kommt es häufig vor, diese verändern/bearbeiten zu müssen, um bestimmte Einstellungen vorzugeben. Dies kann zum Beispiel das automatische Entfernen eines Desktopsymbols sein oder das Installieren/Deinstallieren eines Features.

Für das Bearbeiten und Erstellen von MSI-Paketen wird zusätzliche Software gebraucht. Gängige lizenzpflichtige Werkzeuge sind *WinINSTALL* von OnDemand<sup>1</sup> oder das Produkt *Wise for Windows Installer* von Wise Solutions<sup>2</sup>. Es gibt auch lizenzfreie Alternativen, so bietet OnDemand kostenfrei das Produkt *WinINSTALL LE 2003* an. Dieses besitzt gegenüber der Vollversion einen eingeschränkten Funktionsumfang. Fehlende Funktionen können mit anderen Werkzeugen wie dem MSI-Editor *Orca* aus dem Hause Microsoft ergänzt werden.

Dieses Dokument zeigt exemplarisch an verschiedenen Softwareprodukten, wie mit den eben erwähnten Werkzeugen *WinINSTALL LE 2003* und *Orca* MSI-Pakete erzeugt werden können und wie diese in einer Windows-Domäne über Gruppenrichtlinien an Clientcomputer verteilt werden können. Bewusst wird nur diese kostenfreie Alternative vorgestellt. Wer in größerem Rahmen Softwareverteilung in seiner IT-Infrastruktur einsetzen will, sollte sich mit Produkten, wie dem *Systems Management Server (SMS)* von Microsoft oder der *Desktop Availability Suite (DAS)* von OnDemand auseinander setzen.

---

1 <http://www.ondemandsoftware.com>

2 <http://www.wise.com>

## 2 Erstellen von MSI-Paketen

Viele Softwarehersteller nutzen inzwischen das Microsoft-Installer-Format für ihre Setup-Routinen, jedoch ist dies meistens nicht offensichtlich. Lädt man eine Anwendung wie beispielsweise den *Adobe Acrobat Reader 7*<sup>1</sup> aus dem Internet, liegt dieser als ausführbare Datei mit der Endung *.EXE* vor. Bei dieser Datei handelt es sich aber nicht einfach um ein selbstextrahierendes Zip-Archiv aus dem direkt ein MSI-Paket entpackt werden kann. Die MSI-Dateien sind hier hinter einem weiteren Installer verborgen.

Manche anderen Softwareprodukte bieten leider keine MSI-Pakete zur Installation an, sondern bringen proprietäre Installer mit sich. Hier ist es notwendig, sich MSI-Pakete mit Hilfe von *Win-INSTALL LE 2003* zu erstellen. Die Erstellung basiert im Prinzip auf einem Zustandsvergleich eines Clientcomputers vor und nach einer manuell vom Administrator installierten Software. Die Veränderung am System, die nach einer Installation aufgetreten sind, so z.B. das Hinzufügen von Programmdateien, Ändern von Registrierungswerten, etc., werden in einem MSI-Paket gespeichert. Beide Vorgänge zum Erzeugen der MSI-Dateien werden im Weiteren exemplarisch dargestellt.

Noch ein Hinweis: Für die MSI-Paketerstellung empfiehlt sich die Verwendung eines eigens für diesen Zweck bereitgestellten Computers, der im Weiteren mit „Test-PC“ bezeichnet wird. Dies ist insbesondere erforderlich, wenn man aus proprietären Installern MSI-Pakete erzeugt. Welche Systemkonfigurationen wünschenswert bzw. notwendig sind, wird in den folgenden Kapiteln erläutert.

### 2.1 Extrahieren aus EXE-Installern

Wie schon zuvor erwähnt, steckt hinter vielen EXE-Installern von diversen Softwareprodukten ein MSI-Paket. Ob eine Software bereits im MSI-Format vorliegt, d.h. den *Windows Installer* nutzt, kann bei der Installation erkannt werden. Dialoge wie in *Abbildung 2.1*, aber auch die Optik des Installers weisen auf die Verwendung des *Windows Installers* und somit die Existenz eines MSI-Pakets hin.



Abbildung 2.1: Typischer Dialog eines Windows Installers

1 <http://www.adobe.com>

Ein MSI-Paket besteht aus einer MSI-Datei und eventuell zugehörigen MST- und CAB-Dateien. Für eine administrative Installation oder für die Softwareverteilung werden im Prinzip nur diese benötigt. So sind diese also erst aus dem EXE-Installer zu extrahieren. Da diese EXE-Installer in der Regel keine ausführbaren ZIP-Archive sind, muss oft eine manuelle Installation der Software durchgeführt werden. Je nach Produkt sind dann die gewünschten Dateien während der Installation in temporären Verzeichnissen oder unter den Programmdateien finden. Hier ist das Gespür des Administrators gefordert, aber auch der Blick ins Internet über Suchmaschinen ist sehr zu empfehlen. Am Beispiel vom *Adobe Acrobat Reader 7* und dem *Sun J2SE Runtime Environment 5.0* wird im Folgenden das Extrahieren der MSI-Dateien aufgezeigt.

### 2.1.1 Adobe Acrobat Reader 7

Beim *Adobe Acrobat Reader 7* hat der Administrator es sehr leicht, das MSI-Paket zu extrahieren. Es ist zunächst lediglich notwendig das Programm auf einem Test-PC zu installieren. Für das hier beschriebene Szenario ist keine besondere Konfiguration des Test-PCs erforderlich. Eine Standardinstallation von *Windows 2000/XP* ist ausreichend. *Adobe Acrobat Reader 7* steht in mehreren Varianten auf den Webservern von Adobe<sup>1</sup> zum Download, wobei hier die deutsche Vollversion eingesetzt wird. Auf dem Test-PC ist nun der *Adobe Acrobat Reader 7* zu installieren. Nach einer erfolgreichen Installation navigiert man über den Explorer in das Programmverzeichnis<sup>2</sup>. Da bezüglich des Betriebssystems von einer deutschen Standardinstallation ausgegangen wird, werden auch in den folgenden Kapiteln die Verzeichnispfade in absoluter Form und ohne Verwendung von Systemvariablen dargestellt. Die Installationsroutine von Adobe hat im Programmverzeichnis eine Ordnerstruktur angelegt, deren Wurzelverzeichnis „Adobe“ ist. Das MSI-Paket ist im Adobeinstallationsverzeichnis<sup>3</sup> zu finden. Folgende zwei Dateien sollten nun für eine spätere automatisierte Verteilung aus diesem Verzeichnis kopiert werden: „Adobe Reader 7.0 - Deutsch.msi“ und „Data1.cab“.

### 2.1.2 Sun J2SE Runtime Environment 5.0

Beim *Java Runtime Environment (JRE)* ist das Extrahieren der MSI-Datei nicht ganz so trivial wie beim *Adobe Reader 7*. Zunächst muss das *Java Runtime Environment* als Offline-Installer von den Webservern von Sun heruntergeladen werden. Zum Zeitpunkt der Erstellung dieses Dokumentes ist die aktuelle Version die Version 5.0 Update 3 des *J2SE Runtime Environment*. Die Installation kann auf einem Test-PC, dessen Konfiguration für diese Aufgabe nebensächlich ist, gestartet werden. Nach dem Starten der Installationsdatei wird die Lizenzvereinbarung angezeigt (siehe Abbildung 2.2). An dieser Stelle der Setuproutine ist mittels des Explorer nach Dateien mit der Endung .MSI im Anwendungsdatenordner<sup>4</sup> des Profilverzeichnisses zu suchen. Zu beachten ist, dass in diesem Beispiel von einer Anmeldung mit dem lokalen Administratorenkonto ausgegangen wird und sich somit dieser Verzeichnispfad ergibt. Zudem muss bei den Suchkriterien auch die Option zur Suche in versteckten Ordnern und Systemordnern aktiviert sein. Das Suchergebnis liefert eine Datei namens „J2SE Runtime Environment 5.0 Update 3.msi“. Diese Datei ist für die Softwareverteilung zu kopieren. Das noch aktive Setup kann anschließend abgebrochen werden.

---

1 <http://www.adobe.de/products/acrobat/readermain.html>

2 C:\Programme

3 C:\Programme\Adobe\Acrobat 7.0\Setup Files\RdgBig\GER

4 C:\Dokumente und Einstellungen\Administrator\Lokale Einstellungen\Anwendungsdaten



Abbildung 2.2: Lizenzvereinbarung

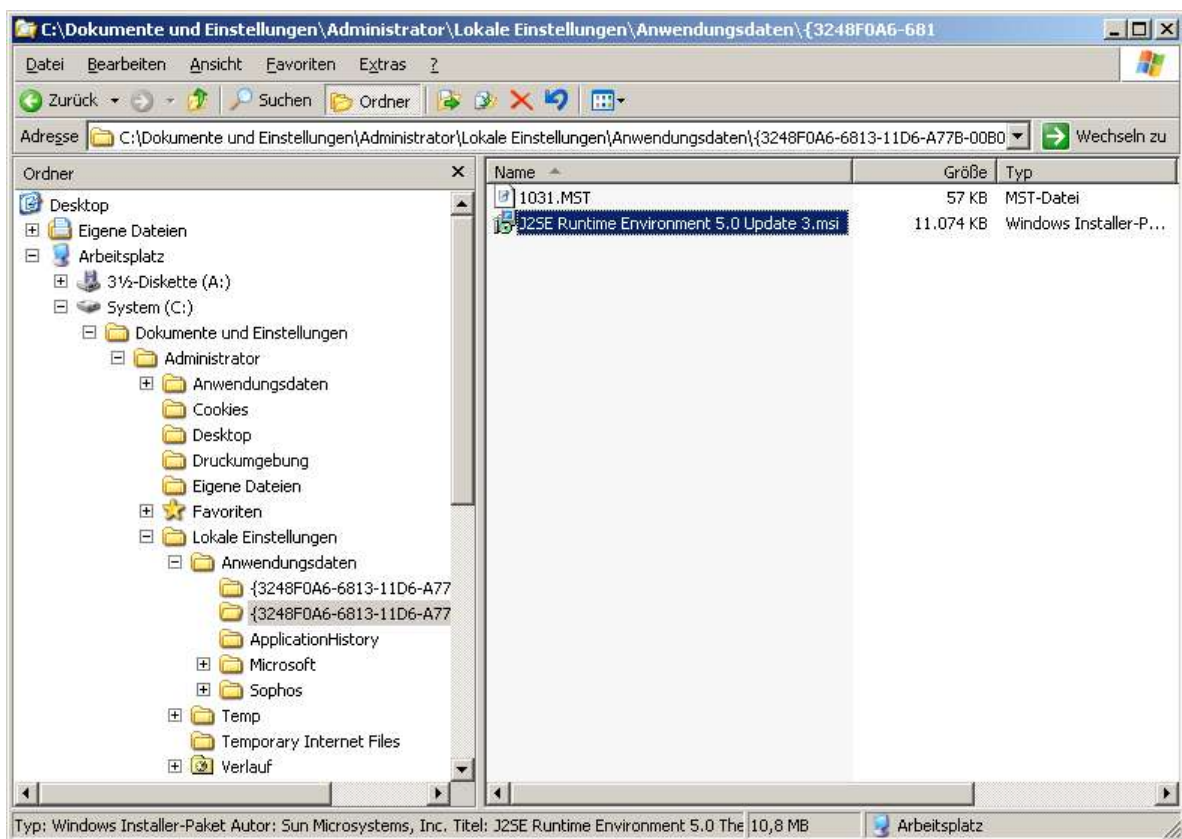


Abbildung 2.3: Pfad zum MSI-Paket des J2SE Runtime Environment 5.0 Update 3

## 2.2 Erstellen aus proprietären Installern

Ausgehend von einem proprietären Installer kann die Paketerstellung mit der Software *WinINSTALL LE 2003* der Firma OnDemand erfolgen. Wie schon anfangs erwähnt, stellt diese Software in der genannten Version eine kostenfreie Alternative dar und soll aus diesem Grund auch exemplarisch die Paketerstellung veranschaulichen. Es sollte ein Test-PC aufgesetzt werden, der nur über das Betriebssystem (in diesem Fall *Windows XP*) und alle verfügbaren ServicePacks und Sicherheitsupdates verfügt. Ansonsten sollten keine weiteren Softwareprodukte oder Komponenten installiert sein, um Abhängigkeiten der Softwarepakete zu vermeiden. Ein solcher Test-PC wird in der Hilfe von *WinINSTALL LE 2003* als „clean reference machine“ bezeichnet. Für weitere Anmerkungen und eine ausführliche Beschreibung der kompletten Prozedur sei an dieser Stelle auf diese Hilfe verwiesen. Auf einem weiteren (anderen) Computer ist *WinINSTALL LE 2003* zu installieren. Dieser muss nicht speziell konfiguriert werden, da der Test-PC lediglich auf eine Freigabe zugreift, die *WinINSTALL LE 2003* während der Installation anlegt hat, um den „Before-Snapshot“ und „After-Snapshot“ auszuführen. Diese zwei Snapshots sind Aufnahmen des Systemzustands vor und nach der Ausführung einer proprietären Installation. Anhand der verschiedenen Änderungen, wie neu hinzugekommene Programmdateien oder neue Systemregistrierungseinträge, die am Ursprungssystem vorgenommen wurden, wird anschließend das MSI-Paket erstellt. Am Beispiel der Software *GSview 4.7* wird im folgenden Abschnitt die Erstellung eines MSI-Paketes mit *WinINSTALL LE 2003* veranschaulicht.

Da für jedes Softwarepaket wieder eine „clean reference machine“ benötigt wird, ist es ratsam Clone-Werkzeuge wie *Symantec's Norton Ghost<sup>1</sup>* zu benutzen oder besser noch mit einer virtuellen Maschine zu arbeiten.

### 2.2.1 GSview 4.7

Es wird im folgenden davon ausgegangen, dass der Test-PC nach den Vorgaben der Hilfedokumentation von *WinINSTALL LE 2003* konfiguriert wurde und dass über eine Freigabe auf einem anderen Rechner auf das Programm „Disco32.exe“, welches für die Snapshots zuständig ist, zugegriffen werden kann. Zunächst ist die Anwendung *GSview 4.7* von der Herstellerwebseite<sup>2</sup> aus dem Internet zu laden. Die heruntergeladene Datei „gsv47w32.exe“ ist auf den Test-PC zu kopieren, an dem man mit administrativen Rechten angemeldet sei.

Vor der Ausführung der Datei „gsv47w32.exe“, der Setup-Routine, ist wie die Abbildung 2.4 zeigt, der Before-Snapshot über den Dialog Ausführen zu starten.

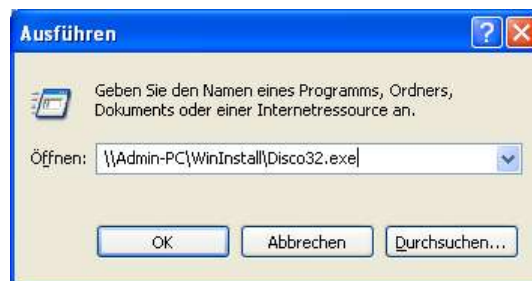


Abbildung 2.4: Aufruf des Before-Snapshots über die WinInstall-Freigabe

1 <http://www.symantec.com>

2 <http://www.ghostscript.com>

Die einzelnen Schritte des Before-Snapshots sollten in der Programmhilfe nachgeschlagen werden. Der letzte Schritt fordert den Anwender zur Eingabe des Installerprogramms auf, hier ist die Datei „gsv47w32.exe“ anzugeben. Nun installiert man wie gewohnt die Anwendung. Nach erfolgreicher Installation können noch gewünschte Veränderungen vorgenommen werden, wie z.B. das Löschen einer Desktopverknüpfung. Allerdings sollte man sich bewusst sein, dass jede Benutzeraktion eventuell als Veränderung des System mit in das MSI-Paket übernommen wird, was vielleicht nicht erwünscht ist. Abschließend ist über den Ausführen-Dialog wieder das Programm „Disco32.exe“ auszuführen, um den After-Snapshot zu initiieren, der zum Erzeugen des MSI-Pakets führt.

## 3 Bearbeiten von MSI-Paketen

Häufig ist es erwünscht, bereits existierende MSI-Pakete zu bearbeiten, um gewünschte Funktionen zu aktivieren oder zu deaktivieren. Dies können Programm-Features sein, aber auch das Entfernen von Desktopsymbolen oder das automatische Akzeptieren einer Lizenzvereinbarung, etc. An dieser Stelle sei erwähnt, dass die Dateierweiterung .MSI keine ausführbare Datei darstellt, sondern mit dem Programm „msiexec.exe“, dem *Windows Installer*, verknüpft ist. Ein Doppelklick auf eine MSI-Datei führt also automatisch zum Aufruf des *Windows Installers*. Dieser kann über Parameter konfiguriert werden. Eine genaue Beschreibung der Parameter kann durch Aufruf von „msiexec.exe“ in einer Eingabeaufforderung abgerufen werden. Mittels der Parameter kann der Ausführungsmodus (administrativ/interaktiv) bestimmt werden, aber es können auch Installationseigenschaften verändert werden. Wichtig ist zu wissen, wenn man die Softwareverteilung über Gruppenrichtlinien nutzt, wird die Installation immer im nichtinteraktiven Modus ausgeführt und man kann keine weiteren Parameter übergeben. Um dennoch die Installationseigenschaften zu ändern, werden Transformationsdateien verwendet. Sie besitzen die Dateierweiterung .MST.

Zur Erstellung der Transformationsdateien bieten einige Hersteller administrative Werkzeuge an. So liefert z.B. Adobe den *Installshield Tuner 7.0* für die *Adobe Acrobat* Produkte. Für die Office-Produkte stellt Microsoft die entsprechenden Resource Kits bereit, die eine Anpassung der MSI-Dateien an die Organisationvorgaben erlauben. Desweiteren gibt es MSI-Editoren, wie den Editor *Orca* von Microsoft, die eine produktunabhängige Bearbeitung zulassen. Generell ist es jedoch zu empfehlen, erst auf die Herstellerwerkzeuge zurückzugreifen, ehe man andere Hilfsmittel einsetzt.

Im nun folgenden Abschnitt wird gezeigt, wie man mit dem Werkzeug *Orca* sich eine MST-Datei erstellt, um gewünschte Änderungen am MSI-Paket „J2SE Runtime Environment 5.0 Update 3.msi“ vorzunehmen.

### 3.1 Bearbeiten mit Orca

Das Programm *Orca* ist ein Werkzeug zum Betrachten und Editieren von Paketen die mit dem Windows Installer ausgeführt werden können, also MSI-Dateien, aber auch MSP, MSM und PCP-Dateien. Letztere sind für diese Dokument nicht relevant. *Orca* ist Teil des *Microsoft Platform Software Development Kits* (PSDK), welches von der Microsoft-Webseite<sup>1</sup> heruntergeladen werden kann. Um an die Installationsdatei „Orca.msi“ zu gelangen ist eine Installation des PSDKs notwendig. Die Datei „Orca.msi“ ist anschließend im Programmverzeichnis<sup>2</sup> zu finden. *Orca* kann nun auf einem PC installiert werden, um für folgendes Szenario eine vernünftige Lösung zu finden:

Für das MSI-Paket „J2SE Runtime Environment 5.0 Update 3.msi“ soll eine Transformationsdatei erzeugt werden, die für eine geplante Verteilung der Software über Gruppenrichtlinien benötigt wird, um einige Änderungen am Softwarepaket vorzunehmen. Folgende Änderungen werden gewünscht: Die Integration des *JRE* in den *Internet Explorer* sowie in den *Mozilla-* und *Netscapebrowser* soll standardmäßig durchgeführt werden, die Update Funktion soll deaktiviert werden, da sich sowieso der Administrator um Aktualisierungen kümmert.

---

1 <http://www.microsoft.com/msdownload/platformsdk/sdkupdate/psdk-full.htm>

2 C:\Programme\Microsoft Platform SDK for Windows XP SP2\Bin

Die Firma Sun gibt auf ihrer Webseite<sup>1</sup> an, welche Setup-Parameter notwendig sind, um die gewünschten Änderungen vorzunehmen. Mit Hilfe des *Orca*-Editors lassen sich diese Parameter in der Tabelle „Property“ wiederfinden und bearbeiten. Da eine Transformation erstellt werden soll, muss im Menü „Transform“, die Auswahl „New Transform“ selektiert werden (siehe Abbildung 3.1).

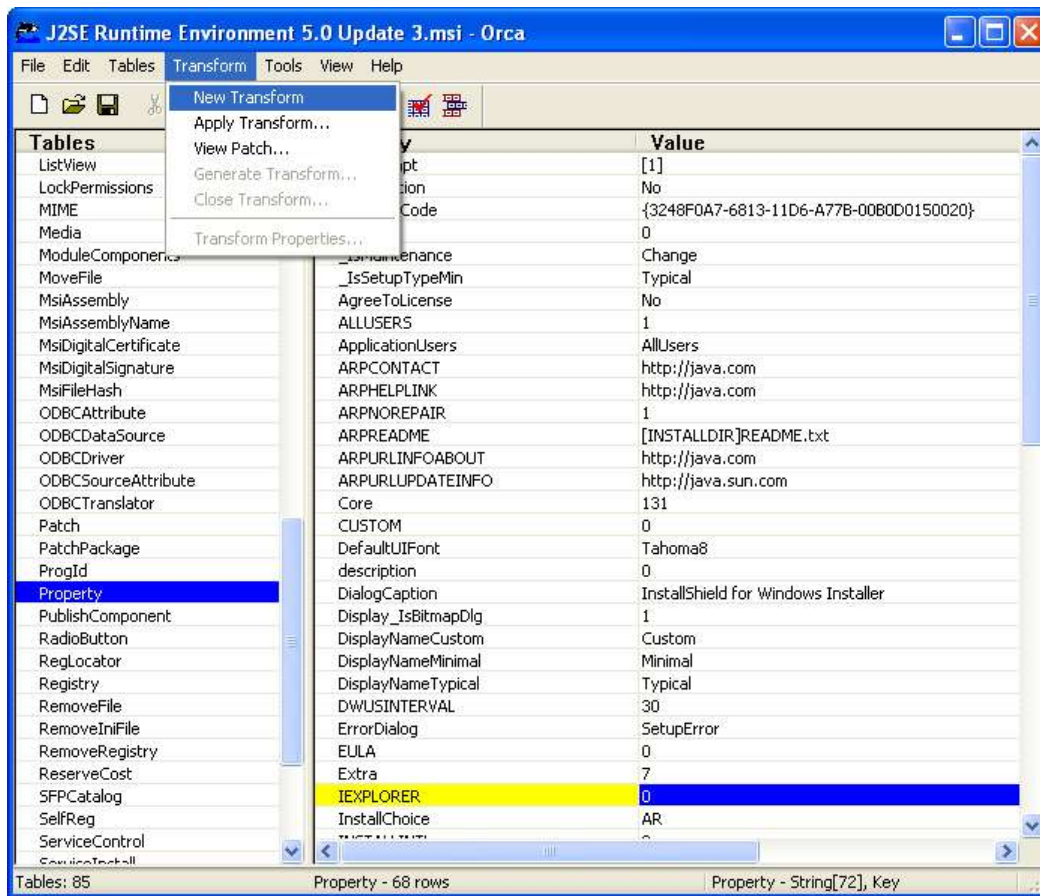


Abbildung 3.1: Erzeugen einer Transformationsdatei mit Orca

Die Abbildung zeigt die Eigenschaft „IEXPLORER“, deren Wert nun von 0 auf 1 geändert werden muss. Ebenso ist mit der Eigenschaft „MOZILLA“ zu verfahren und der Wert für die Eigenschaft „JAVAUPDATE“ ist von 1 auf 0 zu ändern. Die Transformation-Datei kann anschließend wieder über das Menü „Transform“ generiert werden. Wer komplexere Transformationen mit dem *Orca*-Editor erstellen will, sollte über Programmierkenntnisse und ein grundlegendes Verständnis über den Aufbau von MSI-Paketen mitbringen. Deshalb sei nochmals an dieser Stelle betont, dass zuerst auf Herstellerwerkzeuge zurückgegriffen werden sollte, bevor man nativ MSI-Pakete bearbeitet oder Transformationen erstellt.

## 3.2 Bearbeiten mit WinINSTALL LE 2003

*WinINSTALL LE 2003* bietet ebenfalls einen gewissen Funktionsumfang zum Editieren von MSI-Paketen. So können Komponenten wie Registrierungsschlüssel und Dateien hinzugefügt oder ent-

<sup>1</sup> <http://java.sun.com/j2se/1.5.0/docs/guide/deployment/deployment-guide/contents.html>

fernt werden oder ganze Features abgeschaltet werden, allerdings ist *WinINSTALL LE 2003* nicht in der Lage Transformationsdateien zu erstellen. Hierzu ist eine kostenpflichtige Version der OnDemand Produktpalette notwendig. Vorgenommene Änderungen werden in einem neuen MSI-Paket gespeichert. Deshalb empfiehlt sich die Verwendung von *WinINSTALL LE 2003* hauptsächlich zur Gewinnung von MSI-Pakete aus proprietären Installern, wie in Kapitel 2.2 gezeigt oder zur Erstellung eigener „einfacher“ MSI-Pakete wie z.B. ein Installationspaket für eine TrueType-Schriftart. Informationen zur Bedienung und zum Funktionsumfang von *WinINSTALL LE 2003* sind den beiliegenden Hilfedokumenten des Programms zu entnehmen.

## 4 Verteilen von MSI-Paketen

Mit der Einführung von *Windows 2000 Server* und den Microsoft-Verzeichnisdiensten (Active Directory Services) wurden den Administratoren auch mächtige Verwaltungswerkzeuge, die Gruppenrichtlinien, zur Seite gestellt. Gruppenrichtlinien erlauben eine umfangreiche Konfiguration von Benutzer- und Computerobjekten in einer Active-Directory-Domäne oder Domänenhierarchie. Exemplarisch seien hier Kennwortkonventionen, Browserverhalten oder Desktopvorgaben erwähnt. Neben solchen Konfigurationen ermöglichen Gruppenrichtlinien aber auch die Verteilung von Softwarepaketen, die im Windows-Installer-Format vorliegen. Es gibt im Grunde zwei Varianten der Softwareverteilung bzw. Softwareinstallation. Software kann über ein Gruppenrichtlinienobjekt auf Benutzer- oder Computerbasis installiert werden. Bei einer Installation auf Computerbasis, wird die Software während des Startvorgangs installiert und steht anschließend jedem Benutzer dieses Computers zur Verfügung. Bei einer Softwareinstallation auf Benutzerbasis steht nur Benutzern die Software zur Verfügung, auf die das Gruppenrichtlinienobjekt wirkt. Bei letzterer Variante gibt es wiederum einen Unterschied, ob Software veröffentlicht oder zugewiesen wird. Eine Zuweisung bewirkt die Installation während der Anmeldung; eine Veröffentlichung ermöglicht dem Benutzer ein eigenständiges Hinzufügen/Entfernen der Software über die Systemsteuerung.

In den nun folgenden Kapiteln wird bei der Verteilung von Softwarepaketen über Gruppenrichtlinien nur die Softwareinstallation auf Computerobjekte betrachtet, da dies den Verwaltungsmechanismus ausreichend veranschaulicht.

### 4.1 Einrichten einer Freigabe für die Softwareverteilung

Damit Clientcomputer oder Benutzer ein Softwarepaket beziehen können, ist ein zentraler Speicherort notwendig, der über eine Freigabe erreichbar ist. Sinnvollerweise wird hierfür ein Verzeichnis auf einem Dateiserver (Domänenmitgliedsserver) angelegt und freigegeben. Für die weiteren Schritte gehe man davon aus, dass das Verzeichnis „Softwareinstallation“ auf einem lokalen Datenträger angelegt wurde und unter gleichem Namen auch freigegeben wurde. Die Freigabe sollte nur Lesezugriff für die Gruppe authentifizierte Benutzer bereitstellen. Mitglieder dieser vordefinierten Gruppe sind alle Computerobjekte und Benutzerobjekte, die sich, wie der Gruppenname erkennen lässt, an der Domäne authentifiziert haben. Nach Belieben kann auch eine versteckte Freigabe verwendet werden.

In dem Verzeichnis „Softwareinstallation“ werden nun die Softwarepakete abgelegt. Der Übersicht wegen ist es empfehlenswert, für jedes Softwarepaket ein Unterverzeichnis mit beschreibendem Namen anzulegen. Die Abbildung 4.1 dient hierzu als Beispiel.

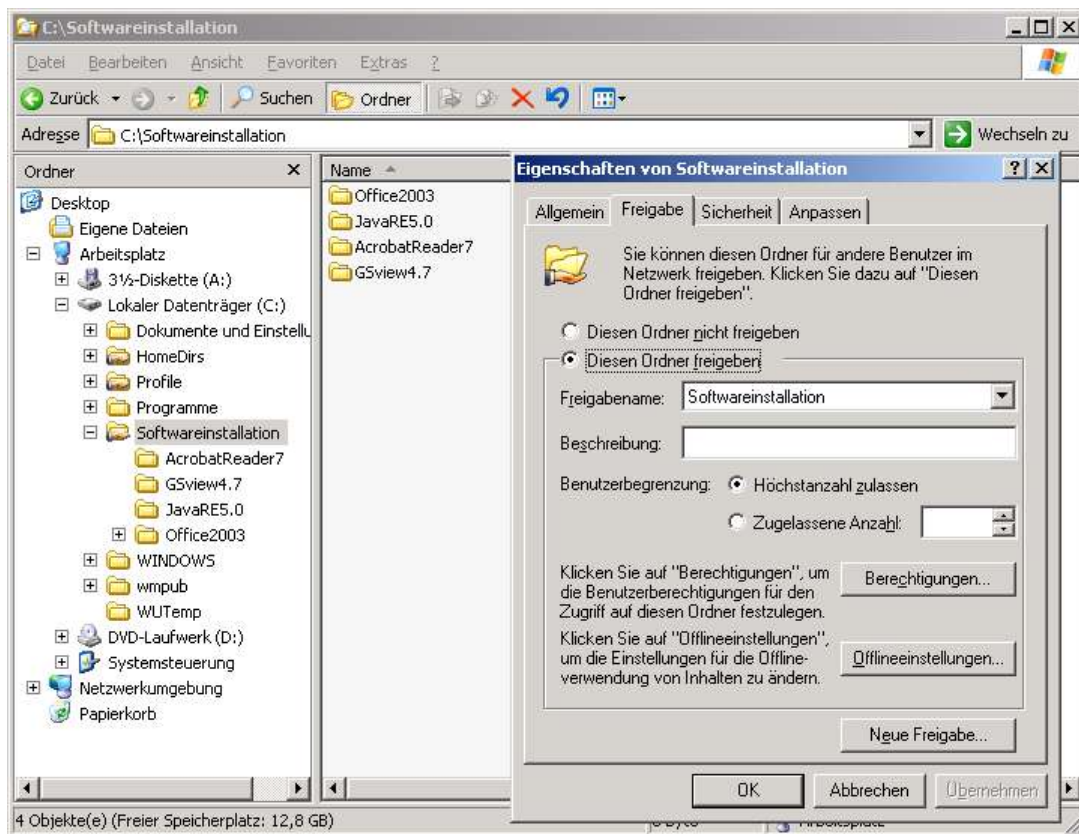


Abbildung 4.1: Beispiel einer Ordnerstruktur für die Softwarepakete

## 4.2 Einrichten von Gruppen

Nachdem die Verzeichnis und Freigabestruktur angelegt wurde, werden noch Gruppen benötigt, in denen alle Computer- und Benutzerobjekte zusammengefasst werden können, auf die letztendlich eine Softwareinstallation über die Gruppenrichtlinien zugeordnet wird. Empfehlenswert ist es für diese Gruppen im Active Directory eine eigene Organisationseinheit (OU) anzulegen und sich bei der Benennung an den Unterverzeichnissen der Softwarepakete zu orientieren (siehe Abbildung 4.2).



Abbildung 4.2: Einrichten von Gruppen

## 4.3 Einrichten der Gruppenrichtlinien

*Windows Server 2000* und *Windows 2003 Server* ohne Service Pack enthalten keine eigenständige Management-Console für die Verwaltung der Gruppenrichtlinien. Für *Windows Server 2003* ist jedoch die Gruppenrichtlinien-Verwaltungskonsolle (GPMC), eine Erweiterung mit vielen komfortablen Funktionen, verfügbar. Sie kann im Microsoft Downloadbereich<sup>1</sup> heruntergeladen werden. In diesem Dokument wird vorausgesetzt, dass Gruppenrichtlinien auf einem *Windows Server 2003* unter Einsatz dieser Konsole verwaltet werden. Ebenfalls beziehen sich die folgenden Abbildungen auf diese Konfiguration.

Folgendes Szenario sei gegeben: Das *Sun J2SE Runtime Environment 5.0* soll auf einigen Computern in einer Domäne über Gruppenrichtlinien verteilt werden. Zudem soll die Installation unter Verwendung einer Transformationsdatei angepasst werden. Es wurde bereits die MSI-Datei extrahiert, eine Transformationsdatei erstellt und beide auf einem Mitgliedsserver abgelegt. Die Dateien sind über eine Freigabe nach der Empfehlung des Kapitels 4.1 erreichbar.

Wichtig zu wissen ist, dass Gruppenrichtlinienobjekte, die Softwareinstallationen steuern, gewöhnlich auf Domänenebene implementiert werden. Ein beliebter Anfängerfehler ist es, OUs für die Wirkungsbereiche solcher Richtlinienobjekte zu verwenden. Die OUs sollen im Allgemeinen die Organisation abbilden. Der Wirkungsbereich von Softwareinstallationsrichtlinienobjekten wird über die Sicherheitsfilter gesteuert. Hier kommen die zuvor angelegten Gruppen ins Spiel. Eine empfehlende Vorgehensweise zeigt die Abbildung 4.3.

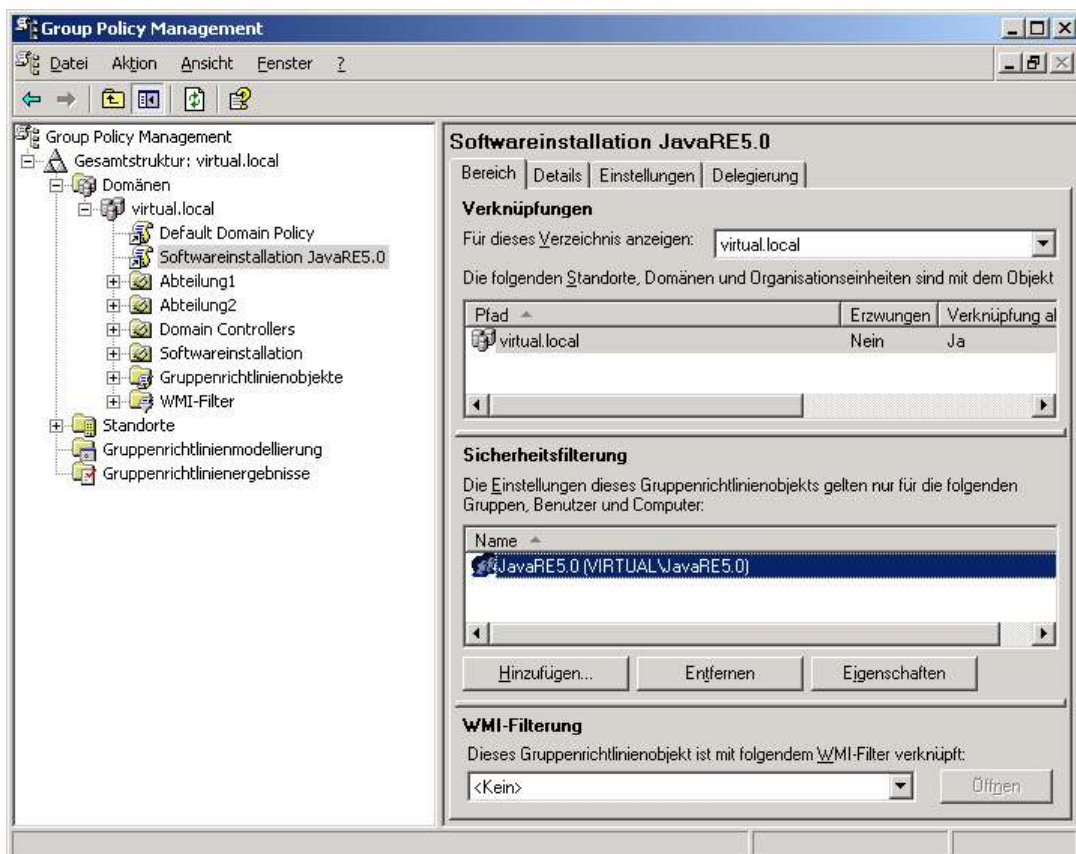


Abbildung 4.3: Einstellen der Sicherheitsfilterung für das Gruppenrichtlinienobjekt

<sup>1</sup> <http://www.microsoft.com/downloads>

Hier wird ein neues Gruppenrichtlinienobjekt mit Namen „Softwareinstallation JavaRE5.0“ angelegt und mit dem Domänencontainer verknüpft. Bevor nun die eigentliche Richtlinie implementiert wird, sollte in der Sicherheitsfilterung, welche unter dem Reiter „Bereich“ zu finden ist, die Gruppe Authentifizierte Benutzer entfernt werden. Im Gegenzug wird die entsprechende Gruppe (siehe Kapitel 4.1) hinzugefügt. Die Definition des Wirkungsbereiches sollte immer der erste Schritt bei Einrichten einer Gruppenrichtlinie sein, um z.B. eine ungewollte Installation einer Software auf sämtlichen Domänenrechnern zu verhindern.

Als nächstes wird das Gruppenrichtlinienobjekt bearbeitet. Ein Rechtsklick der Maus auf die Richtlinienverknüpfung „Softwareinstallation JavaRE5.0“ führt zum Bearbeitungsmodus. Da dieses Szenario von einer Softwareverteilung des *J2SE Runtime Environments* auf Computerobjekte ausgeht, wird nun das in Kapitel 2.1.2 extrahierte MSI-Paket „J2SE Runtime Environment 5.0 Update 3.msi“ über einen Rechtsklick auf die Softwareinstallation im Abschnitt „Computerkonfiguration/Softwareeinstellungen“ gemäß der Abbildung 4.4 hinzugefügt.

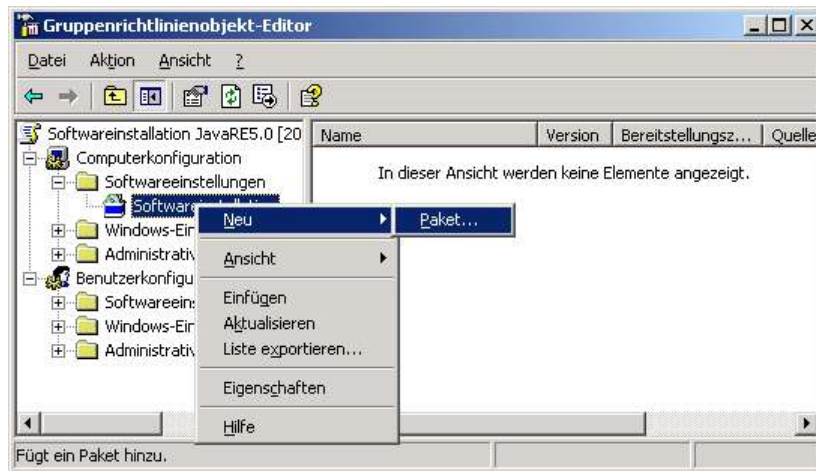


Abbildung 4.4: Hinzufügen eines MSI-Pakets

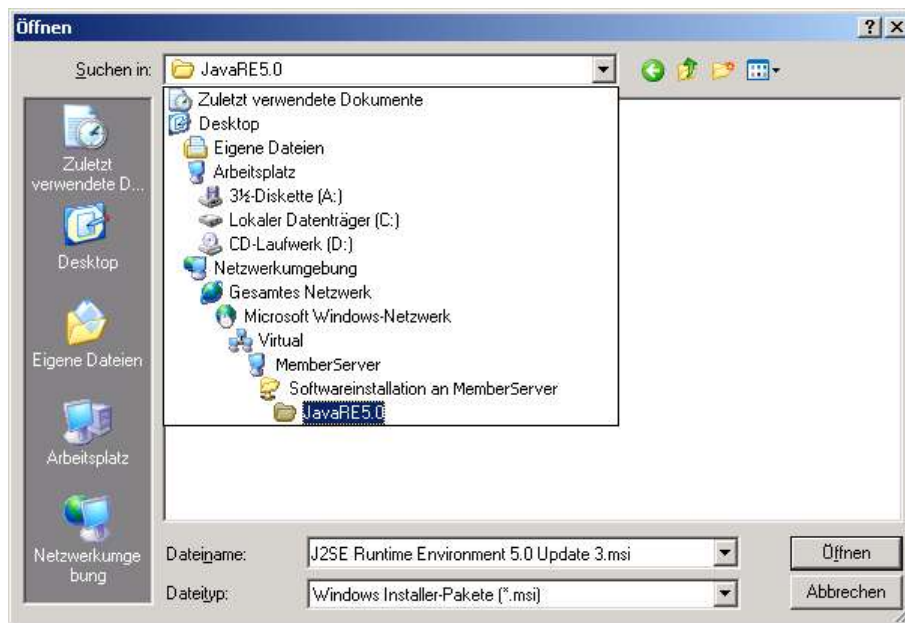


Abbildung 4.5: Öffnen der Datei unter Nutzung eines Netzwerkpfads

Es ist zu beachten, dass die Pfadangabe, in der die MSI-Datei zu finden ist, einen Netzwerkzugriff kennzeichnet, denn diese stellt den Pfad des Zugriffs für die Clients dar. Da sich hier die MSI-Dateien auf einem Mitgliedsserver befinden, der eine Verzeichnisstruktur aufweist, die den Empfehlungen des Kapitel 4.1 folgt, sieht die Pfadangabe entsprechend der Abbildung 4.5 aus.

Nach Auswahl der MSI-Datei erscheint eine Aufforderung, ob das Paket zugewiesen oder ob erweiterte Einstellungen vorgenommen werden wollen. Letzteres ist ratsam, wenn z.B. die Möglichkeit einer Deinstallation der verteilten Software oder die Abhängigkeit zu anderen Produkten beachtet werden muss. Insbesondere dann, wenn Transformationsdateien dem Paket mitgegeben werden sollen, die die Software den Bedürfnissen der Organisation anpassen (gemäß diesem Szenario), müssen erweiterte Einstellungen vorgenommen werden. Abbildung 4.6 zeigt das Einbinden der Transformationsdatei „JavaRE.mst“.

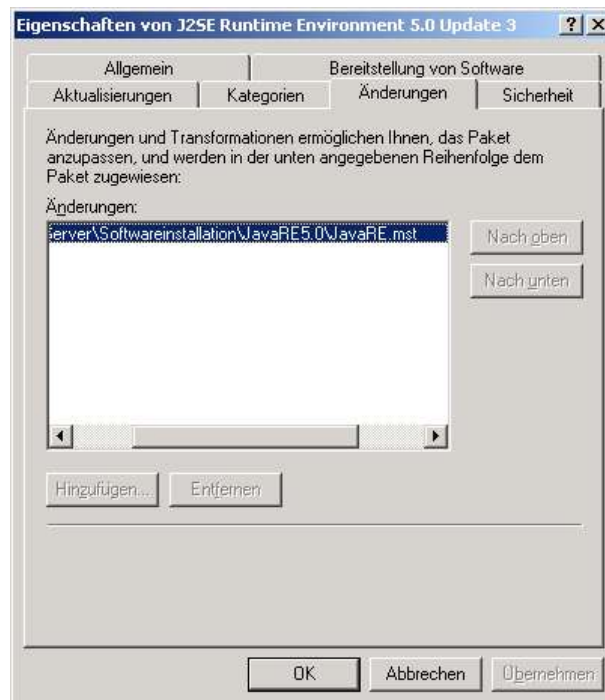


Abbildung 4.6: Hinzufügen einer Transformationsdatei

Im Prinzip ist nun das Gruppenrichtlinienobjekt konfiguriert, und die Software wird auf alle Computerobjekte angewendet, die Mitglied der Gruppe „JRE“ sind. Zwei kleine Optimierungen sind noch zu empfehlen. Zum einen, da die Richtlinie sowieso nur auf Computerobjekte angewendet wird, können die Benutzerkonfigurationseinstellungen dieser Richtlinie deaktiviert werden (siehe Abbildung 4.7), zum anderen sollte für Windows XP Clients noch die Richtlinie „Beim Neustart des Computers und bei der Anmeldung immer auf das Netzwerk warten“ aktiviert werden. Die Abbildung 4.8 zeigt, wo diese Richtlinie zu finden ist. Diese Einstellung verhindert, dass die Konfiguration einer Softwareinstallationsrichtlinie erst nach dem dritten Neustart eines Windows XP-Clients erfolgt. Ebenso ist es empfehlenswert alle Gruppenrichtlinienobjekte zur Softwareinstallation zu erzwingen, damit sie nicht von einer eventuell deligierten OU überschrieben werden können. Unter Windows 2000 heißt die Option zum Erzwingen „Kein Vorrang“.

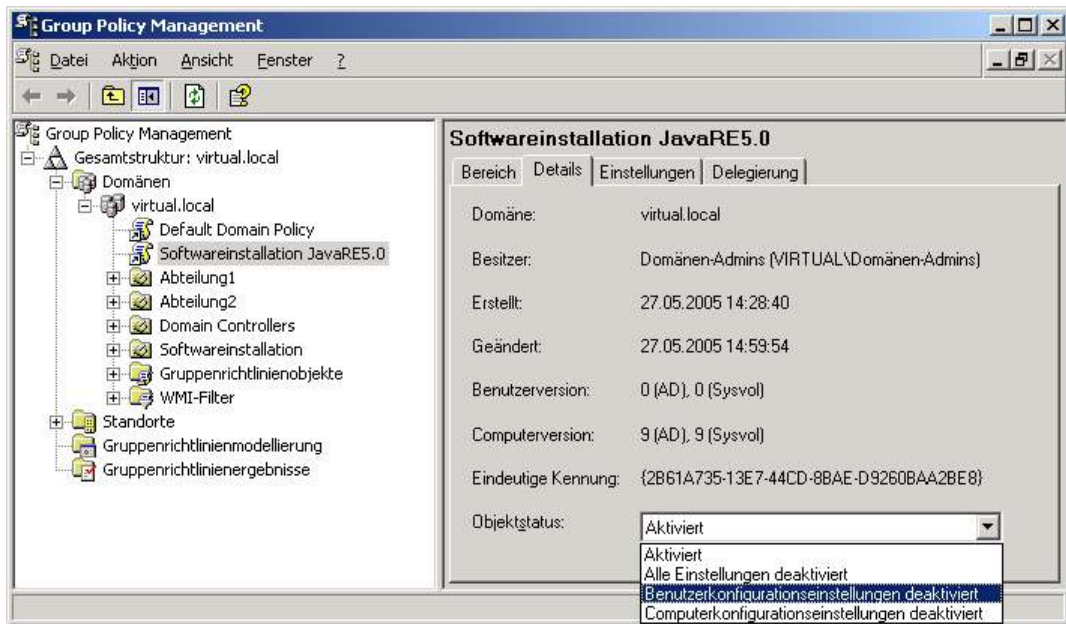


Abbildung 4.7: Benutzerkonfigurationseinstellungen deaktivieren

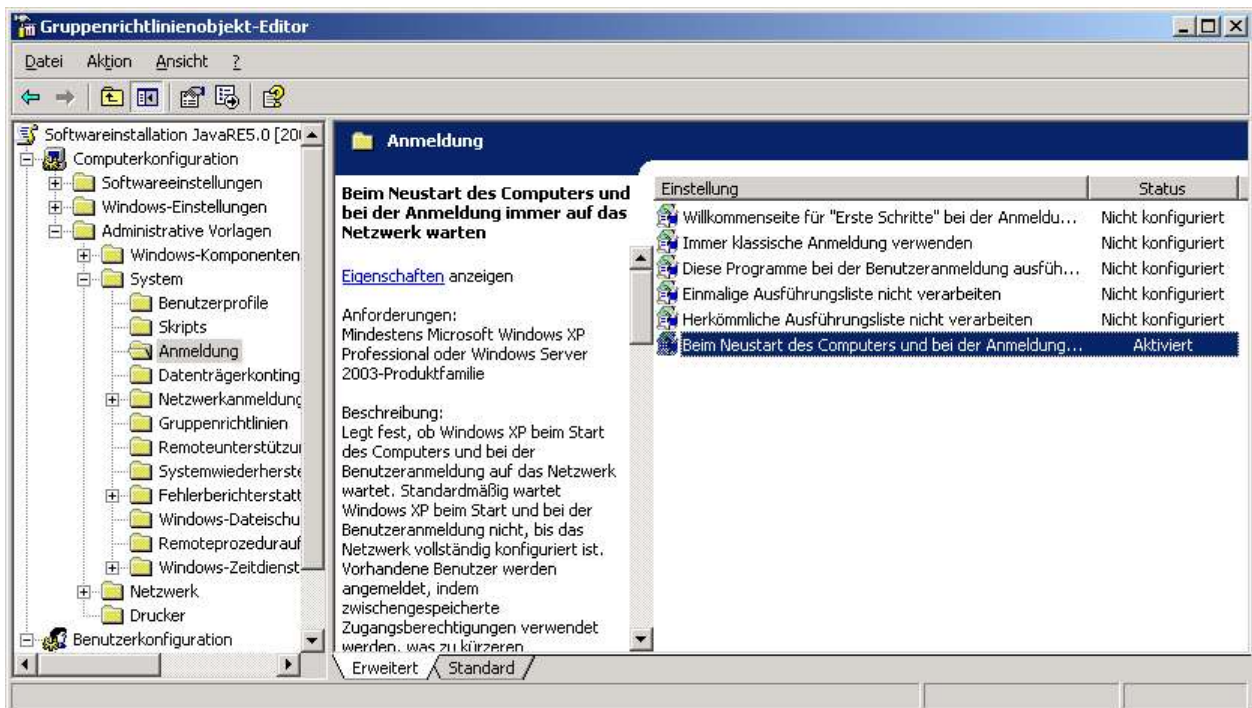


Abbildung 4.8: Weitere empfohlene Richtlinieneinstellung

Weitere Informationen zu Gruppenrichtlinien und deren Konfigurationsmöglichkeiten sind den Hilfedokumenten der Betriebssysteme *Windows 2000 Server* bzw. *Windows Server 2003* oder den bei Microsoft online verfügbaren Best-Practise-Guides und Handbüchern zu entnehmen. Ebenso sei für weitere Informationen über die in diesem Dokument erwähnten Produkte der Firma Microsoft, Sun, Adobe und das Produkt *GSview*, auf die entsprechenden Webseiten der Hersteller verwiesen.

## 5 Zusammenfassung

An zwei Produkten von Adobe und Sun wurde aufgezeigt, wie MSI-Pakete aus den vom Internet heruntergeladenen Dateien extrahieren werden können. Ebenso ist exemplarisch an der Software *GSview* dargestellt worden, wie man aus einem proprietären Installer mit *WinINSTALL LE 2003* ein MSI-Paket erzeugt. Diese drei Beispiele sollten dazu dienen, ein Gespür zu entwickeln, woran Software-Produkte zu erkennen sind, die den *Windows Installer* nutzen bzw. nicht nutzen. Im anschließenden Kapitel wurde kurz dargestellt, mit welchen kostenfreien Varianten sich die MSI-Pakete bearbeiten lassen und welchen Zweck Transformationsdateien erfüllen. Kapitel 4 zeigte die empfohlene Vorgehensweise auf, wie Softwarepakete mittels der Gruppenrichtlinien auf Clientcomputern verteilt werden.

Alle in diesem Dokument verwendeten Logos, Firmennamen und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen ihrer jeweiligen Eigentümer.

Dieses Dokument erhebt keinen Anspruch auf Vollständigkeit.