

Das Windows Server 2003 Codebook

**Unser Online-Tipp
für noch mehr Wissen ...**

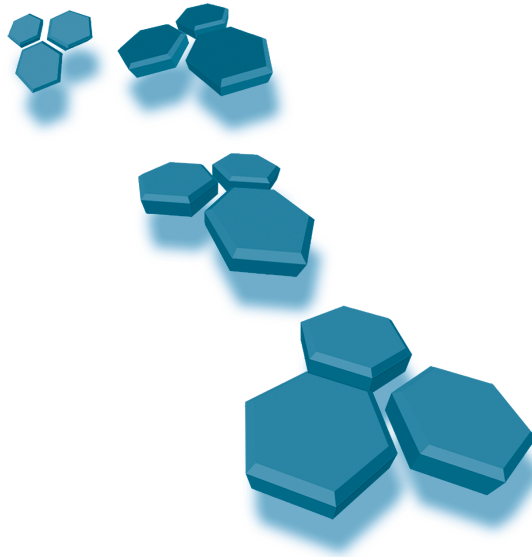


... aktuelles Fachwissen rund
um die Uhr – zum Probelesen,
Downloaden oder auch auf Papier.

www.InformIT.de

Annette Stolz

Das Windows Server 2003 Codebook



ADDISON-WESLEY

An imprint of Pearson Education

München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <<http://dnb.ddb.de>> abrufbar.

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

Fast alle Hardware- und Softwarebezeichnungen und weitere Stichworte und sonstige Angaben, die in diesem Buch verwendet werden, sind als eingetragene Marken geschützt. Da es nicht möglich ist, in allen Fällen zeitnah zu ermitteln, ob ein Markenschutz besteht, wird das ® Symbol in diesem Buch nicht verwendet.

Umwelthinweis:

Dieses Buch wurde auf chlorfrei gebleichtem Papier gedruckt. Die Einschumpffolie – zum Schutz vor Verschmutzung – ist aus umweltverträglichem und recyclingfähigem PE-Material.

10 9 8 7 6 5 4 3 2 1

08 07 06

ISBN-13: 978-3-8273-2277-7

ISBN-10: 3-8273-2277-4

© 2006 by Addison-Wesley Verlag,
ein Imprint der Pearson Education Deutschland GmbH,
Martin-Kollar-Straße 10-12, D-81829 München/Germany
Alle Rechte vorbehalten

Korrektur: Sandra Gottmann, sandra.gottmann@t-online.de
Fachlektorat: Jochen Ruhland, Daniel Frick, Gerd Bastuck
Lektorat: Sylvia Hasselbach, shasselbach@pearson.de
Herstellung: Elisabeth Prümm, epruemmm@pearson.de
Satz: reemers publishing services gmbh, Krefeld (www.reemers.de)
Umschlaggestaltung: Marco Lindenbeck, webwo GmbH (mlindenbeck@webwo.de)
Druck und Verarbeitung: Kösel, Krugzell (www.KoeselBuch.de)

Printed in Germany

Inhaltsverzeichnis

Vorwort	11
Teil I Einführung	15
Betriebssysteminstallation	17
Betriebssystem-Update	57
Eingabeaufforderung	79
Windows Script Host	109
Teil II Rezepte	123
Benutzer und Gruppen	125
1 Existiert der Benutzer bereits?	126
2 Benutzer auflisten	130
3 Neue Benutzer anlegen	134
4 Mehrere Benutzer auf einmal anlegen	139
5 Eigenschaften eines Benutzers auslesen	145
6 Benutzer ändern	153
7 Benutzer deaktivieren	159
8 Benutzer umbenennen	163
9 Benutzer löschen	165
10 Welcher Benutzer ist gerade angemeldet?	168
11 Benutzer abmelden	170
12 Eine sekundäre Anmeldung durchführen	173
13 Wann hat sich ein bestimmter Benutzer zuletzt angemeldet?	174
14 Einzelne Benutzer in andere Organisationseinheiten verschieben	178
15 Welche SID hat der Benutzer?	180
16 Kennwörter zurücksetzen	183
17 Existiert die Gruppe bereits?	186
18 Gruppen auflisten	191
19 Neue Gruppen anlegen	195
20 Eigenschaften einer Gruppe auslesen	200
21 Benutzer in Gruppen aufnehmen	205
22 Welchen Gruppen gehört der Benutzer an?	209
23 Welche Benutzer sind in einer bestimmten Gruppe?	214
24 Benutzer aus Gruppen entfernen	217
25 Gruppen löschen	221
26 Welche SID hat die Gruppe?	224

Dateien und Ordner**227**

27	Dateien und Ordner suchen	227
28	Welche Berechtigung ist auf die Datei oder auf den Ordner vergeben?	233
29	Berechtigungen für Dateien und Ordner einrichten und bearbeiten	243
30	Wer ist der Besitzer der Datei oder des Ordners?	251
31	Besitz von Dateien und Ordner übernehmen	253
32	Einen Dump der Datei- und Ordnerberechtigungen erstellen	256
33	Die aktuelle Dateiversion ermitteln	260
34	Datei- und Ordnergröße ermitteln	263
35	Dateien ab einer bestimmten Größe finden	266
36	Wann wurde die Datei oder der Ordner erstellt?	268
37	Wann wurde die Datei oder der Ordner zuletzt geändert?	272
38	Datums- und Zeitstempel von Dateien und Ordnern ändern	276
39	Datei- und Ordnerattribute ändern	278
40	Erweiterte Datei- und Ordneigenschaften auslesen	283
41	Dateien und Ordner kopieren und verschieben	286
42	Dateien und Ordner umbenennen	289
43	Dateien und Ordner komprimieren und dekomprimieren	292
44	Dateien und Ordner ver- und entschlüsseln	298
45	Dateien splitten und wieder zusammenfügen	301
46	Testdateien mit einer bestimmten Größe anlegen	305
47	Geöffnete Dateien überschreiben	306
48	Neue Dateitypen registrieren	308
49	Kopf- und Fußzeilen aus einer Textdatei entfernen	312
50	Inhalte von Textdateien miteinander vergleichen	316
51	Textdateien nach bestimmten Zeichenfolgen durchsuchen	321
52	Zeichenfolgen suchen und ersetzen	330
53	Leerzeilen entfernen	333
54	Nur die letzten x Zeilen einer Textdatei ausgeben	336
55	Doppelte Einträge finden und löschen	339
56	Textdateien mit einer Zeilennummerierung versehen	343
57	Die Anzahl der Wörter einer Textdatei ermitteln	347
58	Die Zeilenanzahl einer Textdatei ermitteln	348
59	Textdateiinhalte sortieren	350
60	Dateien und Ordner sichern und wiederherstellen	355

Datenträgerverwaltung**363**

61	Laufwerkinformationen auslesen	363
62	Aktuelle Startkonfiguration ermitteln	366
63	Die Startkonfiguration ändern	369
64	Den Datenträgerstatus ermitteln	374
65	Die Auslastung eines Datenträgers ermitteln	376
66	Partitionen und Volumes eines Datenträgers anzeigen lassen	379
67	Basisdatenträger in dynamische Datenträger konvertieren	382
68	Dynamischen Datenträger in Basisdatenträger konvertieren	386

69	Eine primäre Partition anlegen	388
70	Eine erweiterte Partition anlegen	391
71	Logische Laufwerke einrichten	393
72	Partitionen aktivieren	396
73	Volumes auf dynamischen Datenträgern erstellen	398
74	Partitionen, Volumes und logische Laufwerke löschen	402
75	Partitionen, Volumes und logische Laufwerke formatieren	405
76	Laufwerkbuchstaben und -pfade festlegen	409
77	Datenträgerbezeichnungen festlegen	414
78	FAT und FAT32 nach NTFS konvertieren	416
79	Datenträger defragmentieren	418
80	Kontingentverwaltung für Datenträger aktivieren	421
81	Neue Kontingenteinträge einrichten	424
82	Auslastung der Datenträgerkontingente abfragen	427

Netzwerk

431

83	Welche Netzwerkadapter sind installiert?	431
84	Welche MAC-Adressen haben die Netzwerkkarten?	435
85	Die aktuelle IP-Konfiguration ermitteln	437
86	Eine statische IP-Adresse zuweisen	440
87	Standardgateway konfigurieren	443
88	Dynamische IP-Adressen verwenden	446
89	Dynamische IP-Adressen lösen und erneuern	448
90	DNS-Einstellungen festlegen	450
91	WINS-Einstellungen konfigurieren	455
92	TCP/IP-Filter konfigurieren	458
93	Aktive TCP/IP-Verbindungen abfragen	464
94	Welche Freigaben sind eingerichtet?	466
95	Neue Ordnerfreigaben einrichten	469
96	Bestehende Freigaben ändern	474
97	Freigabeberechtigungen bearbeiten	476
98	Auf welche Freigabe wird gerade zugegriffen?	479
99	Aktive Freigabeverbindungen trennen	481
100	Netzlaufwerke verbinden und trennen	483
101	Freigaben entfernen	486
102	Freigaben im Active Directory veröffentlichen	489

Registrierung

493

103	Existiert der Schlüssel bereits?	494
104	Einzelne Schlüssel auflisten	496
105	Neue Schlüssel erstellen	500
106	Schlüssel umbenennen	502
107	Schlüssel löschen	504
108	Existiert der Wert bereits?	506
109	Einzelne Werte auflisten	509

110	Neue Werte erstellen	511
111	Werte umbenennen	515
112	Werte löschen	518
113	Welche Daten enthält der Wert?	520
114	Daten auflisten	523
115	Daten ändern	526
116	Daten löschen	529
117	Besitzer festlegen	531
118	Registrierungsrelevante Ereignisse überwachen	533
119	Registrierungseinstellungen exportieren	537
120	Registrierungseinstellungen importieren	538
121	Einen Dump der Registrierung erzeugen	540
122	Favoriten hinzufügen	542
123	Favoriten entfernen	544
124	Registrierungsdateien auflisten	547
125	Registrierung sichern und wiederherstellen	549
126	Berechtigungen innerhalb der Registrierung bearbeiten	556

Systemmanagement

563

127	Computerkonto erstellen	563
128	Computerkonto löschen	566
129	Computer einer Domäne hinzufügen	568
130	Computer aus Domäne herausnehmen	572
131	Computer- und Systeminformationen auslesen	575
132	Wie lange läuft Windows bereits?	579
133	Datum, Uhrzeit und Zeitzone abfragen und bearbeiten	582
134	Systeminformationen als Hintergrundbild anzeigen	589
135	Windows Update-Service konfigurieren	591
136	Welche Applikationen sind installiert?	596
137	Ereignisprotokollierung für das Herunterfahren deaktivieren	598
138	Systemereignisse abfragen	600
139	Eigenschaften der Ereignisprotokolle ändern	603
140	Ereignisprotokolle exportieren	606
141	Eigene Systemereignisse erstellen	608
142	Aktive Prozesse auslesen	610
143	Speicherauslastung aktiver Prozesse abfragen	612
144	Einzelne Prozesse beenden	614
145	Priorität eines Prozesses festlegen	617
146	Welche Dienste sind installiert?	619
147	Neue Dienste registrieren	623
148	Vorhandene Dienste löschen	626
149	Dienste anhalten und fortsetzen	628
150	Dienste starten und beenden	631
151	Eigenschaften der Dienste ändern	634
152	Größe der Auslagerungsdatei verändern	637

153	Umgebungsvariablen auslesen	639
154	Umgebungsvariablen bearbeiten	642
155	Computer herunterfahren und neu starten	646

Server-Rollen 651

156	Server zum Domänen-Controller heraufstufen	651
157	Domänen-Controller zum Server herabstufen	656
158	Domänenfunktionsebene heraufstufen	659
159	Gesamtstrukturfunktionsebene heraufstufen	663
160	Auf welchem Domänen-Controller ist der globale Katalog aktiviert?	666
161	Globalen Katalog aktivieren und deaktivieren	671
162	Inhaber der FSMO-Rollen suchen und finden	673
163	FSMO-Rollen auf andere Server übertragen	679
164	Welche Vertrauensstellungen sind eingerichtet?	683
165	Neue Vertrauensstellungen einrichten	687
166	Vertrauensstellungen überprüfen	690
167	Eine Vertrauensstellung entfernen	692
168	Einen DHCP-Server autorisieren	695
169	DHCP-Bereiche definieren	697
170	DHCP-Bereiche aktivieren und deaktivieren	701
171	Adressbereiche ausschließen	704
172	DHCP-Adressen reservieren	708
173	Server- und Bereichsoptionen konfigurieren	711
174	Anzahl der Konflikterkennungsversuche festlegen	715
175	DHCP-Statistik abfragen	718
176	Eine primäre Zone einrichten	720
177	Zonen löschen	723
178	Einen neuen Host-Eintrag erzeugen	725
179	DNS-Cache leeren	727
180	Dynamische Updates zulassen und verweigern	729
181	DNS-Ereignisprotokollierung konfigurieren	731

Automatisierung 735

182	Existiert der geplante Task bereits?	735
183	Geplante Tasks einrichten	741
184	Geplante Tasks bearbeiten	748
185	Geplante Tasks vorübergehend deaktivieren	752
186	Geplante Tasks starten und beenden	756
187	Geplante Tasks löschen	758
188	Benutzerkonto für die Task-Ausführung ändern	762
189	Automatischer Programmstart vor der Windows-Anmeldung	764
190	Automatischer Programmstart nach der Windows-Anmeldung	767
191	Automatischer Programmstart vor der Windows-Abmeldung	771
192	Automatischer Programmstart vor dem Herunterfahren	777

Teil III Anhang	781
Referenz	783
Stichwortverzeichnis	813

Vorwort

Es gibt mittlerweile eine ganze Reihe guter Bücher zum Thema *Windows Server 2003*. Da stellt sich einem doch prompt die Frage, warum noch ein weiteres? Nun, einer der wichtigsten Gründe ist, dass ich für meine Tätigkeit als Consultant, die vom Aufgabenbereich ja mehr als vielfältig ist, selbst schon des Öfteren ein Sammel-surium von (Automatisierungs-)Lösungen für die wichtigsten Aufgaben aus der *Windows Server 2003-Administration* benötigt habe.

Besonders dann, wenn man sich auf Grund der vielfältigen Aufgabengebiete nicht konstant mit ein und demselben Thema beschäftigen kann, sind bestimmte Dinge sehr schnell wieder vergessen. Und mit dem nächsten Projekt oder »Feuerwehreinsatz« gilt es dann, diese Dinge zum x-ten Male auf's Neue auszufüteln, womit meiner Meinung nach nur unnötig Zeit verschwendet wird. Daher schien es mir wichtig, die häufigsten Aufgabenstellungen aus der *Windows Server 2003-Administration* einmal aufzustellen und die Lösungen dabei nicht nur unter der grafischen Oberfläche aufzuzeigen, sondern insbesondere die Automatisierungsmöglichkeiten mit Hilfe der Eingabeaufforderung und per VBScript festzuhalten.

Ein Blick in die elektronische Glaskugel namens Google liefert zwar eine ganze Reihe von Lösungsmöglichkeiten für die unterschiedlichen Aufgabenstellungen, allerdings fehlen oftmals die nötigen Hinweise auf mögliche Stolpersteine, weshalb viele Lösungen erst einmal ein aufwändiges Troubleshooting mit sich bringen. Und wenn *Windows Server 2003* keine entsprechenden Möglichkeiten bereitstellt, so bleiben viele Aufgaben schlichtweg ungelöst.

Eine bestimmte Projektsituation zwang mich allerdings einmal dazu, die Grenze des auf den ersten Blick »Unmachbaren« etwas überschreiten zu müssen. Ich war gezwungen, Lösungen für die Automatisierung bestimmter Aufgaben zu schaffen, für die *Windows Server 2003* selbst keine entsprechenden Befehle bereitstellt, für die aber auch VBScript auf Grund des deaktivierten WSH nicht verwendet werden konnte. Nach langem Suchen bin ich auf die GNU-Utilities für Win32 gestoßen, mit denen bestimmte funktionale Lücken (z.B. das automatisierte Bearbeiten von Textdateien) dann letztlich doch noch recht effektiv geschlossen werden konnten.

Ich möchte Ihnen diese Sammlung nun in einer erweiterten Form für Ihre alltägliche Arbeit mit *Windows Server 2003* zur Verfügung stellen. Sie finden die einzelnen Aufgabenstellungen in Teil II des Buches – in verschiedenen Kategorien untergliedert – jeweils als Rezepte vor. Jedes Rezept enthält zu Beginn den Lösungsansatz unter grafischer Oberfläche. Im Anschluss daran werden die Lösungswege per Eingabeaufforderung und auch per VBScript beschrieben. Die Batch- und VBScript-Dateien sind dabei einfach und schlank gehalten und enthalten jeweils nur die Basis-Funktionalitäten, die zur Lösung der einzelnen Aufgaben benötigt werden. Um möglichst viele

Rezepte unterbringen zu können, habe ich aus platztechnischen Gründen auf jegliche Fehlerabfrage verzichtet. Sie können die Batch- und VBScript-Dateien größtenteils mit nur geringen Änderungen Ihren eigenen Anforderungen entsprechend anpassen und direkt einsetzen. Sie finden sämtliche Batch- und VBScript-Dateien, die in diesem Buch vorgestellt werden, auf der CD-ROM und können sie für eigene Zwecke gerne abändern und dann entsprechend verwenden.

Dieses Buch ist folgendermaßen gegliedert:

- ▶ Teil I stellt eine Einführung in die Basis-Themen dar:
 - ▶ Kapitel 1 zeigt, auf welche Art und Weise das Betriebssystem Windows Server 2003 installiert werden kann und wie sich die Betriebssystem-Installation vereinfachen und auch automatisieren läßt.
 - ▶ Kapitel 2 zeigt die Möglichkeiten der manuellen und automatischen Betriebssystem-Aktualisierung auf.
 - ▶ Kapitel 3 behandelt die Grundlagen zur Eingabeaufforderung.
 - ▶ Kapitel 4 beinhaltet eine kleine Einführung in den Windows Script Host und in das Erstellen von VBScript-Dateien.
- ▶ Teil II ist der Hauptteil des Buches und enthält über 180 Rezepte, wobei die Lösungen jeweils unter grafischer Oberfläche, via Eingabeaufforderung und auch per VBScript beschrieben sind:
 - ▶ Die Kategorie *Benutzer und Gruppen* zeigt, wie sich Benutzer- und Gruppenkonten anlegen, modifizieren und löschen lassen und wie Informationen zu Benutzer- und Gruppenkonten abgefragt werden können.
 - ▶ Die Kategorie *Dateien und Ordner* dreht sich um das Datei- und Ordner-Management und zeigt, wie sich Dateien und Ordner abfragen, erstellen, ändern und löschen lassen und auch, wie Text- und Protokolldateien auf automatischem Wege modifiziert werden können.
 - ▶ Die Kategorie *Datenträgerverwaltung* zeigt, wie sich Laufwerksinformationen abfragen, Partitionen und RAID-Konfigurationen einrichten und Datenträger verwalten lassen.
 - ▶ Die Kategorie *Netzwerk* zeigt, wie sich Informationen zur Netzwerkkonfiguration abfragen, Einstellungen ändern, Freigaben einrichten und auch bearbeiten lassen.
 - ▶ Die Kategorie *Registrierung* dreht sich um die Registrierung von Windows Server 2003 und zeigt, wie sich Registrierungseinstellungen abfragen, bearbeiten und auch löschen lassen.
 - ▶ Die Kategorie *System-Management* zeigt, wie sich zahlreiche Computerinformationen auslesen, neue Computerkonten anlegen und Dienste, sowie auch Ereignisse, verwalten lassen.

- ▶ Die Kategorie *Server-Rollen* zeigt unter anderem, wie Standalone- und Member-Server zu einem Domänen-Controller heraufgestuft, Domänen- und Gesamtstrukturfunktionsebenen angehoben, die Inhaber der FSMO-Rollen aufgefunden und Vertrauensstellungen verwaltet werden können.
- ▶ Die Kategorie *Automatisierung* dreht sich um den Taskplaner, mit dem die Ausführung von Batchdateien, VBScript-Dateien und sonstigen Programmen automatisiert werden kann.
- ▶ Teil III enthält den Anhang und bietet folgende Informationen:
 - ▶ Tabellen mit den Eigenschaften vieler WMI-Klassen, die in diesem Buch angesprochen werden.
 - ▶ Tabelle mit den regulären Ausdrücken, die Sie in VBScript verwenden können.
 - ▶ Tabelle mit den Parametern, die Sie für die automatische Einrichtung eines Domänen-Controllers verwenden können.

Danksagung

Seit meiner Ausbildung zur technischen Autorin, die ich 1999 mit großem Erfolg abgeschlossen habe, sind an meinem Schreibtisch viele Fachartikel und auch eine ganze Reihe von Büchern entstanden. Aber mit diesem Buch habe ich zum ersten Mal das Bedürfnis, mich bei einer ganzen Reihe netter Menschen zu bedanken, denn ohne ihre Unterstützung wäre dieses Buch nie zu Stande gekommen.

Zunächst einmal möchte ich Sylvia Hasselbach vom Addison Wesley-Verlag danken, die meine Buch-Idee netterweise aufgegriffen, das Ganze reichlich aufgemöbelt und damit überhaupt den wichtigsten Schritt in Richtung Windows Server 2003-Codebook getan hat. Frau Hasselbach, bei Ihnen bedanke ich mich ganz besonders für die sehr kompetente und sehr nette Beratung und insbesondere für die kontinuierliche Betreuung während der kompletten Projekt-Phase.

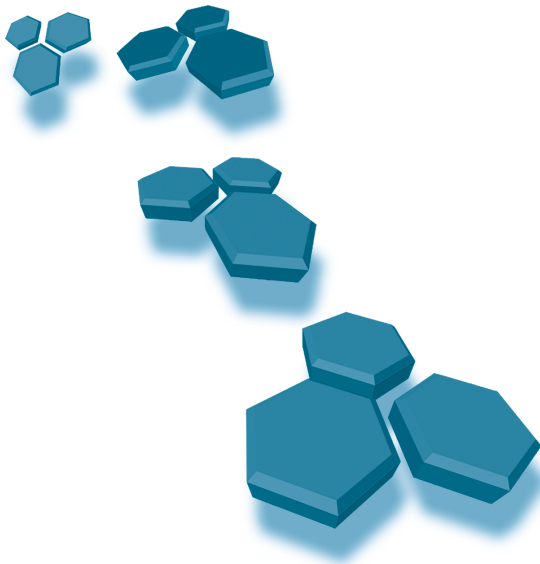
Meinem Manager Thomas Güttler bin ich ebenfalls zu großem Dank verpflichtet. Thomas, ohne Deine Unterstützung hätte ich nie die Zeit gehabt, diese guten 800 Seiten zu produzieren. Dir vielen lieben Dank für den Rückhalt, den du mir während der letzten Monate gegeben hast, für all die Unterstützung, die ich erhalten habe und auch für all die Aufmunterung und für den netten Zuspruch!

Franziska Hess gebührt ebenfalls ein riesengroßes Dankeschön. Denn Du, liebe Fränzi, hast ebenfalls dafür gesorgt, dass ich mich in den letzten Wochen (oder sollte ich besser sagen: Monaten?) voll und ganz meinem Text und auch den Skripten widmen konnte. Du hast damit ebenfalls sehr viel dazu beigetragen, dass es dieses Buch überhaupt gibt. Denn ohne Deine Unterstützung hätte mir einfach die Zeit dazu gefehlt. Mein Dank gilt hierbei natürlich auch Ludwig Neunobel und Jörg Haase, die damit ebenfalls ihren Beitrag zu diesem Buch-Projekt geleistet haben. Und auch Helmut Leist, der mich während der langen Projekt-Phase mit immens viel Humor stets bei Laune gehalten hat, gilt mein Dank.

Und last, but not least möchte ich natürlich auch meine Fach-Lektoren nicht vergessen haben, die mir mit reichlich konstruktiver Kritik zur Seite gestanden haben. Jochen Ruhland und insbesondere meine direkten Kollegen Dr. Günter Seibert, Konstanze Scholz, Daniel Frick und Gerd Bastuck. Auch bei Sandra Gottmann möchte ich mich bedanken, die den ganzen Text noch einmal gründlich gelesen und sprachlich korrigiert hat. Ihr habt einige Nachtschichten eingelegt. Dafür ein großes Dankeschön!

Annette Stolz, Dezember 2005

Teil I Einführung



Betriebssysteminstallation

Betriebssystem-Update

Eingabeaufforderung

Windows Script Host

Betriebssysteminstallation

Die erste Kategorie dieses Buches ist der Installation des Betriebssystems gewidmet, womit Sie die Basis dafür schaffen, um mit Windows Server 2003 arbeiten zu können. Beginnend mit einer kurzen Übersicht der Installationsmöglichkeiten und allgemeinen Vorüberlegungen zu einer Betriebssysteminstallation (Systemvoraussetzungen, Kompatibilitätscheck der verwendeten Hard- und Software etc.) finden Sie hier viele nützliche Hinweise und teils auch fertige Lösungen zu den unterschiedlichen Installationsmöglichkeiten und insbesondere auch zur Optimierung der Betriebssysteminstallation.

Die Erfahrung zeigt, dass eine gut durchdachte Installationsstrategie nicht nur den Grundstein für einen stabilen und sicheren Serverbetrieb legt, sondern auch jede Menge administrativer Zeit einspart, da sich viele Aufgaben, die ohnehin nach der Installation durchgeführt werden müssen, bereits in den Installationsprozess integrieren und damit auch automatisieren lassen. Bestes Beispiel hierfür sind die Einrichtung des Service Packs 1 und die zahlreichen Sicherheitsupdates, womit sich die Installation eines Servers gut und gerne über einen halben Tag lang hinweg erstrecken kann, wenn die Aktualisierungen noch von Hand durchgeführt werden. Was bei einem einzelnen Server vielleicht noch halbwegs im Erträglichen bleibt, kann bei der Einrichtung mehrerer Server oder gar einer größeren Server-Landschaft zu immensen Aufwänden führen.

Und die Erfahrung zeigt ebenfalls, dass gerade beim Neuaufbau oder beim Upgrade einer Server-Landschaft oder auch bei einem größeren Rollout von Windows Server 2003 oft weitaus mehr Systeme aufgesetzt werden müssen, als physikalisch überhaupt vorhanden sind. Denn hin und wieder kommt es zu Fehlinstallationen, oder es treten schier unlösbare Probleme auf, womit einzelne Computersysteme mehrmals installiert werden müssen und sich der Einrichtungsaufwand damit um ein Vielfaches erhöht.

Kurzum: Grundlegende Vorüberlegungen und geeignete Installationsstrategien sind nicht nur für die Verantwortlichen einer größeren IT-Infrastruktur von Interesse, sondern können sich auch für die Betreiber einer kleineren Server-Umgebung bezahlt machen. Eine simple Testumgebung, die des Öfteren neu aufgesetzt werden muss, reicht dabei schon aus.

Installationsmöglichkeiten in der Übersicht

Das Installieren des Betriebssystems erfolgt in den meisten Fällen auf manuellem Wege, indem die Installations-CD von Windows Server 2003 eingelegt, das Setup gestartet und die zur Einrichtung des Betriebssystems erforderlichen Konfigurationsparameter eingegeben werden. Es gibt jedoch noch eine ganze Reihe weiterer Möglichkeiten, wie Sie Windows Server 2003 einrichten können. Sie können die Installation beispielsweise teil- oder auch vollautomatisieren, eine fertige Windows

18 >> Installationsmöglichkeiten in der Übersicht

Server 2003-Installation per Gruppenrichtlinie einem bestimmten Computersystem zuweisen oder die fertige Betriebssysteminstallation per Datenträgerduplizierung auf andere Computersysteme übertragen.

Die nachfolgende Auflistung soll Ihnen einen kleinen Überblick über die wichtigsten und geläufigsten Installationsmethoden geben und die jeweiligen Voraussetzungen aufzeigen.

- ▶ **Manuelle Installation:** Die manuelle Installation von Windows Server 2003 erfolgt in der Regel bei der Einrichtung eines einzelnen oder nur weniger Server (bis max. fünf Computersysteme). Das Computersystem wird entweder mit Hilfe der Installations-CD von Windows Server 2003 gestartet, oder es wird eine Verbindung zu einer Freigabe im Netzwerk hergestellt, damit die Installationsdateien aus einem Distributionsordner bezogen werden können. Bei der manuellen Installation werden alle zur Einrichtung des Betriebssystems erforderlichen Konfigurationsparameter während des Setups von Hand eingegeben, weshalb sich diese Installationsmethode aufgrund der manuellen Eingriffe durch den Administrator nur für die Installation weniger Server eignet.
- ▶ **Unbeaufsichtigte Installation:** Bei der unbeaufsichtigten Installation handelt es sich um eine teil- oder wahlweise auch vollautomatische Betriebssysteminstallation unter Zuhilfenahme einer Antwortdatei, in der alle zur Einrichtung des Betriebssystems erforderlichen Konfigurationsparameter bereits hinterlegt sind. Bei dieser Installationsmethode wird das Computersystem ebenfalls von CD gestartet, oder es wird zunächst eine Verbindung zum Distributionsordner hergestellt, allerdings sind nach dem Start der Installationsroutine keine weiteren Interaktionen mehr erforderlich, da sämtliche Konfigurationsparameter aus der Antwortdatei gelesen und übernommen werden können. Diese Installationsmethode eignet sich für die Installation von mehr als fünf Computersystemen und setzt lediglich das Erstellen und Vorhandensein einer entsprechenden Antwortdatei voraus, in der die setup-spezifischen Einstellungen hinterlegt sind.
- ▶ **Cloning:** Das Klonen einer Betriebssysteminstallation erfolgt durch die Duplizierung der Systempartition oder der Festplatte, womit nicht nur das Betriebssystem als solches auf andere Computersysteme übertragen werden kann, sondern auch sämtliche Einstellungen und auch die installierten Applikationen. Da das Windows Server 2003-System 1:1 kopiert und übertragen wird, setzt das Cloning in der Regel eine identische Hardware-Ausstattung und auch eine nachträgliche Änderung der eindeutigen Systemeinstellungen (SID, Hostname, IP-Adresse ...) voraus. Dies lässt sich mit Hilfe des Deploy-Tools *Sysprep.exe* jedoch vereinfachen, da Sie mit diesem Tool das Computersystem für das Deployment vorbereiten und mit dem ersten Starten des geklonten Systems eine Änderung der SID und des Computernamens wie auch eine erneute Hardware-Erkennung initiieren können.

Das Klonen eines Computersystems erfordert den Einsatz einer speziellen Software (z.B. Drive Image, Ghost oder die Image-Tools von Acronis), damit ein Abbild der Installation erstellt werden kann. Dieses Abbild wird dann auf ein anderes Computersystem übertragen und mit Hilfe der Cloning-Software wieder

restauriert. Aufgrund der schnellen Verteilung einer kompletten Installation eignet sich diese Methode speziell für ein größeres Deployment von Windows Server 2003.

- ▶ Remote-Installationsdienste: Mit Hilfe der Remote-Installationsdienste (RIS) von Windows Server 2003 können Sie – ähnlich wie beim Cloning – ein Abbild einer vollständigen Betriebssysteminstallation (inklusive der bereits eingerichteten Applikationen) erstellen. Allerdings wird dieses Abbild nicht durch eine Datenträgerduplizierung erzeugt, sondern durch das Zusammentragen aller erforderlichen Dateien. Hierzu wird zunächst ein so genanntes Master-System eingerichtet, auf dem das Betriebssystem und wahlweise auch weitere Applikationen installiert werden. Anschließend wird von diesem Master-System ein Abbild erstellt und auf dem Server, auf dem die Remote-Installationsdienste eingerichtet sind, abgelegt. Die Computersysteme, die mit diesem Abbild eingerichtet werden sollen, werden dann über das Netzwerk (mit PXE-fähigen Netzwerkkarten) oder mit Hilfe einer speziellen Diskette gestartet, stellen eine Verbindung zum RIS-Server her und werden via Netzwerk mit dem Abbild betankt.

Die Remote-Installationsdienste von Windows Server 2003 stellen zwar eine recht komfortable und flexible Installationsmethode dar, allerdings setzt der Einsatz der Remote-Installationsdienste einiges voraus. Denn Sie benötigen zunächst einen Windows Server 2003, der als RIS-Server fungiert und auf dem die Remote-Installationsdienste eingerichtet sind. Die Systempartition des RIS-Servers muss mit NTFS formatiert sein, und es muss eine zweite Partition (oder eine zweite Festplatte) für die Ablage der Abbilder vorhanden sein, da die Abbilder nicht auf der Systempartition abgelegt werden können. Des Weiteren benötigen Sie einen funktionstüchtigen DHCP- und DNS-Server, damit die zu installierenden Computersysteme mit entsprechenden IP-Adressen versorgt und eine Verbindung zum RIS-Server herstellen können.

Entscheidend für die Auswahl einer geeigneten Installationsmethode ist zunächst die Anzahl der Server-Systeme, die installiert werden müssen. Sie sollten natürlich auch berücksichtigen, dass einige Installationsmethoden diverse Vorarbeiten erfordern (z.B. Installieren eines Master-Systems, Erzeugen einer Antwortdatei, Einrichten der Remote-Installationsdienste ...), weshalb der Aufwand der Vorarbeiten dem Zeitgewinn der einzelnen Installationen gegenübergestellt werden muss. Und nicht zuletzt ist auch entscheidend, ob die Installationsmethode nur für ein vorübergehendes Deployment oder für künftige Client-Server-Installationen verwendet werden soll.

Vorbereiten der Installation

Bevor Sie mit der Installation von Windows Server 2003 beginnen, sollten Sie die Installation zunächst entsprechend vorbereiten, damit das Setup reibungslos und ohne Unterbrechungen durchgeführt werden kann. Zu dieser Vorbereitung zählt in erster Linie das Überprüfen und Sicherstellen der Systemvoraussetzungen, da Windows Server 2003 einen gewissen Anspruch an die verwendete Hardware stellt.

Des Weiteren sollte die verwendete Hard- und Software auf mögliche Kompatibilitätsprobleme hin getestet werden, um mögliche Probleme während des späteren Serverbetriebes vermeiden zu können. Und zu guter Letzt gehört auch das Zusammentragen der erforderlichen Konfigurationsparameter zur Vorbereitung, damit das Setup zügig durchgeführt werden kann und nicht aufgrund fehlender Informationen unterbrochen oder gar komplett abgebrochen werden muss.

Überprüfen der Systemvoraussetzung

Die Installation von Windows Server 2003 setzt voraus, dass das Computersystem über eine Mindestausstattung verfügt. Die einzelnen Windows Server 2003-Editionen stellen hierbei unterschiedliche Anforderungen an die verwendete Computer-Hardware, weshalb leider keine allgemein gültige Aussage hinsichtlich der Systemvoraussetzungen getroffen werden kann.

Die nachfolgende Tabelle zeigt, welche minimalen und empfohlenen Systemvoraussetzungen gegeben sein müssen, um die einzelnen Windows Server 2003-Editionen (32-Bit) installieren zu können.

Windows Server 2003 Web Edition		
Komponente	Minimum	Empfohlen
Prozessor	133 MHz	550 MHz
Arbeitsspeicher	128 MByte	256 MB
Festplatte	1,5 GByte	2,0 GByte
Windows Server 2003 Standard Edition		
Komponente	Minimum	Empfohlen
Prozessor	133 MHz (einfach)	550 MHz (max. vierfach)
Arbeitsspeicher	128 MByte	256 MB (max. 4 GByte)
Festplatte	1,5 GByte	2,0 GByte
Windows Server 2003 Enterprise Edition		
Komponente	Minimum	Empfohlen
Prozessor	133 MHz (einfach)	550 MHz (achtfach)
Arbeitsspeicher	128 MByte	512 MB (max. 32 GByte)
Festplatte	1,5 GByte	2,0 GByte
Windows Server 2003 Datacenter Edition		
Komponente	Minimum	Empfohlen
Prozessor	400 MHz (achtfach)	733 MHz (max. 64-fach)
Arbeitsspeicher	512 MByte RAM	1 GByte (max. 1 TByte)
Festplatte	1,5 GByte	2,0 GByte

Tabelle 1: Die Systemvoraussetzungen der einzelnen Windows Server 2003-Editionen

Darüber hinaus muss das Computersystem über eine VGA-Grafikkarte (empfohlene Auflösung: 800 x 600 oder höher) verfügen und auch ein CD- oder DVD-Laufwerk beinhalten, wenn das Betriebssystem von CD installiert werden soll. Bei einer Installation via Netzwerk muss das Computersystem zusätzlich über eine funktionstüchtige Netzwerkanbindung verfügen, da anderenfalls keine Verbindung zum Distributionsordner hergestellt und somit nicht auf die Installationsdateien zugegriffen werden kann.

Hardware- und Treiber-Check

Vor der Installation von Windows Server 2003 sollten Sie der verwendeten Hardware und insbesondere auch deren Treiber entsprechende Beachtung schenken. Denn gerade Hardware-Komponenten, die entweder nicht Plug-&-Play-fähig sind oder eine Plug-&-Play-Unterstützung eines Drittanbieters verwenden, verursachen häufig Probleme oder müssen zumindest manuell nachkonfiguriert werden.

Microsoft testet die Treiber gängiger Hardware-Komponenten in einem speziellen Testlabor und unterzieht die Gerätetreiber einem WHQL-Test (*Windows Hardware Quality Labs*). Hat ein Treiber den Test bestanden, wird er von Microsoft zertifiziert und in die Windows Server 2003-Treiberdatenbank aufgenommen. Das Gerät wird damit auch in der Hardware-Kompatibilitätsliste (engl. = *Hardware Compatibility List*), zu finden unter www.microsoft.com, als unterstützte Komponente deklariert.

Prüfen Sie also noch vor der Installation, ob Sie Hardware-Komponenten verwenden, die nicht in der HCL enthalten sind, und testen Sie deren Funktionstüchtigkeit (zumindest vor einem größeren Rollout) entsprechend aus. Prüfen Sie auch, dass für alle verwendeten Hardware-Komponenten zertifizierte Treiber vorhanden sind. Besorgen Sie sich gegebenenfalls die entsprechenden Gerätetreiber für Windows Server 2003. In der Regel werden die aktuellen Treiber auf den Internetseiten der Gerätehersteller zum Download angeboten, meist auch mit dem entsprechenden Hinweis, dass die Treiber WHQL-zertifiziert sind.

Bereitstellen der Konfigurationsparameter

Last but not least sollten Sie vor der Installation von Windows Server 2003 noch alle erforderlichen Konfigurationsparameter und – falls erforderlich – auch die benötigten Treiber (z.B. für SCSI- oder RAID-Adapter) bereithalten, damit die Installation nicht aufgrund fehlender Informationen oder Treiber unterbrochen werden muss.

Welche Informationen zur Einrichtung von Windows Server 2003 erforderlich sind und welche Entscheidungen Sie während des Setups (für die Basisinstallation) treffen müssen, zeigt die nachfolgende Tabelle.

Parameter	Beschreibung
Systempartition	Während der Installation müssen Sie die Systempartition angeben, auf der Windows Server 2003 installiert werden soll. Falls bereits mehrere Partitionen eingerichtet wurden, so müssen Sie wissen, welche Partition Sie auswählen müssen oder welche Partition Sie gegebenenfalls löschen können, um eine neue Systempartition anlegen zu können. Letzteres erfordert auch die Angabe der Partitionsgröße.
Dateisystem	Nach der Auswahl der Systempartition müssen Sie das Dateisystem (FAT oder NTFS) auswählen, das verwendet werden soll. Entscheiden Sie also vorab, ob FAT oder NTFS verwendet werden soll (auf einem Server sollte NTFS gewählt werden, da dieses Dateisystem wesentlich mehr Sicherheit bietet).
Regions- und Sprachoptionen	Zu Beginn des grafischen Installationsteils können Sie die Regions- und Sprachoptionen festlegen, die später verwendet werden sollen. Hierzu zählen die Zahlen-, Währungs- und Uhrzeitformate wie auch die Standardinformation, die Sprachversion und das Tastaturlayout. Diese Informationen können bei Bedarf auch nach der Installation noch konfiguriert werden.
Name und Organisation	Während der Installation müssen Sie einen Namen und auch die Bezeichnung der Firma oder der Organisation eingeben. Legen Sie fest, welche Informationen an dieser Stelle verwendet werden sollen.
Produktschlüssel	Die Installation von Windows Server 2003 erfordert, dass Sie einen gültigen Produktschlüssel eingeben. Halten Sie also den Schlüssel für die Einzel- oder Firmenlizenz bereit.
Lizenzierungsmodus	Nach der Eingabe des Produktschlüssels müssen Sie zwischen dem Lizenzierungsmodus <i>Pro Server</i> und <i>Pro Gerät</i> oder Benutzer unterscheiden. Treffen Sie die entsprechende Wahl bezüglich der Clientzugriffslizenzen, und halten Sie die Information bereit.
Computername/Administratoren-Kennwort	Während der Installation müssen Sie einen gültigen und eindeutigen Computernamen und auch das Kennwort für den lokalen Administrator vergeben. Überlegen Sie vorab, welchen Computernamen Sie verwenden und welches Kennwort Sie vergeben möchten.

Tabelle 2: Die erforderlichen Informationen für die Betriebssysteminstallation

Parameter	Beschreibung
Datum- und Uhrzeiteinstellungen	Im grafischen Installationsteil können Sie das Datum und auch die Uhrzeit für den Server festlegen wie auch die entsprechende Zeitzone auswählen. Diese Konfigurationseinstellungen sind jedoch nicht zwingend erforderlich, da sich diese Einstellungen auch nach der Installation noch ändern lassen.
Netzwerkeinstellungen	Wenn Sie die Netzwerkeinstellungen bereits während der Installation anpassen möchten, so müssen Sie wissen, welche Client-, Dienst- und Protokollkomponenten benötigt werden und mit welchen Parametern sie jeweils konfiguriert werden müssen. Bei Verwendung des TCP/IP-Protokolls benötigen Sie beispielsweise eine statische IP-Adresse, die Subnetzmaske, gegebenenfalls das Standard-Gateway wie auch die DNS- und wahlweise auch die WINS-Einstellungen.
Arbeitsgruppen- oder Domänenmitgliedschaft	Während der Installation müssen Sie festlegen, ob der Server Mitglied einer Arbeitsgruppe oder aber einer Domäne werden soll, und auch die entsprechende Arbeitsgruppe oder Domäne angeben.

Tabelle 2: Die erforderlichen Informationen für die Betriebssysteminstallation (Forts.)

In Ergänzung zu den Windows Server 2003-spezifischen Konfigurationsparametern sollten Sie vor Beginn der Installation auch überlegen, welche Service Packs (derzeit SP1), Hotfixes, Fixpacks und Sicherheitsupdates nach dem Setup eingerichtet werden müssen und die Updates auch entsprechend bereithalten, damit sie im Anschluss an die Betriebssysteminstallation direkt eingerichtet werden können.

Slipstream-Versionen erzeugen

Wenn Sie mehrere Computersysteme mit Windows Server 2003 einrichten und nach dem Setup auch direkt aktualisieren müssen, so können Sie sich die Installation bzw. die anschließende Aktualisierung des Betriebssystems erheblich vereinfachen, indem Sie eine entsprechende Slipstream-Version erzeugen, in der die Updates bereits enthalten sind. Denn wenn Sie das Betriebssystem mit den bereits aktualisierten Dateien installieren, so brauchen Sie die Updates später nicht mehr einzurichten, da sie bereits während der Installation des Betriebssystems verwendet und damit auch eingerichtet wurden.

Sie können beispielsweise eine Slipstream-Version *Windows Server 2003 inklusive Service Pack 1* erstellen und die Installation des Betriebssystems mit den aktualisierten Dateien des Service Packs 1 vornehmen, womit die nachträgliche Installation des Service Packs 1 dann überflüssig wird, da die aktualisierten Dateien bereits mit der Betriebssysteminstallation eingerichtet werden können.

24 >> Slipstream-Versionen erzeugen

Um nun eine solche Slipstream-Version von Windows Server 2003 zu erstellen und die entsprechenden Updates in den Installationsvorgang einzubeziehen, kopieren Sie zunächst den Inhalt des *i386*-Ordners der Installations-CD in einen Ordner, entpacken dann die Dateien des Service Packs 1 in einen separaten Ordner und führen eine manuelle Aktualisierung der Installationsdateien von Windows Server 2003 durch. Bei Bedarf können Sie danach noch weitere Fixpacks, Hotfixes und Sicherheitsupdates integrieren. Anschließend geben Sie den Ordner mit den aktualisierten Installationsdateien frei und führen eine manuelle oder automatische Installation von Windows Server 2003 durch, womit die Einrichtung des Betriebssystems unter Verwendung der aktualisierten Dateien erfolgt. Das Betriebssystem ist damit bereits nach Abschluss aller Installationsarbeiten auf dem gewünschten Stand und braucht nicht noch einmal von Hand auf den gewünschten Patch-Level gebracht zu werden.

Achtung

Wenn Sie das Service Pack 1 und gegebenenfalls noch weitere Fixpacks, Hotfixes und auch Sicherheitsupdates in die Slipstream-Version von Windows Server 2003 einbinden, so ersparen Sie sich damit zwar das nachträgliche Patchen des Servers, doch können Sie die Aktualisierungspakete später nicht mehr deinstallieren, da für die Installation des Betriebssystems bereits die aktuellen Dateien verwendet wurden und die ursprünglichen Dateien damit nicht mehr zur Verfügung stehen. Vor den ersten produktiven Installationen sollten Sie die Slipstream-Version von Windows Server 2003 also unbedingt testen.

Das Service Pack 1 integrieren

Wenn Sie in künftige Betriebssysteminstallationen automatisch das Service Pack 1 einfließen lassen und sich die nachträgliche Einrichtung des Service Packs 1 damit ersparen möchten, so erzeugen Sie hierfür eine entsprechende Slipstream-Version und ersetzen die Installationsdateien von Windows Server 2003 durch die Service Pack 1-Dateien. Folgendes benötigen Sie dazu:

- ▶ 1,3 GByte freier Festplattenspeicher (zumindest temporär)
- ▶ Installations-CD von Windows Server 2003
- ▶ Service Pack 1

Erstellen Sie zunächst zwei Ordner für die Windows Server 2003- und auch die Service Pack 1-Dateien. Legen Sie beispielsweise im Root des Laufwerks C einen Ordner *Software* an, und erzeugen Sie darin die Ordner *Win2003Srv* und *Win2003SrvSP1*. Kopieren Sie dann den Ordner *i386* der Installations-CD von Windows Server 2003 in den Ordner *C:\Software\Win2003Srv*, und stellen Sie das Service Pack 1 (*Windows-Server2003-KB889101-SP1-x86-DEU.exe*) im Ordner *C:\Software\Win2003SrvSP1* bereit.

Für die Aktualisierung der Installationsdateien von Windows Server 2003 können Sie vorab den Inhalt des Service Packs 1 extrahieren. Hierzu rufen Sie die Eingabeaufforderung auf und wechseln in den Ordner, der das Service Pack 1 bereithält. Dort starten Sie die *exe*-Datei des Service Packs 1 mit dem Parameter */x*. Geben Sie in der Eingabeaufforderung also den Befehl `WindowsServer2003-KB889101-SP1-x86-DEU.exe /x` ein. Daraufhin wird ein kleines Dialogfenster angezeigt, worüber Sie den Zielordner für die extrahierten Dateien festlegen können. Geben Sie den Ordner an, der auch die *exe*-Datei des Service Packs 1 enthält, und klicken Sie auf OK. Die Dateien werden nun entpackt.

Wenn die Dekomprimierung der Service Pack 1-Dateien abgeschlossen ist, wechseln Sie in den Ordner `C:\Software\Win2003SrvSP1\i386\Update` und führen dort den Befehl `Update /Integrate:C:\Software\Win2003Srv` (ohne Angabe des darunter liegenden *i386*-Ordners) aus, woraufhin die Installationsdateien von Windows Server 2003 mit den Dateien aus dem Service Pack 1 aktualisiert werden. Sobald alle Windows Server 2003-Installationsdateien durch die neueren Versionen ersetzt wurden, erscheint die Meldung **DIE INTEGRIERTE INSTALLATION WURDE ERFOLGREICH ABGESCHLOSSEN**, und die Slipstream-Version ist erstellt.

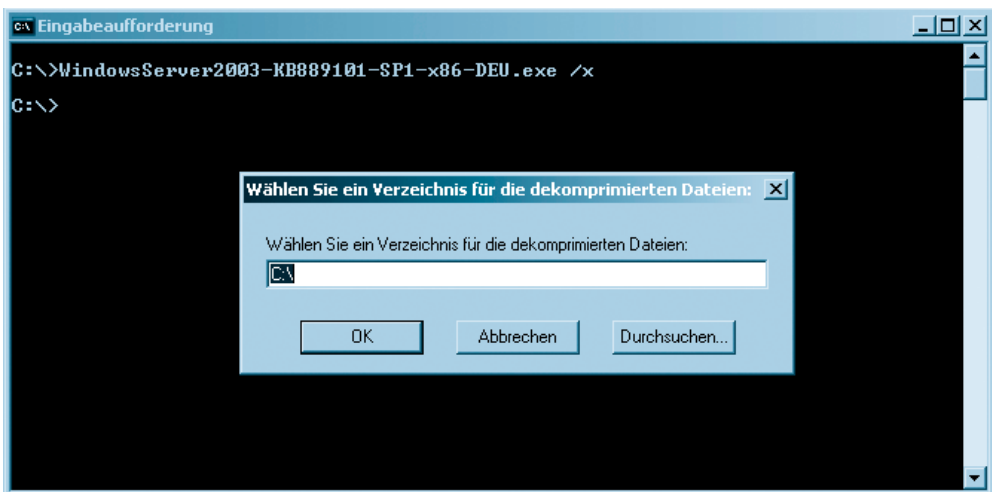


Abbildung 1: Den Pfad für die komprimierten Dateien festlegen

Für den Fall, dass Sie keine weiteren Updates integrieren und die künftigen Betriebssysteminstallationen aus dem Ordner heraus starten möchten, der die aktualisierten Installationsdateien enthält, so legen Sie nun die erforderlichen Berechtigungen für den Ordner fest, geben den Ordner frei und richten die Freigabeberechtigungen ein. Danach können Sie die Slipstream-Version durch Aufruf der Datei `C:\Software\Win2003Srv\i386\Winnt.exe` über eine Netzwerkverbindung starten.

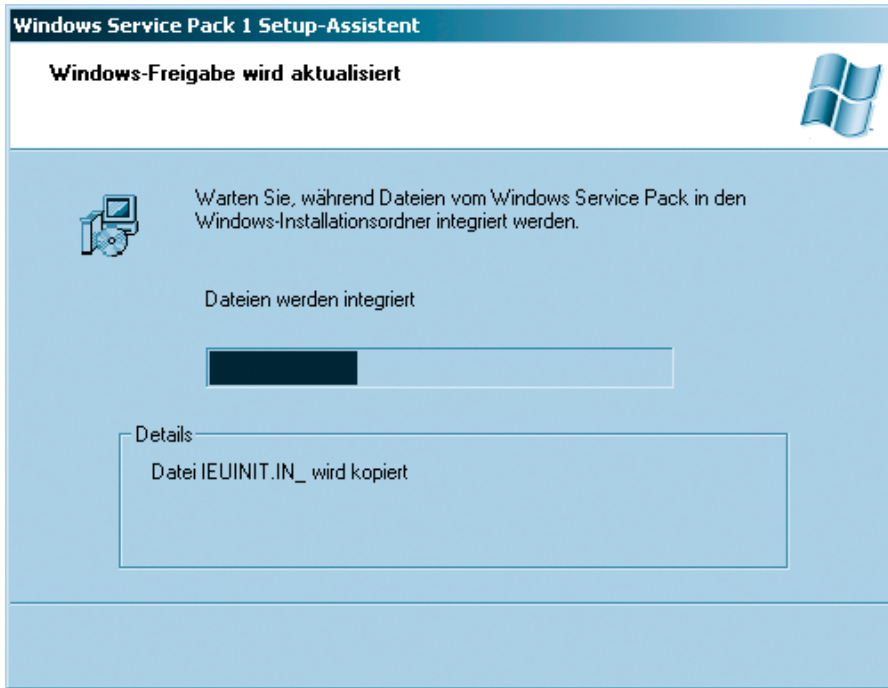


Abbildung 2: Die Installationsdateien werden mit den Service Pack1-Dateien aktualisiert.

Weitere Fixpacks, Hotfixes und Sicherheitsupdates hinzufügen

Auf der Internetseite www.microsoft.com werden in regelmäßigen Abständen aktuelle Fixpacks, Hotfixes und auch Sicherheitsupdates für den Download angeboten, womit bekannte Fehler behoben und Sicherheitsschwachstellen geschlossen werden können. Diese zusätzlichen Aktualisierungspakete können natürlich ebenfalls in die Slipstream-Version eingebunden werden, womit Sie das nachträgliche Patchen des Server-Systems bereits in die Betriebssysteminstallation integrieren können.

Wenn Sie nun eine existierende Slipstream-Version um weitere Fixpacks, Hotfixes und Sicherheitsupdates erweitern möchten, so laden Sie zunächst die entsprechenden Aktualisierungspakete von der Internetseite www.microsoft.com herunter und extrahieren die Inhalte der jeweiligen Pakete in entsprechende (zumindest temporäre) Ordner, indem Sie die *exe*-Datei des Aktualisierungspaketes mit dem Parameter */x* aufrufen.

Das weitere Vorgehen hängt nun davon ab, ob das Aktualisierungspaket über eine Datei mit der Bezeichnung *Update.exe* verfügt und ob der Inhalt des Aktualisierungspaketes *Driver.cab*-relevante Dateien enthält, da aktualisierte Treiber explizit eingebunden werden müssen, weil sie anderenfalls bei der nächsten Betriebssysteminstallation ignoriert werden.

Slipstreaming mit Update.exe

Wenn Sie ein Aktualisierungspaket von der Internetseite *www.microsoft.com* heruntergeladen und in einem (zumindest temporären) Ordner durch Aufruf der exe-Datei mit dem Parameter `/x` entpackt haben, so prüfen Sie zunächst, ob das Aktualisierungspaket eine Datei mit der Bezeichnung *Update.exe* der Version 5.4.15.0 oder höher enthält.

Wenn dies der Fall ist, so können Sie die aktuellen Dateien ganz einfach in die Slipstream-Version von Windows Server 2003 einbinden, indem Sie die Befehlszeile `Update/Integrate:C:\Software\W2003Srv` aufrufen, wobei Sie den Pfad zur Slipstream-Version von Windows Server 2003 natürlich durch den entsprechenden Pfad ersetzen. Damit werden die Dateien, die in der Slipstream-Version von Windows Server 2003 enthalten sind, durch die Dateien aus dem Aktualisierungspaket ersetzt.

Slipstreaming ohne Update.exe

Sollte das Aktualisierungspaket jedoch keine Datei mit der Bezeichnung *Update.exe* oder aber eine *Update.exe* einer älteren Version enthalten, so lässt sich die Slipstream-Version von Windows Server 2003 nicht ganz so einfach aktualisieren, sondern erfordert ein paar manuelle Handgriffe.

Wechseln Sie zunächst in den Distributionsordner, der die bereits bestehende Slipstream-Version von Windows Server 2003 (z.B. inklusive Service Pack 1) enthält, und legen Sie unterhalb von `\i386` einen weiteren Ordner mit der Bezeichnung `\Svc-pack` an. Kopieren Sie nun alle Aktualisierungspakete in den Ordner `\i386\Svcpack` des Slipstream-Distributionsordners, und benennen Sie die Dateien gemäß der 8.3-Namenskonvention (z.B. *KB892313.exe*) um. Extrahieren Sie dann die einzelnen Aktualisierungspakete in temporäre Ordner unterhalb von `\i386\Svcpack`, indem Sie die exe-Dateien mit dem Parameter `/x` aufrufen und die entsprechenden Zielordner festlegen.

Öffnen Sie anschließend die temporären Ordner mit den extrahierten Aktualisierungsdateien, und kopieren Sie jeweils die Catalog-Datei (*.cat) in den Ordner `\i386\Svcpack` (ausgenommen hiervon sind die Dateien mit der Bezeichnung *Empty.cat*) und die aktualisierten Binärdateien (*.sys, *.dll, *.exe) sowie mögliche Unterordner in den Ordner `\i386`. Falls Sie mehrere Aktualisierungspakete integrieren möchten und die eine oder andere Datei vielleicht in unterschiedlichen Aktualisierungspaketen enthalten ist, so prüfen Sie zunächst die Version der Datei und kopieren jeweils die aktuellste Dateiversion in den *i386*-Ordner.

Wechseln Sie nun in den Ordner `\i386` der Slipstream-Version, und laden Sie die Datei *Dosnet.inf* in einen Editor (NOTEPAD, WORDPAD oder EDIT). Prüfen Sie, ob die Datei eine Sektion mit der Bezeichnung `[OptionalSrcDirs]` enthält, und legen Sie die Sektion gegebenenfalls am Ende der Datei an, falls sie noch nicht vorhanden sein sollte. Unterhalb von `[OptionalSrcDirs]` fügen Sie die Zeichenfolge `Svcpack` ein.

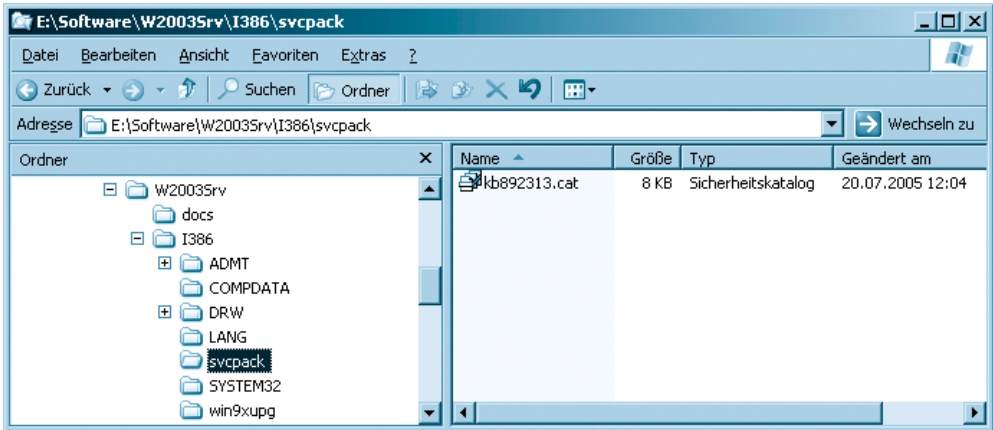


Abbildung 3: Der SvcPack-Ordner enthält ausschließlich Katalogdateien.

Fügen Sie – falls noch nicht vorhanden – eine weitere Sektion mit der Bezeichnung [Files] ein, und nehmen Sie unterhalb davon alle Dateien auf, die Sie zuvor in den *i386*-Ordner der Slipstream-Version kopiert haben. Beginnen Sie jede Zeile mit *d1*, und nehmen Sie dann den Namen der Datei auf.

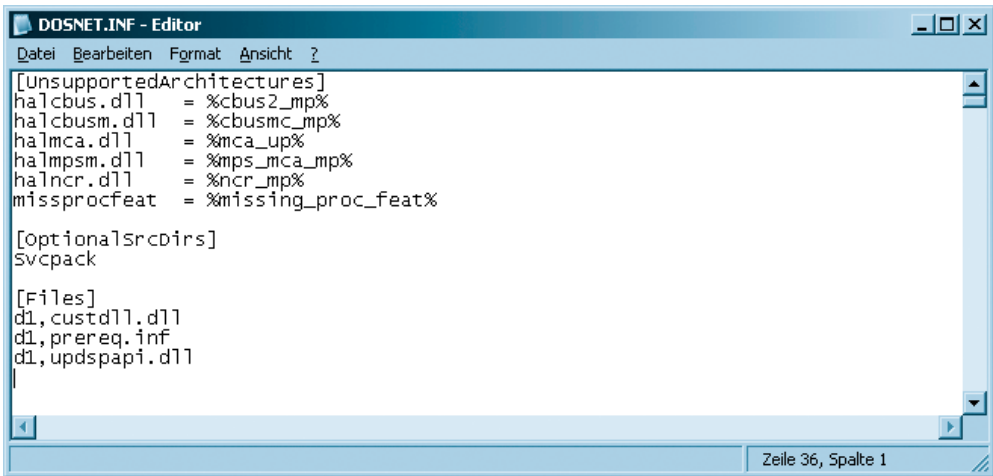


Abbildung 4: Der Inhalt der Datei »Dosnet.inf«

Beenden Sie den Editor wieder, und speichern Sie die geänderte Datei namens *Dosnet.inf* ab. Prüfen Sie daraufhin, ob im Ordner *i386* der Slipstream-Version eine Datei mit der Bezeichnung *Svcpack.in_* vorhanden ist. Wenn die Datei existiert, löschen Sie die Datei aus dem Ordner.

Danach erzeugen Sie mit Hilfe eines Editors eine neue Datei mit der Bezeichnung *Svcpack.inf*, die Sie im *i386*-Ordner der Slipstream-Version ablegen. Der Datei fügen Sie nun folgenden Inhalt hinzu, wobei Sie die Zeichenfolge *892313* durch die ent-

sprechende Paketnummer ersetzen. Sie können – wenn Sie mehrere Aktualisierungspakete hinzufügen möchten – natürlich auch zusätzliche Zeilen aufnehmen.

```
[Version]
Signature="$Windows NT$"
MajorVersion=5
MinorVersion=2
BuildNumber=3790

[SetupData]
CatalogSubDir="\i386\svcpack"

[ProductCatalogsToInstall]
KB8892313.cat

[SetupHotfixesToRun]
KB8892313.exe
```

Speichern Sie die Datei anschließend ab, das Slipstreaming ist damit abgeschlossen, und die aktualisierte Version von Windows Server 2003 steht nun für aktuelle Betriebssysteminstallationen zur Verfügung.

Slipstreaming aktueller Treiberdateien

Für den Fall, dass Sie eine Slipstream-Version von Windows Server 2003 erstellen und ein Aktualisierungspaket integrieren möchten, womit einige der in *Driver.cab* enthaltenen Dateien aktualisiert werden sollen, so müssen Sie auch in diesem Fall einen manuellen Eingriff vornehmen. Denn per Default werden immer die Dateien aus der Datei *Driver.cab* und damit die alten Versionen verwendet.

Um Windows Server 2003 dazu zu bewegen, die aktualisierten Dateien zu verwenden, extrahieren Sie zunächst den Inhalt des Aktualisierungspaketes in einen temporären Ordner, wofür Sie die *exe*-Datei des Aktualisierungspaketes mit dem Parameter */x* aufrufen. Anschließend kopieren Sie die *cat*-Datei (aber nicht *Empty.cat*) in den Ordner *\i386\Svcpack* und die aktualisierten Binärdateien in den *i386*-Ordner der Slipstream-Version von Windows Server 2003.

Danach editieren Sie die Datei *Dosnet.inf*, fügen (falls noch nicht vorhanden) die Sektion *[OptionalSrcDirs]* hinzu, nehmen unterhalb davon die Zeichenfolge *Svcpack* auf und fügen eine weitere Sektion mit der Bezeichnung *[ForceCopyDriverCabFiles]* ein. Unterhalb dieser Sektion nehmen Sie nun die aktualisierten Treiberdateien auf.

Speichern Sie die Änderung ab, und erzeugen Sie anschließend eine Datei mit der Bezeichnung *Svcpack.inf*, die folgende Sektionen und Einträge enthalten muss, wobei Sie *Q333333.exe* aber durch die Datei ersetzen, die integriert werden soll:

```
[Version]
Signature="$Windows NT$"

[SetupData]
CatalogSubDir="\i386\svcpack"
```

```
[ProductCatalogsToInstall]  
Q333333.cat
```

```
[SetupHotfixesToRun]  
Q333333.exe /q /n /z
```

Speichern Sie die Datei zu guter Letzt im *i386*-Ordner der Slipstream-Version ab. Mit der nächsten Installation wird dann das Fixpack *Q333333.exe* installiert, wobei aber nicht die Treiber aus der Datei *Driver.cab* verwendet werden, sondern die aktualisierten Dateien aus dem *i386*-Ordner.

Eine manuelle Installation durchführen

Wenn Sie lediglich einen einzelnen Server von Grund auf neu installieren müssen, so nehmen Sie in der Regel eine manuelle Installation vor und richten das Betriebssystem entweder mit Hilfe der Windows Server 2003-CD ein, oder aber Sie kopieren den Inhalt der CD (zumindest den Ordner *\i386*) zunächst auf ein Netzlaufwerk, geben dann den Ordner frei und starten die Installation anschließend von der Freigabe aus, indem Sie *Winnt.exe* aufrufen.

Um die Installation von CD zu starten, legen Sie die Installations-CD von Windows Server 2003 in das CD-Laufwerk und booten das Computersystem von CD. Möchten Sie die Installation stattdessen von einer Freigabe aus aufrufen, so stellen Sie zunächst eine Verbindung zu der entsprechenden Freigabe her und rufen dann im Ordner *\i386* die Datei *Winnt.exe* auf, woraufhin das Setup-Programm dann ebenfalls gestartet wird. Sobald das Setup-Programm aufgerufen wurde, wird automatisch die aktuelle Hardware-Konfiguration des Computersystems ermittelt und der textbasierte Teil der Installation beginnt.

Falls das Computersystem über einen SCSI- oder RAID-Controller verfügt, für den der Hersteller einen speziellen Treiber bereitstellt, so müssen Sie mit dem Einblenden der Meldung ZUR INSTALLATION EINES SCSI-/RAID-TREIBERS EINES DRITTANBIETERS F6 DRÜCKEN... am unteren Bildschirmrand die Taste **[F6]** betätigen und die Diskette mit dem entsprechenden Treiber einlegen, damit dieser Treiber in den Installationsvorgang eingebunden werden kann. Denn anderenfalls – wenn Sie diesen Treiber nicht bereitstellen und das Betriebssystem selbst keinen passenden Treiber bereithält – kommt es meist zu einem Abbruch der Installation, da die Festplatte nicht gemountet und damit auch nicht angesprochen werden kann. Kommen Sie dieser Aufforderung (sofern erforderlich) nach, und setzen Sie den Installationsvorgang fort, woraufhin die grundlegenden Systemdateien und auch die Basistreiber geladen werden.

Sie gelangen dann in das Dialogfenster WILLKOMMEN und können zwischen der Installation des Betriebssystems, dem Reparieren einer fehlerhaften Installation und auch dem Abbruch der Installation auswählen. Drücken Sie an dieser Stelle die **[Enter]**-Taste, um mit der Installation fortzufahren.

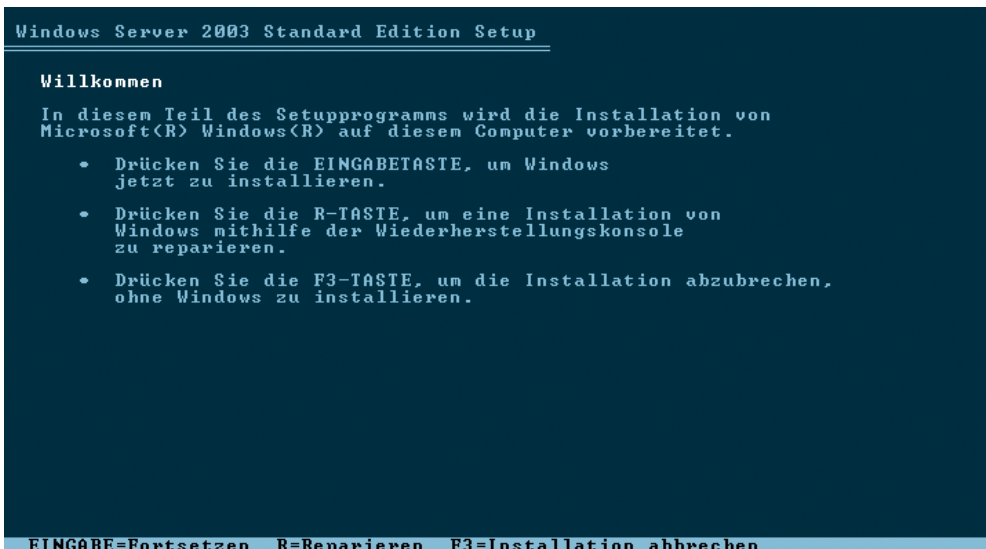


Abbildung 5: Das Dialogfenster »Willkommen« im textbasierten Teil des Setups

Daraufhin wird der Windows-Lizenzvertrag geöffnet, den Sie durch Betätigen der Taste **[F8]** manuell bestätigen müssen. Als Nächstes geht es weiter mit der Auswahl der Partition, auf der Windows Server 2003 installiert werden soll. Das Installationsprogramm zeigt dabei eine Aufstellung der verfügbaren Festplatte(n) unter Angabe der Partitionen bzw. des noch nicht partitionierten Speicherplatzes an. Sie können die Windows Server 2003-Installation auf einer bereits vorhandenen Partition vornehmen, einen noch unpartitionierten Speicherbereich neu partitionieren oder vorhandene Partitionen auch löschen und anschließend neu partitionieren. Wählen Sie die gewünschte Partition aus, oder markieren Sie den unpartitionierten Speicherbereich, und legen Sie mit der Taste **[E]** fest, dass Sie eine neue Partition anlegen möchten. Wenn Sie eine neue Partition anlegen wollen, so müssen Sie daraufhin die gewünschte Größe der Partition festlegen. Die Größe darf die Mindestanforderung von 1,5 GByte allerdings nicht unterschreiten, da sich das Betriebssystem sonst nicht installieren lässt.

Anschließend werden Sie aufgefordert, die gewünschte Formatierungsoption auszuwählen. Wählen Sie aus, ob Sie die Partition mit FAT oder besser: mit NTFS formatieren möchten und ob Sie die Formatierung mit dem etwas schnelleren Quick-Format oder mit der gewöhnlichen Formatierung durchführen möchten. Ersteres geht zwar etwas schneller, dafür wird die Partition aber nicht auf fehlerhafte Sektoren überprüft, was sich negativ auf die Systemstabilität auswirken kann. Denn werden die Sektoren nicht auf Fehler überprüft, so werden fehlerhafte Sektoren auch nicht als solche gekennzeichnet, und die Ablage von Daten auf diesen fehlerhaften Speicherbereichen ist die Folge. Wählen Sie der Stabilität wegen die gewöhnliche Formatierung aus, und markieren Sie das zu verwendende Dateisystem (FAT oder NTFS).

32 >> Eine manuelle Installation durchführen

Nach der Formatierung werden die Installationsdateien auf die Partition kopiert, und das Computersystem wird neu gestartet. Damit ist der textbasierte Teil des Setups abgeschlossen, und der grafische Teil der Installation beginnt. Dieser zweite Installations-Part startet dabei mit der Ermittlung und der Einrichtung der plug- und play-fähigen Hardware-Komponenten, wofür jedoch keinerlei Interaktion erforderlich ist, da das Betriebssystem diese Einrichtung automatisch vornimmt.

Nach der Geräteinstallation gelangen Sie in das Dialogfenster REGIONS- UND SPRACH-OPTIONEN, worüber Sie regionale Einstellungen für Zahlen-, Währungs-, Datums- und Uhrzeitformate festlegen und auch das zu verwendende Tastaturlayout bestimmen können. Nehmen Sie die gewünschten Einstellungen vor, und setzen Sie die Einrichtung von Windows Server 2003 dann fort.

Im nächsten Schritt werden Sie aufgefordert, einen Namen und auch die Firmenbezeichnung einzugeben. Legen Sie hier beispielsweise den Namen und auch die Firma des Lizenzinhabers fest, und geben Sie im nächsten Dialogfenster den Produktschlüssel für die Windows Server 2003-Installation ein, der sich entweder auf dem CD-Cover oder aber in der Lizenz-Order befindet. Der eingegebene Produktschlüssel wird daraufhin verifiziert und auf seine Gültigkeit hin überprüft.

Nach erfolgreicher Überprüfung des Produktschlüssels werden Sie nun aufgefordert, den gewünschten Lizenzierungsmodus für die Clientzugriffslizenzen (engl. = *Client Access Licence*) auszuwählen. Folgende Optionen stehen dabei zur Auswahl:

- ▶ **Pro Server:** Der Lizenzierungsmodus *Pro Server* erfordert, dass für jede Client-Server-Verbindung eine entsprechende Clientzugriffslizenz vorhanden ist. Wenn Sie beispielsweise über fünf Clientzugriffslizenzen verfügen und den Lizenzierungsmodus *Pro Server* auswählen, so akzeptiert der Server maximal fünf gleichzeitige Verbindungen zum Server, wobei für die jeweiligen Clients, die diese Verbindung nutzen, keine gesonderten Lizenzen mehr erforderlich sind. Diese Lizenzierungsart wird z.B. vorzugsweise für RAS- und Terminal-Server eingesetzt, da diese Clients nicht als Netzwerkclients registriert bzw. lizenziert sein müssen. Dieser Lizenzierungsmodus ermöglicht auch eine spätere (einmalige und kostenlose) Änderung zum Lizenzierungsmodus *Pro Gerät oder pro Benutzer*, weshalb Sie diesen Modus gerade dann auswählen sollten, wenn Sie noch nicht ganz sicher sind, welchen Modus Sie benötigen.
- ▶ **Pro Gerät oder pro Benutzer:** Der Lizenzierungsmodus *Pro Gerät oder pro Benutzer* erfordert, dass für jeden Client oder für jeden Benutzer eine entsprechende Clientzugriffslizenz vorhanden ist, womit von den lizenzierten Clients aus mehrere bzw. unterschiedliche Server-Verbindungen hergestellt werden können. Dieser Lizenzierungsmodus kommt vorzugsweise dann zum Einsatz, wenn von einzelnen Clients aus unterschiedliche Server-Verbindungen erstellt werden, wie es beispielsweise in größeren Netzwerken der Fall ist.

Wählen Sie also den entsprechenden Lizenzierungsmodus (im Zweifel *Pro Server*) aus, und klicken Sie anschließend auf WEITER, um mit der Einrichtung von Windows Server 2003 fortzufahren.

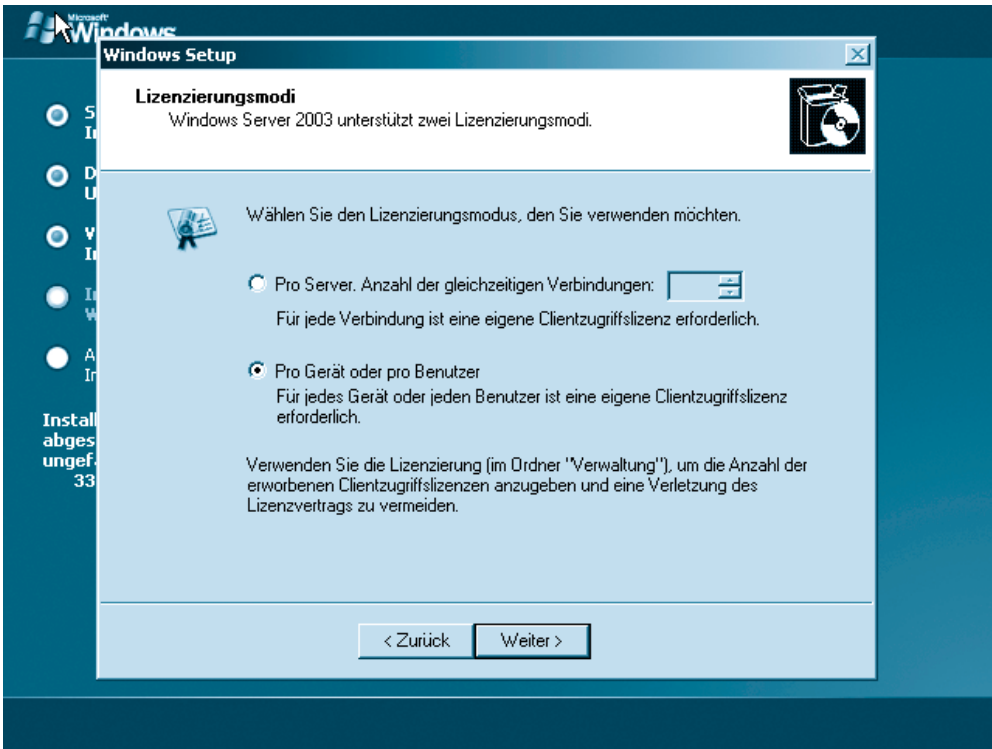


Abbildung 6: Den gewünschten Lizenzierungsmodus auswählen

Sie gelangen nun in das Dialogfenster **COMPUTERNAME UND ADMINISTRATORKENNWORT**. Geben Sie im Feld **COMPUTERNAME** eine aussagekräftige und eindeutige Bezeichnung für den Server ein, und berücksichtigen Sie die maximale Länge von 15 Zeichen. Sie können die Buchstaben **A** bis **Z** wie auch die Zahlen von **0** bis **9** und den Bindestrich (**-**) verwenden. Allerdings darf der Computername nicht aus reinen Zahlen bestehen, sondern muss auch Buchstaben enthalten.

Im Feld **ADMINISTRATORKENNWORT** legen Sie ein sicheres Kennwort für den Benutzer *Administrator* fest und geben das gleiche Kennwort im Feld **KENNWORT BESTÄTIGEN** noch einmal ein, um mögliche Fehler bei der Eingabe zu vermeiden. Sichere Kennwörter umfassen mindestens sechs Zeichen und bestehen aus Buchstaben, Zahlen und Sonderzeichen, die weder einen leicht zu erratenden Namen darstellen und für die es auch keinerlei Eselsbrücken gibt, womit sich das Kennwort leichter merken lässt.

Danach wird das Dialogfenster **DATUM- UND UHRZEITEINSTELLUNGEN** aufgerufen. Überprüfen (und korrigieren Sie gegebenenfalls) die Einstellungen für das Datum und die Uhrzeit, und stellen Sie sicher, dass die richtige Zeitzone verwendet wird. Über dieses

34 >> Eine manuelle Installation durchführen

Dialogfenster können Sie auch die automatische Umstellung der Sommer- und Winterzeit aktivieren bzw. deaktivieren (die Einstellung ist per Default aktiviert), womit die Systemzeit beim Wechsel der Sommer-/Winterzeit automatisch angepasst wird, sofern die Option UHR AUTOMATISCH AUF SOMMER-/WINTERZEIT UMSTELLEN aktiviert ist.

Klicken Sie anschließend auf WEITER, um die NETZWERKEINSTELLUNGEN aufzurufen und zwischen STANDARDEINSTELLUNGEN und BENUTZERDEFINIERTEN EINSTELLUNGEN zu wählen. Ersteres bedeutet, dass der *Client für Microsoft-Netzwerke*, die *Datei- und Druckerfreigabe für Microsoft-Netzwerke* und auch das *TCP/IP-Protokoll* mit automatischer IP-Adressvergabe (DHCP) eingerichtet wird, wobei Sie das TCP/IP-Protokoll aber erst nach Abschluss der Installation detailliert konfigurieren können. Wenn Sie die Netzwerkeinstellungen und insbesondere das TCP/IP-Protokoll sofort konfigurieren möchten, so wählen Sie stattdessen BENUTZERDEFINIERT EINSTELLUNGEN aus, womit ein weiteres Dialogfenster aufgerufen wird, über das Sie die Netzwerkkomponenten hinzufügen und entfernen sowie das TCP/IP-Protokoll konfigurieren (z.B. Vergabe einer statischen IP-Adresse, Festlegen des DNS-Servers etc.) können.

Im nächsten Schritt gilt es, zwischen einer Arbeitsgruppen- und Domänenmitgliedschaft auszuwählen. Wählen Sie an dieser Stelle die Option NEIN, DIESER COMPUTER IST ENTWEDER NICHT IM NETZWERK ODER OHNE DOMÄNE IM NETZWERK. DIESEN COMPUTER DER FOLGENDEN ARBEITSGRUPPE HINZUFÜGEN: aus, wenn Sie einen Standalone-Server installieren möchten und keine Domäne verwenden. Legen Sie anschließend den Namen der Arbeitsgruppe fest. Wählen Sie stattdessen die Option JA, DIESEN COMPUTER ZUR FOLGENDEN DOMÄNE HINZUFÜGEN: aus, wenn Sie bereits eine Domäne eingerichtet haben und diesen Server als weiteren Domänen-Controller oder aber als Member-Server der Domäne nutzen möchten, und geben Sie daraufhin den Namen der Domäne ein, die um diesen Server erweitert werden soll. Sie müssen daraufhin den Benutzernamen und auch das Kennwort eines Benutzers eingeben, der die Berechtigung zum Hinzufügen eines Computersystems zur Domäne hat. Die Auswahl der zuvor genannten Option erfordert, dass sich der Server bereits im Netzwerk der Domäne befindet und dass die erforderliche TCP/IP-Verbindung im vorherigen Schritt schon konfiguriert wurde. Ist der Server physikalisch noch nicht im Domänennetzwerk, so konfigurieren Sie den Server zunächst als Mitglied der Arbeitsgruppe und konfigurieren die Domänenmitgliedschaft nach Abschluss der Installation.

Damit haben Sie nun alle grundlegenden Konfigurationsparameter ausgewählt, und das Kopieren der Betriebssystemdateien sowie das eigentliche Installieren von Windows Server 2003 beginnt. Dieser Vorgang kann – je nach Leistung des Computersystems – einige Minuten dauern. Sobald dieser Kopier- und Einrichtungsvorgang beendet ist, wird das Computersystem automatisch neu gestartet, und die Installation des Betriebssystems ist damit beendet. Sie können sich nun an Windows Server 2003 anmelden und die finale Konfiguration (z.B. Einrichtung der Benutzer- und Gruppenkonten, Konfigurieren spezieller Server-Rollen, Update des Betriebssystems etc.) vornehmen.

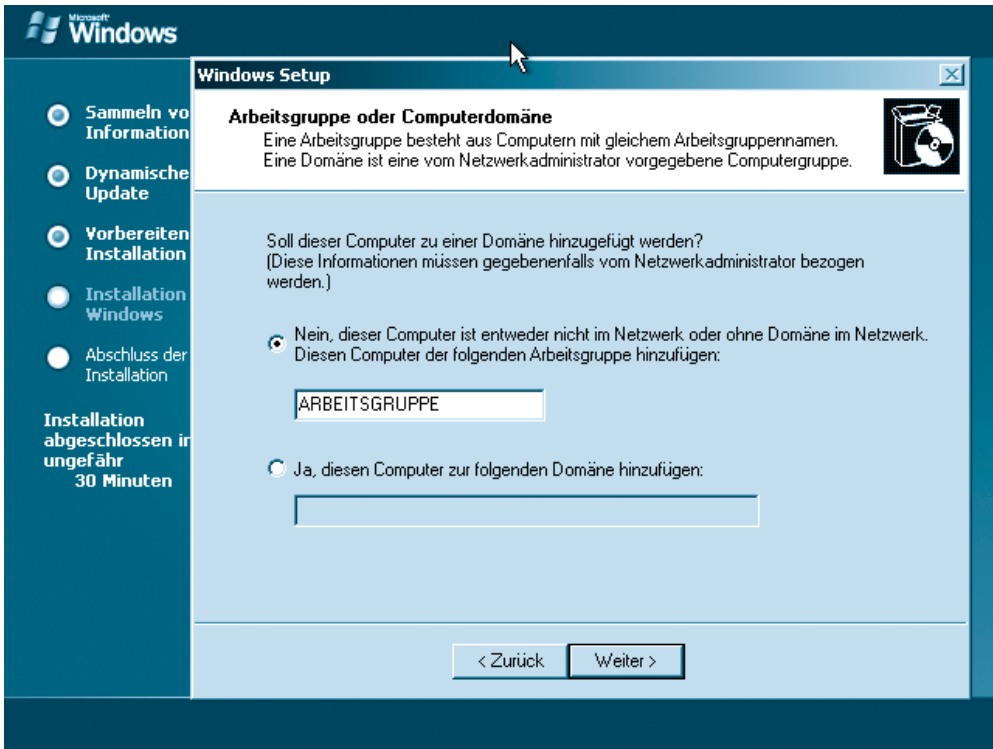


Abbildung 7: Die Arbeitsgruppen- oder Domänenmitgliedschaft auswählen

Automatische Betriebssysteminstallationen erstellen

Für den Fall, dass Sie das Öffnen von Windows Server 2003-Installationen vornehmen müssen, so ist die manuelle Betriebssysteminstallation natürlich recht zeitaufwändig. Sie können sich das Deployment von Windows Server 2003 aber ein Stück weit vereinfachen, indem Sie sämtliche Einstellungen, die während dem Setup konfiguriert werden müssen, in einer Antwortdatei hinterlegen. Damit können Sie die Installation größtenteils und bei Bedarf auch vollständig automatisieren.

Eine solche unbeaufsichtigte Installation (engl. = *Unattended Setup*) macht jegliche Interaktion während des Setup-Prozesses überflüssig, da alle erforderlichen Angaben und Einstellungen, die normalerweise während des textbasierten und auch grafischen Teils der Installation konfiguriert werden müssen, aus der Antwortdatei ausgelesen und automatisch übernommen werden. Diese unbeaufsichtigte Installation kann sowohl für das Setup von CD wie auch für eine Netzwerkinstallation eingerichtet werden und erfordert lediglich das Anlegen einer entsprechenden Antwortdatei und das Aufrufen des Setups unter Angabe der zu verwendenden Antwortdatei.

Die nachfolgenden Abschnitte zeigen, wie Sie mit dem INSTALLATIONS-MANAGER eine funktionsfähige Antwortdatei erstellen und die Installation von Windows Server 2003 von CD und aus einem Distributionsordner heraus vollständig automatisieren können.

Eine Antwortdatei erstellen

Die einfachste und auch sicherste Methode, eine funktionstüchtige Antwortdatei zu erstellen, bietet der INSTALLATIONS-MANAGER, der Bestandteil der Deploy-Tools von Windows Server 2003 ist. Dieses Tool stellt einen Assistenten dar, der das einfache Auswählen und Setzen der jeweiligen Konfigurationseinstellungen ermöglicht und damit eine schnelle und fehlerfreie Erstellung der Antwortdatei erlaubt.

Um nun eine Antwortdatei für eine unbeaufsichtigte Installation des Betriebssystems zu erstellen, legen Sie zunächst die Installations-CD von Windows Server 2003 in das Laufwerk und extrahieren die Deploy-Tools (*Deploy.cab*), die im Ordner *\Support\Tools* abgelegt sind, in einen beliebigen Ordner auf der Festplatte. Anschließend starten Sie das Tool *Setupmgr.exe*, um den INSTALLATIONS-MANAGER aufzurufen.

Schließen Sie den Willkommensbildschirm mit einem Klick auf WEITER, und wählen Sie im nächsten Schritt aus, dass Sie eine neue Antwortdatei anlegen möchten. Klicken Sie auf WEITER, und wählen Sie im Dialogfenster INSTALLATIONSTYP die Option UNBEAUF SICHTIGTE INSTALLATION aus, um eine Antwortdatei für das *Unattended Setup* zu erzeugen.

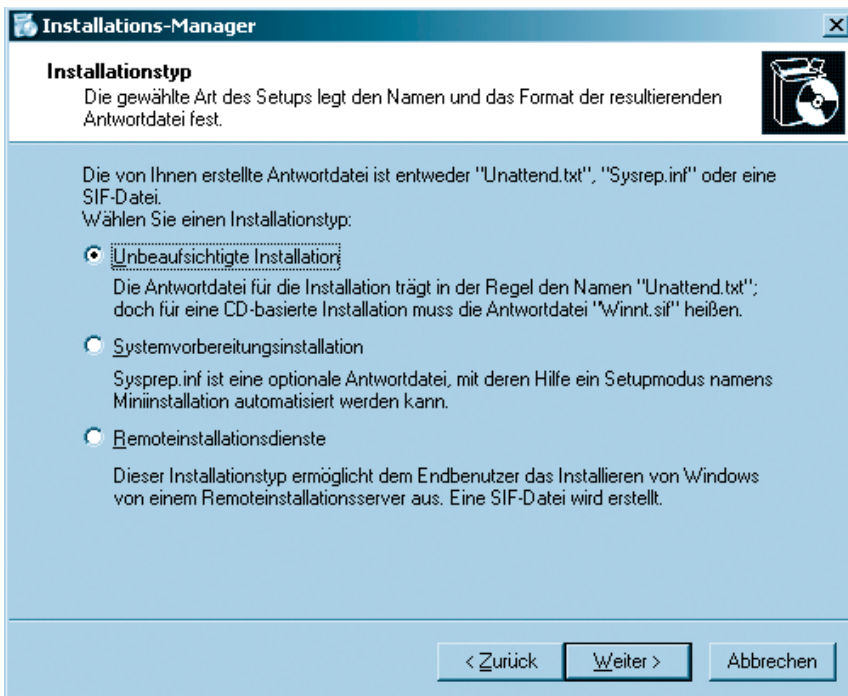


Abbildung 8: Den Installationstyp »unbeaufsichtigte Installation« auswählen

Wählen Sie anschließend das Betriebssystem (z.B. WINDOWS SERVER 2003 STANDARD EDITION) aus, das Sie künftig mit Hilfe der Antwortdatei unbeaufsichtigt installieren möchten, und klicken Sie auf WEITER. Sie gelangen nun in das Dialogfenster BENUTZER-

EINGRIFF und können auswählen, inwieweit die Installation automatisiert werden soll und ob eine Interaktion während des Setups erwünscht ist oder nicht. Folgende Optionen stehen zur Auswahl:

- ▶ **Benutzergesteuert:** Die Option **BENUTZERGESTEUERT** erlaubt das Erstellen einer vollständigen Antwortdatei, womit alle Konfigurationseinstellungen, die während des textbasierten und grafischen Teils des Setups gesetzt werden müssen, hinterlegt werden können. Die hinterlegten Einstellungen werden jedoch nicht automatisch konfiguriert, sondern nur als Standardwerte angezeigt und lassen sich daher während der Installation jederzeit ändern.
- ▶ **Vollautomatisiert:** Die Option **VOLLAUTOMATISIERT** erlaubt das Erstellen einer vollständigen Antwortdatei, womit alle Konfigurationseinstellungen hinterlegt und auch automatisch angewendet werden können. Während der Installation ist keine Interaktion mit dem Administrator erforderlich.
- ▶ **Ausgeblendete Seiten:** Die Option **AUSGEBLENDETE SEITEN** ermöglicht lediglich die Automatisierung bestimmter Setup-Teile und nicht der kompletten Installation. Hierbei werden die Dialogseiten, für die Konfigurationseinstellungen hinterlegt werden, automatisch konfiguriert und daher während des Setups auch nicht mehr angezeigt. Alle noch nicht definierten Konfigurationseinstellungen (z.B. TCP/IP-Parameter) müssen dann während des Installationsvorgangs konfiguriert werden.
- ▶ **Schreibgeschützt:** Die Option **SCHREIBGESCHÜTZT** erfordert, dass alle benötigten Konfigurationseinstellungen hinterlegt werden, wobei die Einstellungen während des Installationsvorgangs angezeigt, aber nicht geändert werden können.
- ▶ **GUI-gesteuert:** Die Option **GUI-GESTEUERT** erlaubt die vollständige Automatisierung des textbasierten Setup-Teils. Alle Konfigurationseinstellungen, die während des grafischen Teils des Setups benötigt werden, werden nicht hinterlegt und müssen daher während der Installation von Hand konfiguriert werden.

Wenn Sie die Installation von Windows Server 2003 also vollständig automatisieren möchten, so wählen Sie die Option **VOLLAUTOMATISIERT** aus und klicken anschließend auf **WEITER**. Im nächsten Schritt müssen Sie nun angeben, ob Sie die Installation via Netzwerk oder von CD durchführen möchten. Ersteres verlangt die Angabe darüber, ob Sie bereits einen entsprechenden Distributionsordner angelegt und freigegeben haben oder ob Sie einen neuen Distributionsordner erstellen möchten. Wählen Sie die entsprechende Option aus, und klicken Sie auf **WEITER**.

Die vollständige Automatisierung der Windows Server 2003-Installation verlangt, dass Sie dem Lizenzvertrag zustimmen. Aktivieren Sie hierfür die Option **DEM LIZENZVERTRAG ZUSTIMMEN**, und führen Sie einen Klick auf **WEITER** aus, um zu den eigentlichen Konfigurationseinstellungen zu gelangen. Auf der linken Seite sehen Sie nun die Kategorien und die jeweiligen Einstellungsmöglichkeiten, die Sie jetzt definieren und damit automatisch in der Antwortdatei hinterlegen können. Nehmen Sie in der rechten Fensterhälfte also die entsprechenden Einstellungen vor, und klicken Sie jeweils auf **WEITER**, um zur nächsten Einstellung zu gelangen.

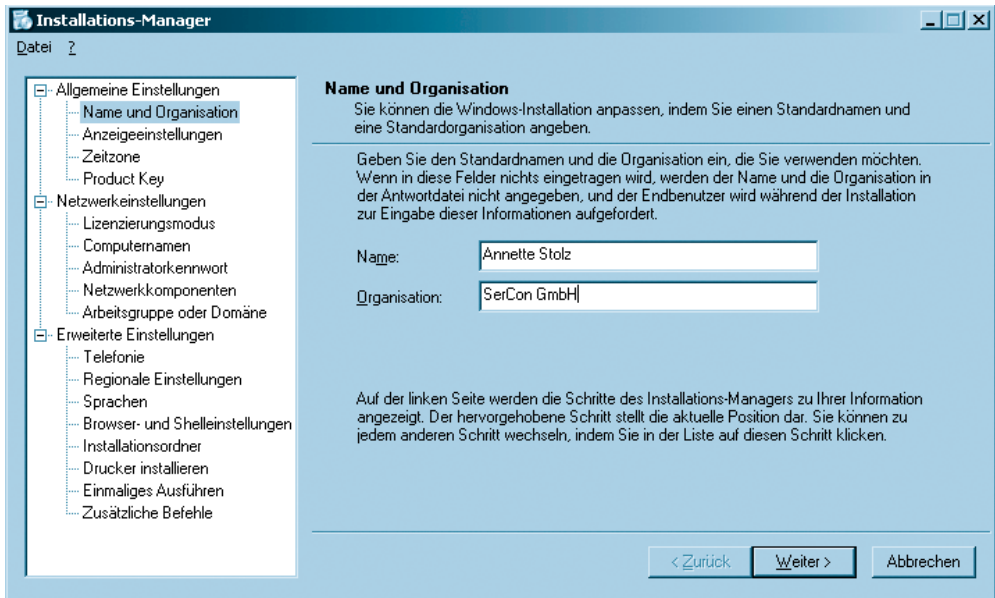


Abbildung 9: Die Angaben zu »Name« und »Organisation« hinterlegen

Wenn Sie alle Konfigurationseinstellungen hinterlegt haben, gelangen Sie automatisch in ein Dialogfenster, worüber Sie den Pfad und auch den Namen der Antwortdatei hinterlegen können. Wählen Sie den gewünschten Pfad aus, und übernehmen Sie den Dateinamen *Unattend.txt*, um die Antwortdatei dann mit einem Klick auf FERTIG STELLEN anzulegen. Wenn Sie die soeben erstellte Antwortdatei mit einem Editor öffnen, dann können Sie die Einstellungen einsehen und gegebenenfalls auch noch einmal anpassen, der Inhalt der Datei sieht (zumindest in etwa) folgendermaßen aus:

```
[Data]
AutoPartition=1
MsDosInitiated="0"
UnattendedInstall="Yes"

[Unattended]
UnattendMode=FullUnattended
OemSkipEula=Yes
OemPreinstall=No
TargetPath=\WINDOWS

[GuiUnattended]
AdminPassword=12345678
EncryptedAdminPassword=Yes
```

Listing 1: Ein Beispiel für den Inhalt der »Unattended.txt«

```
OEMSkipRegional=1
TimeZone=110
OemSkipWelcome=1

[UserData]
ProductKey=xxxxx-xxxxx-xxxxx-xxxxx-xxxxx
FullName="Annette Stolz"
OrgName="SerCon GmbH"
ComputerName=FRA-MD-FPS-001

[Display]
BitsPerPel=32
Xresolution=1024
YResolution=768
Vrefresh=75

[LicenseFilePrintData]
AutoMode=PerSeat

[TapiLocation]
CountryCode=49
AreaCode=069

[RegionalSettings]
LanguageGroup=1
Language=00000407

[GuiRunOnce]
Command0=FinalKonfiguration.bat

[Identification]
JoinWorkgroup=ARBEITSGRUPPE

[Networking]
InstallDefaultComponents=No

[NetAdapters]
Adapter1=params.Adapter1

[params.Adapter1]
INFID=*

[NetClients]
MS_MSClient=params.MS_MSClient

[NetServices]
```

Listing 1: Ein Beispiel für den Inhalt der »Unattended.txt« (Forts.)

40 >> Automatische Betriebssysteminstallationen erstellen

```
MS_SERVER=params.MS_SERVER

[NetProtocols]
MS_TCPIP=params.MS_TCPIP

[params.MS_TCPIP]
DNS=No
UseDomainNameDevolution=No
EnableLMHosts=Yes
AdapterSections=params.MS_TCPIP.Adapter1

[params.MS_TCPIP.Adapter1]
SpecificTo=Adapter1
DHCP=No
IPAddress=192.168.0.120
SubnetMask=255.255.255.0
WINS=No
NetBIOSOptions=0
```

Listing 1: Ein Beispiel für den Inhalt der »Unattended.txt« (Forts.)

Die automatische Installation starten

Wenn Sie sämtliche Konfigurationseinstellungen hinterlegt und die Antwortdatei *Unattended.txt* erzeugt haben, so können Sie nun die erste automatische Installation starten. Wie Sie dabei vorgehen, hängt nun davon ab, ob Sie die Installation von CD oder via Netzwerk aus einem Distributionsordner heraus starten wollen.

Automatische Installation von CD

Um eine automatische Installation mit Hilfe der Installations-CD von Windows Server 2003 zu starten, kopieren Sie die soeben erstellte Antwortdatei mit der Bezeichnung *Unattended.txt* auf eine Diskette und benennen sie in *Winnt.sif* um. Anschließend legen Sie die Diskette, auf der sich die Antwortdatei *Winnt.sif* befindet, in das Diskettenlaufwerk und legen die Windows Server 2003-CD in das CD-Laufwerk.

Starten Sie das Computersystem vom CD-Laufwerk, woraufhin automatisch das Installationsprogramm gestartet wird. Das Installationsprogramm prüft dabei zu Beginn, ob sich im Diskettenlaufwerk ein Medium befindet, und sucht automatisch nach der Datei *Winnt.sif*. Wird diese Datei gefunden, so wird der Inhalt der Datei eingelesen und zur automatischen Konfiguration der Einstellungen verwendet.

Automatische Installation aus einem Distributionsordner heraus

Gesetzt den Fall, Sie haben die Installationsdateien von Windows Server 2003 (zumindest den Inhalt des *i386*-Ordners) in einen Ordner auf der lokalen Festplatte kopiert und diesen Ordner auch entsprechend freigeben, so kopieren Sie die soeben

erstellte Antwortdatei mit der Bezeichnung *Unattended.txt* in den *i386*-Ordner des Distributionsordners.

Starten Sie anschließend das zu installierende Computersystem, und stellen Sie eine Netzwerkverbindung zu dem entsprechenden Distributionsordner her. Wechseln Sie dann in den Ordner *i386*, und geben Sie den Befehl `Winnt /s \\Server\Share /u:Unattend.txt` ein, wobei Sie `\\Server\Share` durch den Pfad ersetzen, der den freigegebenen Distributionsordner darstellt und damit die Installationsdateien enthält. Mit diesem Befehl wird nun das Installationsprogramm von Windows Server 2003 unter Verwendung der Antwortdatei *Unattend.txt* gestartet, womit alle Konfigurationseinstellungen, die in der Antwortdatei hinterlegt sind, eingelesen und automatisch angewendet werden.

Betriebssysteminstallationen klonen

Eine weitere Möglichkeit des Betriebssystem-Deployments stellt das Cloning dar. Hierbei wird zunächst ein einzelner Server (das Master-System) installiert und in der Regel auch vollständig konfiguriert. Anschließend wird die Partition, auf der sich die Betriebssysteminstallation befindet, in eine Image-Datei kopiert und auf den restlichen Server-Systemen, die es zu installieren und zu konfigurieren gilt, wiederhergestellt. Mit dieser Partitions- oder auch Datenträger-Duplizierung lässt sich eine vollständige Installation mit nur wenigen Schritten und auch innerhalb kurzer Zeit auf mehrere Computersysteme verteilen, wobei eine aufwändige Nachkonfiguration der Zielsysteme in den meisten Fällen entfällt, da die finale Konfiguration meist schon im Abbild enthalten ist.

Das Problem am Cloning ist aber, dass aufgrund der Partitions- oder auch Datenträger-Duplizierung sämtliche Konfigurationseinstellungen mit eingeschlossen sind und damit natürlich auch auf die Zielsysteme übertragen werden, weshalb das Cloning in der Regel nur dann eingesetzt wird, wenn sämtliche Zielsysteme über exakt die gleiche Hardware-Ausstattung verfügen. Des Weiteren werden natürlich auch alle Sicherheitseinstellungen 1:1 übertragen, womit die Zielsysteme, auf denen das Abbild später restauriert wird, allesamt über die gleiche SID (siehe *Welche SID hat der Benutzer?* in der Kategorie *Benutzer und Gruppen* im Teil II dieses Buches) und auch über den gleichen Hostnamen verfügen. Dies führt natürlich zu Problemen im Netzwerk, da weder die Computersysteme noch die Computer- und Benutzerkonten eindeutig sind.

Diese Cloning-Probleme lassen sich aber mit dem Deploy-Tool namens *Sysprep.exe*, das im Ordner *Support\Tools* der Installations-CD von Windows Server 2003 enthalten ist, sehr schnell aus der Welt schaffen. Denn mit *Sysprep* können Sie eine fertige Windows Server 2003-Installation (inklusive der bereits eingerichteten Server-Rollen und Applikationen) für das Deployment vorbereiten und mit dem ersten Hochfahren des Computersystems den Start eines Mini-Setups erzwingen, womit elementare Systemeinstellungen wie SID, Hostname, Hardwareerkennung, Netzwerkeinstellungen, Lizenzierung etc. geändert werden können, ohne dass alle anderen Einstellungen dabei verloren gehen. Und dieses Mini-Setup lässt sich mit Hilfe einer Antwortdatei

sogar automatisieren, weshalb das Cloning nach wie vor eine sehr einfache, aber dennoch effiziente Möglichkeit des Server-Deployments darstellt.

Das Klonen eines Serversystems setzt voraus, dass Sie zunächst ein entsprechendes Master-System installieren und auch entsprechend konfigurieren. Anschließend versiegeln Sie die Installation mit Hilfe von *Sysprep* und erzwingen so den Aufruf des Mini-Setups beim nächsten Starten, womit diverse Systemeinstellungen noch während des Startvorgangs modifiziert werden können. Anschließend erstellen Sie das Image, übertragen die Image-Datei auf die entsprechenden Zielsysteme und restaurieren die Image-Datei dann auf den entsprechenden Computersystemen. Mit dem ersten Computerstart wird dann das Mini-Setup ausgeführt, worüber Sie bestimmte Systemeinstellungen verändern können, ohne dass das Betriebssystem vollständig gestartet wurde.

Die nachfolgenden Abschnitte zeigen nun, wie Sie das bereits vollständig installierte und auch konfigurierte Master-System mit *Sysprep* versiegeln und für das Deployment vorbereiten können. Des Weiteren wird auch erwähnt, wie Sie im Fall der Fälle die SID auf den Zielsystemen mit *NewSID* (www.sysinternals.com) ändern können, falls Sie die Versiegelung mit *Sysprep* nicht einsetzen möchten oder die Versiegelung versehentlich vergessen haben.

Die Betriebssysteminstallation mit Sysprep versiegeln

Wenn Sie das Master-System so weit eingerichtet und auch entsprechend konfiguriert haben, dann können Sie das System nun mit Hilfe von *Sysprep* versiegeln und für das Deployment vorbereiten. Hierzu legen Sie die Installations-CD von Windows Server 2003 in das Laufwerk und extrahieren die Deploy-Tools, die sich im Ordner *Support\Tools* befinden, in den Ordner *Sysprep* der Systempartition (z.B. C:\Sysprep). Die Angabe eines alternativen Pfades ist hier allerdings nicht möglich. Denn die *Sysprep*-Dateien werden nur dann angezogen, wenn sie im Ordner *Sysprep* der Systempartition abgelegt wurden.

Dann rufen Sie per Doppelklick das Tool *Sysprep.exe* auf, woraufhin erst eine kurze Information zum Systemvorbereitungsprogramm eingeblendet wird. Klicken Sie auf OK, um das Dialogfenster zu schließen und in die eigentliche GUI von *Sysprep* zu gelangen. Aktivieren Sie nun die Option KULANZFRIST FÜR AKTIVIERUNG NICHT ZURÜCKSETZEN, wenn die Windows Server 2003-Installation bereits aktiviert wurde, anderenfalls lassen Sie die Einstellung deaktiviert, damit Windows Server 2003 nach dem Deployment aktiviert werden muss. Lassen Sie die Option SICHERHEITSKENNUNGEN NICHT NEU GENERIEREN deaktiviert, damit während des ersten Windows Server 2003-Starts neue SIDs erzeugt und angewendet werden. Aktivieren Sie die Option NICHT PLUG & PLAY-FÄHIGE HARDWARE ERKENNEN, damit während des ersten Windows Server 2003-Starts eine automatische Hardware-Erkennung und -Einrichtung durchgeführt wird.

Wenn Sie das Computersystem nun herunterfahren und anschließend duplizieren möchten, ohne eine Antwortdatei für das Mini-Setup zu erstellen, dann klicken Sie nun auf ERNEUT VERSIEGELN. Das Computersystem wird daraufhin heruntergefahren,

und die Systempartition kann per Imaging kopiert und auf andere Computersysteme übertragen werden. Wenn die Systempartition übertragen wurde und das Zielsystem zum ersten Mal gestartet wird, dann wird automatisch das Mini-Setup aufgerufen, und folgende Einstellungen müssen konfiguriert werden:

- ▶ Annehmen des Lizenzvertrages
- ▶ Konfigurieren der Regions- und Sprachoptionen
- ▶ Konfigurieren der Benutzerinformationen (Name und Organisation)
- ▶ Eingeben des Produktschlüssels
- ▶ Festlegen des Lizenzierungsmodus (pro Server/pro Gerät oder pro Benutzer)
- ▶ Festlegen des Computernamens und Setzen des Administratoren-Kennwortes
- ▶ Konfigurieren der Datum- und Uhrzeiteinstellungen
- ▶ Konfigurieren der Netzwerkeinstellungen (Standardeinstellungen oder benutzerdefinierte Einstellungen)
- ▶ Festlegen der Arbeitsgruppen- oder Domänenmitgliedschaft

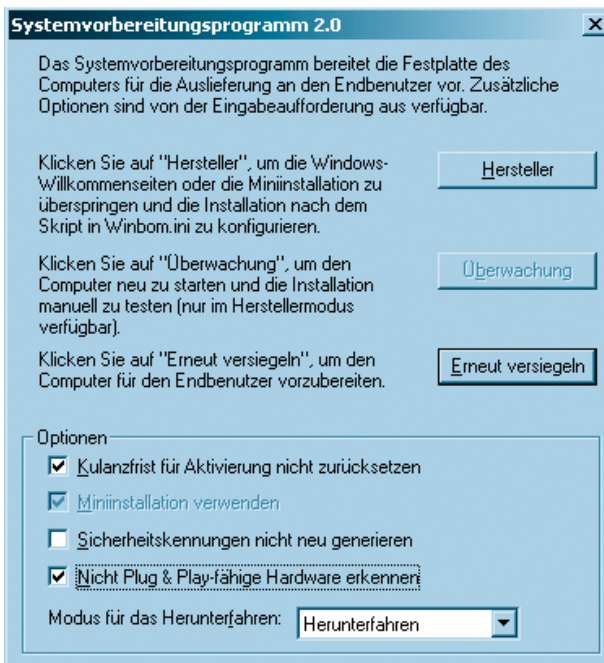


Abbildung 10: Die Sysprep-Optionen konfigurieren

Die Konfigurationseinstellungen, die während der Ausführung des Mini-Setups festgelegt werden müssen, lassen sich mit Hilfe einer Antwortdatei (*Sysprep.inf*) natürlich auch schon im Vorfeld hinterlegen, womit Sie das Mini-Setup vollständig automatisieren können.

44 >> Betriebssysteminstallationen klonen

Nachfolgend finden Sie eine Beispieldatei, womit Sie die grundlegenden Konfigurationseinstellungen für das Mini-Setup hinterlegen können. Neben den zuvor genannten Konfigurationseinstellungen, die während des Mini-Setups durchgeführt werden müssen, können Sie natürlich darüber hinaus noch etliche Einstellungen mehr vornehmen. Eine Auflistung aller erlaubten Sektionen und Konfigurationseinstellungen können Sie der Datei *Ref.chm* entnehmen, die ebenfalls Bestandteil der Deploy-Tools (*Deploy.Cab*) ist.

```
[Unattended]
    OemSkipEula=Yes
    InstallFilesPath=C:\sysprep\i386

[GuiUnattended]
    AdminPassword=12345678
    EncryptedAdminPassword=Yes
    OEMSkipRegional=1
    TimeZone=110
    OemSkipWelcome=1

[UserData]
    ProductKey=XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
    FullName="Annette Stolz"
    OrgName="SerCon GmbH"
    ComputerName=FRA-MD-FPS-001

[LicenseFilePrintData]
    AutoMode=PerSeat

[TapiLocation]
    CountryCode=49
    AreaCode=069

[SetupMgr]
    DistFolder=C:\sysprep\i386
    DistShare=windist

[Identification]
    JoinWorkgroup=ARBEITSGRUPPE

[Networking]
    InstallDefaultComponents=Yes
```

Listing 2: Ein Beispiel für den Inhalt der »Sysprep.inf«

Wenn Sie das Mini-Setup automatisieren möchten, so erzeugen Sie eine *Sysprep.inf*-Datei mit den entsprechenden Konfigurationseinstellungen und legen die Datei dann im Ordner *C:\Sysprep* ab. Bei Bedarf können Sie auch noch eine zusätzliche *Unatten-*

ded.txt erzeugen, um weitere Optionen für die Betriebssystemkonfiguration (z.B. Hostnamen, IP-Einstellungen ...) zu hinterlegen, die nicht in der Datei *Sysprep.inf* aufgenommen werden können. Anschließend rufen Sie das Tool *Sysprep* auf und klicken auf ERNEUT VERSIEGELN, woraufhin das Computersystem dann heruntergefahren wird. Anschließend können Sie ein Image des Datenträgers erstellen und auf andere Computersysteme übertragen. Wenn die duplizierte Windows Server 2003-Installation dann das erste Mal gestartet wird, so wird automatisch das Mini-Setup mit den in der *Sysprep.inf*-Datei hinterlegten Einstellungen konfiguriert. Nach Abschluss aller Konfigurationsarbeiten wird der Ordner *C:\Sysprep* automatisch gelöscht, und Windows Server 2003 wird gestartet.

Das Cloning

Wenn Sie die Windows Server 2003-Installation so weit vorgenommen und das Computersystem vorzugsweise mit *Sysprep* versiegelt haben, so können Sie nun ein Abbild der Betriebssysteminstallation und damit ein so genanntes Image erstellen. Hierfür benötigen Sie eine zusätzliche Software wie beispielsweise *Norton Ghost* (www.symantec.com), *Paragon Drive Backup* (www.drive-backup.com), *Acronis True Image* (www.acronis.de) oder *R-TT Drive Image* (www.drive-image.com).

Die meisten der zuvor genannten Software-Produkte bieten ein »On the fly-Imaging« an. Das bedeutet, dass Sie die Software unter grafischer Oberfläche installieren und während des laufenden Windows Server 2003-Betriebs ein Abbild der Systempartition oder wahlweise auch der gesamten Festplatte erzeugen können. Durch dieses »On the fly-Imaging« steht der Server auch während des Imaging-Vorgangs zur Verfügung und muss für den Imaging-Vorgang nicht heruntergefahren werden.

Der Inhalt der ausgewählten Partition bzw. der Festplatte wird während des Imaging-Vorgangs in eine spezielle Image-Datei kopiert und kann dann auf eine andere Festplatte übertragen werden. Die Image-Datei wird anschließend auf dem Zielsystem wieder restauriert, womit sich die Partition bzw. der komplette Festplatteninhalt 1:1 auf ein anderes Computersystem übertragen lässt. Auf welche Art und Weise das Übertragen und Restaurieren der Image-Datei erfolgt, ist von Produkt zu Produkt unterschiedlich. In der Regel enthalten die Software-Produkte jedoch entsprechende Utilities, womit Sie das Zielsystem booten, auf die Image-Datei zugreifen (z.B. über eine gesonderte Partition, von CD oder via Netzwerk) und den Inhalt der Image-Datei dann wieder zurückspielen können.

Die SID ändern

Für den Fall, dass Sie ein Abbild (ein Image) einer vollständigen Windows Server 2003-Installation erstellt und dieses Abbild bereits auf einem anderen Computersystem restauriert haben, ohne dass Sie das Systemvorbereitungsprogramm zur Änderung der SID eingesetzt haben, so können Sie die SID des geklonten Computersystems natürlich auch im Nachhinein noch ändern.

Auf der Internetseite *www.sysinternals.com* finden Sie hierfür ein Freeware-Tool mit der Bezeichnung *NewSID*, womit Sie eine per Zufall generierte SID erzeugen oder auch eine bestimmte SID festlegen und diese dann auf alle Sicherheitsobjekte anwenden können.

Sollten Sie also eine vollständige Betriebssysteminstallation kopiert haben, ohne dass die SID geändert wurde, so laden Sie zunächst das Tool *NewSID* von der zuvor genannten Internetseite herunter und extrahieren die Dateien auf dem betreffenden Computersystem in einen beliebigen Ordner. Wechseln Sie dann in den Ordner, der die *NewSID*-Dateien enthält, und rufen Sie das Tool *NewSID* durch einen Doppelklick auf. Schließen Sie den Willkommensbildschirm mit einem Klick auf NEXT, und wählen Sie im nächsten Schritt die Option RANDOM SID aus, um eine neue SID per Zufall zu generieren.

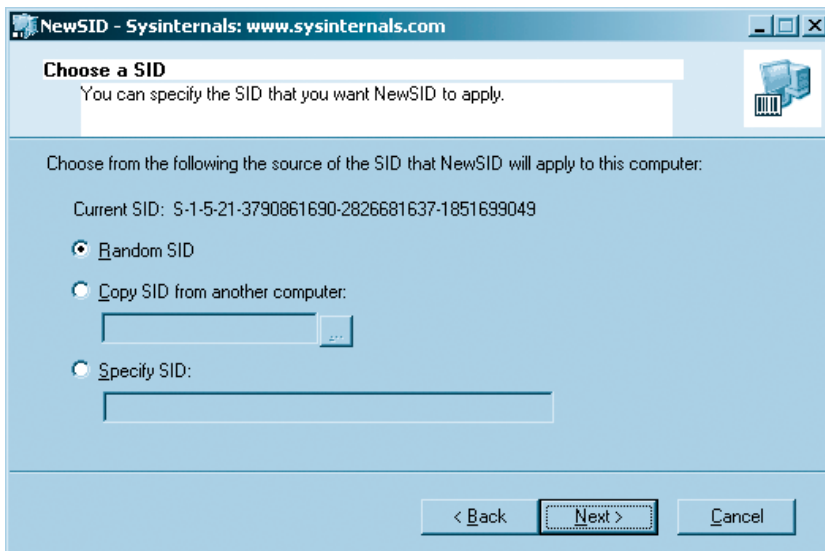


Abbildung 11: Eine neue SID generieren

Klicken Sie auf NEXT, und aktivieren Sie die Option RENAME THE COMPUTER TO:, wenn Sie bei dieser Gelegenheit auch den Hostnamen des Computersystems ändern möchten. Geben Sie den entsprechenden Hostnamen ein, und klicken Sie auf NEXT, woraufhin die aktuelle und auch die neue SID des Computersystems zunächst nur angezeigt werden.

Wenn Sie die SID anwenden möchten, so aktivieren Sie die Option AUTOMATICALLY REBOOT AFTER SID IS APPLIED und klicken auf NEXT, womit die SID aller Sicherheitsobjekte angepasst und das Computersystem daraufhin neu gestartet wird. Nach dem Computerneustart können Sie den Server in das Netzwerk integrieren, da die SIDs sämtlicher Sicherheitsobjekte geändert und auch aktiviert wurden.

Remoteinstallationsdienste nutzen

Windows Server 2003 enthält eine optionale Windows-Komponente mit der Bezeichnung *Remoteinstallationsdienste* (engl. = *Remote Installation Services*), womit Sie in einer Domänenumgebung einen professionellen Installationsserver für Windows 2000-/XP-/2003-Systeme einrichten können.

Diese Remoteinstallationsdienste werden zunächst auf einem Member-Server innerhalb der Domäne eingerichtet, womit der eigentliche RIS-Server entsteht. Anschließend werden die entsprechenden Master-Systeme installiert und den Anforderungen entsprechend eingerichtet, deren Installation und Konfigurationseinstellungen dann mit Hilfe des Tools *Riprep.exe* abgebildet und auf den RIS-Server übertragen werden können. Diese Abbilder stellen jedoch keine gewöhnlichen Images in Form einer Datenträgerduplizierung dar, sondern enthalten ausschließlich die zur Einrichtung weiterer Computersysteme erforderlichen Dateien und Konfigurationseinstellungen.

Sobald entsprechende Abbilder erzeugt und auf dem RIS-Server bereitgestellt wurden, können die zu installierenden Computersysteme mit Hilfe einer PXE-fähigen Netzwerkkarte (*Pre-Boot Execution Environment*) oder wahlweise auch mit einer RIS-spezifischen Bootdiskette gebootet werden, woraufhin sie per DHCP eine Verbindung mit dem RIS-Server herstellen und das zur Ausführung des Installation-Wizards erforderliche Boot-Image herunterladen und ausführen können. Über den Installation-Wizard kann das zu installierende Abbild dann ausgewählt und auf dem betreffenden Computersystem restauriert werden.

Zur Einrichtung der Remoteinstallationsdienste ist es erforderlich, dass der künftige RIS-Server:

- ▶ Mitglied einer Domäne ist
- ▶ Keine Domänen-Controller-Funktion ausführt
- ▶ Auf der Systempartition das NTFS-Dateisystem verwendet
- ▶ Über eine zweite Partition oder eine zweite Festplatte verfügt, die ebenfalls mit NTFS formatiert ist (die Abbilder können nicht auf der Systempartition abgelegt werden)
- ▶ Über genügend freien Speicherbereich zum Ablegen der Abbilder verfügt (für eine Standardinstallation von Windows Server 2003 werden – ohne zusätzliche Applikationen – ungefähr 1,3 GByte benötigt)

Einrichten der Remoteinstallationsdienste

Um die Remoteinstallationsdienste (den RIS-Server) verwenden zu können, müssen Sie die optionale Windows-Komponente zunächst nachinstallieren, da diese Komponente während der Standardinstallation von Windows Server 2003 per Default nicht mit eingerichtet wird.

Legen Sie auf dem Server-System, auf dem Sie die Remoteinstallationsdienste einrichten möchten, die Installations-CD von Windows Server 2003 in das Laufwerk,

48 >> Remoteinstallationsdienste nutzen

oder stellen Sie eine Verbindung zum Distributionsordner her, und öffnen Sie in der SYSTEMSTEUERUNG das Objekt SOFTWARE. Klicken Sie auf WINDOWS-KOMPONENTEN HINZUFÜGEN/ENTFERNEN, und aktivieren Sie die Option REMOTEINSTALLATIONSDIENSTE.

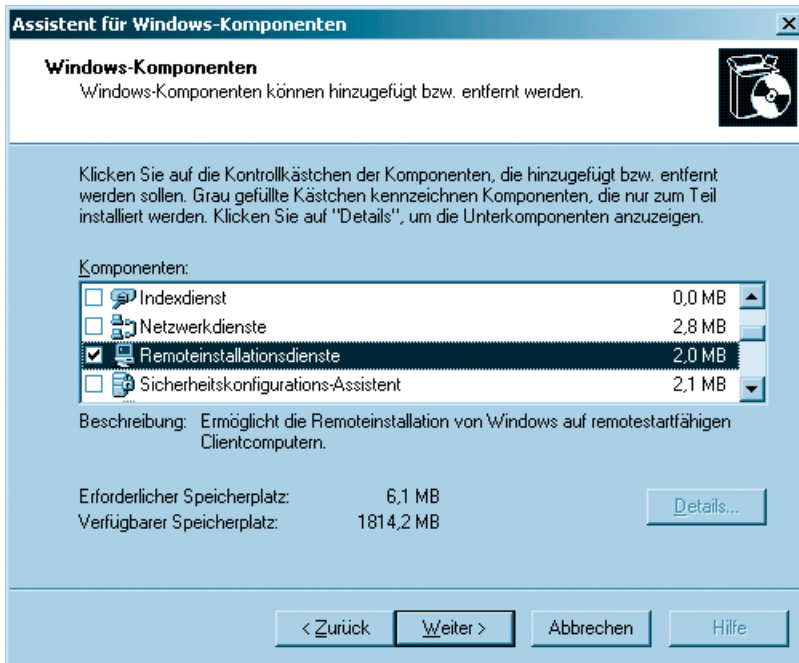


Abbildung 12: Die »Remoteinstallationsdienste« auswählen

Klicken Sie auf WEITER, die erforderlichen Dateien werden nun auf die Festplatte kopiert. Sobald alle Dateien kopiert wurden, können Sie den ASSISTENTEN FÜR WINDOWS-KOMPONENTEN mit einem Klick auf FERTIG STELLEN schließen.

Rufen Sie nun im STARTMENÜ, genauer gesagt unter VERWALTUNG, den Eintrag REMOTEINSTALLATIONSDIENSTE-SETUP auf, um mit der Konfiguration des RIS-Servers fortzufahren. Schließen Sie den Willkommensbildschirm mit einem Klick auf WEITER, und wählen Sie daraufhin den Pfad für die RIS-Struktur aus, wobei Sie hier nicht die Systempartition von Windows Server 2003 verwenden dürfen. Wenn Sie beispielsweise eine weitere Partition oder eine zusätzliche Festplatte *Laufwerk D* eingerichtet haben, so wählen Sie das Root des Laufwerks aus und geben die Bezeichnung für die RIS-Struktur (standardmäßig wird *RemoteInstall* verwendet) ein.

Im nächsten Schritt wählen Sie aus, ob der RIS-Server mit sofortiger Wirkung auf Anfragen aus dem Netzwerk reagieren soll oder erst, wenn der RIS-Server vollständig eingerichtet wird. Da zu diesem Zeitpunkt noch keine Abbilder erstellt und bereitgestellt worden sind, lassen Sie die Option AUF DIENSTANFRAGEN VON CLIENTS ANTWORTEN deaktiviert und klicken auf WEITER.

Als Nächstes müssen Sie den Pfad zu den Windows Server 2003-Installationsdateien auswählen, damit die zur Installation erforderlichen Dateien in die RIS-Struktur kopiert werden können. Wählen Sie an dieser Stelle den bisherigen Distributionsordner (*i386*-Ordner) aus, oder legen Sie den Pfad zur Installations-CD von Windows Server 2003 fest. Klicken Sie auf WEITER, und legen Sie anschließend den Namen für den Ordner fest, der künftig die Installationsdateien für Windows Server 2003 bereithält (per Default wird *Windows* verwendet).

Klicken Sie auf WEITER, um eine kurze Beschreibung für den RIS-Server einzugeben. Im nächsten Schritt werden die zuvor festgelegten Konfigurationseinstellungen noch einmal angezeigt. Klicken Sie auf FERTIG STELLEN, um die automatische Konfiguration des RIS-Servers zu starten.

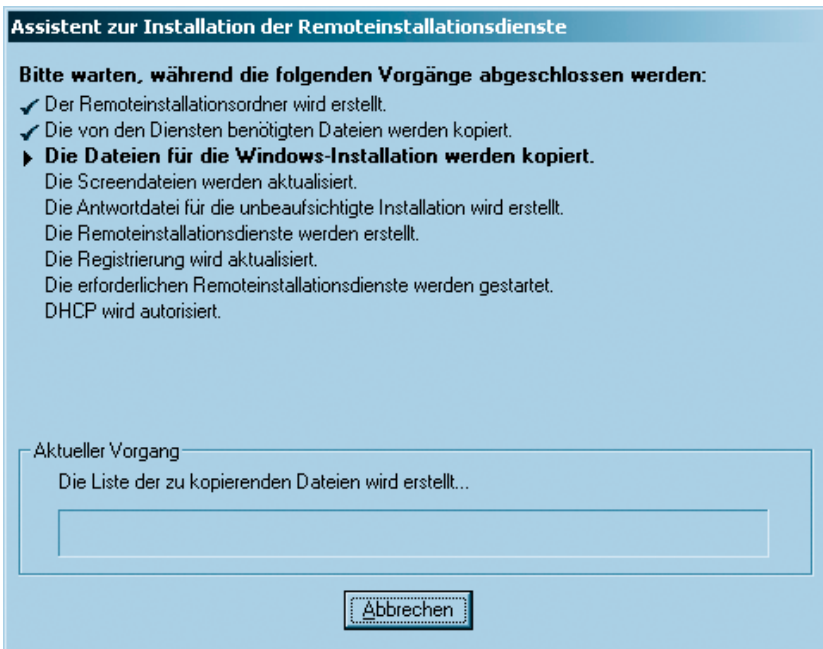


Abbildung 13: Der RIS-Server wird konfiguriert.

Dieser Vorgang kann nun einige Minuten in Anspruch nehmen. Sobald der RIS-Server vollständig konfiguriert ist, beenden Sie den ASSISTENT ZUR INSTALLATION DER REMOTEINSTALLATIONSDIENSTE und führen einen Computerneustart durch, um ein erstes Abbild erstellen und auch bereitstellen zu können.

Erstellen eines Abbildes

Sofern Sie die Remoteinstallationsdienste, sprich: den RIS-Server, vollständig eingerichtet und auch konfiguriert haben, so können Sie nun mit dem Erstellen entsprechender Abbilder mit *RipRep.exe* beginnen. Installieren Sie auf dem Master-System, von dem Sie später ein Abbild erzeugen möchten, Windows Server 2003, und konfi-

gurieren Sie das Betriebssystem gemäß den Anforderungen. Sie können natürlich auch zusätzliche Applikationen einrichten und diese später mit dem Abbild auf andere Computersysteme übertragen.

Wenn das Master-System so weit eingerichtet ist, dann beginnen Sie mit dem Erstellen des Abbildes, das Sie dann auf dem RIS-Server bereitstellen und auf andere Computersysteme übertragen können. Stellen Sie zunächst eine Netzwerkverbindung zwischen dem Master-System und dem RIS-Server her, verbinden Sie sich dann vom Master-System aus mit der Freigabe `\\RIS-Server\Reminst\Admin\i386`, wobei Sie *RIS-Server* durch den Hostnamen oder die IP-Adresse des entsprechenden RIS-Servers ersetzen, und starten Sie das Tool *Riprep.exe*.

Für den Fall, dass das Service Pack 1 auf dem Master-System installiert ist, so erhalten Sie nun eine Sicherheitswarnung und müssen die Ausführung des Tools *Riprep.exe* durch einen Klick auf **ÖFFNEN** manuell bestätigen. Daraufhin wird der **ASSISTENT ZUR VORBEREITUNG DER REMOTEINSTALLATION** gestartet. Schließen Sie den Willkommensbildschirm mit einem Klick auf **WEITER**, und geben Sie im nächsten Schritt den Hostnamen des RIS-Servers an, auf den das Abbild kopiert werden soll.

Klicken Sie auf **WEITER**, und geben Sie anschließend einen Namen für den Ordner an, in den das Abbild kopiert werden soll. Legen Sie im nächsten Schritt eine kurze Beschreibung und bei Bedarf auch einen kleinen Hilfetext fest, um zu einem späteren Zeitpunkt das Abbild besser identifizieren zu können.

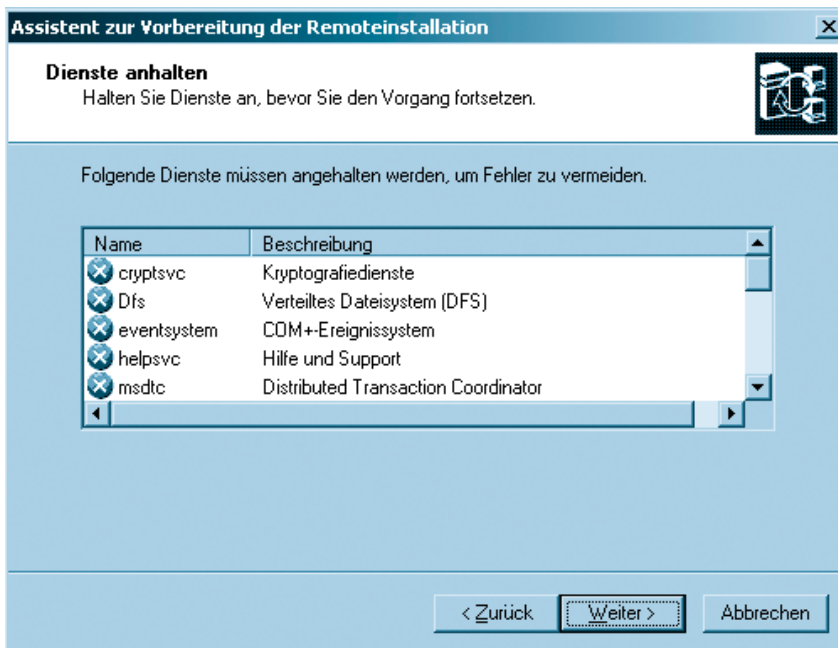


Abbildung 14: Die Dienste, die temporär gestoppt werden müssen

Sie gelangen nun in ein Dialogfenster, worüber gestartete Dienste angezeigt werden, die während der Abbilderstellung temporär gestoppt werden müssen. Klicken Sie auf WEITER, um das Stoppen der Dienste zu bestätigen und mit der Abbilderstellung fortzufahren.

Für den Fall, dass während der Ausführung des Assistenten einzelne Applikationen ausgeführt werden, so erhalten Sie nun eine Auflistung der Applikationen, die gerade gestartet sind und die nun beendet werden müssen. Klicken Sie auf WEITER, um eine kurze Zusammenfassung der ausgewählten Konfigurationseinstellungen zu erhalten, und beginnen Sie mit der Erstellung des Abbildes, indem Sie erneut auf WEITER klicken.

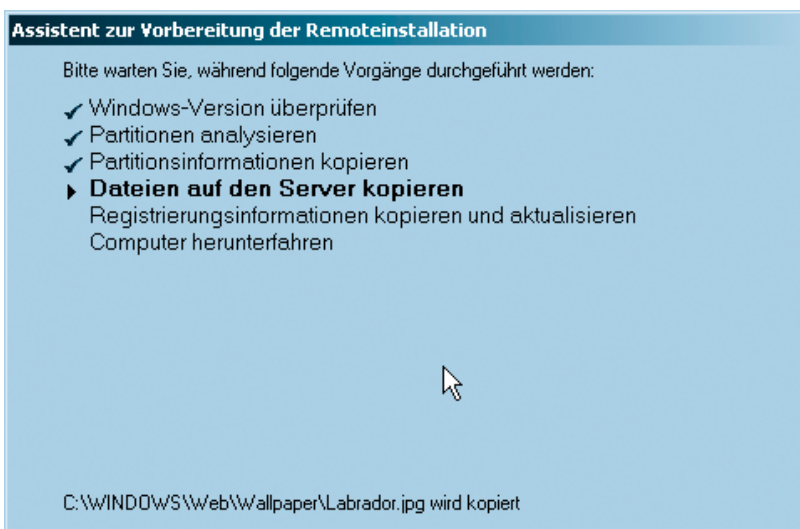


Abbildung 15: Das Abbild wird erstellt.

Sobald das Abbild vollständig auf den RIS-Server übertragen und dort abgelegt wurde, können Sie die automatisch generierte Antwortdatei entsprechend anpassen oder wahlweise auch direkt mit einer teilautomatischen Installation beginnen.

Durchführen der Installation

Die Installation eines mit *Riprep.exe* erzeugten Abbildes erfordert eine funktionstüchtige Netzwerkverbindung zwischen dem zu installierenden Computersystem und dem RIS-Server. Sofern das Computersystem über eine PXE-fähige Netzwerkkarte verfügt, so booten Sie das Computersystem via Netzwerkkarte, anderenfalls erstellen Sie erst eine entsprechende Bootdiskette, damit das zu installierende Computersystem mit dem RIS-Server Verbindung aufnehmen kann. Um eine solche Bootdiskette zu erstellen, legen Sie auf dem RIS-Server eine leere, aber formatierte Diskette ein, rufen im Ordner `\RemoteInstall\Admin\i386` das Tool *Rqfg.exe* auf und klicken auf DISKETTE ERSTELLEN. Die erforderlichen Dateien werden daraufhin auf die Diskette übertragen.

52 >> Remoteinstallationsdienste nutzen

Um ein RIS-Abbild auf einem anderen Computersystem zu restaurieren, starten Sie das betreffende Computersystem nun via Netzwerkkarte oder mit Hilfe der RIS-Bootdiskette. Dem zu installierenden Computersystem wird daraufhin eine IP-Adresse zugewiesen, und die Verbindung zum RIS-Server wird hergestellt. Daraufhin wird das Boot-Image ausgeführt, und der CLIENT INSTALLATION WIZARD wird gestartet.

Schließen Sie den Willkommensbildschirm mit , und geben Sie im nächsten Schritt die Account-Informationen (Benutzername und Kennwort) für den Benutzer ein, der über ausreichend Berechtigungen verfügt, um auf den RIS-Server zugreifen und ein Abbild installieren zu können. Den Benutzernamen müssen Sie dabei im Format *Benutzername@Domäne* eingeben. Das @-Zeichen fügen Sie durch Druck von + ein, da zu diesem Zeitpunkt noch der englische Tastaturtreiber geladen ist.

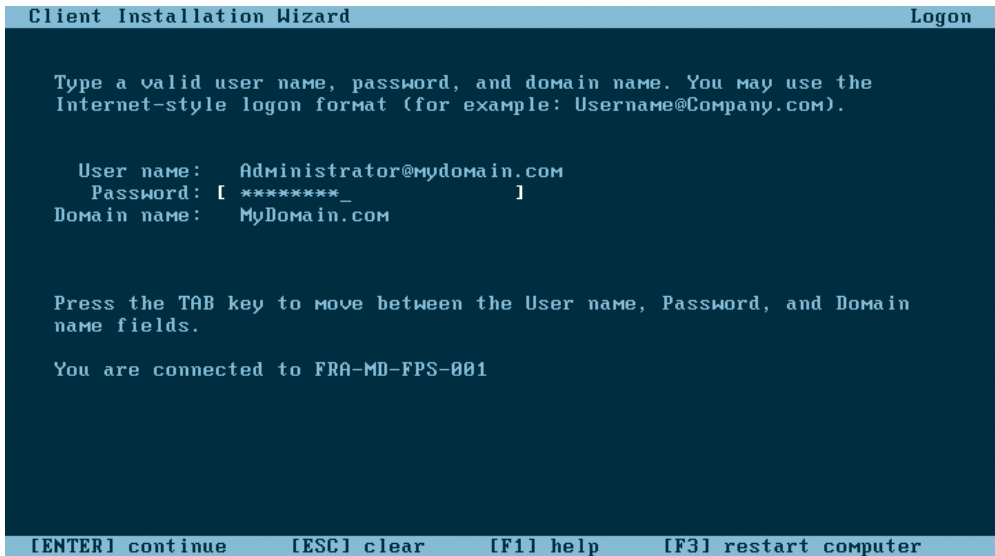


Abbildung 16: Die Anmeldung durchführen

Drücken Sie , und wählen Sie im nächsten Dialogfenster die Option AUTOMATIC SETUP aus, um ein vorhandenes Abbild zu übertragen und zu installieren. Selektieren Sie im Anschluss das Abbild, das Sie restaurieren möchten, und bestätigen Sie die Warnung, dass nun alle Daten auf dem lokalen Datenträger unwiderruflich gelöscht werden. Drücken Sie , woraufhin die RIS-Informationen (*Benutzername*, *GUID*, *RIS-Server*) noch einmal angezeigt werden, und starten Sie die Installation des Abbildes mit .

Die Dateien des Abbildes werden nun auf das zu installierende Computersystem übertragen, und Windows Server 2003 wird eingerichtet. Sofern Sie die Antwortdatei unattend.txt bereits entsprechend angepasst haben, wird die Installation nun vollautomatisch durchgeführt, anderenfalls müssen Sie während der Betriebssysteminstallation die systemspezifischen Informationen wie Produktschlüssel, Computernamen, Administratoren-Kennwort, Domänenmitgliedschaft, Netzwerkeinstellungen etc. von Hand konfigurieren.

Die Windows-Produktaktivierung

Ein letzter, aber dennoch wichtiger Punkt hinsichtlich der Betriebssysteminstallation gilt der Windows-Produktaktivierung (engl. = *Windows Product Activation*). Denn seit Windows XP/Windows Server 2003 muss die Betriebssysteminstallation bei Microsoft aktiviert werden, was der Bekämpfung der Software-Piraterie zugute kommt.

Sofern Sie keine Volume License-Version aus dem Select- oder Open-Programm (Großkundenprogramm von Microsoft) verwenden, müssen Sie die Windows Server 2003-Installation also innerhalb von 30 Tagen bei Microsoft aktivieren, anderenfalls können Sie sich nicht mehr an Windows Server 2003 anmelden. Lediglich die Aktivierung der Betriebssysteminstallation ist dann noch möglich. Die Produktaktivierung kann dabei wahlweise über eine bestehende Internetverbindung oder auch per Telefon durchgeführt werden. Während des Aktivierungsvorgangs wird – basierend auf der Hardware-Ausstattung des zu aktivierenden Computersystems und auch dem während der Installation eingegebenem Produktkey – eine Installations-ID erzeugt, die an Microsoft übermittelt wird. Microsoft verifiziert die übergebene ID und übermittelt daraufhin den eigentlichen Aktivierungscode, womit die Betriebssysteminstallation dann freigeschaltet werden kann. Dieses Prozedere ist immer dann erforderlich, wenn Windows Server 2003 von Grund auf neu installiert wird oder wenn sich größere Änderungen in der Hardware-Ausstattung ergeben.

Wenn Sie nun einen einzelnen oder zumindest nur wenige Computersysteme mit Windows Server 2003 installieren und das Setup manuell vornehmen, so stellt die Windows-Produktaktivierung weniger ein Problem dar, da Sie die Aktivierung der Betriebssysteminstallation dann im Anschluss über eine bestehende Internetverbindung oder via Telefon vornehmen und die Installations-ID, die Sie zurückerhalten, direkt eingeben können.

Für den Fall, dass Sie aber ein größeres Deployment von Windows Server 2003 planen und die Installationen größtenteils auf automatischem Wege vornehmen wollen, so stellt sich natürlich die Frage, wie die einzelnen Betriebssysteminstallationen aktiviert werden können, ohne dass nach Abschluss der Installation ein manueller Eingriff erforderlich wäre.

Hierfür gibt es nun zwei Möglichkeiten:

- ▶ Produktaktivierung per Antwortdatei: Gesetzt den Fall, dass sich die zu installierenden Computersysteme in einem Netzwerk befinden, das über eine Internetanbindung (z.B. über einen Proxy) verfügt, und die Installation mit Hilfe einer Antwortdatei (*Unattend.txt*, *Winnt.sif*, *Sysprep.inf*) durchgeführt wird, so können Sie in der Antwortdatei eine entsprechende Befehlszeile einfügen und eine automatische Produktaktivierung durchführen.
- ▶ Produktaktivierung per Skript: Die Windows-Produktaktivierung lässt sich – sofern eine Internetanbindung vorhanden ist – auch per VBScript durchführen. Die WMI-Klasse *Win32_WindowsProductActivation* stellt hierfür die *ActivateOnline*-Methode bereit, womit Sie die Freischaltung auch per Skript automatisieren können.

54 >> Die Windows-Produktaktivierung

Die nachfolgenden Abschnitte zeigen, wie Sie die Produktaktivierung mit Hilfe einer Antwortdatei und auch per VBScript automatisieren können.

Produktaktivierung per Antwortdatei

Wenn Sie die Computersysteme mit Hilfe einer Antwortdatei teil- oder auch vollautomatisch installieren, so können Sie in der Antwortdatei eine entsprechende Befehlszeile hinterlegen, womit Windows Server 2003 dann nach Abschluss der Installation automatisch aktiviert wird. Dies setzt allerdings voraus, dass das zu installierende Computersystem über eine funktionstüchtige Internetverbindung verfügt.

Um die automatische Produktaktivierung in der Antwortdatei zu hinterlegen, öffnen Sie die betreffende Datei (*Unattended.txt*, *Winnt.sif* oder *Sysprep.inf*) mit einem Editor und fügen unterhalb der Sektion [Unattended] die Zeile `AutoActivate = Yes` ein.

```
[Unattended]
UnattendMode=FullUnattended
OemSkipEula=Yes
OemPreinstall=No
TargetPath=\WINDOWS
AutoActivate = Yes
```

Listing 3: Ein Beispiel für die Unattended-Sektion

Des Weiteren muss natürlich auch der eigentliche Produktschlüssel in der Antwortdatei vorhanden sein. Dieser wird jedoch nicht in der Sektion [Unattended] hinterlegt, sondern in der Sektion [UserData]. Erstellen Sie also – falls noch nicht geschehen – unterhalb von [UserData] die Zeile `ProductKey=xxxxx-xxxxx-xxxxx-xxxxx-xxxxx`, wobei Sie die x-Zeichen durch den eigentlichen Produktschlüssel ersetzen.

```
[UserData]
ProductKey=xxxxx-xxxxx-xxxxx-xxxxx-xxxxx
FullName="Annette Stolz"
OrgName="SerCon GmbH"
ComputerName=FRA-MD-FPS-001
```

Listing 4: Ein Beispiel für die UserData-Sektion

Für den Fall, dass die Internetverbindung über einen Proxy-Server hergestellt wird, so fügen Sie zusätzlich die Sektion [Branding] und unterhalb davon die Zeile `BrandIEUsingUnattended = Yes` ein, um die Erstinitialisierung des Internet Explorers einzuleiten, und hinterlegen dann in der Sektion [Proxy] die eigentlichen Proxy-Einstellungen. Wenn der Proxy-Server beispielsweise unter der URL *http://MyProxy:80* erreichbar ist, sofort aktiviert werden und die URL für alle Protokolle gelten soll, so können Sie das folgende Fragment verwenden:

```
[Branding]
BrandIEUsingUnattended = Yes

[Proxy]
Proxy_Enable = 1
HTTP_Proxy_Server = http://MyProxy:80
Proxy_Override = <local>
Use_Same_Proxy = 1
```

Listing 5: Ein Beispiel für die Branding- und Proxy-Sektion

Produktaktivierung per Skript

Um die Windows-Produktaktivierung mit Hilfe von VBScript vorzunehmen, stellen Sie per `GetObject` zunächst eine Verbindung zum WMI-Namespace `root\cimv2` her, setzen dabei den Impersonation-Level auf `Impersonate` und führen im Anschluss eine `ExecQuery`-Abfrage gegen `Win32_WindowsProductActivation` durch, um das WPA-Objekt zu selektieren.

Anschließend prüfen Sie, ob der Wert der Eigenschaft `ActivationRequired` größer 0 ist, womit Windows Server 2003 aktiviert werden muss, und wenden – sofern die Aktivierung erforderlich ist – die `ActivateOnline`-Methode gegen das WPA-Objekt an, womit die Produktaktivierung über eine bestehende Internetverbindung durchgeführt wird.

Für den Fall, dass die Internetverbindung über einen Proxy-Server aufgebaut werden muss, so können Sie mit der `SetProxySetting`-Methode der WMI-Klasse `Win32_Proxy` die Verbindung konfigurieren, indem Sie der `SetProxySetting`-Methode den Namen des Proxy-Servers (`strProxyServer`) und auch den Port (`strProxyPort`) als Argument übergeben.

```
Option Explicit

Dim oWMI
Dim objWPA
Dim objProxy
Dim colWPA
Dim colProxy
Dim strProxyServer
Dim strProxyPort
Dim Result
Dim strComputer

strProxyServer = "MyProxy"
strProxyPort = "80"
strComputer = "."
```

Listing 6: 1_01_018.vbs

56 >> Die Windows-Produktaktivierung

```
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colWPA = oWMI.ExecQuery ("SELECT * FROM Win32_WindowsProductActivation")
Set colProxy = oWMI.ExecQuery ("SELECT * FROM Win32_Proxy")

For Each objWPA in colWPA
    If objWPA.ActivationRequired > 0 Then
        For Each objProxy in colProxy
            Result = objProxy.SetProxySetting(strProxyServer,strProxyPort)
        Next
        objWPA.ActivateOnline()
    End If
Next

Set oWMI = Nothing
Set objWPA = Nothing
Set objProxy = Nothing
```

Listing 6: 1_01_018.vbs (Forts.)

Betriebssystem-Update

Windows Server 2003 stellt das mit Abstand sicherste und auch stabilste Betriebssystem aus dem Hause Microsoft dar. Nichtsdestotrotz muss auch dieses Betriebssystem regelmäßig aktualisiert und stets auf dem neuesten Stand gehalten werden, um langfristig einen sicheren und zuverlässigen Serverbetrieb gewährleisten zu können. Denn im Laufe der Zeit werden immer wieder neue Sicherheitsschwachstellen bekannt, die ein potenzielles Risiko für die Sicherheit und auch für die Stabilität des Serversystems darstellen und die es daher so schnell wie möglich auszumerzen gilt.

Gerade Viren und Würmer, die darauf getrimmt sind, bekannte Sicherheitslücken auszunutzen und entweder Hintertürchen für weitere Angriffe zu schaffen, Informationen auszulesen und zu übertragen oder einfach nur das Computersystem zu sabotieren, stellen hier eine besonders große Gefahr dar. Mit Hilfe von Sicherheitsupdates, Updates und Update-Rollups, die regelmäßig unter www.microsoft.com bzw. windowupdate.microsoft.com zum Download angeboten werden, lassen sich solche Sicherheitslücken schnell wieder schließen. Das Serversystem gewinnt dadurch zunehmend an Stabilität und auch an Sicherheit.

Diese Kategorie soll Ihnen zeigen, mit welchen Möglichkeiten Sie Windows Server 2003 up to date halten können und wie Sie die Aktualisierung des Betriebssystems auch weitestgehend automatisieren können. Denn gerade in größeren IT-Infrastrukturen kann sich die Aktualisierung der vielen Serversysteme als sehr zeitaufwändig erweisen, da meist mehrere Serversysteme zur gleichen Zeit aktualisiert und oftmals auch mehrere Updates hintereinander installiert werden müssen.

Die Update-Kategorien in der Übersicht

Wenn von der Betriebssystemaktualisierung die Rede ist, so werden des Öfteren Begriffe wie Hotfix, Service Pack, wichtige Updates, Sicherheitsupdates und weitere mehr verwendet. Diese Begriffe bezeichnen die jeweilige Update-Kategorie und geben zum einen Auskunft über die Schwere sicherheitsbezogener und nicht sicherheitsbezogener Fehler, zum anderen aber auch über den Umfang des Updates.

Nachfolgend finden Sie eine kurze Übersicht der gängigsten und wichtigsten Update-Kategorien unter Angabe der wesentlichen Merkmale:

- ▶ **Feature Packs:** Mit der Installation eines *Feature Packs* werden neue Funktionen implementiert, über die das Betriebssystem (oder eine Applikation) bis dato noch nicht verfügt und die offiziell erst in der nächsten Betriebssystem- oder Applikationsversion enthalten sein werden, per Feature Pack aber schon vorab implementiert werden können.
- ▶ **Service Packs:** Unter einem *Service Pack* ist eine Sammlung aller Sicherheitsupdates, wichtiger Updates und Updates sowie Hotfixes zu verstehen, die seit der letzten Betriebssystemversion oder seit dem letzten Service Pack entwickelt und publiziert wurden.

- ▶ Sicherheitsupdates: Ein *Sicherheitsupdate* enthält Aktualisierungen, womit sicherheitsbezogene Fehler eines Betriebssystems oder einer Applikation behoben werden können. Die Kategorie *Sicherheitsupdate* untergliedert sich dabei in *kritische*, *schwere*, *mittlere* und *niedrige* Security-Updates, womit der Schweregrad des Fehlers angegeben wird.
- ▶ Tools: Die Kategorie *Tools* enthält Funktionen, womit die Ausführung bestimmter Aufgaben unterstützt wird.
- ▶ Treiber: Die Kategorie *Treiber* enthält zusätzliche Treiber für neue Hardware-Komponenten oder Treiber-Updates für bereits vorhandene Gerätetreiber.
- ▶ Update-Rollups: Ein *Update-Rollup* enthält meist mehrere *Updates*, *wichtige Updates* und *Sicherheitsupdates*, die der vereinfachten Aktualisierung wegen zu einem Update-Paket zusammengefasst wurden.
- ▶ Updates: Ein gewöhnliches *Update* enthält neuere Versionen eines Betriebssystems oder einer Applikation, womit unkritische und nicht sicherheitsbezogene Fehler (die so genannten Bugs) behoben werden.
- ▶ Wichtige Updates: Ein *wichtiges Update* behebt einen kritischen, aber nicht sicherheitsbezogenen Betriebssystem- oder Applikationsfehler.

Die einzelnen Update-Möglichkeiten

Es gibt eine ganze Reihe von Möglichkeiten, wie Sie Windows Server 2003 aktualisieren und damit auf einen bestimmten Patch-Level (Stand der Betriebssystemversion) bringen können. Denn Sie können das Betriebssystem sowohl manuell aktualisieren wie auch auf teil- oder vollautomatischem Wege. Die nachfolgende Auflistung zeigt eine kurze Übersicht dieser Update-Möglichkeiten:

- ▶ Manuelles Update: Wenn Sie nur wenige (bis max. fünf Computersysteme) Serversysteme aktualisieren müssen und primär die Basissicherheit gewährleisten möchten, so können Sie das Betriebssystem auf manuellem Wege aktualisieren. Hierbei laden Sie die entsprechenden Updates entweder von der Internetseite *www.microsoft.com* herunter oder nutzen den Update-Catalog unter *update.microsoft.com/windowsupdate*, womit die noch nicht installierten Updates automatisch ermittelt und zum Download angeboten werden. Anschließend installieren Sie die Updates dann von Hand. Dies ist jedoch relativ aufwändig, weshalb die manuelle Aktualisierung nur dann durchgeführt werden sollte, wenn nur wenige Serversysteme gepatcht werden müssen. Da Sie die Updates manuell herunterladen, können Sie selbst entscheiden, welche Update-Kategorien Sie downloaden und installieren möchten.
- ▶ Automatische Updates: Windows Server 2003 ist standardmäßig mit der Funktion *Automatische Updates* ausgestattet, womit Sie das Betriebssystem dazu veranlassen können, von selbst auf der Internetseite *www.microsoft.com* nach neuen Updates für das Betriebssystem zu suchen. Sie können hierbei festlegen, ob Sie nur über das Vorhandensein neuer Updates informiert werden möchten oder ob

die Updates direkt heruntergeladen und auch installiert werden sollen. Der Vorteil dieser Funktion ist, dass Sie – wenn Sie die Updates automatisch herunterladen und installieren lassen – sich nicht mehr um die Aktualisierung des Betriebssystems kümmern müssen, da das Betriebssystem diese Aufgabe von selbst übernimmt. Der Nachteil ist jedoch, dass zunächst einmal sämtliche Server über eine Internetverbindung (z.B. über einen Proxy) verfügen müssen und dass Sie kaum eine Kontrolle darüber haben, welche Updates heruntergeladen und installiert werden. Wenn Sie die Funktion *Automatische Updates* einsetzen und alle verfügbaren Update-Pakete automatisch herunterladen und installieren, so besteht die Gefahr, dass aufgrund der Installation von nicht getesteten Update-Paketen das Betriebssystem an Stabilität verliert, da es hin und wieder zu Unverträglichkeiten zwischen den einzelnen Update-Paketen kommen kann. Außerdem bewirkt die automatisch Installation diverser Update-Pakete einen automatischen Computerneustart, was sich gerade in der produktiven Server-Umgebung sehr negativ auswirken kann. Sie sollten diese Funktion also mit größter Vorsicht einsetzen. Die Funktion *Automatische Updates* berücksichtigt dabei alle Updates, Update-Rollups, Sicherheitsupdates, wichtige Updates, Treiber, Feature Packs und Service Packs für das Betriebssystem und auch für Exchange, SQL und Office.

- ▶ **Windows Server Update-Services:** Microsoft stellt auf seiner Internetseite (www.microsoft.com) einen kostenlosen Service namens *Windows Server Update-Service* (kurz *WSUS* genannt) bereit, womit Sie einen eigenen Distributionsserver für Betriebssystem-Updates einrichten können. Dieser Update-Service wird innerhalb des eigenen Netzwerkes auf einem Windows Server 2003 eingerichtet, auf dem die IIS-Dienste (*Internet Information Services*) und auch eine Internetverbindung vorhanden sind, und hat primär die Aufgabe, in regelmäßigen Abständen eine Synchronisierung mit dem Microsoft-eigenen Update-Server durchzuführen, womit dann die zuvor ausgewählten Update-Kategorien (Sicherheitsupdate, Treiber ...) auf den WSUS-Server übertragen werden. Alle anderen Serversysteme innerhalb des eigenen Netzwerkes können ihre Aktualisierung dann vom internen WSUS-Server beziehen und benötigen somit keine eigene Internetverbindung mehr. Der Vorteil des WSUS-Servers liegt darin, dass der Update-Service zunächst über das Vorhandensein neuer Updates informiert. Die auf den WSUS-Server heruntergeladenen Updates können dann sorgfältig geprüft und getestet werden. Erst wenn die Updates zur Distribution freigegeben sind, werden sie auf die restlichen Server übertragen und dort – auf Wunsch – automatisch installiert. Der Windows Server Update-Service berücksichtigt ebenfalls alle Feature Packs, Service Packs, Sicherheitsupdates, Tools, Treiber, Update-Rollups, Updates und wichtige Updates. Lediglich bei den Treiber-Updates gibt es eine Einschränkung, da WSUS nur kritische Treiber-Updates beachtet.

Für welche Update-Möglichkeit Sie sich entscheiden, hängt zunächst von der Anzahl der zu aktualisierenden Serversysteme und auch von der Regelmäßigkeit der Betriebssystemaktualisierung ab. Des Weiteren sind aber auch die Update-Kategorien (siehe vorheriger Abschnitt) relevant, da nicht jede Update-Möglichkeit alle Update-Kategorien berücksichtigt.

Das Betriebssystem manuell aktualisieren

Für den Fall, dass Sie nur wenige Computersysteme mit Windows Server 2003 installiert haben und nur einige der verfügbaren Updates einrichten müssen, so können Sie die Aktualisierung des Betriebssystems manuell vornehmen. Will heißen, dass Sie die Internetseite *www.microsoft.com* aufrufen, die entsprechenden Updates dann von Hand suchen und herunterladen oder dass Sie den Update-Katalog unter *update.microsoft.com/windowsupdate* verwenden, womit die noch fehlenden Updates automatisch ermittelt und zum Download angeboten werden können. Letzteres erfordert, dass beim ersten Aufruf der Internetseite *update.microsoft.com/windowsupdate* ein ActiveX-Steuerelement heruntergeladen und installiert wird, womit die bereits installierten (bzw. die noch fehlenden) Updates ermittelt werden können.

Wenn Sie beispielsweise ein einzelnes Serversystem auf einen bestimmten Patch-Level bringen müssen und dies vorzugsweise manuell durchführen möchten, so rufen Sie der Einfachheit halber die Internetseite *update.microsoft.com/windowsupdate* auf und stimmen der Installation des ActiveX-Steuerelements durch einen Klick auf JA zu. Sobald das Steuerelement installiert wurde, werden zwei Schaltflächen (SCHNELL-SUCHE und BENUTZERDEFINIERTESUCHE) eingeblendet, worüber Sie den Update-Modus auswählen können.

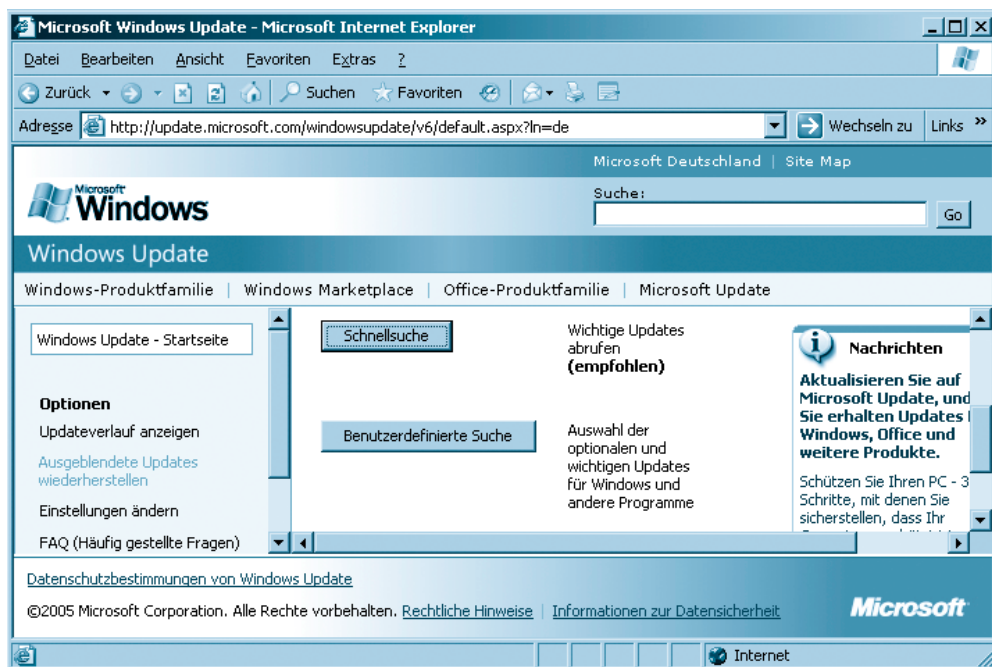


Abbildung 17: Zwischen »Schnellsuche« und »Benutzerdefinierte Suche« auswählen

- ▶ Schnellsuche: Klicken Sie auf SCHNELLSUCHE, um die noch fehlenden Updates automatisch zu ermitteln und herunterzuladen. Beachten Sie, dass Sie bei Auswahl dieser Update-Funktion keine Möglichkeit haben, die zur Verfügung stehenden Update-Pakete auszuwählen.
- ▶ Benutzerdefinierte Suche: Klicken Sie stattdessen auf BENUTZERDEFINIERTE SUCHE, wenn Sie eine Übersicht der zur Verfügung stehenden Update-Pakete erhalten und die entsprechenden Update-Pakete von Hand auswählen möchten.

Wenn Sie Ihre Auswahl entsprechend getroffen haben, so werden die jeweiligen Update-Pakete heruntergeladen und auf das lokale Computersystem übertragen. Die Update-Pakete werden dabei im Ordner `C:\Windows\SoftwareDistribution\Download` zwischengespeichert und anschließend installiert. Die Zwischenspeicherung der Update-Pakete hat hierbei den entscheidenden Vorteil, dass Sie den Inhalt des Ordners auf ein anderes Computersystem übertragen und die Update-Pakete dort installieren können, ohne dass Sie die einzelnen Updates ein weiteres Mal downloaden müssen.

Mehrere Updates und Sicherheitsupdates auf einmal installieren

Gesetzt den Fall, dass Sie mehrere Windows Server 2003-Systeme auf einen bestimmten Patch-Level bringen müssen und die Installation der Updates und Security Updates von Hand durchführen möchten, so können Sie sich die Arbeit ein ganzes Stück vereinfachen, in dem Sie die zu installierenden Update-Pakete in einer Batch-Datei aufnehmen und die ausführbaren Dateien hintereinander und jeweils mit dem Parameter `/NoRestart` aufrufen. Denn damit können Sie mehrere Update-Pakete auf einmal installieren und brauchen den erforderlichen Computerneustart nur einmal (nämlich nach Abschluss sämtlicher Installationen) auszuführen, da Sie mit dem Parameter `/NoRestart` den Computerneustart zwischen den einzelnen Installationen unterdrücken können. Das Betriebssystem-Update lässt sich so wesentlich schneller durchführen, und das Serversystem steht umso schneller zur Verfügung.

Wenn Sie beispielsweise die Updates *WindowsServer2003-KB896727-x86-DEU.exe*, *WindowsServer2003-KB883939-x86-DEU.exe* und *WindowsServer2003-KB867801-x86-DEU.exe* von der Internetseite www.microsoft.com heruntergeladen haben und nun in einem Rutsch installieren möchten, so legen Sie die Dateien entweder auf der Festplatte des betreffenden Computersystems oder aber in einem freigegebenen Distributionsordner ab, erzeugen dann eine Batch-Datei und rufen aus der Batch-Datei heraus die einzelnen Updates mit dem Parameter `/NoRestart` auf. Damit werden die Updates dann automatisch installiert, ohne dass jedes Mal ein Computerneustart durchgeführt wird. Und mit dem zusätzlichen Parameter `/Quiet` können Sie ganz nebenbei auch die Anzeige des Installationsvorgangs verbergen.

Nachfolgend finden Sie ein Beispiel für eine Batch-Datei, womit Sie die zuvor genannten Updates automatisch installieren können, ohne dass das Computersystem jeweils nach der Einrichtung eines Updates neu gestartet wird. Lediglich nach der

Installation des letzten Updates wird ein Computerneustart durchgeführt, damit die zu aktualisierenden Dateien durch die neuen Dateiversionen ersetzt werden. Die Batch-Datei setzt dabei voraus, dass die einzelnen Update-Pakete im Ordner `C:\Temp` abgelegt sind.

```
@Echo off
C:\Temp\WindowsServer2003-KB896727-x86-DEU.exe /NoRestart /Quiet
C:\Temp\WindowsServer2003-KB883939-x86-DEU.exe /NoRestart /Quiet
C:\Temp\WindowsServer2003-KB867801-x86-DEU.exe /Quiet
```

Listing 7: 1_02_003.bat

Beim Anlegen der Batch-Datei müssen Sie nicht unbedingt auf die chronologische Reihenfolge der einzelnen Update-Pakete (zu erkennen an der KB-Nummer) achten. Denn die ausführbaren Hotfix- und Update-Dateien für Windows Server 2003 beinhalten bereits von Haus aus die Funktionen des *Qchain.exe*-Utilities (dieses Tool musste bis Windows 2000 noch manuell aufgerufen werden), womit die aktualisierten Dateien automatisch auf ihre Version hin überprüft werden. Und wenn mehrere Updates installiert werden, die sich möglicherweise auf gleiche Dateien auswirken, so wird aus den einzelnen Updates automatisch die neueste Version ermittelt und nach dem nächsten Computerneustart auch angewendet.

Wenn Sie ein Update-Paket von Hand installieren, so wird am Ende des Installationsvorgangs ein Dialogfenster geöffnet, wo Sie über den Status der Installation informiert werden. Das Dialogfenster gibt Auskunft darüber, ob die Installation erfolgreich durchgeführt wurde oder ob das Setup aus einem bestimmten Grund fehlgeschlagen ist. Wenn Sie ein einzelnes oder auch mehrere Update-Pakete per Batch-Datei im *Quiet*-Modus installieren, so bleibt diese Meldung aus, weshalb Sie erst einmal nicht wissen, ob die Installation erfolgreich war oder misslungen ist.

Das Betriebssystem generiert jedoch während der automatischen Installation eines Update-Paketes eine Log-Datei mit der Bezeichnung *KBxxxxx.log*, die per Default im Windows-Ordner abgelegt wird. Anhand der Einträge, die in der Log-Datei enthalten sind, können Sie den Status der Installation erkennen und erfahren so, ob die Installation fehlerfrei durchgelaufen oder abgebrochen ist. Und für den Fall, dass Sie die einzelnen Logfiles in einem anderen Pfad ablegen möchten, so fügen Sie am Ende der Befehlszeile, worüber das Update-Paket installiert wird, den Parameter */Log:Pfad* hinzu, wobei Sie Pfad durch den Laufwerk-, Ordner- und gegebenenfalls auch Dateinamen ersetzen.

Automatische Updates-Funktion verwenden

Windows Server 2003 ist standardmäßig mit der Funktion AUTOMATISCHE UPDATES ausgestattet, womit Sie das Betriebssystem auf recht einfache Art und Weise aktualisieren können. Sie haben dabei die Möglichkeit, sowohl den Download wie auch das Installieren der Updates vollständig zu automatisieren.

Um die AUTOMATISCHE UPDATES-Funktion einsetzen zu können, muss das Serversystem entweder über eine funktionstüchtige Internetverbindung verfügen, oder innerhalb des lokalen Netzwerkes muss ein WSUS-Server (siehe Windows Server Update-Service einrichten auf Seite 66) vorhanden sein, der die Updates bereithält. Wenn diese Voraussetzungen gegeben sind, dann können Sie die AUTOMATISCHE UPDATES-Funktion nun aktivieren und den Anforderungen entsprechend einrichten.

Wechseln Sie hierzu in die SYSTEMSTEUERUNG, und rufen Sie dort das Objekt AUTOMATISCHE UPDATES auf. Aktivieren Sie nun eine der nachfolgenden Optionen, um die Funktion generell zu aktivieren und das Verhalten der AUTOMATISCHE UPDATES-Funktion zu bestimmen:

- ▶ **AUTOMATISCH (EMPFOHLEN):** Aktivieren Sie diese Option, wenn Windows Server 2003 eine automatische Verbindung zur Internetseite *update.microsoft.com/windowsupdate* herstellen und nach verfügbaren Updates suchen soll. Bei Auswahl dieser Option werden die Updates automatisch heruntergeladen und an dem angegebenen Tag und zur angegebenen Uhrzeit installiert. Die Installation hat unter Umständen einen automatischen Computerneustart zur Folge, weshalb diese Option für die produktive Serverumgebung nicht zu empfehlen ist.
- ▶ **UPDATES DOWNLOADEN, ABER INSTALLATIONSZEITPUNKT MANUELL FESTLEGEN:** Aktivieren Sie diese Option, wenn Windows Server 2003 selbstständig nach neuen Updates suchen und diese auch automatisch heruntergeladen soll. Sobald neue Updates heruntergeladen wurden, erscheint im Gerätefeld der Taskleiste eine Meldung, dass neue Updates vorhanden sind und diese nun installiert werden können. Sie haben dann die Möglichkeit, die Installation zu starten oder auch auf einen späteren Zeitpunkt zu verschieben.
- ▶ **BENACHRICHTIGEN, ABER NICHT AUTOMATISCH DOWNLOADEN ODER INSTALLIEREN:** Aktivieren Sie diese Option, wenn Windows Server 2003 selbstständig nach neuen Updates suchen soll, diese aber nicht downloaden und erst recht nicht installieren soll. Sobald auf der Internetseite *update.microsoft.com/windowsupdate* neue Updates verfügbar sind, erhalten Sie ein Dialogfenster, wo Sie die Updates auswählen und nach entsprechender Markierung auch herunterladen (und bei Bedarf direkt installieren) können.

Wenn Sie die gewünschte Option aktiviert und gegebenenfalls auch den Wochentag und die Uhrzeit für die automatische Installation der Updates festgelegt haben, dann klicken Sie auf **ÜBERNEHMEN**, woraufhin die Änderungen gespeichert werden und das Dialogfenster wieder geschlossen wird.

Die AUTOMATISCHE UPDATES-Funktion lässt Sie auch direkt in der Registrierung aktivieren und konfigurieren und daher mit dem `Reg`-Befehl auch von der Eingabeaufforderung (z.B. aus einer Batch-Datei heraus) einschalten und einrichten. Hierbei gilt jedoch zu berücksichtigen, dass die AUTOMATISCHE UPDATES-Einstellungen, die in der Registrierung hinterlegt sind, an zwei verschiedenen Stellen konfiguriert werden können. Denn Sie können die AUTOMATISCHE UPDATES-Funktion zum einen direkt für das lokale Computersystem konfigurieren, zum anderen aber auch die Einstellungen der Gruppenrichtlinie für die AUTOMATISCHE UPDATES-Funktion modifizieren. Die direkten Einstellungen

für die AUTOMATISCHE UPDATES-Funktion des lokalen Computersystems sind im Schlüssel HKLM\SOFTWARE\MICROSOFT\CURRENTVERSION\WINDOWSUPDATE\AUTO UPDATE zu finden, während die Einstellungen der Gruppenrichtlinie im Schlüssel HKLM\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\WINDOWSUPDATE\AU enthalten sind.

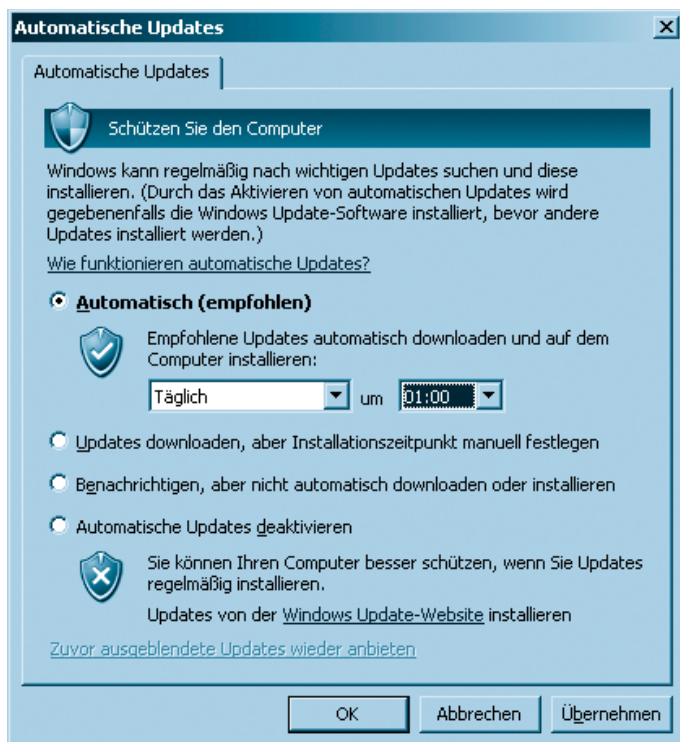


Abbildung 18: Die Updates automatisch herunterladen und installieren

Tipp

Für den Fall, dass Sie die AUTOMATISCHE UPDATES-Funktion auf mehreren Serversystemen innerhalb des Active Directory aktivieren und auch konfigurieren möchten, so können Sie die Einstellungen auch in den Gruppenrichtlinien vornehmen und anschließend zuweisen. Damit müssen Sie die Einstellung nur einmal an zentraler Stelle vornehmen und ersparen sich das vermehrte Konfigurieren auf den einzelnen Serversystemen. Sie finden die betreffenden Richtlinien unter COMPUTERKONFIGURATION\ADMINISTRATIVE VORLAGEN\WINDOWS UPDATE. Wenn Sie einen internen WSUS-Server installiert haben, so können Sie dort auch die Adresse des WSUS-Servers hinterlegen.

Wenn Sie die AUTOMATISCHE UPDATES-Funktion aktivieren und die Updates direkt von Microsoft beziehen möchten, so nehmen Sie die Einstellungen unter HKLM\SOFTWARE\MICROSOFT\CURRENTVERSION\WINDOWSUPDATE\AUTO UPDATE vor. Wenn Sie dagegen einen WSUS-Server einsetzen, so hinterlegen Sie die Einstellungen im Schlüssel

HKLM\SOFTWARE\POLICIES\MICROSOFT\WINDOWSUPDATE\AU, da Sie in diesem Schlüssel auch die Adresse für den WSUS-Server hinterlegen können (siehe Windows Server Update-Service einrichten auf Seite 66).

Für den Fall, dass Sie die Konfiguration der AUTOMATISCHEN UPDATES-Funktion per Batch-Datei vornehmen und die Updates direkt von Microsoft beziehen möchten, so erzeugen Sie mit dem `Reg Add`-Befehl im Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\WINDOWSUPDATE\AUTO UPDATE einen neuen DWORD-Wert mit der Bezeichnung *AUOptions* und weisen dem DWORD-Wert den entsprechenden Wert zu. Folgende Werte bzw. Optionen sind möglich:

Wert	Beschreibung
1	Automatische Updates deaktivieren
2	Benachrichtigen, aber nicht automatisch downloaden oder installieren
3	Updates downloaden, aber Installationszeitpunkt manuell festlegen
4	Automatisch (empfohlen)

Tabelle 3: Die Werte des DWORD-Wertes »AUOptions«

Um beispielsweise die Option BENACHRICHTIGEN, ABER NICHT AUTOMATISCH DOWNLOADEN ODER INSTALLIEREN zu konfigurieren, können Sie folgende `Reg Add`-Befehlszeile verwenden, womit Sie den DWORD-Wert *AUOptions* erzeugen (bzw. überschreiben, da er schon vorhanden ist) und diesem den Wert 2 zuweisen können:

```
@Echo off
Reg Add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto _
Update" /v AUOptions /t REG_DWORD /d 2 /f
```

Listing 8: 1_02_004a.bat

Falls Sie Option AUTOMATISCH (EMPFOHLEN) verwenden möchten, so müssen Sie auch den Tag, an dem die Updates installiert werden sollen, wie auch die Uhrzeit angeben. Um die Auswahl *Täglich* oder einen bestimmten Wochentag festzulegen, fügen Sie dem zuvor genannten Schlüssel einen neuen DWORD-Wert mit der Bezeichnung *ScheduledInstallDay* hinzu und weisen den entsprechenden Wert (siehe nachfolgende Tabelle) zu.

Wert	Beschreibung
0	Täglich
1	Jeden Sonntag
2	Jeden Montag
3	Jeden Dienstag

Tabelle 4: Die Werte des DWORD-Wertes »ScheduledInstallDay«

Wert	Beschreibung
4	Jeden Mittwoch
5	Jeden Donnerstag
6	Jeden Freitag
7	Jeden Samstag

Tabelle 4: Die Werte des DWORD-Wertes »ScheduledInstallDay« (Forts.)

Des Weiteren fügen Sie den DWORD-Wert *ScheduledInstallTime* hinzu, um die Uhrzeit anzugeben, wann mit der Installation der Updates begonnen werden soll. Mögliche Werte sind hierbei die Uhrzeiten 00:00 bis 23:00, wobei Sie jedoch nur volle Stunden angeben können. Um beispielsweise die Installation um 02:00 zu starten, weisen Sie dem DWORD-Wert *ScheduledInstallTime* den dezimalen Wert 2 zu. Soll die Installation dagegen um 23:00 starten, so setzen Sie stattdessen den dezimalen Wert 23.

Das nachfolgende Batch-Datei-Beispiel aktiviert die AUTOMATISCHE UPDATES-Funktion mit der Option *Automatisch (empfohlen)* und installiert die verfügbaren Updates jeweils *täglich* um 23:00 Uhr.

```
@Echo off
Reg Add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto _
Update" /v AUOptions /t REG_DWORD /d 4 /f
Reg Add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto _
Update" /v ScheduledInstallDay /t REG_DWORD /d 0 /f
Reg Add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto _
Update" /v ScheduledInstallTime /t REG_DWORD /d 23 /f
```

Listing 9: 1_02_004b.bat

Windows Server Update-Service einrichten

Wenn Sie mehrere Windows Server 2003-Systeme up to date halten müssen und die Aktualisierung so einfach wie möglich gestalten möchten, so können Sie einen eigenen Distributionsserver für die Updates einrichten. Denn Microsoft stellt auf seiner Internetseite (*www.microsoft.com*) ein kostenloses Add-on für Windows Server 2003 bereit, womit Sie einen internen Update-Server konfigurieren können: den **Windows Server Update-Service**, kurz auch **WSUS** genannt.

Die Einrichtung des Windows Server Update-Services stellt folgende Systemanforderungen an das Serversystem, auf dem der WSUS-Service eingerichtet wird:

- ▶ Die IIS-Dienste müssen eingerichtet sein.
- ▶ Der Server benötigt eine Internetverbindung.

- Auf dem Server muss ein mit NTFS formatiertes Laufwerk mit mindestens 6 GByte freiem Speicherplatz vorhanden sein, um die Updates lokal zwischenspeichern zu können.

Um einen solchen Update-Server einzurichten und zur Aktualisierung der internen Windows Server 2003-Systeme nutzen zu können, laden Sie zunächst das Tool *WSUSSetup.exe* von der zuvor genannten Internetseite herunter und installieren es auf einem Serversystem, das über eine funktionstüchtige Internetverbindung verfügt. Anschließend richten Sie den WSUS-Server den Anforderungen entsprechend ein und legen zumindest fest, welche Update-Kategorien berücksichtigt werden sollen. Anschließend konfigurieren Sie die WSUS-Client-Funktionalität auf den restlichen Serversystemen, womit die Updates dann nicht mehr von Microsoft direkt bezogen, sondern innerhalb des Netzwerkes vom WSUS-Server verteilt werden.

Einrichten des WSUS-Servers

Der erste Schritt zur Einrichtung eines eigenen Update-Servers liegt darin, den Windows Server Update-Service herunterzuladen und auf einem mit Windows Server 2003 installierten Computersystem, das über die Internet-Informationendienste und über eine Internetverbindung verfügt, einzurichten. Laden Sie das Tool *WSUSSetup.exe* also von der Internetseite www.microsoft.com herunter, und starten Sie die Installation mit einem Doppelklick auf die Datei *WSUSSetup.exe*. Daraufhin werden die Dateien in einen temporären Ordner extrahiert, und der SETUP-ASSISTENT FÜR MICROSOFT WINDOWS SERVER UPDATE SERVICES wird gestartet.

Schließen Sie den Willkommensbildschirm mit einem Klick auf WEITER, und stimmen Sie im nächsten Schritt dem Lizenzvertrag zu. Legen Sie anschließend fest, ob Sie die Updates lokal zwischenspeichern möchten und – wenn Sie die Option UPDATES LOKAL SPEICHERN aktivieren – wo die jeweiligen Updates abgelegt werden sollen. Wenn Sie die Updates lokal zwischenspeichern möchten, so muss das betreffende Laufwerk mit NTFS formatiert sein und mindestens über 6 GByte freien Speicherplatz verfügen. Wenn Sie die Updates nicht zwischenspeichern möchten, so deaktivieren Sie die Option UPDATES LOKAL SPEICHERN, womit die Updates dann mit dem ersten Zugriff von der Microsoft-Internetseite heruntergeladen werden.

Bestimmen Sie nun die Datenbank-Optionen. Wählen Sie die Option SQL SERVER-DESKTOPMODUL (WINDOWS) AUF DIESEM COMPUTER INSTALLIEREN aus, um eine Art Mini-SQL Server einzurichten, oder wählen Sie – falls Sie bereits einen Datenbankserver installiert haben – den zu verwendenden Datenbankserver aus. Sofern sich auf dem Serversystem kein entsprechender Datenbankserver befindet, so müssen Sie zwangsläufig die Option SQL SERVER-DESKTOPMODUL (WINDOWS) AUF DIESEM COMPUTER INSTALLIEREN auswählen.

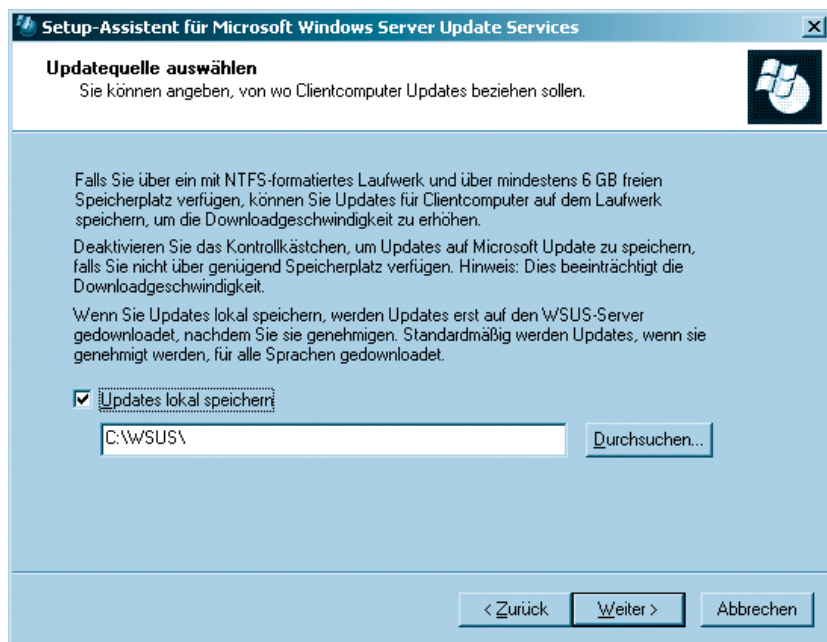


Abbildung 19: Den Ordner für die Updates festlegen

Im nächsten Schritt gilt es, die bevorzugte Webseite festzulegen. Wählen Sie der Einfachheit halber die Option **DIE VORHANDENE IIS-STANDARDWEBSEITE VERWENDEN (EMPFOHLEN)** aus, womit zur Administration des WSUS-Servers die IIS-Standardwebseite, die über Port 80 erreichbar ist, verwendet wird. Wenn Sie dagegen eine gesonderte WSUS-Seite einrichten möchten, so wählen Sie stattdessen die Option **EINE NEUE MICROSOFT WINDOWS SERVER UPDATE SERVICE-WEBSEITE ERSTELLEN** aus, die per Default über Port 8530 erreichbar sein wird. Letzteres muss gerade dann berücksichtigt werden, wenn die Serversysteme mit einer Firewall ausgestattet sind, und beispielsweise müssen Sie Paketfilter einsetzen, da der Port 8530 in den meisten Fällen explizit freigeschaltet werden muss.

Klicken Sie auf **WEITER**, um in das Dialogfenster **UPDATEEINSTELLUNGEN SPIEGELN** zu gelangen. Sofern Sie bereits einen WSUS-Server eingerichtet haben und die gleichen Konfigurationseinstellungen auch auf dem neuen WSUS-Server nutzen möchten, so können Sie die Einstellungen des vorhandenen WSUS-Servers auf den neuen Server übertragen. Aktivieren Sie hierzu die Option **DIESER SERVER ERBT DIE EINSTELLUNGEN VON FOLGENDEM SERVER**, und geben Sie den Hostnamen oder die IP-Adresse wie auch den Port ein. Klicken Sie anschließend auf **WEITER**, woraufhin die zuvor festgelegten Konfigurationseinstellungen noch einmal angezeigt werden. Klicken Sie dann auf **FERTIG STELLEN**, um mit der eigentlichen Installation des WSUS-Servers zu beginnen.

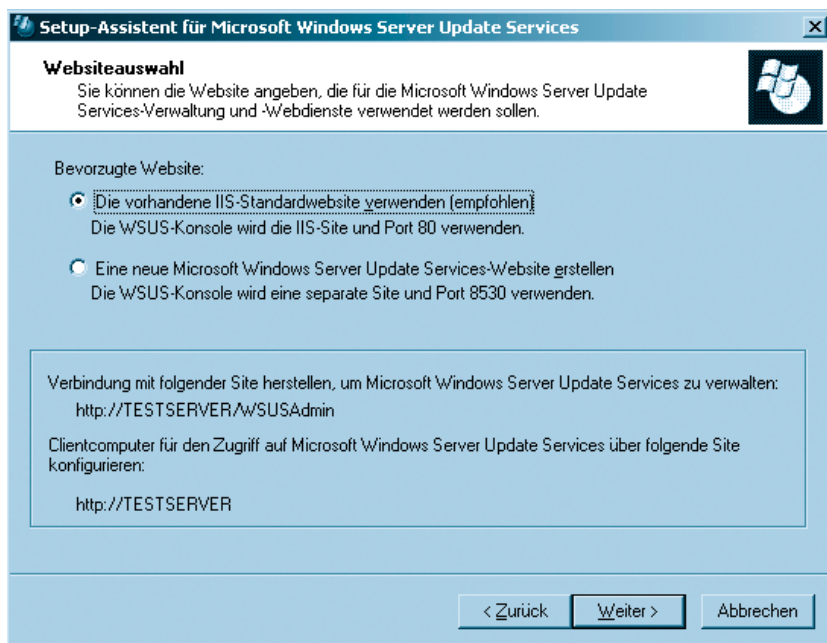


Abbildung 20: Die bevorzugte Webseite festlegen

Konfigurieren des WSUS-Servers

Wenn die eigentliche Installation des WSUS-Servers so weit abgeschlossen ist, geht es mit der Einrichtung, sprich: mit der Konfiguration, des Windows Server Update-Services weiter. Hierzu rufen Sie im STARTMENÜ, genauer gesagt unter VERWALTUNG, den Eintrag MICROSOFT WINDOWS SERVER UPDATE SERVICES auf oder geben im Browser die URL `http://Server/WSUSAdmin/` (wobei *Server* durch den eigentlichen WSUS-Servernamen oder dessen IP-Adresse zu ersetzen ist) ein, um die Administrationskonsole des WSUS-Servers zu starten. Anschließend klicken Sie im unteren Teil des Fensters (in der Sektion AUFGABENLISTE) auf ALS ERSTES DEN SERVER SYNCHRONISIEREN, um den Zeitplan festzulegen, gegebenenfalls die Proxy-Einstellungen vorzunehmen und auch die Microsoft-Produkte und gewünschten Update-Kategorien auszuwählen.

Für den Fall, dass der WSUS-Server eine automatische Synchronisierung mit dem Microsoft-eigenen Update-Server vornehmen soll, so legen Sie zunächst den gewünschten Zeitplan der Synchronisierung fest. Anschließend wählen Sie in der Sektion PRODUKTE UND KLASSIFIZIERUNGEN die Betriebssysteme und auch die Update-Kategorien aus, die während der Synchronisierung berücksichtigt werden sollen.

Wenn die Verbindung zwischen dem WSUS-Server und dem Microsoft-eigenen Update-Server über einen Proxy-Server hergestellt wird, so hinterlegen Sie in der Sektion PROXYSERVER die URL (z.B. `http://myproxy`) und auch den Port des Proxy-Servers und legen – falls erforderlich – den Benutzernamen und auch das Kennwort für den Proxy-Zugang fest, damit der WSUS-Server ohne eine weitere Interaktion eine Verbindung zum Microsoft-Server herstellen kann.

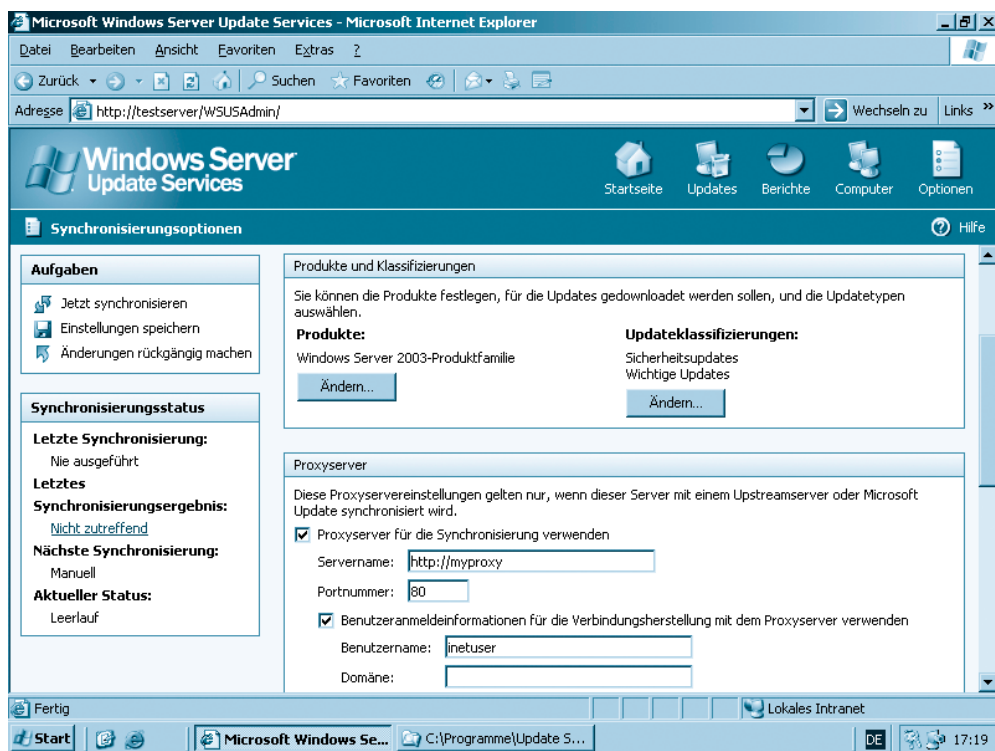


Abbildung 21: Die grundlegenden Konfigurationseinstellungen für den WSUS-Server

Des Weiteren geben Sie in der Sektion UPDATEQUELLE an, ob die Synchronisierung direkt mit dem Microsoft-eigenen Update-Server durchgeführt werden soll (wofür eine Internetverbindung erforderlich ist) oder ob die Updates von einem bereits vorhandenen WSUS-Server (ein so genannter Upstream-Server), der sich im Netzwerk befindet, bezogen werden sollen. Klicken Sie anschließend in der Sektion DATEIEN UND SPRACHE AKTUALISIEREN auf ERWEITET, um die Auswahl der Zwischenspeicherung (auf dem Microsoft-Server oder lokal) und auch eine Einschränkung der Sprachversionen vorzunehmen.

Wenn Sie alle Einstellungen entsprechend vorgenommen haben, klicken Sie im linken Fensterbereich auf EINSTELLUNGEN SPEICHERN, um die Änderung zu übernehmen.

Führen Sie nun eine erste Synchronisierung mit dem Update-Server von Microsoft oder mit dem internen Upstream-Server durch, woraufhin die Updates auf den lokalen WSUS-Server übertragen und damit auch von den WSUS-Clients abgerufen werden können.

Genehmigen der Updates

Wenn auf dem Update-Server von Microsoft oder aber auf dem internen Upstream-Server neue Update-Pakete für den Download angeboten und diese auf den WSUS-Server übertragen wurden, so müssen Sie die Update-Pakete daraufhin für die Ermittlung und auch für die Installation genehmigen. Damit haben Sie die Möglichkeit, einzelne Updates von der Verteilung auszuschließen. Mit dieser Genehmigung legen Sie also fest, wie WSUS mit den jeweiligen Update-Paketen verfahren soll. Folgende Möglichkeiten stehen dabei zur Auswahl:

Genehmigungseinstellung	Beschreibung
Installieren	Die Ermittlung und auch die Installation werden genehmigt. Das Update-Paket wird damit an die Clients weitergereicht, auf denen das Update noch nicht eingerichtet wurde (und die es benötigen), und dort auch installiert.
Nur ermitteln	Lediglich die Ermittlung wird genehmigt. Damit wird auf den WSUS-Clients geprüft, ob das Update benötigt wird, das Update wird jedoch nicht installiert.
Entfernen	Sofern ein bereits installiertes Update die Deinstallation unterstützt, so können Sie das betreffende Update für das Entfernen genehmigen. Das Update wird nach der nächsten Synchronisierung zwischen WSUS-Client und -Server automatisch deinstalliert.
Update ablehnen	Wenn Sie bestimmte Update-Pakete aus dem Synchronisierungsprozess ausschließen und damit deren Verteilung unterbinden möchten, so wählen Sie <i>Update ablehnen</i> aus.

Tabelle 5: Die Genehmigungseinstellungen des WSUS-Servers

Die Genehmigung können Sie für jedes Update-Paket manuell vornehmen, Sie können aber auch entsprechende Regeln erstellen, womit die Update-Pakete gemäß ihrer Kategorien automatisch zur Ermittlung und/oder zur Installation genehmigt werden. Per Default ist der WSUS-Server so eingestellt, dass alle Update-Pakete der Kategorien *Wichtige Updates* und *Sicherheitsupdates* automatisch für die Ermittlung genehmigt werden, jedoch nicht für die Installation.

Um Updates von Hand zu genehmigen, rufen Sie die Verwaltungskonsole des WSUS-Servers aus dem STARTMENÜ (Untermenü VERWALTUNG) auf und klicken auf UPDATES. Sie erhalten nun eine Liste der Updates und können der Spalte GENEHMIGUNG den Genehmigungsstatus entnehmen. Markieren Sie die noch nicht genehmigten Updates (eine Mehrfachauswahl mit gedrückter -Taste ist hierbei möglich), und wählen Sie in der linken Fensterhälfte (Feld GENEHMIGUNG) die entsprechende Genehmigungseinstellung aus.

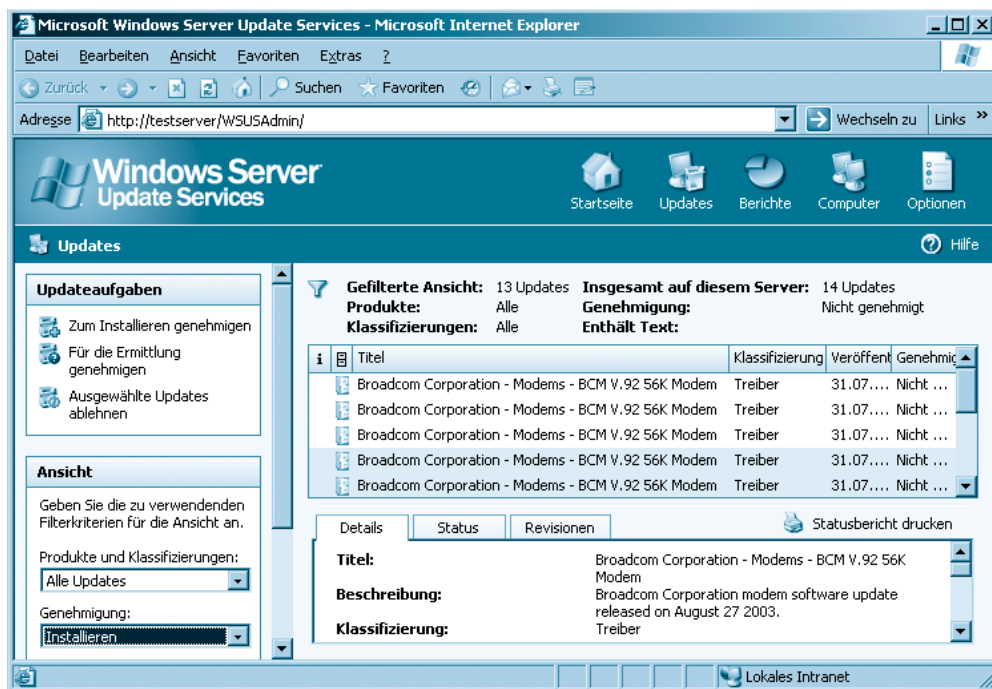


Abbildung 22: Die manuelle Genehmigung

Um entsprechende Regeln zu erstellen, womit neue Updates automatisch zur Ermittlung und wahlweise auch zur Installation genehmigt werden können, klicken Sie auf **OPTIONEN** und anschließend auf **OPTIONEN FÜR AUTOMATISCHE GENEHMIGUNGEN**. Sie können nun durch einen Klick auf die Schaltfläche **KLASSIFIZIERUNGEN HINZUFÜGEN/ENTFERNEN** und die anschließende Auswahl der jeweiligen Kategorien bestimmen, welche Update-Kategorien automatisch zur Ermittlung und zur Installation genehmigt werden sollen.

Einrichten der WSUS-Clients

Abschließend gilt es nun, auf den zu aktualisierenden Computersystemen (den WSUS-Clients) die **AUTOMATISCHE UPDATES**-Funktion zu aktivieren und die Einstellung der Funktion dahingehend zu ändern, dass die Updates nicht mehr über eine Internetverbindung von dem Microsoft-eigenen Update-Server bezogen werden, sondern vom internen WSUS-Server. Diese Einstellung lässt sich nun auf zwei unterschiedlichen Wegen vornehmen:

- ▶ **Per Richtlinie:** Sie können die Einstellungen zur **AUTOMATISCHEN UPDATES**-Funktion und insbesondere auch die Verbindung zum WSUS-Server mit Hilfe einer Richtlinie konfigurieren.
- ▶ **Direkt in der Registrierung:** Sie können die Einstellungen aber auch direkt in der Registrierung vornehmen und die Einstellungen per Ex- und Import der Registrierungseinstellungen oder mit Hilfe einer Batch-Datei verteilen.

Client-Konfiguration per Richtlinie

Wenn Sie die Konfiguration per Richtlinie (lokal oder im Active Directory) vornehmen möchten, so rufen Sie den Gruppenrichtlinienobjekt-Editor auf und erweitern zunächst COMPUTER-KONFIGURATION, dann ADMINISTRATIVE VORLAGEN und anschließend WINDOWS-KOMPONENTEN. Unterhalb davon markieren Sie den Ordner WINDOWS UPDATE, woraufhin in der rechten Fensterhälfte die entsprechenden Richtlinien angezeigt werden.

Führen Sie einen Doppelklick auf die Richtlinie AUTOMATISCHE UPDATES KONFIGURIEREN aus, um in die eigentlichen Einstellungen zu gelangen, und wählen Sie die gewünschte Option (siehe Automatische Updates-Funktion verwenden auf Seite 62) aus. Öffnen Sie anschließend die Richtlinie INTERNEN PFAD FÜR DEN MICROSOFT UPDATEDIENST ANGEBEN, die sich im gleichen Ordner befindet, und hinterlegen Sie die URL des WSUS-Servers (z.B. <http://Server/WSUSAdmin>) wie auch die URL für den Statistikserver (die URL ist in der Regel die gleiche wie die des WSUS-Servers). Führen Sie anschließend einen Computerneustart durch, damit die computerspezifischen Richtlinien angezogen und die Updates damit nun nicht mehr vom Microsoft-eigenen Update-Server bezogen werden, sondern vom internen WSUS-Server. Um die neuen Einstellungen dagegen ohne einen Computerneustart zu übernehmen, können Sie mit dem Befehl `GPUpdate /Force` das Aktualisieren der Gruppenrichtlinien erzwingen.

Client-Konfiguration per Registrierung

Um die Einstellungen der AUTOMATISCHE UPDATES-Funktion und auch die des WSUS-Servers direkt in der Registrierung vorzunehmen, konfigurieren Sie die lokalen Richtlinien (siehe vorheriger Abschnitt) gewissermaßen direkt »unter der Haube«. Das heißt, dass Sie die gleichen Einstellungen vornehmen wie auch per Richtlinie, nur dass Sie die Konfiguration nicht über den Gruppenrichtlinienobjekt-Editor vornehmen, sondern eben direkt in die Registrierung schreiben.

Um die Funktion AUTOMATISCHE UPDATES zu aktivieren, erzeugen Sie im Schlüssel `HKLM\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\WINDOWSUPDATE\AU` einen neuen DWORD-Wert mit der Bezeichnung *AUOptions*, dem Sie den entsprechenden Wert (siehe nachfolgende Tabelle) zuweisen.

Wert	Beschreibung
2	Vor Download und Installation benachrichtigen
3	Automatisch downloaden, aber vor Installation benachrichtigen
4	Automatisch downloaden und laut Zeitplan installieren
5	Lokalem Administrator ermöglichen, Einstellungen zu verändern

Tabelle 6: Die Werte des DWORD-Wertes »AUOptions«

Des Weiteren legen Sie (im gleichen Schlüssel) einen neuen DWORD-Wert mit der Bezeichnung *NoAutoUpdate* an, dem Sie den Wert 0 (0 = Aktivieren, 1 = Deaktivieren) zuweisen.

74 >> Welche Updates und Sicherheitsupdates sind bereits installiert?

Um die Einstellungen des WSUS-Servers zu hinterlegen, legen Sie zudem im Schlüssel `HKLM\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\WINDOWSUPDATE` eine neue Zeichenfolge mit der Bezeichnung `WUServer` an, der Sie die URL des WSUS-Servers (z.B. `http://server`) als Wert zuweisen. Den gleichen Wert weisen Sie zudem der Zeichenfolge `WUStatusServer`, die in dem gleichen Schlüssel abgelegt wird, zu.

Das nachfolgende Beispiel aktiviert die AUTOMATISCHE UPDATES-Funktion mit der Option `VOR DOWNLOAD UND INSTALLATION BENACHRICHTIGEN` und setzt die URL für den WSUS-Server und auch für den WSUS-Status-Server auf `http://MyServer`.

```
@Echo off
Reg Add "HKLM\SOFTWARE\Policies\Microsoft\Windows\
WindowsUpdate" /v WUServer /t REG_SZ /d http://MyServer
Reg Add "HKLM\SOFTWARE\Policies\Microsoft\Windows\
WindowsUpdate" /v WUStatusServer /t REG_SZ /d http://MyServer
Reg Add "HKLM\SOFTWARE\Policies\Microsoft\Windows\
WindowsUpdate\AU" /v AUOptions /t REG_DWORD /d 2
Reg Add "HKLM\SOFTWARE\Policies\Microsoft\Windows\
WindowsUpdate\AU" /v NoAutoUpdate /t REG_DWORD /d 0
```

Listing 10: 1_02_005.bat

Welche Updates und Sicherheitsupdates sind bereits installiert?

Wenn Sie herausfinden müssen, welche Updates und Sicherheitsupdates bereits auf einem bestimmten Computersystem installiert worden sind, so können Sie zum einen das Systemsteuerungsobjekt `SOFTWARE` öffnen und sich die Liste der Updates anzeigen lassen. Sie können zum anderen aber auch einen Blick in die Registrierung werfen und die installierten Updates direkt in der Registrierung abfragen, da mit jeder Update-Installation ein entsprechender Eintrag in der Registrierung erzeugt wird.

Microsoft stellt auf seiner Internetseite (www.microsoft.com) zwar ein Tool mit der Bezeichnung `Qfecheck.exe` bereit, womit sich die installierten Updates sehr gut ermitteln lassen, allerdings gibt es zum jetzigen Zeitpunkt noch keine entsprechende Version, womit die Updates eines Windows Server 2003-Systems abgefragt werden können. Die verfügbaren `Qfecheck`-Versionen sind bis dato nur für Windows 95/ME/2000/XP verfügbar und können leider nicht auf einem Windows Server 2003-System eingesetzt werden.

Installierte Updates per Systemsteuerung ermitteln

Für den Fall, dass Sie gerade an Windows Server 2003 angemeldet sind oder zumindest eine Remotedesktop-Verbindung mit einem Serversystem hergestellt haben und nun wissen möchten, welche Updates und Sicherheitsupdates bereits installiert worden sind, so wechseln Sie einfach in die `SYSTEMSTEUERUNG` und öffnen dort das Objekt `SOFTWARE`. Anschließend aktivieren Sie im oberen Bereich des Dialogfensters `SOFT-`

WARE die Option UPDATES ANZEIGEN. Unterhalb von WINDOWS SERVER 2003 – SOFTWAREUPDATES finden Sie nun eine Aufstellung aller Updates, die bisher eingerichtet wurden (siehe Abbildung 23).

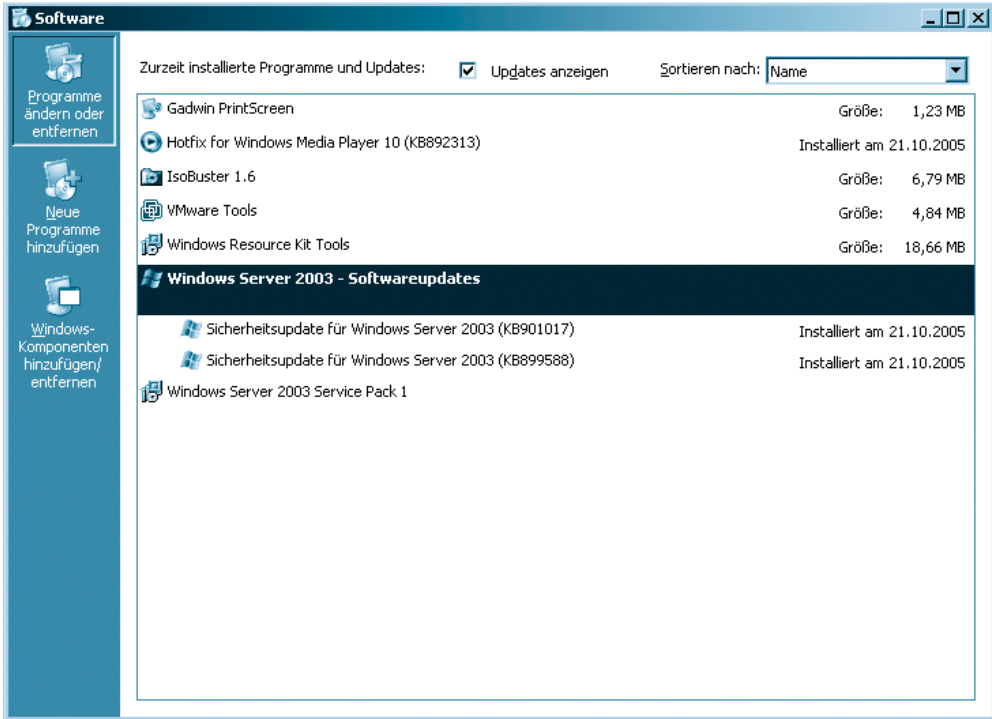


Abbildung 23: Die installierten Updates anzeigen lassen

Installierte Updates per Registrierung abfragen

Wenn Sie ein Update oder Sicherheitsupdate installieren, so wird in der Registrierung automatisch ein neuer Schlüssel erstellt, der eine ganze Reihe wichtiger Informationen (KB-Nummer, Installationsdatum, betroffene Dateien etc.) zu dem entsprechenden Update enthält.

Um herauszufinden, welche Updates auf einem bestimmten Computersystem bereits eingerichtet wurden, so öffnen Sie den Registrierungs-Editor (*regedit.exe*) und erweitern den Schlüssel HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\UPDATES\WINDOWS SERVER 2003. Darunter befinden sich ein oder auch mehrere Schlüssel mit der Bezeichnung SP1, SP2 ..., die jeweils die in der Service-Pack-Nummer enthaltenen Einzel-Updates beinhalten.

Haben Sie beispielsweise das Service Pack 1 installiert und auch einige neuere Einzel-Updates, so finden Sie den Schlüssel SP1 und SP2 vor, wobei die neueren Einzel-Updates im Schlüssel SP2 zu finden sind, da sie im nächsten Service Pack (Service Pack 2) bereits enthalten sein werden. Wenn Sie nun den Schlüssel SP2 erweitern, so

finden Sie für jedes neuere Einzel-Update, das Sie installiert haben, einen untergeordneten Schlüssel vor, der die jeweilige KB-Nummer als Bezeichnung enthält. Anhand dieser Schlüssel können Sie nun gut erkennen, welche Updates nach der Einrichtung von Service Pack 1 installiert wurden.

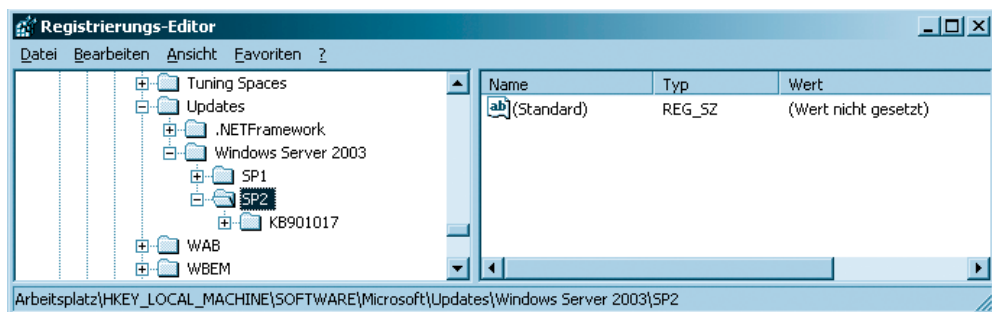


Abbildung 24: Die Anzeige der installierten Updates in der Registrierung

Die Abfrage der installierten Updates lässt sich mit dem `Reg Query`-Befehl natürlich auch von der Eingabeaufforderung aus durchführen und noch dazu automatisieren. Wenn Sie beispielsweise mehrere Computersysteme mit Windows Server 2003 und Service Pack 1 installiert und im Anschluss unterschiedliche Einzel-Updates eingerichtet haben, so können Sie mit der folgenden `Reg Query`-Befehlszeile den Inhalt von `HKLM\SOFTWARE\MICROSOFT\UPDATES\WINDOWS SERVER 2003\SP2` abfragen und das Ergebnis in die Datei `Result.txt` schreiben. Anhand der Schlüsselbezeichnungen können Sie dann erkennen, welche Updates bereits eingerichtet wurden.

```
@Echo off
Reg Query "HKLM\SOFTWARE\Microsoft\Updates\Windows Server 2003\SP2" _
> Result.txt
```

Listing 11: 1_02_006.bat

Im Notfall: Updates und Sicherheitsupdates wieder deinstallieren

Die Updates und Sicherheitsupdates, die auf der Internetseite www.microsoft.com zum Download angeboten werden, dienen natürlich dem Beheben bekannter Fehler und auch dem Schließen potenzieller Sicherheitslücken. Nichtsdestotrotz kann es aber einmal vorkommen, dass sich die Installation eines Updates negativ auf die Systemstabilität auswirkt oder dass es zu Problemen bei der Ausführung von Diensten oder Applikationen kommt.

Wenn derartige Probleme bekannt und unter www.microsoft.com publiziert werden, so stellt Microsoft in der Regel weitere Updates bereit, womit die Problematiken, die sich durch ein bestimmtes Update ergeben, wieder behoben werden können. Bis zur

Veröffentlichung eines neuen Updates kann es nun erforderlich sein, das fehlerhafte Update erst einmal zu deinstallieren, um die Stabilität des Serversystems kurzfristig wieder herzustellen.

Achtung

Wenn Sie ein Update installieren, so werden die betroffenen Dateien durch neuere Dateiversionen ersetzt. Die ursprünglichen Dateiversionen werden dabei aber nicht unwiderruflich überschrieben, sondern während der Einrichtung des Updates in den Ordner `%WinDir%\$NTUninstallKBxxxxxx$` (die x-Zeichen sind durch die jeweilige KB-Nummer zu ersetzen) kopiert, damit sie bei Bedarf wieder restauriert werden können. Dies ist jedoch nur der Fall, wenn Sie das Update nachträglich installieren und das betreffende Update nicht in eine Slipstream-Version (siehe *Slipstream-Versionen erzeugen* in der Kategorie *Das Betriebssystem-Update*) integriert haben. Denn bei Verwendung einer Slipstream-Version werden die aktualisierten Dateien mit der Installation des Betriebssystems eingerichtet. Updates, die in einer Slipstream-Version enthalten sind, lassen sich daher nicht mehr so ohne weiteres deinstallieren.

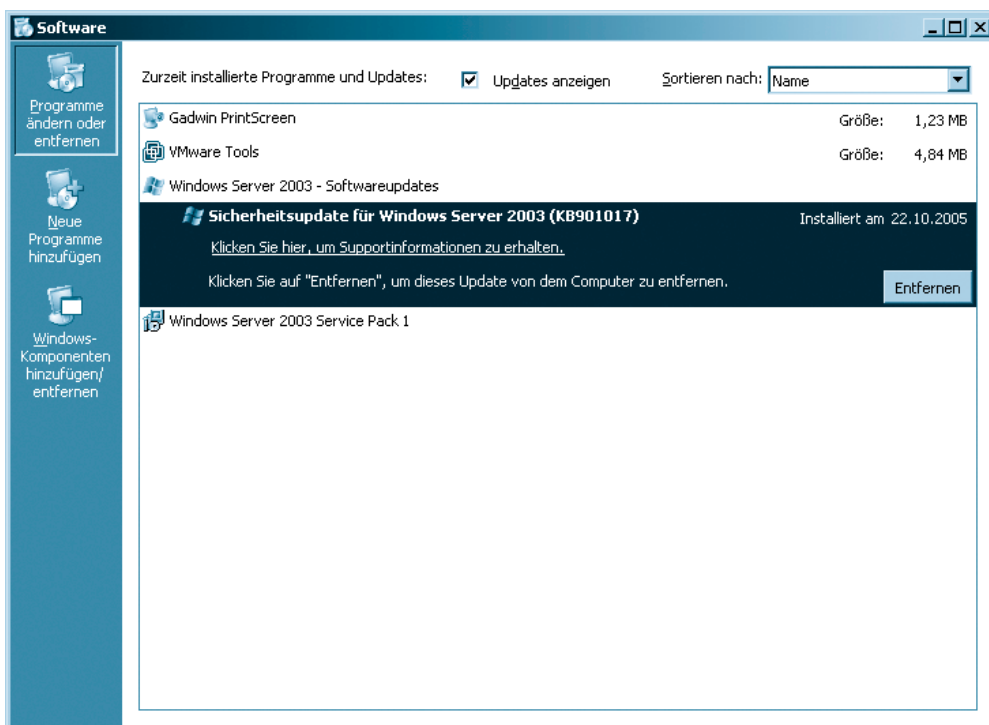
Betriebssystem-Update

Abbildung 25: Ein Security-Update wieder deinstallieren

Um ein Update oder Sicherheitsupdate zu deinstallieren, rufen Sie im einfachsten Fall das Systemsteuerungsobjekt SOFTWARE auf, aktivieren im oberen Bereich des Dialogfensters die Option UPDATES ANZEIGEN, wählen dann das betreffende Update aus und

klicken auf ENTFERNEN. Das Update wird daraufhin entfernt, in dem die vorherigen Dateiversionen aus dem Ordner `C:\Windows\${NTUninstallKBxxxxr}$` restauriert werden. Starten Sie das Computersystem dann neu, damit die wiederhergestellten Dateiversionen vom Betriebssystem verwendet werden können.

Ein nachträglich installiertes Update (das nicht Bestandteil einer Slipstream-Version ist) lässt sich natürlich auch von der Eingabeaufforderung aus wieder deinstallieren. Hierzu wechseln Sie lediglich in den Ordner `%Windir%\${NTUninstallKBxxxxr}$\Spuninst` und rufen das Tool *Spuninst.exe* auf. Mit dem Parameter `/Quiet` können Sie die Deinstallation des Updates automatisch und damit ohne jegliche Interaktion durchführen und mit den Parametern `/ForceRestart` und `/NoRestart` den anschließenden Computerneustart erzwingen oder wahlweise auch unterdrücken.

Das nachfolgende Batch-Datei-Beispiel zeigt, wie Sie das Update *KB901017* von der Eingabeaufforderung aus ohne Interaktion deinstallieren und nach dem Entfernen des Updates einen automatischen Computerneustart durchführen können.

```
@Echo off
%Windir%\${NTUninstallKB901017}$\Spuninst\Spuninst /Quiet /ForceRestart
```

Listing 12: 1_02_007.bat

Der Einsatz des Tools *Spuninst.exe* eignet sich insbesondere dann, wenn Sie ein fehlerhaftes Update auf mehreren Computersystemen eingerichtet haben und die Deinstallation ein Stück weit vereinfachen möchten. Denn Sie können sich beispielsweise mit der administrativen Freigabe C\$ der betreffenden Computersysteme verbinden und die *Spuninst*-Befehlszeile damit auch remote aufrufen. Oder Sie erstellen eine Batch-Datei, nehmen darin die entsprechende *Spuninst*-Befehlszeile auf und starten die Deinstallation des Updates dann per Batch-Datei-Aufruf. Letzteres lässt sich durch Einsatz des `AT`-Befehls oder mit `GEPLANTE TASKS` auch zeitlich steuern und damit beispielsweise in die Nachtstunden verlegen, womit der erforderliche Computerneustart dann während der Nacht durchgeführt wird.

Eingabeaufforderung

Die Eingabeaufforderung (engl. = *Command Prompt*) von Windows Server 2003 stellt ein sehr nützliches Werkzeug für die schnelle Konfiguration vieler Betriebssystem-Einstellungen, für die alltägliche Systemadministration und sogar für Überwachungs- und Troubleshooting-Zwecke dar und gewinnt aufgrund des umfangreichen Befehlsvorrats gerade in der Systemadministration einer Windows Server 2003-Umgebung zunehmend an Bedeutung. Denn durch die direkte Eingabe der Befehle lassen sich viele Aufgaben sehr schnell ausführen und zudem auch sehr gut automatisieren.

Letzteres wird durch den Einsatz so genannter Batch-Dateien (zu Deutsch: Stapelverarbeitungsdateien) ermöglicht, womit Sie mehrere Befehlszeilen inklusive der erforderlichen Parameter in einer gewöhnlichen Textdatei mit der Endung **.bat* oder **.cmd* hinterlegen und die einzelnen Befehle dann automatisch ausführen lassen können. Die Batch-Dateien können Sie dabei von Hand ausführen, in der AUTOSTART-Gruppe oder im RUN-Schlüssel der Registrierung (siehe Kategorie *Registrierung*) hinterlegen und damit beim Starten von Windows Server 2003 oder nach der Anmeldung automatisch ausführen lassen oder mit Hilfe von GEPLANTE TASKS (siehe Kategorie *Automatisierung*) auch zu einem bestimmten Termin oder bei Auftreten eines bestimmten Ereignisses zum automatischen Start bewegen.

Der Einsatz der Eingabeaufforderung hat sich gerade im Bereich der Final-Konfiguration im Anschluss an die Betriebssysteminstallation sehr gut bewährt, da sich mit Hilfe entsprechender Befehlszeilen bzw. durch Verwendung von Batch-Dateien die grundlegende Konfiguration des Betriebssystems sehr gut automatisieren lässt, ohne dass hierfür tief greifende Programmierkenntnisse erforderlich sind.

Die Eingabeaufforderung im Überblick

Wenn von der *Eingabeaufforderung* die Rede ist, so werden Sie des Öfteren auch die Bezeichnungen *Shell*, *Command Shell*, *Kommandozeile*, *DOS-Fenster* oder auch *Command Prompt* hören, die jedoch nichts anderes als ein Synonym für *Eingabeaufforderung* sind und allesamt das gleiche Programm benennen: nämlich den Windows Server 2003-integrierten Befehlsinterpreter *Cmd.exe*.

Der Befehlsinterpreter *Cmd.exe* (von nun an nur noch *Eingabeaufforderung* genannt), der fester Bestandteil von Windows Server 2003 ist, ermöglicht die Eingabe von Befehlen und erlaubt so eine direkte Kommunikation zwischen dem Benutzer und dem Betriebssystem. Mit Hilfe der zur Verfügung stehenden Befehle können Informationen zum Betriebssystem, zur aktuellen Arbeitsumgebung und auch zu den installierten Applikationen abgefragt und natürlich auch Änderungen an der Systemkonfiguration vorgenommen werden. Damit lassen sich viele Aufgaben, für die unter grafischer Oberfläche zahlreiche und zudem noch manuelle Klicks erforderlich sind, weitaus schneller durchführen und obendrein noch automatisieren, da Befehle in einer Batch-Datei hin-

terlegt und automatisch ausgeführt werden können, ohne dass hierfür eine Interaktion mit dem Benutzer erforderlich wäre.

Neben dem Befehlsinterpreter *Cmd.exe* ist auch noch der aus früheren DOS-Zeiten bekannte Befehlsinterpreter *Command.com* vorhanden, wobei dieser Befehlsinterpreter aber nur noch aus Kompatibilitätsgründen mit älteren 16-Bit-Anwendungen noch Bestandteil des Betriebssystems ist. Wenn Sie *Command.com* eingeben, so wird automatisch eine virtuelle DOS-Machine (*NTVDM*) gestartet, worüber der Befehlsinterpreter *Command.com* geladen wird. Die Verarbeitung der Befehle erfolgt dabei aber nicht mit *Command.com*, sondern mit *Cmd.exe*.

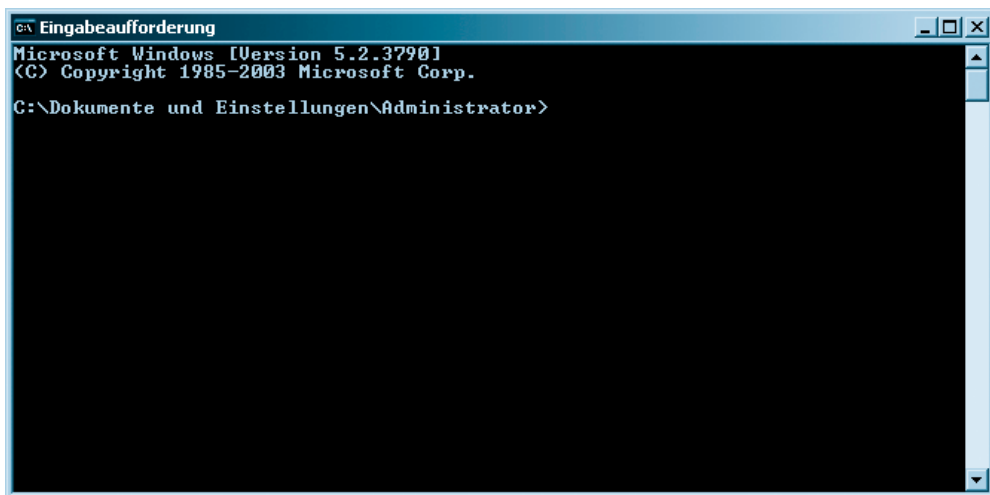


Abbildung 26: Die Eingabeaufforderung

Arbeiten mit der Eingabeaufforderung

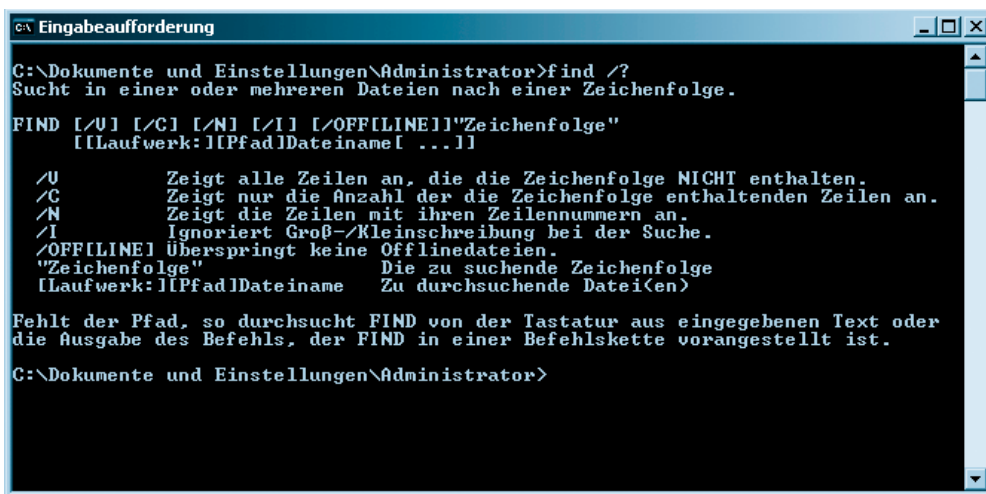
Die Eingabeaufforderung steht nach dem Bootvorgang von Windows Server 2003 bzw. nach der Anmeldung nicht unmittelbar zur Verfügung, sondern muss erst von Hand geöffnet werden. Sie können die Eingabeaufforderung aus dem STARTMENÜ heraus öffnen, indem Sie auf START klicken, dann ALLE PROGRAMME erweitern und im Menü ZUBEHÖR auf EINGABEAUFFORDERUNG klicken. Alternativ dazu können Sie die Eingabeaufforderung auch direkt aufrufen, indem Sie im STARTMENÜ auf AUSFÜHREN klicken und *Cmd.exe* eingeben.

Sie gelangen daraufhin in das Eingabeaufforderungsfenster und befinden sich per Default im Ordner *C:\Dokumente und Einstellungen\Benutzername*, wobei *Benutzername* durch den Anmeldenamen des gerade angemeldeten Benutzers zu ersetzen ist. Dies können Sie an der Systemmeldung, die links des Cursors eingeblendet ist, erkennen. Der blinkende Cursor selbst stellt die Eingabemöglichkeit für Befehle dar, Sie können also direkt nach dem Öffnen der Eingabeaufforderung mit dem Eingeben der Befehle beginnen. Welche Befehle zur Verfügung stehen, können Sie sich durch Eingabe des Befehls *Help* und einen anschließenden Druck auf die Enter-Taste anzeigen lassen.

Mit `Help` werden allerdings nur die internen Befehle angezeigt, die seitens des Befehlsinterpreters `Cmd.exe` bereitgestellt werden. Das Betriebssystem enthält darüber hinaus noch eine Vielzahl weiterer Befehle (die so genannten externen Befehle), die in Form einer `*.com`- oder `*.exe`-Datei im `Windows\System32`-Ordner abgelegt sind. Welche externen Befehle bereitgestellt werden, können Sie der Befehlszeilenreferenz, die Bestandteil der Windows-Hilfe ist, entnehmen.

Wenn Sie einen Befehl ausführen möchten, so geben Sie den entsprechenden Befehl ein und drücken anschließend die `[Enter]`-Taste. Hierbei spielt es keine Rolle, ob Sie den Befehl in Groß- oder Kleinbuchstaben eingeben, da die internen und externen Befehle von Windows Server 2003 nicht case-sensitiv sind. Das Kommando wird daraufhin verarbeitet, und das Ergebnis wird am Bildschirm ausgegeben. Die meisten internen und auch externen Befehle stellen dabei zahlreiche Parameter zur Verfügung, die Sie dem Befehl anfügen und mit denen Sie die Anweisung genauer spezifizieren können. Um herauszufinden, welche Parameter und zusätzlichen Optionen ein Befehl akzeptiert, führen Sie den entsprechenden Befehl mit dem Parameter `/?` aus, woraufhin in den meisten Fällen ein kurzer Hilfetext zu dem jeweiligen Befehl angezeigt wird. Der Hilfetext enthält dabei meist auch eine Übersicht der möglichen Parameter unter Angabe einer kurzen Beschreibung.

Die nachfolgende Abbildung zeigt ein solches Beispiel. Mit dem Befehl `Find /?` wird der Hilfetext zum `Find`-Befehl aufgerufen. Als Ergebnis wird eine kurze Beschreibung zum Befehl selbst, wie auch zu dessen Parameter, ausgegeben.



```
C:\Dokumente und Einstellungen\Administrator>find /?
Sucht in einer oder mehreren Dateien nach einer Zeichenfolge.

FIND [/U] [/C] [/N] [/L] [/OFF[LINE]] "Zeichenfolge"
[[Laufwerk:]IPfad]Dateiname[ ...]

/U      Zeigt alle Zeilen an, die die Zeichenfolge NICHT enthalten.
/C      Zeigt nur die Anzahl der die Zeichenfolge enthaltenden Zeilen an.
/N      Zeigt die Zeilen mit ihren Zeilennummern an.
/L      Ignoriert Groß-/Kleinschreibung bei der Suche.
/OFF[LINE] Überspringt keine Offlinedateien.
"Zeichenfolge" Die zu suchende Zeichenfolge
[[Laufwerk:]IPfad]Dateiname Zu durchsuchende Datei(en)

Fehlt der Pfad, so durchsucht FIND von der Tastatur aus eingegebenen Text oder
die Ausgabe des Befehls, der FIND in einer Befehlskette vorangestellt ist.

C:\Dokumente und Einstellungen\Administrator>
```

Abbildung 27: Den Hilfetext für den »Find«-Befehl aufrufen

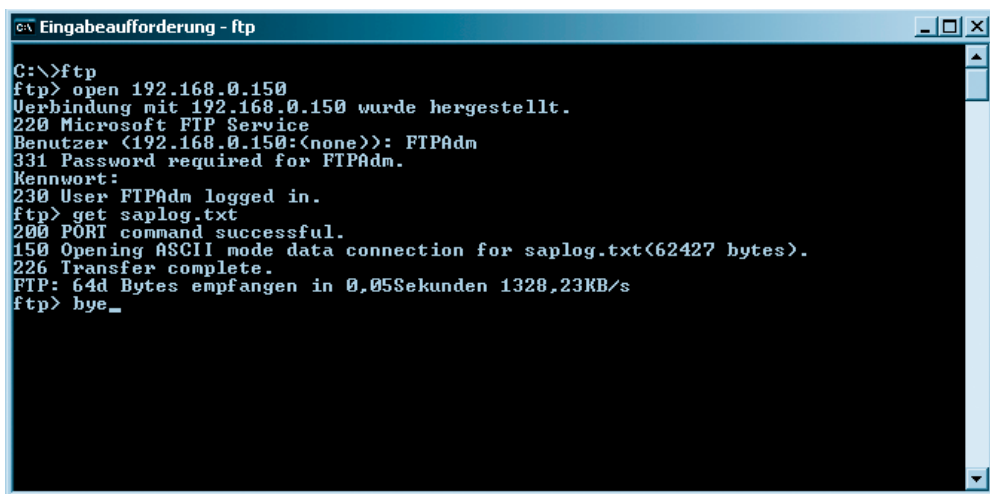
Wenn Sie die Eingabeaufforderung wieder beenden möchten, so geben Sie den Befehl `Exit` ein. Der Befehlsinterpreter wird daraufhin beendet, und das Fenster wird geschlossen.

Ein- und Ausgabeumleitung

Die Befehle, die Sie in der Eingabeaufforderung anwenden, geben Sie für gewöhnlich über die Tastatur ein. Und wenn nach der Befehlsverarbeitung Ergebnisse zurückgeliefert werden, so werden die Ergebnisse normalerweise am Bildschirm ausgegeben. Die Tastatur gilt daher als das Standard-Eingabegerät (*StdIn*), der Bildschirm dagegen als das Standard-Ausgabegerät (*StdOut*).

Mit Hilfe der Ein- und Ausgabeumleitung können Sie die einzugebenden Befehle aber auch aus einer anderen Eingabequelle einlesen und die Ausgabe ebenso an ein anderes Gerät senden. So können Sie beispielsweise in einer Textdatei ein oder auch mehrere Befehle hinterlegen und mit dem <-Zeichen den Befehlsinterpretier dazu veranlassen, die Befehle nicht über die Tastatur einzulesen, sondern aus einer Datei. Und für den Fall, dass die aus der Datei eingelesenen Befehle ein Ergebnis zurückliefern, so können Sie das Ergebnis mit dem >-Zeichen in eine Datei umleiten, an einen Drucker senden oder an das NUL-Device übermitteln. Letzteres führt dazu, dass überhaupt keine Ergebnisse ausgegeben werden, da das NUL-Device alle Informationen förmlich »schluckt«.

Bestes Beispiel für den Einsatz einer Eingabeumleitung stellt das FTP-Skripting dar. Denn wenn Sie z.B. von der Eingabeaufforderung aus einen FTP-Transfer starten, so wechseln Sie in der Regel durch Eingabe von FTP zunächst in die FTP-eigene Eingabeaufforderung und stellen dann mit Hilfe der FTP-spezifischen Befehle (open, user, pwd, get, put, bye ...) die Verbindung her, führen die Anmeldung durch, übertragen dann die entsprechenden Daten und schließen die FTP-Sitzung wieder. Die nachfolgende Abbildung zeigt eine solche FTP-Sitzung via Eingabeaufforderung, wie sie in vielen Fällen durchgeführt wird.



```
C:\>ftp
ftp> open 192.168.0.150
Verbindung mit 192.168.0.150 wurde hergestellt.
220 Microsoft FTP Service
Benutzer (192.168.0.150:(none)): FTPAdm
331 Password required for FTPAdm.
Kennwort:
230 User FTPAdm logged in.
ftp> get saplog.txt
200 PORT command successful.
150 Opening ASCII mode data connection for saplog.txt(62427 bytes).
226 Transfer complete.
FTP: 64d Bytes empfangen in 0,05Sekunden 1328,23KB/s
ftp> bye_
```

Abbildung 28: Eine klassische FTP-Sitzung via Eingabeaufforderung

Statt die FTP-spezifischen Befehle nun interaktiv einzugeben, können Sie die Befehle auch in eine Textdatei aufnehmen und den FTP-Befehl dazu veranlassen, die FTP-spezifischen Kommandos aus der Textdatei zu lesen. Ein FTP-Skript, womit Sie eine Verbindung zum Host *192.168.0.150* herstellen, sich mit *FTPAdm* anmelden und dann die Datei *SAPLog.txt* aus dem Root-Verzeichnis auf das lokale Computersystem herunterladen können, sieht in etwa so aus:

```
Open 192.168.0.150
User FTPAdm 123QWEasd
Get SAPLog.txt
Bye
```

Listing 13: *FtpSkript.txt*

Anschließend wenden Sie in der Eingabeaufforderung den FTP-Befehl mit dem Parameter *-n* (dieser Parameter deaktiviert die automatische FTP-Anmeldung) an, aktivieren mit dem *<*-Zeichen die Eingabeumleitung und übergeben den Namen der Textdatei, in der die FTP-spezifischen Befehle hinterlegt sind. Die FTP-spezifischen Befehle werden daraufhin zeilenweise aus der Datei *FtpSkript.txt* gelesen und entsprechend verarbeitet. Hierbei ist natürlich darauf zu achten, dass am Ende des FTP-Skriptes der FTP-spezifische Befehl *Quit* oder *Bye* stehen muss, da die FTP-Sitzung sonst nicht automatisch beendet werden kann.

```
@Echo off
Ftp -n < FtpSkript.txt
```

Listing 14: *1_03_002a.bat*

Auf ähnliche Art und Weise funktioniert auch die Ausgabeumleitung. Wenn Sie beispielsweise in der Eingabeaufforderung den Befehl *Dir C:\Windows*.** eingeben, so werden alle Dateien und Ordner, die in *C:\Windows* enthalten sind, am Bildschirm angezeigt. Mit der Ausgabeumleitung können Sie die Ausgabe aber auch in eine Datei schreiben lassen. Hierzu wenden Sie den zuvor genannten Befehl an, aktivieren mit dem *>*-Zeichen die Ausgabeumleitung und fügen die Datei, die das Ergebnis später enthalten soll, als Parameter an.

```
@Echo off
Dir C:\Windows\*. * > Result.txt
```

Listing 15: *1_03_002b.bat*

Die Ausgabedatei muss dabei noch nicht existieren, da sie – wenn sie noch nicht vorhanden ist – automatisch angelegt wird. Für den Fall, dass Sie eine Befehlszeile mit einer Ausgabeumleitung mehrmals anwenden, so wird die Ausgabedatei mit jedem Befehlsaufruf neu angelegt bzw. die vorherige Version überschrieben. Wenn Sie die

Ausgabe also nicht in eine neue Datei schreiben möchten, sondern den Inhalt einer vorhandenen Datei um weitere Ausgaben ergänzen möchten, so müssen geben Sie das >-Zeichen doppelt an. Damit wird die angegebene Ausgabedatei nicht überschrieben, sondern ergänzt.

```
@Echo off
Dir C:\Windows\*. * >> Result.txt
```

Listing 16: 1_03_002c.bat

Nach der Ausführung eines Befehls wird der Umgebungsvariablen `ErrorLevel` ein so genannter Rückgabewert zugewiesen, der Auskunft darüber gibt, ob die letzte Befehlsausführung erfolgreich war oder fehlgeschlagen ist. Die meisten Befehle liefern jeweils den Rückgabewert 0 zurück, wenn der Befehl erfolgreich ausgeführt wurde, anderenfalls ist der Wert gleich 1. Wenn Sie nun nach einer Befehlsausführung ausschließlich den Rückgabewert überprüfen möchten, ohne dass das Ergebnis der Befehlsausführung am Bildschirm angezeigt wird, so können Sie die Ausgabeumleitung einsetzen, dabei das Ergebnis nach `NUL` umleiten und lediglich den Wert von `ErrorLevel` abfragen.

Das nachfolgende Beispiel leitet die Ausgabe aller Dateien und Ordner, die im Ordner `C:\Windows` enthalten sind, nach `NUL` um (und unterdrückt damit die vollständige Ausgabe) und gibt lediglich den Wert von `ErrorLevel` am Bildschirm aus. Dieser ist gleich 0, wenn der Ordner-Inhalt erfolgreich abgefragt werden konnte, anderenfalls ist der Wert 1.

```
@Echo off
Dir C:\Windows\*. * > NUL
Echo %ErrorLevel%
```

Listing 17: 1_03_002d.bat

Batch-Dateien erzeugen

Bei einer Batch-Datei handelt es sich um eine gewöhnliche Textdatei mit der Dateinamen-Erweiterung `*.bat` oder `*.cmd`, worin einzelne Befehle und auch vollständige Befehlszeilen hinterlegt werden können. Mit dem Aufruf einer solchen Batch-Datei liest der Befehlsinterpreter den Inhalt der Datei dann automatisch zeilenweise ein und führt die darin enthaltenen Befehle und Befehlszeilen in chronologischer Reihenfolge aus, ohne dass für die Ausführung der jeweiligen Kommandos eine Interaktion mit dem Benutzer erforderlich ist. Wenn das Ende der Datei erreicht ist und keine weiteren Kommandos mehr ausgeführt werden müssen, so wird die Batch-Datei automatisch geschlossen, und der Befehlsinterpreter wird beendet.

Batch-Dateien stellen somit eine sehr komfortable Möglichkeit der Aufgabenautomatisierung dar, da Sie die zur Ausführung einer Aufgabe erforderlichen Befehle und

Befehlszeilen der Reihe nach in der Datei aufführen und anschließend automatisch ausführen lassen können. Und da Batch-Dateien vom Betriebssystem als ausführbare Dateien behandelt werden, können Sie sogar den Aufruf einer Batch-Datei automatisieren, indem Sie die Batch-Datei mit einem bestimmten Systemereignis verknüpfen oder schlichtweg nur den Ausführungszeitpunkt festlegen.

Zum Erstellen einer Batch-Datei benötigen Sie nichts weiter als einen gewöhnlichen Texteditor. Sie können beispielsweise die Windows Server 2003-eigenen Editoren *Notepad.exe* und *Wordpad.exe* einsetzen oder auch auf alternative Editoren wie *Lemmy*, *WinVi* und viele weitere mehr zurückgreifen. Beim Aufbau einer Batch-Datei gilt zu beachten, dass jeder auszuführende Befehl und jede vollständige Befehlszeile in einer separaten Zeile stehen muss, die jeweils mit der `[Enter]`-Taste abgeschlossen wird. Sie können natürlich Leerzeilen verwenden, um das Listing etwas übersichtlicher zu gestalten und »Funktionsblöcke« zu gruppieren. Des Weiteren können Sie die Batch-Datei mit Kommentaren versehen, um das Listing zu dokumentieren. Eine Kommentarzeile wird hierbei mit der Zeichenfolge `REM` (für *Remark* = Bemerkung) oder alternativ mit `::` eingeleitet. Alle Zeichen, die hinter `REM` oder `::` stehen, werden vom Befehlsinterpreter ignoriert und damit nicht in die Befehlsverarbeitung mit einbezogen.

Des Weiteren können Sie Sprungmarken verwenden, um den Ablauf beim Ausführen der Befehle und Befehlszeilen zu steuern. Eine Sprungmarke wird mit einem vorangestellten Doppelpunkt (z.B. `:Sprungmarke1`) deklariert und kann mit der `Goto`-Anweisung angesteuert werden.

Nachfolgend finden Sie ein Beispiel für eine Batch-Datei, in der die zuvor angesprochene Syntax verwendet wird. Die Hauptfunktion der Batch-Datei liegt darin, mit dem `CacIs`-Befehl die Berechtigungen aller Dateien und Ordner, die im Root des Laufwerks `C:` abgelegt sind, abzufragen und per Ausgabeumleitung in die Datei `C:\Result.txt` zu schreiben. Nach der Ausführung des `CacIs`-Befehls wird der Rückgabewert geprüft. Ist dieser gleich 0, so wurde der Befehl fehlerfrei ausgeführt. In diesem Fall wird zur Sprungmarke `:Success` gewechselt, worüber die Meldung Die Ausführung des `CacIs`-Befehls war erfolgreich ... ausgegeben wird. Anschließend wird zur Sprungmarke `:Ende` gewechselt, um die Batch-Datei zu beenden und die Sprungmarke `:Failure` zu übergehen (ohne diesen Sprung würde der Befehlsinterpreter automatisch die Befehle unterhalb von `:Failure` verarbeiten).

Wenn der `CacIs`-Befehl jedoch nicht erfolgreich ausgeführt werden konnte, so ist der Rückgabewert gleich 1. In diesem Fall wird automatisch zur Sprungmarke `:Failure` gewechselt, und es wird die Meldung Die Ausführung des `CacIs`-Befehls war fehlerhaft ... ausgegeben.

```
::  
:: MyBatch.bat  
:: 22.11.2005 / 14:30
```

Listing 18: 1_03_003a.bat

```

::

@Echo off
Cls
CacIs C:\*.* > C:\Result.txt
If ErrorLevel 0 Goto Success Else Goto Failure
Goto Ende

:Success
Echo Die Ausfuehrung des CacIs-Befehls war erfolgreich ...
Goto Ende

:Failure
Echo Die Ausfuehrung des CacIs-Befehls war fehlerhaft ...

:Ende
Echo ...Ende...

```

Listing 18: 1_03_003a.bat (Forts.)

Batch-Dateien werden normalerweise dazu benutzt, um Aufgaben zu automatisieren und bestimmte Aktionen im Hintergrund ablaufen zu lassen. Es gibt jedoch eine ganze Reihe von Situationen, die während der Ausführung einer Batch-Datei eine Interaktion mit dem Benutzer erfordern. Denken Sie beispielsweise an ein Menü, worüber verschiedene Optionen zur Auswahl angeboten werden und der Benutzer zwischen den einzelnen Optionen wählen muss. Je nach gewählter Option wird dann zur entsprechenden Sprungmarke gewechselt oder eine separate Batch-Datei aufgerufen, worüber die entsprechenden Befehle dann aufgerufen werden.

Um nun einen Benutzer während der Ausführung einer Batch-Datei zur Eingabe bestimmter Zeichen aufzufordern und diese als Entscheidungskriterium für die Ablaufsteuerung verwenden zu können, stellt Windows Server 2003 den Befehl *Choice.exe* bereit, worüber Sie die Tastatureingaben abfragen und in die Verarbeitung mit einbeziehen können. Der *Choice*-Befehl akzeptiert standardmäßig nur die Eingabe der Zeichen J und N, da grundsätzlich von einer Ja-/Nein-Abfrage ausgegangen wird. Mit dem Parameter /C können Sie jedoch auch die Eingabe anderer Zeichen ermöglichen. Um die Eingabe des Benutzers, sprich: dessen Auswahl, anschließend auswerten zu können, nutzen Sie die Umgebungsvariable *ErrorLevel*. Dieser Variablen wird mit der Auswahl des ersten per /C übergebenen Zeichens der Wert 1, dem zweiten der Wert 2 usw. zugewiesen. Je nachdem, welcher Wert der Variablen *ErrorLevel* zugewiesen wurde, können Sie also ganz leicht herausfinden, welche Taste der Benutzer gedrückt hat.

Das nachfolgende Listing soll den Einsatz des *Choice*-Befehls verdeutlichen. Mit dieser Batch-Datei wird ein Menü eingeblendet, worüber der Benutzer zwischen den

möglichen Servertypen *Standalone-Server*, *Member-Server* und *Domänen-Controller* auswählen kann und das Menü auch beenden darf, ohne eine Auswahl vorzunehmen.

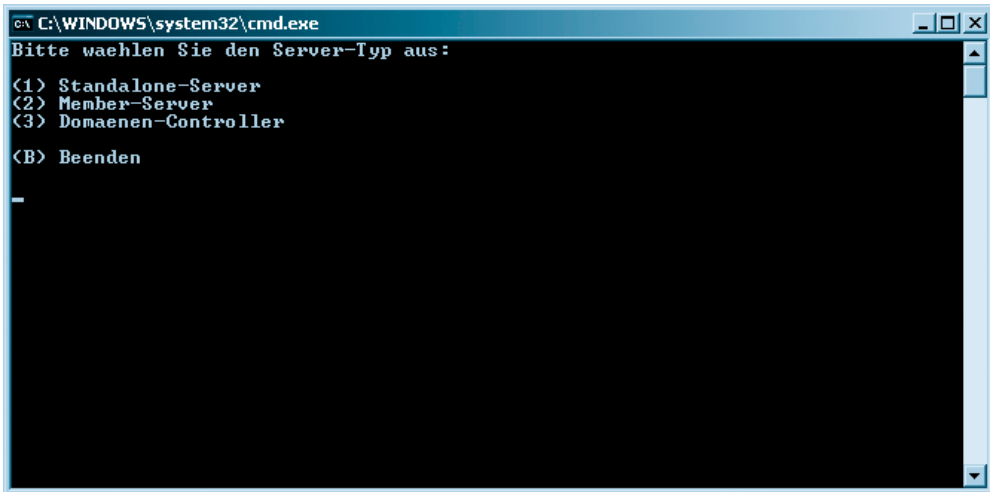


Abbildung 29: Das Auswahlmenü aus Listing 1_03_003b.bat

Hierzu wird im ersten Schritt mit `@Echo off` die Befehlsanzeige ausgeschaltet, um unnötige Bildschirmmeldungen zu unterdrücken. Des Weiteren wird mit `CLS` der Inhalt des Eingabeaufforderungsfensters gelöscht, damit das Menü übersichtlich dargestellt werden kann. Dann folgen die einzelnen Echo-Anweisungen, worüber das Menü dargestellt wird. Im vorletzten Schritt wird der `Choice`-Befehl eingesetzt, womit der Benutzer zur Eingabe der Zeichen 1, 2, 3 oder B und damit zur Auswahl des entsprechenden Menü-Eintrages aufgefordert wird. Und zu guter Letzt wird mit der `IF`-Anweisung der Wert der Umgebungsvariablen `ErrorLevel` ermittelt. Je nach Auswahl wird zur entsprechenden Sprungmarke gewechselt, der Bildschirminhalt gelöscht und die getroffene Auswahl angezeigt. Statt die Meldung Menü-Eintrag xyz ausgewählt auszugeben, können Sie natürlich weitere Befehle hinterlegen oder auch separate Batch-Dateien aufrufen, um beispielsweise die automatisierte Konfiguration des jeweiligen Servertyps durchzuführen.

```
@Echo off
CLS
Echo Bitte waehlen Sie den Servertyp aus:
Echo.
Echo (1) Standalone-Server
Echo (2) Member-Server
Echo (3) Domaenen-Controller
Echo.
Echo (B) Beenden
```

Listing 19: 1_03_003b.bat

```

Echo.
Choice /C 123B /N
If ErrorLevel 1 Goto Auswahl1
If ErrorLevel 2 Goto Auswahl2
If ErrorLevel 3 Goto Auswahl3
Goto Ende

:Auswahl1
Cls
Echo Menue-Eintrag "Standalone-Server" ausgewaehlt
Goto Ende

:Auswahl2
Cls
Echo Menü-Eintrag "Member-Server" ausgewaehlt
Goto Ende

:Auswahl3
Cls
Echo Menue-Eintrag "Domaenen-Controller" ausgewaehlt

:Ende

```

Listing 19: 1_03_003b.bat (Forts.)

Konfiguration der Eingabeaufforderung

Die Eingabeaufforderung verfügt über eine Reihe von Eigenschaften (z.B. Anzeigeoptionen, Cursorgröße, Größe des Befehlsspeichers, Schriftart- und Layout-Optionen ...), die Sie zum Teil schon mit dem Aufruf der Datei *Cmd.exe* via Parameter-Übergabe setzen können, die Sie aber der Einfachheit halber auch im Dialogfenster EIGENSCHAFTEN DER EINGABEAUFFORDERUNG wahlweise für das aktuelle und auch für alle künftigen Eingabeaufforderungsfenster konfigurieren können.

Um die Eigenschaften der Eingabeaufforderung festzulegen, öffnen Sie zunächst das Eingabeaufforderungsfenster und öffnen dann das Kontextmenü der Titelleiste. Wählen Sie im Kontextmenü den Eintrag STANDARDWERTE aus, wenn die bevorstehenden Änderungen für alle künftigen Eingabeaufforderungsfenster gelten sollen, oder klicken Sie stattdessen auf EIGENSCHAFTEN, um nur das aktuell geöffnete Fenster zu konfigurieren. Je nach Auswahl gelangen Sie dann in das Dialogfenster EIGENSCHAFTEN VON »EINGABEAUFFORDERUNG« oder EIGENSCHAFTEN VON KONSOLENFENSTER.

Gleich welche Auswahl Sie zuvor getroffen haben, können Sie auf der Registerkarte OPTIONEN nun die Größe des Cursors festlegen und auch bestimmen, ob die Eingabeaufforderung im Fenster- oder Vollbildmodus dargestellt werden soll. Des Weiteren können Sie die Puffergröße für den Befehlsspeicher auswählen (1–999) und damit

festlegen, wie viele der zuletzt eingegebenen Befehle im Speicher gehalten werden sollen, die so mit den **[a]**- und **[A]**-Tasten zurückgeholt und wieder eingefügt werden können, ohne sie neu eingeben zu müssen. Neben der Puffergröße können Sie auch die Anzahl der Puffer (1-999) und damit die Anzahl der Prozesse mit je einem eigenen Befehlsspeicher festlegen. Und um den Befehlsspeicher so effizient wie möglich zu nutzen, können Sie mit dem Aktivieren der Option **ALTE DUPLIKATE LÖSCHEN** dafür sorgen, dass nur eindeutige Befehle und Befehlszeilen im Speicher gehalten werden und keine doppelten oder mehrfach eingegebenen Befehle und Befehlszeilen zwischengespeichert werden.

Mit der Option **QUICKEDIT-MODUS** können Sie das schnelle Kopieren und Einfügen mit der Maus aktivieren, womit Sie jederzeit eine Zeichenfolge im Eingabeaufforderungsfenster mit der Maus markieren und mit einem Doppelklick der linken Maustaste an der aktuellen Cursor-Position wieder einfügen können, ohne hierfür erst das Kontextmenü öffnen zu müssen. Mit dem Aktivieren der Option **EINFÜGEMODUS** können Sie die Einfüge-Funktion einschalten, die das Einfügen neuer Zeichen in eine bereits eingegebene Befehlszeile erlaubt. Wenn diese Funktion nicht aktiviert ist, so werden die Zeichen, die sich rechts neben der aktuellen Cursor-Position befinden, durch die Eingabe neuer Zeichen überschrieben.

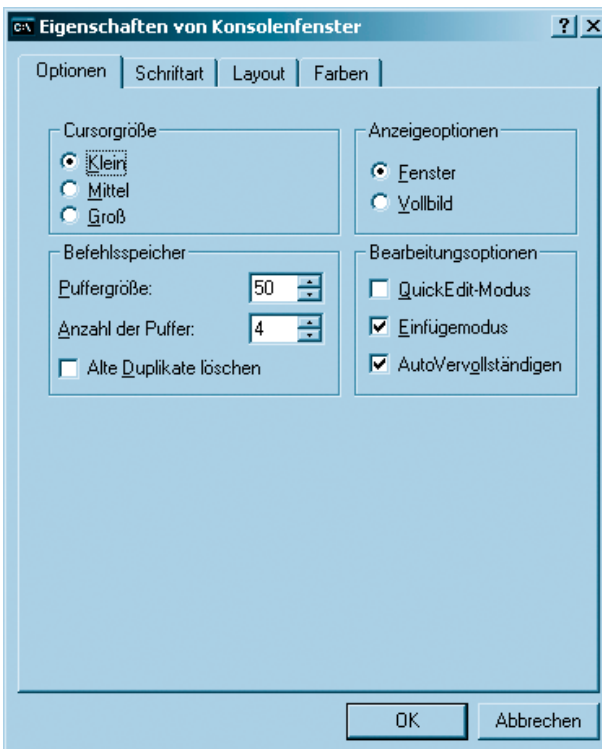


Abbildung 30: Die Eigenschaften der Eingabeaufforderung

Unter der Voraussetzung, dass Sie beim Öffnen der Eigenschaften STANDARDWERTE und nicht EIGENSCHAFTEN ausgewählt haben, finden Sie im Dialogfenster EIGENSCHAFTEN VON KONSOLENFENSTER zusätzlich die Option AUTOVERVOLLSTÄNDIGEN vor. Das Aktivieren dieser Option ermöglicht es Ihnen, die Eingabe von Datei- und Ordernamen zu vereinfachen. Denn Sie brauchen nur noch die ersten Zeichen des Datei- und Ordernamens einzugeben und können dann mit Hilfe der -Taste den vollständigen Namen der Datei oder des Ordners automatisch einfügen.

Auf der Registerkarte SCHRIFTART haben Sie die Möglichkeit, die Schriftart und auch die Schriftgröße zu ändern. Die Größe des Eingabeaufforderungsfensters dagegen können Sie auf der Registerkarte LAYOUT verändern, und auf der Registerkarte FARBEN können Sie die Hintergrundfarbe und auch die Textfarbe für das Eingabeaufforderungsfenster Ihren Wünschen entsprechend anpassen.

Tipp

Wenn Sie des Öfteren mit der Eingabeaufforderung arbeiten, so empfiehlt sich, die Hintergrundfarbe auf Weiß und die Textfarbe auf Schwarz zu setzen, da dies für die Augen weitaus angenehmer ist als umgekehrt.

Die Umgebungsvariablen von Windows Server 2003

Windows Server 2003 verwendet so genannte Umgebungsvariablen (engl. = *Environment Variables*), worin dynamische Daten zur aktuellen Arbeitsumgebung hinterlegt sind. Mit diesen Umgebungsvariablen können Sie von der Eingabeaufforderung aus (und natürlich auch unter grafischer Oberfläche oder aus einem Skript heraus) Informationen zu den Laufwerken, zu diversen Standardordnern (z.B. Benutzerprofil, Windows-Ordner, Temp-Ordner ...), zur verwendeten Betriebssystemversion, zur Computer-Hardware etc. abfragen und einen Teil der Umgebungsvariablen auch ändern.

Der Vorteil dieser Umgebungsvariablen ist, dass Sie die jeweiligen Umgebungsvariablen in einer Befehlszeile oder auch innerhalb eines Skriptes verwenden können, ohne den Inhalt der Variablen »hard-codiert« hinterlegen zu müssen. Wenn Sie z.B. eine Batch-Datei zur Automatisierung diverser Aufgaben erstellen möchten und das Systemlaufwerk und vielleicht auch den Pfad zum Windows-Ordner angeben müssen, so können Sie statt C:\ die Variable %SystemDrive% und statt C:\Windows die Variable %WinDir% verwenden. Das hat den entscheidenden Vorteil, dass die Batch-Datei auch auf einem Computersystem ausgeführt werden kann, auf dem das Systemlaufwerk beispielsweise D:\ oder E:\ und der Windows-Ordner vielleicht E:\WinNT lautet. Denn der Inhalt der Umgebungsvariablen lässt sich zur Laufzeit einer Batch-Datei, eines Skriptes oder sonstigen Programms problemlos abfragen und für Anweisungen und auch als Parameter verwenden.

Bei Verwendung von Umgebungsvariablen gilt es zwischen benutzerspezifischen und systemspezifischen Umgebungsvariablen zu unterscheiden. Die benutzerspezifischen Umgebungsvariablen sind innerhalb der Registrierung im Schlüssel HKEY_CURRENT_USER\ENVIRONMENT abgelegt und stehen nur dann zur Verfügung, wenn der Benutzer angemeldet ist. Die systemspezifischen Umgebungsvariablen dagegen sind im Schlüssel

HKEY_LOCAL_MACHINE\SYSTEM\CURRENTCONTROLSET\CONTROL\SESSION MANAGER\ENVIRONMENT zu finden und stehen für alle Prozesse zur Verfügung. Selbst dann, wenn kein Benutzer angemeldet ist.

Die nachfolgende Tabelle soll Ihnen einen Überblick über die per Default verfügbaren Umgebungsvariablen unter Angabe einer kurzen Beschreibung geben und soll Ihnen auch zeigen, welche Umgebungsvariablen benutzer- und systemspezifisch sind.

Umgebungsvariable	Benutzer	System	Beschreibung
AllUsersProfile	•		Pfad zum All Users-Benutzerprofil (z.B. C:\Dokumente und Einstellungen\All Users)
AppData	•		Pfad zum Ablageordner der Anwendungsdaten (z.B. C:\Dokumente und Einstellungen\Administrator\Anwendungsdaten)
CD	•		Pfad des aktuellen Ordners (z.B. C:\)
CMDCMDLine	•		Befehlszeile für die Ausführung der aktuellen Cmd.exe-Datei (z.B. C:\Windows\System32\Cmd.exe)
CMDEXTVERSION		•	Version der Befehlszeilenerweiterung (z.B. 2)
COMPUTERNAME		•	Name des Computers (z.B. FRA-MD-SRV-001)
COMSPEC		•	Pfad zum Befehlszeileninterpreter (z.B. C:\Windows\System32\Cmd.exe)
DATE		•	Aktuelles Systemdatum (z.B. 11.11.2005)
ERRORLEVEL		•	Rückgabewert der letzten Befehlsausführung (z.B. 0)
HOME		•	Laufwerksbuchstabe zum Home-Laufwerk des gerade angemeldeten Benutzers (z.B. C:)
USERPROFILE		•	Pfad zum Home-Laufwerk des gerade angemeldeten Benutzers (z.B. C:\Dokumente und Einstellungen\Administrator)
NETWORK		•	Netzwerkpfad zum Home-Laufwerk des gerade angemeldeten Benutzers (z.B. \\FRA-MD-DC-001\Users\AdmTask)
LOGONSERVER	•		Name des Computersystems bzw. des Domänen-Controllers, der die Anmeldung der aktuellen Windows-Sitzung verifiziert hat (z.B. \\FRA-MD-DC-001)
NUMBER_OF_PROCESSORS		•	Anzahl der Prozessoren (z.B. 1)

Tabelle 7: Die Standard-Umgebungsvariablen von Windows Server 2003

Umgebungsvariable	Benutzer	System	Beschreibung
OS		•	Name des Betriebssystems (z.B. <i>Windows_NT</i>)
Path		•	Suchpfad für ausführbare Dateien (z.B. <i>C:\Windows\System32;C:\Windows;C:\Windows\System32\Wbem</i>)
PathExt		•	Dateinamenerweiterungen der ausführbaren Dateien (z.B. <i>.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH</i>)
Processor_Architecture		•	Architekturtyp des Prozessors (z.B. <i>x86</i>)
Processor_Identifier		•	Kurzbeschreibung des Prozessors (z.B. <i>x86 Family 6 Model 6 Stepping 2, Authentic-AMD</i>)
Processor_Level		•	Modellnummer des Prozessors (z.B. <i>6</i>)
Processor_Revision		•	Revisionsnummer des Prozessors (z.B. <i>0602</i>)
Prompt	•		Prompt-Konfigurationseinstellung (z.B. <i>\$P\$G</i>)
Random		•	Zufallszahl zwischen 0 und 32767 (z.B. <i>19585</i>)
SystemDrive		•	Laufwerkbuchstabe des Laufwerks, auf dem sich die Betriebssystem-Installation befindet (z.B. <i>C:</i>)
SystemRoot		•	Pfad des Betriebssystem-Basisordners (z.B. <i>C:\Windows</i>)
Temp	•	•	Pfad des benutzer- und systemspezifischen Temp-Ordners (z.B. <i>C:\Dokumente und Einstellungen\Administrator\Lokale Einstellungen\Temp</i> und <i>C:\Windows\Temp</i>)
TMP	•	•	Pfad des benutzer- und systemspezifischen Temp-Ordners (z.B. <i>C:\Dokumente und Einstellungen\Administrator\Lokale Einstellungen\Temp</i> und <i>C:\Windows\Temp</i>)
Time		•	Aktuelle Systemzeit (z.B. <i>16:17:49,37</i>)
UserDomain	•		Name des Computersystems oder der Domäne, auf dem bzw. in der das Benutzerkonto des gerade angemeldeten Benutzers angelegt wurde (z.B. <i>FRA-MD-SRV-001</i>)
UserName	•		Name des gerade angemeldeten Benutzers (z.B. <i>Administrator</i>)

Tabelle 7: Die Standard-Umgebungsvariablen von Windows Server 2003 (Forts.)

Umgebungsvariable	Benutzer	System	Beschreibung
UserProfile	•		Pfad zum Benutzerprofil des gerade angemeldeten Benutzers (z.B. C:\Dokumente und Einstellungen\Administrator)
WinDir		•	Pfad des Betriebssystem-Basisordners (z.B. C:\Windows)

Tabelle 7: Die Standard-Umgebungsvariablen von Windows Server 2003 (Forts.)

Umgebungsvariablen abfragen und bearbeiten

Die benutzer- und systemspezifischen Umgebungsvariablen lassen sich jederzeit von der Eingabeaufforderung aus abfragen und (zumindest größtenteils) auch ändern. Hierfür stellt Windows Server 2003 den Set- und Setx-Befehl zur Verfügung. Um beispielsweise eine Aufstellung aller Umgebungsvariablen (mit Ausnahme der versteckten Umgebungsvariablen CD, Date, Time, Random, ErrorLevel, CMDEXTVERSION und CMDLINE) unter Angabe ihrer jeweiligen Werte zu erzeugen, geben Sie in der Eingabeaufforderung Set (ohne Angabe weiterer Parameter) ein.

Eingabeaufforderung

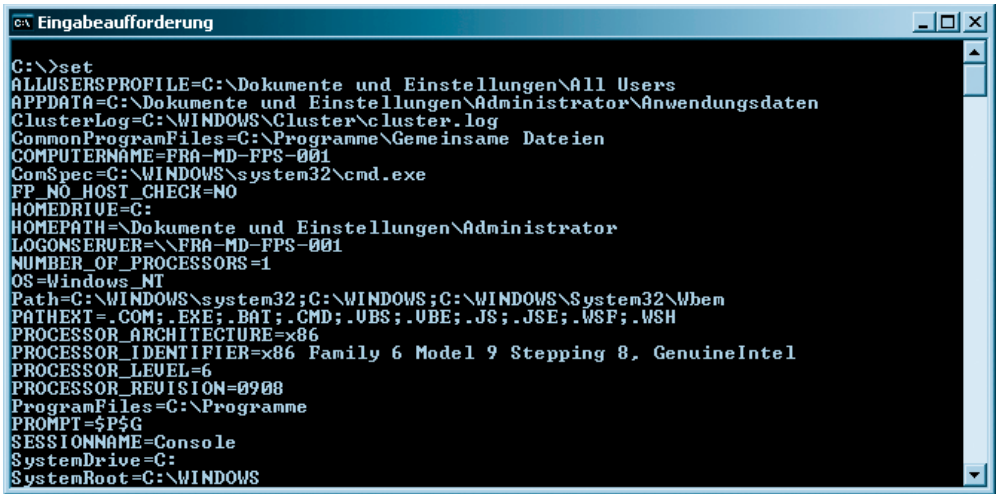


Abbildung 31: Das Ergebnis nach Eingabe des »Set«-Befehls

Möchten Sie dagegen nur den Wert einer bestimmten Umgebungsvariablen abfragen, so wenden Sie ebenfalls den Set-Befehl an und übergeben den Namen der Umgebungsvariablen als Parameter (z.B. Set WinDir). Alternativ dazu können Sie auch den Echo-Befehl verwenden, der ebenfalls den Wert einer Umgebungsvariablen ermittelt und am Bildschirm ausgibt. Allerdings müssen Sie bei Verwendung des Echo-Befehls die Umgebungsvariable in das %-Zeichen einschließen, da der Echo-Befehl ansonsten nur die Eingabe, die dem Echo-Befehl folgt, ausgibt. Mit dem Befehl Echo %WinDir% erhalten Sie also das gleiche Ergebnis wie mit Set WinDir.

Wenn Sie einer Umgebungsvariablen einen neuen Wert zuweisen möchten, so wenden Sie im einfachsten Fall wieder den `Set`-Befehl an und übergeben mit `Umgebungsvariable=Wert` den Namen der Variablen und auch den neuen Inhalt, wobei Sie hier `Umgebungsvariable` durch den Variablennamen und `Wert` durch den neuen Inhalt ersetzen. Wenn Sie beispielsweise den Pfad für die benutzerspezifische Umgebungsvariable `Temp` ändern und künftig `C:\Temp` statt `C:\%UserProfile%\Lokale Einstellungen\Temp` verwenden möchten, so können Sie hierfür die nachfolgende `Set`-Befehlszeile verwenden:

```
@Echo off
Set Temp=C:\Temp
```

Listing 20: 1_03_004a.bat

Hierbei sei angemerkt, dass beim Einsatz des `Set`-Befehls die benutzerspezifischen Umgebungsvariablen Vorrang vor den systemspezifischen Umgebungsvariablen haben. Wenn Sie mit dem `Set`-Befehl beispielsweise die Umgebungsvariablen `Temp` und `Tmp` ändern möchten, so werden hierbei die benutzerspezifischen Umgebungsvariablen geändert. Das Ändern der systemspezifischen Umgebungsvariablen `Temp` und `Tmp` ist daher also nicht möglich.

Um systemspezifische Umgebungsvariablen zu ändern, können Sie den `Setx`-Befehl einsetzen, der standardmäßig im Befehlsumfang von Windows Server 2003 enthalten ist. Der `Setx`-Befehl hat gegenüber dem `Set`-Befehl den Vorteil, dass Sie selbst bestimmen können, ob benutzer- oder systemspezifische Umgebungsvariablen geändert werden sollen.

Wenn Sie beispielsweise die systemspezifischen Umgebungsvariablen `Temp` und `Tmp` ändern möchten und künftig nicht mehr den Pfad `C:\Windows\Temp`, sondern `C:\Temp` nutzen wollen, so wenden Sie den `Setx`-Befehl an, übergeben den Namen der Variablen sowie den neuen Wert und fügen zusätzlich den Parameter `/M` an, mit dessen Hilfe Sie festlegen können, dass nicht die benutzerspezifischen, sondern die systemspezifischen Umgebungsvariablen geändert werden sollen.

```
@Echo off
Setx Temp C:\Temp /M
Setx Tmp C:\Temp /M
```

Listing 21: 1_03_004b.bat

Den `Set`- und `Setx`-Befehl können Sie auch dazu benutzen, um neue Umgebungsvariablen anzulegen und vorhandene Umgebungsvariablen zu löschen. Wenn Sie beispielsweise eine neue Umgebungsvariable mit der Bezeichnung `MyData` erstellen möchten, die in den Ordner `C:\Daten\MyData` verweist, so wenden Sie – sofern es sich um eine benutzerspezifische Umgebungsvariable handelt – den `Set`-Befehl an und übergeben den Namen der Umgebungsvariablen wie auch den dazugehörigen Wert.

```
@Echo off
Set MyData=C:\Daten\MyData
```

Listing 22: 1_03_004c.bat

Um eine systemspezifische Umgebungsvariable mit der Bezeichnung *MyData* zu erzeugen, die in den zuvor genannten Ordner verweist, wenden Sie stattdessen den *Setx*-Befehl an, übergeben den Namen der Variablen, deren Wert und zusätzlich auch den Parameter */M*, da sich der *Setx*-Befehl per Default auf benutzerspezifische und nicht auf systemspezifische Umgebungsvariablen auswirkt.

```
@Echo off
Setx MyData C:\Daten\MyData /M
```

Listing 23: 1_03_004d.bat

Wenn Sie eine Umgebungsvariable löschen möchten, so können Sie hierfür ebenfalls den *Set*- und *Setx*-Befehl einsetzen. Um beispielsweise den Inhalt der systemspezifischen Umgebungsvariablen *MyData* zu entfernen, geben Sie in der Eingabeaufforderung den Befehl *Set MyData* oder *Setx MyData "" /M* ein. Die Variable wird daraufhin wieder entfernt.

Achtung

Beim Löschen von benutzer- und systemspezifischen Umgebungsvariablen sollten Sie größte Vorsicht walten lassen. Denn viele der Standardumgebungsvariablen werden vom Betriebssystem und auch von vielen Applikationen genutzt. Wenn Sie Umgebungsvariablen wahllos löschen, so kann dies zu Instabilitäten des Betriebssystems und einiger Applikationen führen. Löschen Sie Umgebungsvariablen also nur dann, wenn Sie sicherstellen können, dass die Variablen nicht vom System oder von irgendeiner Applikation benötigt werden.

Die WMI-Konsole

Die WMI-Konsole (*WMIC* = *Windows Management Instrumentation Console*) von Windows Server 2003 stellt ein wichtiges Werkzeug für die alltägliche Systemadministration und auch für die Automatisierung von Verwaltungsaufgaben dar. Denn mit *WMIC* haben Sie von der Eingabeaufforderung aus Zugriff auf die Verwaltungsschnittstelle des Betriebssystems und können damit unendlich viele Informationen abfragen und auch etliche Änderungen vornehmen, ohne dass hierfür eine aufwändige Programmierung oder der Einsatz von Systemmanagement-Tools wie z.B. SMS erforderlich wäre.

Mit *WMIC* können Sie sowohl im interaktiven Modus arbeiten wie auch im nicht interaktiven Modus. Ersteres bedeutet, dass Sie durch Eingabe von *WMIC* in die *WMIC*-eigene

Eingabeaufforderung gelangen, worüber Sie dann die WMIC-spezifischen Unterbefehle eingeben können. Der nicht interaktive Modus dagegen erlaubt die Nutzung der WMIC-spezifischen Unterbefehle von der Eingabeaufforderung aus, womit Sie die WMI-Möglichkeiten z.B. auch aus einer Batch-Datei heraus nutzen können. Denn im nicht interaktiven Modus können Sie die Unterbefehle von WMIC direkt an den eigentlichen WMIC-Befehl anhängen. Damit lassen sich auch außerhalb der WMIC-eigenen Eingabeaufforderung Informationen abfragen und Einstellungen ändern.

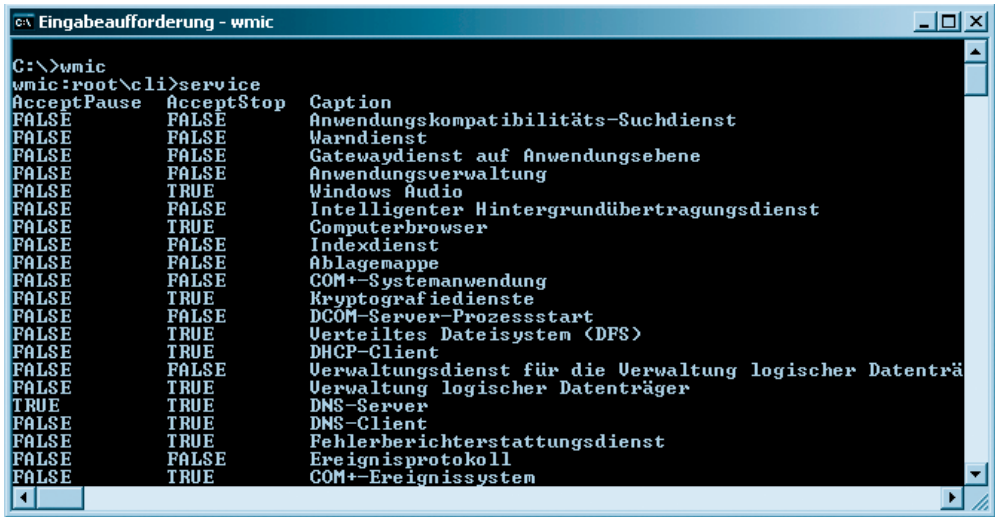
Einfache WMI-Abfragen durchführen

WMIC enthält bereits von Haus aus eine ganze Reihe vorgefertigter WMI-Abfragen, womit Sie viele Informationen zum Betriebssystem, zur verwendeten Hardware wie auch zu den installierten Applikationen abfragen können. Diese WMI-Abfragen werden in Form von Alias bereitgestellt, die Sie nur aufrufen müssen, um die gewünschten Informationen zu erhalten.

Hinter jedem der enthaltenen Alias verbirgt sich eine WQL-Abfrage (*WMI Query Language*) einer bestimmten WMI-Klasse. So können Sie beispielsweise durch die simple Eingabe von `Service` die WQL-Abfrage `Select * From Win32_Services` durchführen und erhalten daraufhin alle Eigenschaften der vorhandenen Systemdienste von Windows Server 2003 zurück. Das Komfortable dabei ist, dass Sie sich nicht um das Herstellen einer Verbindung zum WMI-Namespace `root\cimv2` und zur WMI-Klasse `WIN32_Service` sowie um die anschließende `Select`-Anweisung kümmern müssen, da dies bereits alles im Alias hinterlegt ist.

Um einmal alle Eigenschaften der Systemdienste von Windows Server 2003 am Bildschirm ausgeben zu lassen, wechseln Sie also zunächst in die Eingabeaufforderung, rufen mit `WMIC` die WMIC-eigene Eingabeaufforderung auf und geben dann den Alias `Service` ein (interaktiver Modus). Sie erhalten daraufhin eine mehrspaltige Ausgabe mit den Eigenschaften der Systemdienste (siehe `Win32_Service` in *Teil III* dieses Buches), die weitaus mehr Informationen zu den Diensten enthält, als Sie in der Dienst-Übersicht unter grafischer Oberfläche je angezeigt bekommen werden. Alternativ dazu können Sie auch außerhalb der WMIC-eigenen Eingabeaufforderung den Befehl `WMIC Service` verwenden, um die Eigenschaften abzufragen. Allerdings ist die Ausgabe im nicht interaktiven Modus aufgrund der vielen Spalten oftmals etwas verschoben und damit recht unübersichtlich. Mit Hilfe der Ausgabeumleitung (`> Datei.txt` oder `/Output:Datei.txt`) können Sie das Ergebnis aber auch in eine Datei ausgeben, womit die Spalten dann wieder richtig und damit auch übersichtlicher angezeigt werden.

Neben dem Alias `Service`, der mit der WMI-Klasse `Win32_Service` verknüpft ist, gibt es noch eine ganze Reihe weiterer Aliase, die Sie zum Auslesen von Informationen verwenden können. Die nachfolgende Tabelle enthält eine Übersicht aller Alias, die standardmäßig zur Verfügung stehen und die Sie im interaktiven Modus durch Eingabe des Aliasnamens und im nicht interaktiven Modus durch Eingabe von `WMIC Aliasname` zum Abfragen der Systeminformationen verwenden können.



Eingabe-
aufforderung

Abbildung 32: Eine einfache WMI-Abfrage durch Eingabe des Aliasnamens durchführen

Alias	Beschreibung
Alias	Alias-Verwaltung
BaseBoard	Hauptplatinen-Verwaltung
BIOS	BIOS-Verwaltung
BootConfig	Startkonfigurationsverwaltung (Boot.ini)
CDRom	CD-Verwaltung
Computersystem	Computersystem-Verwaltung
CPU	CPU-Verwaltung
CSProduct	SMBIOS-Verwaltung
DataFile	Datei-Verwaltung
DCOMApp	DCOM-Anwendungsverwaltung
Desktop	Desktop-Verwaltung
DesktopMonitor	Desktop-Bildschirmverwaltung
DeviceMemoryAddress	Gerätespeicheradress-Verwaltung
DiskDrive	Physikalische Laufwerksverwaltung
DiskQuota	Kontingentverwaltung
DMACHannel	DMA-Kanalverwaltung
Environment	Systemumgebungsverwaltung
FSDir	Ordner-Verwaltung

Tabelle 8: Die Standard-Alias von WMIC

Alias	Beschreibung
Group	Gruppenverwaltung
IDEController	IDE-Controller-Verwaltung
IRQ	IRQ-Verwaltung
Job	Taskplaner-Verwaltung
LoadOrder	Verwaltung der Lade-Reihenfolge bei Abhängigkeiten zwischen mehreren Systemdiensten
LogicalDisk	Lokale Speichergeräte-Verwaltung
Logon	Anmelde-/Sitzungsverwaltung
MemCache	Cache-Verwaltung
MemLogical	Systemspeicherverwaltung
MemoryChip	Speicherchip-Informationen
MemPhysical	Physikalische Speicherverwaltung
NetClient	Netzwerkclient-Verwaltung
NetLogin	Netzwerkanmelde-Informationen
NetProtocol	Netzwerkprotokoll-Verwaltung
NetUse	Netzwerkverbindungsverwaltung
Nic	Netzwerkadapter-Verwaltung
NicConfig	Netzwerkadapter-Konfigurationsverwaltung
NTDomain	NT-Domänenverwaltung
NTEvent	Ereignisverwaltung
NTEventLog	Ereignisdateiverwaltung
OnboardDevice	Verwaltung der auf der Hauptplatine befindlichen Geräte
OS	Betriebssystemverwaltung
PageFile	Auslagerungsdateiverwaltung
PageFileSet	Auslagerungsdateikonfiguration
Partition	Partitionsverwaltung
Port	Port-Verwaltung
PortConnector	Physikalische Verbindungsverwaltung
Printer	Druckerverwaltung
PrinterConfig	Druckerkonfiguration
PrintJob	Druckauftragsverwaltung
Process	Prozessverwaltung
Product	Aufgabenverwaltung für Anwendungsinstallationen
QFE	Hotfix-Verwaltung

Tabelle 8: Die Standard-Alias von WMIC (Forts.)

Alias	Beschreibung
QuotaSetting	Kontingent-Konfiguration
RDAccount	Berechtigungsverwaltung für Remotedesktop-Verbindungen
RDNic	Remotedesktop-Verbindungsverwaltung für bestimmte Netzwerkadapter
RDPermissions	Berechtigungsverwaltung für bestimmte Remotedesktop-Verbindungen
RDToggle	Aktivierungs- und Deaktivierungsfunktion für den Remotedesktop-Listener
RecoverOS	Speicherabbild beim Betriebssystemabsturz
Registry	Registrierungsverwaltung
SCSIController	SCSI-Controller-Verwaltung
Server	Server-Prozessverwaltung
Service	Dienste-Verwaltung
ShadowCopy	Schattenkopieverwaltung
ShadowStorage	Schattenkopiespeicherverwaltung
Share	Freigabenverwaltung
SoftwareElement	Software-Verwaltung
SoftwareFeature	Verwaltung spezieller Software-Funktionen
SoundDev	Soundkarten-Verwaltung
Startup	Autostart-Verwaltung
SysAccount	Systemkonto-Verwaltung
SysDriver	Systemtreiber-Verwaltung
SystemEnclosure	Systemgehäuse-Verwaltung
SystemSlot	Physikalische Verbindungspunkte-Verwaltung
TapeDrive	Bandlaufwerk-Verwaltung
Temperature	Temperatur-Informationen
TimeZone	Zeitonenverwaltung
UPS	Unterbrechungsfreie Stromversorgungsverwaltung
UserAccount	Benutzerkontoverwaltung
Voltage	Spannungssensor-Datenverwaltung
Volume	Lokale Speichervolume-Verwaltung
VolumeQuotaSetting	Kontingent-Konfiguration für bestimmte Volumes
VolumeUserQuota	Kontingent-Verwaltung für bestimmte Benutzer
WMISet	Einstellungen des WMI-Dienstes

Tabelle 8: Die Standard-Alias von WMIC (Forts.)

Wenn Sie im interaktiven Modus den Aliasnamen oder im nicht interaktiven Modus `WMIC Aliasname` eingeben, so werden per Default alle Eigenschaften, die durch die jeweilige WMI-Klasse bereitgestellt werden, ausgegeben. Welche Eigenschaften jeweils zur Verfügung stehen, können Sie sich mit `Aliasname Get /?` bzw. mit `WMIC Aliasname Get /?` jederzeit anzeigen lassen. Für den Fall, dass Sie jedoch nur bestimmte Eigenschaften einer WMI-Klasse abfragen und das Ergebnis damit etwas reduzieren möchten, so können Sie die Ausgabe des Ergebnisses mit den Verben `List` und `Get` Ihren Anforderungen entsprechend anpassen.

Das Verb `List` erlaubt die Nutzung eines vorgefertigten Ausgabeformats, womit nur bestimmte Eigenschaften ausgegeben werden. Je nach Alias stehen dabei unterschiedliche Ausgabeformate zur Verfügung, die jeweils eine unterschiedliche Selektion der Eigenschaften beinhalten. Welche Ausgabeformate durch die einzelnen Aliase bereitgestellt werden und welche Eigenschaften jeweils in den Ausgabeformaten enthalten sind, können Sie sich im interaktiven Modus mit dem Befehl `Aliasname List /?` und im nicht interaktiven Modus mit `WMIC Aliasname List /?` anzeigen lassen können. Die nachfolgende Tabelle zeigt z.B. eine Übersicht der Ausgabeformate des Alias `Service`:

List-Format	Beschreibung
Brief	Dieses Format gibt die am häufigsten benötigten Eigenschaften eines Dienstes aus (z.B. <i>ProcessID</i> , <i>Name</i> , <i>StartMode</i> , <i>State</i> ...).
Config	Das <i>Config</i> -Format gibt nur die Eigenschaften eines Dienstes aus, die unmittelbar mit der Dienstkonfiguration in Verbindung stehen (z.B. <i>ErrorControl</i> , <i>Name</i> , <i>PathName</i> , <i>ServiceType</i> ...).
Control	In diesem Format sind alle Kontrolleigenschaften wie <i>AcceptPause</i> , <i>AcceptStop</i> , <i>ErrorControl</i> , <i>Name</i> , <i>StartMode</i> und <i>StartName</i> enthalten.
Full	Dieses Format gibt alle Eigenschaften aus, die durch die WMI-Klasse bereitgestellt werden.
Instance	Das <i>Instance</i> -Format gibt lediglich die Eigenschaft <i>Name</i> zurück.
Status	Dieses Format gibt nur die Eigenschaften <i>Name</i> , <i>Status</i> und <i>ExistCode</i> aus und dient der schnellen Dienst-Übersicht.
System	Dieses Format gibt die Dienste unter Angabe des vollständigen WMI-Pfades aus (z.B. <code>\\Computername\\Root\\Cimv2:Win32_Service.Name=»Browser«</code>).

Tabelle 9: Die List-Formate des Alias »Service«

Wenn Sie beispielsweise alle konfigurationsspezifischen Eigenschaften der Windows Server 2003-Dienste abfragen möchten, so geben Sie im interaktiven Modus den Befehl `Service List Config` und im nicht interaktiven Modus den Befehl `WMIC Service List Config` ein. Sie erhalten daraufhin eine Übersicht aller Dienste unter Angabe der Eigenschaften *DesktopInteract*, *ErrorControl*, *Name*, *PathName*, *ServiceType* und *StartMode*.

Falls Sie die Selektion der Eigenschaften selbst vornehmen und nicht auf ein vorgefertigtes Ausgabeformat zurückgreifen möchten, so können Sie hierfür das Verb `Get` verwenden, dem Sie die auszugebenden Eigenschaften als Parameter übergeben müssen. Um beispielsweise eine Übersicht aller Dienste zu erstellen und dabei nur den Namen des Dienstes und den Pfad zur ausführbaren Dienstdatei auszugeben, geben Sie im interaktiven Modus den Befehl `Service Get Name, PathName` und im nicht interaktiven Modus den Befehl `WMIC Service Get Name,PathName` ein. Mit dem Verb `Get` können Sie also selbst bestimmen, welche Eigenschaften ausgegeben werden sollen.

Und mit Hilfe der `Where`-Klausel können Sie zudem eine Selektion der einzelnen Instanzen vornehmen. Wenn Sie beispielsweise nur die Eigenschaften des Dienstes *Browser* abfragen und alle anderen Dienste von der Abfrage unberücksichtigt lassen möchten, so können Sie die Abfrage mit `Where "Name='Browser'"` genauer spezifizieren und damit nur die Eigenschaften der mit `Where` selektierten Instanz ausgeben. Geben Sie im interaktiven Modus also `Service Where "'Name=Browser'"` oder im nicht interaktiven Modus den Befehl `WMIC Service Where "'Name=Browser'"` ein, um alle Eigenschaften des Dienstes *Browser* abzufragen. Mit dem `AND`- und `OR`-Operator können Sie natürlich auch mehrere Selektionen miteinander verknüpfen. Und um die Ausgabe zu beeinflussen, können Sie natürlich auch hier wieder das Verb `Get` verwenden und damit nur die Eigenschaften abfragen, die Sie benötigen.

WMI-Methoden aufrufen

Viele der per Default eingerichteten Aliase erlauben den Aufruf so genannter WMI-Methoden, womit Sie einzelne Instanzen erstellen, ändern und auch löschen können. Dadurch können Sie von der Eingabeaufforderung aus die Systemkonfiguration verändern und z.B. aus einer Batch-Datei heraus die Dienstkonfiguration bearbeiten, die IP-Konfiguration vornehmen und vieles, vieles mehr.

Ob ein Alias entsprechende Methoden bereitstellt, können Sie sich im interaktiven Modus mit `Aliasname /?` und im nicht interaktiven Modus mit `WMIC Aliasname /?` anzeigen lassen. Wenn mit dieser Abfrage das Verb `Call` angezeigt wird, so erlaubt der entsprechende Alias den Aufruf von WMI-Methoden. Welche dies im Einzelnen sind und welche Parameter mit dem Methoden-Aufruf übergeben werden müssen, erfahren Sie durch Eingabe von `Aliasname Call /?` (interaktiver Modus) bzw. durch `WMIC Aliasname Call /?` (nicht interaktiver Modus).

Der Alias `Service` stellt z.B. eine ganze Reihe nützlicher Methoden bereit, womit Sie Dienste einrichten, löschen, starten, beenden, anhalten und auch fortsetzen können. Um beispielsweise den Dienst *Taskplaner* (Dienstname *Schedule*) mit Hilfe der WMI-Methode `StopService` zu stoppen und daraufhin mit der WMI-Methode `StartService` wieder zu starten, wenden Sie den Alias `Service` an, selektieren mit der `Where`-Klausel den betreffenden Dienst und rufen dann mit `Call` die Methode `StopService` auf, woraufhin der Dienst dann beendet wird.

Geben Sie im interaktiven Modus also `Service Where "Name='Schedule'" Call StopService` und im nicht interaktiven Modus den Befehl `WMIC Service Where "Name='Schedule'" Call StopService` ein, um den Dienst zu beenden. Während Sie im interaktiven

Modus die Ausführung der Methode von Hand bestätigen müssen, erfolgt das Beenden des Dienstes im nicht interaktiven Modus automatisch. Mit `Service Where "Name='Schedule'" Call StartService` bzw. mit `WMIC Service Where "Name='Schedule'" Call StartService` können Sie den Dienst daraufhin wieder starten.

```

Eingabeaufforderung - wmic
wmic:root\cli>service call /?

Methodenausführungsvorgänge.
SYNTAX:
CALL <Methodenname> [<Tatsächliche Parameterliste>]
HINWEIS: <Tatsächliche Parameterliste> ::= <Tatsächlicher Parameter> ! <Tatsächlicher Parameter> ! <Tatsächlicher Parameter> ! ...

Die folgenden Aliasverben/-methoden sind verfügbar:

Aufruf      [ Ein-/Aus- ]Parameter&typ      Status
=====
Change      [IN ]DesktopInteract<BOOLEAN>      <null>
            [IN ]DisplayName<STRING>
            [IN ]StartPassword<STRING>
            [IN ]ErrorControl<UINT8>
            [IN ]LoadOrderGroup<STRING>
            [IN ]LoadOrderGroupDependencies<ARRAY OF STRING>

Drücken Sie zum Fortsetzen eine beliebige Taste oder die ESC-Taste zum Beenden.
  
```

Abbildung 33: Die Anzeige der möglichen Methoden unter Angabe der anzugebenden Parameter

Viele Methoden erwarten die Angabe bestimmter Parameter, die Sie mit dem Methodenaufruf übergeben müssen. Wenn Sie z.B. mit der `ChangeStartMode`-Methode den Startmodus eines Dienstes verändern möchten, so müssen Sie den gewünschten Modus als Parameter übergeben. In diesem Fall lauten die möglichen Werte *Boot*, *System*, *Automatic*, *Manual* und *Disable*. Um die Startkonfiguration für den Dienst *Taskplaner* von *Automatic* auf *Manual* zu setzen, geben Sie im interaktiven Modus den Befehl `Service Where "Name='Schedule'" Call ChangeStartMode Manual` ein und bestätigen daraufhin die Ausführung der Methode. Im nicht interaktiven Modus dagegen geben Sie in der Eingabeaufforderung den Befehl `WMIC Service Where "Name='Schedule'" Call ChangeStartMode Manual` ein, die Änderung wird daraufhin sofort ausgeführt.

Tip

Viele Parameter, die Sie mit dem Aufruf einer WMI-Methode übergeben müssen, erlauben nur bestimmte Werte in Form einer Zahl oder einer Zeichenfolge. Die erlaubten Werte können Sie sich leider nicht mit `/?` anzeigen lassen. Sie finden jedoch auf der Internetseite msdn.microsoft.com eine Online-Referenz aller WMI-Klassen, Methoden und Eigenschaften und können (wenn Sie den Namen der Methode als Suchbegriff eingeben) jederzeit nachsehen, welche Parameter im Einzelnen angegeben werden müssen und welche Werte erlaubt sind. Außerdem finden Sie zu allen Klassen, Methoden und Eigenschaften eine kurze Beschreibung und auch eine Aufstellung sämtlicher Rückgabewerte, die seitens der Methodenaufrufe zurückgeliefert werden.

Die WMIC-Schalter

WMIC erlaubt die Verwendung so genannter Schalter, womit Sie das Verhalten von WMIC beeinflussen können. Mit Hilfe dieser Schalter können Sie beispielsweise eine WMI-Abfrage über mehrere Computersysteme hinweg durchführen, den Authentifizierungs- und Personifizierungslevel festlegen, den Namespace wechseln, die Ausgabe der WMI-Abfragen formatieren und vieles mehr.

Die nachfolgende Tabelle zeigt, welche Schalter erlaubt sind und was sie im Einzelnen bedeuten:

Schalter	Beschreibung
Aggregate	Wenn Sie eine WMI-Abfrage über mehrere Computersysteme hinweg durchführen, so können Sie mit <code>/Aggregate:On</code> festlegen, dass die Ausgabe des Ergebnisses erst dann erfolgt, wenn alle mit <code>/Node</code> angegebenen Computersysteme auf die WMI-Abfrage geantwortet haben. Die Einstellung <code>/Aggregate:Off</code> dagegen bewirkt, dass das Ergebnis sofort nach Erhalt durch das entsprechende Computersystem ausgegeben wird.
Append	Mit <code>/Append</code> können Sie das Ergebnis einer WMI-Abfrage an eine bereits vorhandene Ergebnisauflistung (Bildschirm-, Zwischenablage- oder Datei-Ausgabe) anhängen.
AuthLevel	Mit <code>/AuthLevel</code> können Sie den Authentifizierungslevel <i>Connect</i> , <i>Call</i> , <i>Pkt</i> , <i>Pktintegrity</i> oder <i>Pktprivacy</i> festlegen.
FailFast	Mit <code>/FailFast:On</code> können Sie festlegen, dass ein mit <code>/Node</code> angegebenes Computersystem zunächst per Ping auf seine Erreichbarkeit hin überprüft wird, bevor die eigentliche WMI-Abfrage erfolgt. Mit <code>/FailFast:Off</code> wird die Erreichbarkeit nicht geprüft.
ImpLevel	Der Schalter <code>/ImpLevel</code> erlaubt die Angabe des Personifizierungslevel <i>Anonymous</i> , <i>Identify</i> , <i>Impersonate</i> (dies ist die Standardeinstellung) und <i>Delegate</i> .
Interactive	Mit <code>/Interactive:On</code> können Sie WMIC dazu veranlassen, vor der Ausführung von Aktionen eine Bestätigung des Benutzers zu erzwingen. Mit <code>/Interactive:Off</code> werden die Befehle ohne eine Benutzerbestätigung ausgeführt.
Locale	Der Schalter <code>/Locale</code> erlaubt das Festlegen der regionalen Einstellungen. So können Sie mit <code>/Locale:MS_409</code> z.B. die englischen Einstellungen aktivieren.
Namespace	Mit <code>/Namespace</code> können Sie zwischen den einzelnen WMI-Namespace wechseln.
Node	Der Schalter <code>/Node</code> erlaubt die Angabe eines (oder auch mehrerer) Remote-Systeme, auf denen die WMI-Abfrage durchgeführt werden soll.

Tabelle 10: Die WMIC-Schalter und ihre Bedeutung

Schalter	Beschreibung
Output	Mit <code>/Output</code> können Sie die Ausgabe des Ergebnisses nach StdOut, Clipboard oder wahlweise auch in eine Datei umleiten. Mit <code>/Output:Datei.txt</code> wird das Ergebnis z.B. in eine Datei geschrieben, mit <code>/Output:Clipboard</code> dagegen in die Zwischenablage kopiert.
Password	Erlaubt die Angabe eines Kennwortes, wenn die WMI-Abfrage auf einem Remote-System durchgeführt werden soll, für das eine gesonderte Anmeldung erforderlich ist (siehe auch <i>User</i>).
Privileges	Mit <code>/Privileges:On</code> können Sie zusätzliche WMI-Privilegien aktivieren, um beispielsweise ein Computersystem herunterfahren zu dürfen.
Record	Mit <code>/Record:Datei.txt</code> können Sie festlegen, dass alle Ein- und Ausgaben in einer XML-Datei mitprotokolliert werden.
Role	Mit <code>/Role</code> können Sie den Pfad zu der Funktion definieren, worin die Alias-Definitionen hinterlegt sind. Standardmäßig wird <code>\\root\\cli</code> verwendet.
Trace	Mit <code>/Trace:On</code> können Sie das Aufzeichnen erweiterter Tracing-Informationen aktivieren und mit <code>/Trace:Off</code> wieder deaktivieren.
User	Erlaubt die Angabe eines Benutzernamens, wenn die WMI-Abfrage auf einem Remote-System durchgeführt werden soll, für das eine gesonderte Anmeldung erforderlich ist (siehe auch <i>Password</i>).

Tabelle 10: Die WMIC-Schalter und ihre Bedeutung (Forts.)

Wenn Sie z.B. die Dienstinformationen unterschiedlicher Computersysteme abfragen möchten, so wechseln Sie in die WMIC-eigene Eingabeaufforderung, legen mit `/Node:"Computername1, Computername2, Computername3"` die Hostnamen der abzufragenden Computersysteme fest und wenden dann den Alias *Service* an, um die Eigenschaften der einzelnen Instanzen abzufragen.

Möglichkeiten der Befehlserweiterung

Im Gegensatz zu früheren Windows-Betriebssystemversionen verfügt Windows Server 2003 über einen recht umfangreichen Befehlsvorrat, womit Sie einen Großteil der administrativen Aufgaben, die Sie über die grafische Oberfläche lösen können, auch von der Eingabeaufforderung aus durchführen können. Nichtsdestotrotz gibt es immer noch eine ganze Reihe von Aufgaben, für die selbst Windows Server 2003 keine eigenen Bordmittel bereitstellt. Denken Sie hierbei z.B. an das automatisierte Bearbeiten und Formatieren von Textdateien, an das Splitten großer Dateien in kleinere Dateifragmente, an das Abfragen und Bearbeiten der Berechtigungen und dergleichen mehr.

Um Aufgaben, für die Windows Server 2003 keine Befehle bereitstellt, dennoch von der Eingabeaufforderung aus lösen zu können, bleibt nur die Erweiterung des Befehlsvorrats durch Einsatz externer Tools oder aber die Erstellung von Visual Basic- und Java-Skripten, die Sie mit dem *Windows Scripting Host* (kurz *WSH* genannt) auch von der Eingabeaufforderung aus starten können.

Es gibt eine ganze Reihe von Möglichkeiten, wie Sie den Befehlsumfang von Windows Server 2003 um nützliche Tools erweitern können. Denn im Internet werden einige Tool-Kits angeboten, die viele leistungsfähige und insbesondere auch arbeits-erleichternde Tools enthalten. Die nachfolgende Tabelle nennt die wichtigsten Erweiterungsmöglichkeiten unter Angabe der jeweiligen Bezugsquelle.

Tools/Tool-Kit	Bezugsquelle
Windows Server 2003 Support-Tools	Windows Server 2003-CD
Windows Server 2003 Resource Kit-Tools	www.microsoft.com
Windows Services für Unix	www.microsoft.com
GNU-Utilities für Windows 32	unxutils.sourceforge.net

Tabelle 11: Die wichtigsten Erweiterungsmöglichkeiten und ihre Bezugsquellen

Windows Server 2003 Support-Tools

Die Windows Server 2003 Support-Tools enthalten zahlreiche Befehle für die Eingabeaufforderung, womit Sie Fehler in der Betriebssystem- und Netzwerkkonfiguration aufspüren und auch beseitigen können. Diese Tools stellen ein nützliches Add-On zu den Standardbefehlen des Betriebssystems dar und erlauben die Ausführung von Aufgaben, für die Windows Server 2003 per Default keine Werkzeuge bereithält.

Mit den Support-Tools können Sie beispielsweise die Replikation innerhalb des Active Directory überwachen, Diskrepanzen in den Zugriffskontrolllisten der Active Directory-Objekte aufspüren, die Funktionalität der Domänen-Controller testen, DNS-Konfigurationseinstellungen abfragen und verändern, den Datentransfer im Netzwerk überwachen und vieles mehr.

Die Support-Tools sind zwar Bestandteil der Installations-CD von Windows Server 2003, werden jedoch während der Betriebssysteminstallation nicht automatisch mit eingerichtet. Wenn Sie die Support-Tools einsetzen möchten, so müssen Sie die Tool-Sammlung von Hand nachinstallieren, Sie finden die Tools im Ordner `\Support\Tools` auf der Installations-CD von Windows Server 2003.

Windows Server 2003 Resource Kit-Tools

In Ergänzung zu den Windows Server 2003 Support-Tools stellt Microsoft die Resource Kit-Tools für Windows Server 2003 zur Verfügung, die Sie kostenlos von der Internetseite www.microsoft.com herunterladen und nach dem Extrahieren der Tools sofort einsetzen können.

Die Resource Kit-Tools beinhalten ebenfalls Befehle für die Eingabeaufforderung, womit Sie viele Aufgaben aus der Systemadministration durchführen können, für die das Betriebssystem standardmäßig keine entsprechenden Befehle bereithält. Mit Hilfe der Resource Kit-Tools können Sie beispielsweise viele Informationen zur Betriebssystem- und Netzwerkkonfiguration abfragen, die Konfiguration des Active Directory

verändern, die Software-Verteilung überwachen, sicherheitsspezifische Konfigurationseinstellungen abfragen und auch bearbeiten und zahlreiche Diagnosevorgänge durchführen.

Die Befehle, die in den Resource Kit-Tools enthalten sind, tragen dabei nicht nur zur vereinfachten Fehlersuche bei Betriebssystem- oder Netzwerkproblemen bei, sondern erleichtern die Systemadministration immens. Denn viele Befehle können zur Durchführung und auch zur Automatisierung komplexer Aufgaben eingesetzt werden, die sonst nur mit aufwändigen Skripten zu lösen wären.

Windows Services für Unix

Die Windows Services für Unix (auch SFU genannt) stellen ein komplettes Unix-Subsystem für Windows-Betriebssysteme dar, womit Sie zum einen eine sehr komfortable Schnittstelle zwischen Windows- und Unix-Systemen schaffen können und zum anderen auch eine Möglichkeit haben, die zahlreichen und sehr leistungsfähigen Unix-Tools auf Windows-Systemen einsetzen zu können.

Dieses Add-On, das Sie kostenlos von der Internetseite www.microsoft.com herunterladen können, erleichtert Ihnen die Arbeit in heterogenen Umgebungen immens. Denn mit den Windows Services für Unix können Sie auf recht einfache Art und Weise einen Datenaustausch zwischen Windows- und Unix-Systemen vornehmen, Unix-Systeme von einem Windows-System aus administrieren, mehr als 300 Unix-Tools unter der Windows-Betriebssystemumgebung ausführen, Skripte erzeugen und ausführen und vieles mehr.

GNU-Utilities für Windows 32

Bei den GNU-Utilities für Win32 handelt es sich um eine Sammlung von über hundert GNU-Tools, die von Unix zu Win32 portiert wurden und die keine gesonderte Umgebung (z.B. der Einsatz einer Emulation) benötigen, um auf einem Windows-System eingesetzt werden zu können. Die GNU-Utilities für Win32 setzen lediglich das Vorhandensein der C-Runtime (*Msvcrt.dll*) voraus, die jedoch standardmäßig auf allen neueren Windows-Betriebssystemversionen vorhanden ist.

In dieser Tool-Sammlung sind leistungsfähige Tools wie *Agrep.exe*, *Egrep.exe*, *Fgrep.exe*, *Sed.exe*, *Head.exe*, *Tail.exe*, *Wc.exe*, *Wget.exe*, *Split.exe*, *Sort.exe* und viele, viele mehr enthalten, womit Sie die aus der Unix-Welt bekannten Befehle auch unter Windows Server 2003 verwenden können, ohne eine zusätzliche Software installieren zu müssen. Denn Sie brauchen die kostenlose Tool-Sammlung nur von der Internetseite unxutils.sourceforge.net herunterzuladen und in einen beliebigen Ordner zu extrahieren und können die Befehle direkt einsetzen. Und im Gegensatz zu den *Windows Services für Unix* und *Cygwin* können Sie die Befehle auch per Copy & Paste auf andere Computersysteme übertragen und dort einsetzen. Das hat den Vorteil, dass Sie die Tools auch im Anschluss an ein Unattend-Setup oder für das schnelle To do zwischendurch verwenden können, da die GNU-Utilities für Win32 keine Installation im herkömmlichen Sinne erfordern. Die Befehle lassen sich problemlos auf einer Dis-

kette oder auf eine CD unterbringen und von dort aus ausführen, womit Sie sich oftmals – gerade in größeren Produktivumgebungen – das aufwändige Erstellen eines Change-Requests ersparen, da keine Veränderungen am Server-System vorgenommen werden müssen.

Einige dieser GNU-Utilities für Win32 kommen in den Rezepten in Teil II des Buches zum Einsatz. Sicherlich hätten die Aufgabenstellungen problemlos mit den Windows Services für Unix und auch mit Cygwin gelöst werden können, aber aufgrund des einfachen und unkomplizierten Einsatzes (und insbesondere der Verzicht auf die Installation einer zusätzlichen Software) dieser Befehlspportierungen werden primär die GNU-Utilities für Win32 verwendet.

Windows Script Host

Mit dem Windows Script Host gibt Ihnen Microsoft ein nützliches Werkzeug an die Hand, womit Sie Visual Basic- und Java-Skripte direkt unter Windows ausführen können, ohne dass Sie hierfür eine gesonderte Programmierungsumgebung einrichten müssen. Damit haben Sie die Möglichkeit, eigene Programme zu erstellen, mit deren Hilfe Sie häufig wiederkehrende Aufgaben vollständig automatisieren können.

Durch das Erstellen entsprechender Skriptdateien können Sie beispielsweise Informationen zur Betriebssystemumgebung, zur Netzwerkkonfiguration, zu den Registrierungseinstellungen und auch zum Active Directory abfragen und einen Großteil der Einstellungen natürlich auch verändern.

Der WSH im Überblick

Bis Windows 98 gab es kaum keine Möglichkeit, mit betriebssystemeigenen Bordmitteln eine wiederkehrende Aufgabe zu automatisieren. Zwar gab es bis Windows 95 noch den Makrorekorder, womit Sie bestimmte Tasten- und Mausklicks aufzeichnen und anschließend wiedergeben konnten, doch war das Einsatzgebiet des Makrorekorders auf Desktop-Aktionen beschränkt und stellte somit kein Hilfsmittel für die Administration dar.

Mit entsprechenden Batch-Dateien, worüber Sie mehrere Befehle der Eingabeaufforderung »stapeln« und automatisiert ausführen können, haben Sie schon weit mehr Möglichkeiten der Aufgabenautomatisierung. Allerdings lassen sich auch via Batch-Datei nicht alle Aufgaben automatisieren.

Es bestand also der Bedarf, eine Art Programmiersprache zu implementieren, die nicht von einer gesonderten Programmierungsumgebung (z.B. Visual Basic, C++) abhängig ist, sondern direkt unter Windows funktioniert. Dies wurde mit dem Einführen des *Windows Script Hosts* (nachfolgend nur noch *WSH*) realisiert, der in der Version 1.0 zum ersten Mal mit Windows 98 ausgeliefert wurde (die weitere Chronologie des WSH finden Sie in der WSH-Referenz unter msdn.microsoft.com). In Windows Server 2003 ist der WSH in der Version 5.6 per Default enthalten, Sie müssen also keine nachträgliche Installationen durchführen, um die Möglichkeiten des WSH nutzen zu können.

VBScript und JScript

Der WSH unterstützt zwei Programmiersprachen: *VBScript* und *JScript*. Die Syntax von *VBScript* ist der von Visual Basic sehr ähnlich. Wenn Sie also schon mit Visual Basic programmiert haben, so wird Ihnen der Einstieg mit *VBScript* sehr leicht fallen. Allerdings ist die Funktionsvielfalt etwas eingeschränkt. Denn Sie können beispielsweise keine Variablentypen (String, Integer, Boolean ...) definieren, Declare-Anwei-

sungen einsetzen und einiges mehr. Dennoch reichen die Möglichkeiten aus, um selbst komplexe Aufgaben lösen zu können (alle Skriptdateien, die in diesem Buch vorgestellt werden, sind mit *VBScript* erstellt).

Haben Sie bereits in Java programmiert, so wird Ihnen wahrscheinlich *JScript* eher zusagen, da *JScript* die Java-Syntax verwendet und daher keine große Umgewöhnung erfordert.

Die Differenzierung der *VBScript*- und *JScript*-Dateien erfolgt dabei einzig und alleine über die Dateinamenerweiterung. Das bedeutet, dass eine **.vbs*-Datei automatisch als *VBScript*-Datei und eine **.js*-Datei automatisch als *JScript*-Datei interpretiert wird. Beim Ausführen der Skriptdateien müssen also keine Parameter zur Differenzierung angegeben werden.

CScript und WScript: die Executables

Der WSH stellt zwei unterschiedliche Executables zur Verfügung: *CScript.exe* und *WScript.exe*. Die Datei *CScript.exe* ist das Executable, das Sie zur Ausführung von *VBScript*- und *JScript*-Dateien innerhalb der Eingabeaufforderung einsetzen, und *WScript.exe* kommt mit der Skriptausführung unter grafischer Oberfläche zum Einsatz.

Während *CScript.exe* eine ganze Reihe von Eigenschaften bereitstellt, die durch Angabe von Parametern gesetzt werden können, stellt *WScript.exe* lediglich zwei Eigenschaften bereit, die in den Datei-Eigenschaften einer Skriptdatei geändert werden können (siehe Skripteneigenschaften setzen auf Seite 114).

Der Unterschied dieser beiden Executables macht sich insbesondere mit der Ausgabe von Informationen bemerkbar. Denn wenn Sie beispielsweise mit der *WScript.Echo*-Anweisung Informationen am Bildschirm ausgeben möchten und die Skriptdatei mit *CScript.exe* ausführen, so wird die Meldung (wie auch beim Einsatz der *Echo*-Anweisung innerhalb einer Batch-Datei) in der Eingabeaufforderung angezeigt. Wenn Sie die Skriptdatei dagegen mit *WScript.exe* ausführen, so wird die Meldung dagegen in Form eines gewöhnlichen Windows-Dialogfensters angezeigt. Das Unschöne daran: Mit jeder *WScript.Echo*-Befehlszeile wird ein Windows-Dialogfenster geöffnet. Wenn Sie also in einer Skriptdatei mehrere *WScript.Echo*-Befehlszeilen hintereinander aufführen und unterschiedliche Informationen ausgeben (wie dies z.B. beim Abfragen von Eigenschaften oft der Fall ist), so wird jedes Ergebnis in einem eigenen Fenster angezeigt und daher mit je einem Klick auf die OK-Schaltfläche wieder von Hand geschlossen werden.

Der WSH nutzt per Default das Executable *WScript.exe* als Standard-Host. Wenn Sie des Öfteren mit der Eingabeaufforderung arbeiten und die Skripte vorzugsweise via Eingabeaufforderung starten, so können Sie jederzeit das Executable *CScript.exe* als Standard-Host festlegen (siehe CScript oder WScript: den Standard-Host festlegen auf Seite 114).

Eine Skriptdatei erzeugen

Um ein neues Skript zu erzeugen, benötigen Sie lediglich einen gewöhnlichen Texteditor. Sie können natürlich die Windows Server 2003-eigenen Editoren NOTEPAD und WORDPAD verwenden oder wahlweise auch externe Editoren oder gar ein Textverarbeitungssystem einsetzen. Wenn Sie ein Textverarbeitungssystem verwenden, so sollten Sie jedoch darauf achten, dass gerade Anführungszeichen verwendet werden und keine typografischen. Denn typografische Anführungszeichen werden während der Skriptausführung nicht als Anführungszeichen interpretiert, womit die Skriptausführung aufgrund von Syntaxfehlern unterbrochen wird.

Hinweis

Windows Script-Host-Referenz

Der WSH stellt eine ganze Reihe von ActiveX-Schnittstellen zur Verfügung, mit deren Hilfe Sie grundlegende Aufgaben zur Dateioperation, zu Netzwerkfunktionen, zur Registrierung und einiges mehr durchführen können. Die hierfür erforderlichen Objekte, Methoden und Eigenschaften sind in der *Windows Script-Host-Referenz* beschrieben, die auf der Internetseite *msdn.microsoft.com* publiziert wurde.

Wenn Sie beispielsweise ein Skript zum Verbinden eines Netzwerklaufwerkes erstellen möchten, so rufen Sie zunächst einen Editor (oder wahlweise auch ein Textverarbeitungssystem) auf und beginnen mit einer neuen und damit noch leeren Datei.

Sie sollten ein neues Skript immer mit entsprechenden Kommentarzeilen beginnen lassen, worin beispielsweise der Dateiname des Skriptes und auch eine kurze Beschreibung zum Skript selbst enthalten sind. Denn gerade in längeren Skripten lässt sich in der Regel nicht auf Anhieb erkennen, welche Aufgaben durch das jeweilige Skript gelöst werden. Es ist daher einfacher, eine kurze Beschreibung zur Funktion einzufügen und das Skript damit zu dokumentieren. Die Kommentarzeilen dienen lediglich der Übersichtlichkeit und werden von der Script-Engine während der Skriptausführung ignoriert.

Eine Kommentarzeile leiten Sie dabei mit dem ' -Zeichen oder wahlweise auch mit dem Kürzel REM (für **Remark**) ein. Wenn Sie beispielsweise in der ersten Zeile des Skriptes den Namen der Skriptdatei als Kommentar einfügen möchten, so beginnen Sie das Skript mit ' MyScript.vbs oder mit REM MyScript.vbs. Sie können nun weitere Kommentarzeilen einfügen und beispielsweise noch die Aufgabe, die durch das Skript gelöst werden soll, beschreiben. Wie viele Kommentarzeilen Sie einfügen, bleibt natürlich Ihnen selbst überlassen. Sie sollten das Skript jedoch nicht mit überflüssigen Sternchen-Reihen (dies ist eine allseits beliebte Vorgehensweise zur Auflockerung des Quellcodes) unnötig aufblähen und die Kommentarzeilen daher nur zur reinen Dokumentation des Skriptes verwenden. Ein Skript könnte beispielsweise wie folgt beginnen:

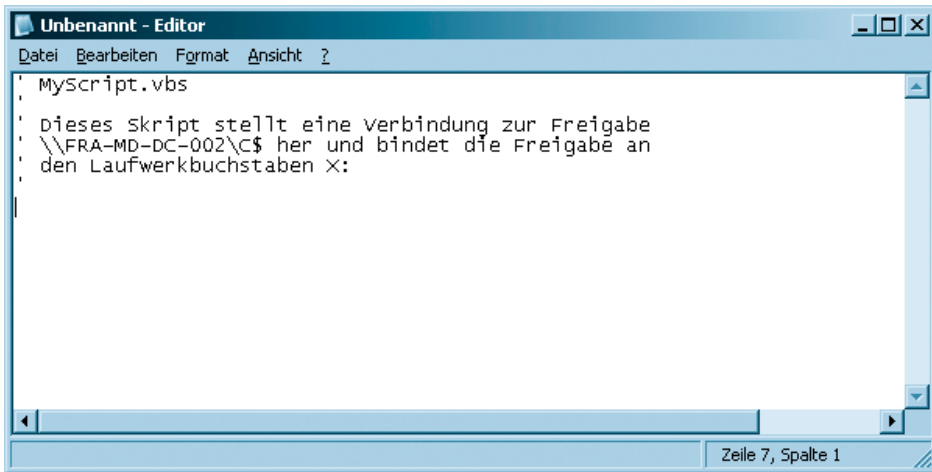


Abbildung 34: Kommentarseiten zu Beginn des Skriptes

Als Nächstes könnten Sie beispielsweise eine kurze Meldung ausgeben, dass nun versucht wird, eine Verbindung zur Freigabe herzustellen. Die Ausgabe von Informationen nehmen Sie mit der `WScript.Echo`-Anweisung vor, die lediglich den in Anführungszeichen eingeschlossenen Text, der ausgegeben werden soll, als Argument erwartet. Wenn Sie z.B. die Meldung »Stelle Verbindung mit Freigabe xyz her ...« ausgeben möchten, so fügen Sie die Anweisung `WScript.Echo "Stelle Verbindung mit Freigabe xyz her ..."` ein.

Um nun tatsächlich die Verbindung zur Freigabe herzustellen, erzeugen Sie mit der `CreateObject`-Methode ein Automatisierungsobjekt für `WScript.Network` und binden das zurückgelieferte Automatisierungsobjekt an eine so genannte Objektvariable.

```
Set oNetwork = WScript.CreateObject("WScript.Network")
```

Damit haben Sie eine Verbindung zum `WScript.Network`-Objekt hergestellt und können nun die zur Verfügung gestellten Methoden einsetzen und Eigenschaften abfragen. Um beispielsweise ein Netzwerklaufwerk zu verbinden, setzen Sie die `MapNetworkDrive`-Methode gegen die zuvor erstellte Objektvariable ein und übergeben den Laufwerksbuchstaben und auch den Pfad zur Freigabe als Argument.

```
oNetwork.MapNetworkDrive "X:", "\\FRA-MD-DC-001\C$"
```

Das komplette Skript, womit Sie zunächst eine Meldung am Bildschirm ausgeben und dann die Freigabe `\\FRA-MD-DC-001\C$` an den Laufwerksbuchstaben `X:` binden können, sieht nun folgendermaßen aus:

```
WScript.Echo "Stelle Verbindung mit Freigabe xyz her ..."
Set oNetwork = WScript.CreateObject("WScript.Network")
oNetwork.MapNetworkDrive "X:", "\\FRA-MD-DC-001\C$"
```

Damit haben Sie das erste Skript erstellt und können es nun ausführen und damit testen. Speichern Sie die Datei unter einem beliebigen Namen (jedoch mit der Endung `*.vbs`) ab, und beenden Sie den Editor. Wechseln Sie daraufhin in den WINDOWS

EXPLORER, und führen Sie einen Doppelklick auf die Skriptdatei aus, oder wechseln Sie in die Eingabeaufforderung, und geben Sie dort den Befehl `CScript MyScript.vbs` oder `WScript MyScript.vbs` ein, um das Skript zu starten. Sie sollten nun die Meldung *Stelle Verbindung mit Freigabe xyz her ...* sehen und daraufhin auch (sofern der Pfad zur Freigabe gültig ist) ein zusätzliches Laufwerk mit dem Buchstaben *X* vorfinden, das zu der angegebenen Freigabe verweist.

Skriptdateien ausführen

Es gibt mehrere Möglichkeiten, wie Sie eine Skriptdatei ausführen können. Im einfachsten Fall öffnen Sie unter grafischer Oberfläche den WINDOWS EXPLORER und führen einen Doppelklick auf die auszuführende Skriptdatei aus. Der Inhalt der Skriptdatei wird daraufhin eingelesen und an den Standard-Host (dies ist per Default *WScript.exe*) zur Verarbeitung weitergegeben.

Eine weitere Möglichkeit bietet der Startmenü-Eintrag AUSFÜHREN. Gleich ob Sie *CScript.exe* oder *WScript.exe* als Standard-Host verwenden: Öffnen Sie das Startmenü, klicken Sie auf AUSFÜHREN, und geben Sie den Pfad und auch den Namen (inklusive der Dateinamenerweiterung) ein. Die angegebene Skriptdatei wird daraufhin geöffnet, und der Inhalt wird an den Standard-Host zur Verarbeitung übergeben. Wenn Sie per Default *WScript.exe* benutzen und auf die zuvor genannte Art und Weise eine Skriptdatei an *Cscript.exe* übergeben wollen, so brauchen Sie nur das entsprechende Executable voranzustellen.

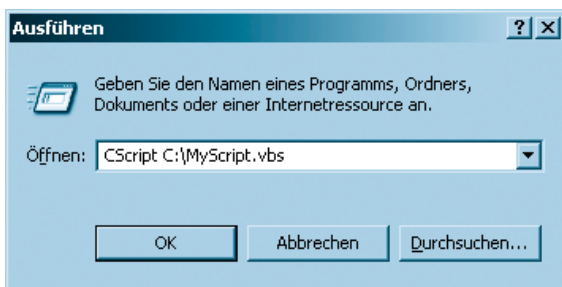


Abbildung 35: Eine Skriptdatei mit »CScript.exe« ausführen

Eine Skriptdatei können Sie auch direkt in der Eingabeaufforderung ausführen lassen. Wenn Sie beispielsweise die Skriptdatei *MyScript.vbs*, die im Root des Laufwerks *C* abgelegt ist, mit *CScript.exe* starten möchten, so geben Sie in der Eingabeaufforderung den Befehl `CScript C:\MyScript.vbs` ein. Der Inhalt der Skriptdatei wird daraufhin eingelesen und an *CScript.exe* zur Verarbeitung übergeben. Auf die gleiche Art und Weise können Sie eine Skriptdatei auch an *WScript.exe* übergeben. Sie müssen lediglich das entsprechende Executable verwenden: `WScript C:\MyScript.vbs`.

CScript oder WScript: den Standard-Host festlegen

Der WSH enthält zwei unterschiedliche Host-Versionen. So gibt es zum einen die Windows-basierte Host-Version (*WScript.exe*), zum anderen die eingabeaufforderungs-basierte Host-Version (*CScript.exe*). Der WSH nutzt per Default die Windows-basierte Version zur Skriptausführung. Wenn Sie beispielsweise im Windows Explorer eine Visual Basic- oder Java-Skriptdatei doppelklicken, so wird der Inhalt der Skriptdatei automatisch von *WScript.exe* verarbeitet.

Wenn Sie nun vorzugsweise in der Eingabeaufforderung arbeiten und daher eher mit der eingabeaufforderungs-basierten Host-Version arbeiten möchten, so können Sie den Standard-Host natürlich jederzeit neu definieren und *CScript.exe* (statt *WScript.exe*) zur Skriptausführung nutzen. Um den Standard-Host von *WScript.exe* zu *CScript.exe* zu ändern, öffnen Sie die Eingabeaufforderung und geben folgenden Befehl ein:

```
CScript //H:CScript
```

Möchten Sie zu einem späteren Zeitpunkt wieder die Windows-basierte Host-Version als Standard-Host nutzen, so geben Sie stattdessen folgenden Befehl ein:

```
CScript //H:WScript
```

Skripteigenschaften setzen

Mit dem Aufruf einer Skriptdatei via *CScript.exe* können Sie zeitgleich eine Reihe von Parametern übergeben und damit bestimmte Skripteigenschaften setzen, womit Sie die Ausführung der Skriptdatei (zumindest ein Stück weit) beeinflussen können.

Die nachfolgende Tabelle gibt Auskunft über die möglichen Parameter unter Angabe einer kurzen Beschreibung:

Parameter	Beschreibung
//B	Der Parameter //B aktiviert den Batch-Modus, womit keine Skriptfehler und Aufforderungen angezeigt werden.
//D	Mit dem Parameter //D können Sie das aktive Debuggen der Skriptdatei einschalten.
//E:Engine	Dieser Parameter erlaubt die Angabe des zu verwendenden Moduls.
//H:CScript.exe WScript.exe	Der Parameter //H: erlaubt das Festlegen des Standard-Hosts. Übergeben Sie CScript.exe, wenn Skriptdateien automatisch von CScript.exe verarbeitet werden sollen, oder übergeben Sie stattdessen WScript.exe, wenn die Skriptdateien automatisch mit WScript.exe verarbeitet werden sollen. Standardmäßig wird WScript.exe verwendet.

Tabelle 12: Die Skripteigenschaften

Parameter	Beschreibung
//I	Mit dem Parameter //I aktivieren Sie wieder den interaktiven Modus, wenn Sie zuvor mit //B den Batch-Modus eingeschaltet haben.
//Job:Datei.wsf	Dieser Parameter erlaubt die Ausführung einer *.wsf-Datei.
//Logo	Die Angabe dieses Parameters sorgt dafür, dass zu Beginn der Skriptausführung der WSH-Banner eingeblendet wird (dies ist die Standardeinstellung).
//NoLogo	Mit diesem Parameter deaktivieren Sie das Einblenden des WSH-Banners.
//S	Der Parameter //S ermöglicht die dauerhafte Speicherung der Skripteigenschaften. Die übergebenen Parameter werden in einer *.wsh-Datei gespeichert und erhalten automatisch den Namen der Skriptdatei (lediglich die Endung lautet *.wsh). Damit müssen die Parameter mit dem erneuten Aufruf der Skriptdatei nicht noch einmal übergeben werden, sondern können aus der *.wsh-Datei gelesen werden.
//T:x	Mit diesem Parameter können Sie den Timeout für die Skriptausführung bestimmen, indem Sie dem Parameter //T: die maximal zulässige Ausführungszeit in Sekunden übergeben.
//X	Dieser Parameter legt fest, dass das Skript im Debugger ausgeführt wird.
//U	Mit diesem Parameter können Sie veranlassen, dass Unicode für umgeleitete Ein- und Ausgabevorgänge verwendet wird.

Tabelle 12: Die Skripteigenschaften (Forts.)

Wenn Sie beispielsweise in der Eingabeaufforderung eine Skriptdatei mit der Bezeichnung *MyScript.vbs* mit *CScript.exe* ausführen und auf die Anzeige des WSH-Banners verzichten möchten, so fügen Sie der Befehlszeile, worüber Sie die Skriptdatei aufrufen, einfach den Parameter `//NoLogo` an. Damit werden die ersten beiden Bannerzeilen, die Auskunft über die verwendete WSH-Version geben und den Copyright-Vermerk von Microsoft enthalten, nicht mehr angezeigt. Und wenn Sie vermeiden möchten, dass das Skript länger als zehn Sekunden ausgeführt wird und damit nach Erreichen der maximalen Ausführungszeit von zehn Sekunden von selbst stoppt, so fügen Sie zudem den Parameter `//T:10` hinzu.

Für den Fall, dass Sie die zuvor genannte Skriptdatei des Öfteren ausführen müssen und mit jedem Skriptauf Ruf auch die beiden Parameter übergeben möchten, so ist es weitaus einfacher, die Eigenschaften zu speichern. Damit ersparen Sie sich das ständige Eingeben der beiden Parameter. Um die Eigenschaften dauerhaft zu setzen, rufen Sie zunächst die Skriptdatei auf gewohnte Art und Weise (sprich: unter Angabe der erforderlichen Parameter) auf und fügen der Befehlszeile den zusätzlichen Parameter `//S` hinzu. Damit werden die übergebenen Parameter (mit Ausnahme von `//S`) gespeichert, und Sie brauchen die Parameter künftig nicht mehr mit anzugeben.

Wenn Sie die Skriptdateien vorzugsweise unter grafischer Oberfläche per Doppelklick ausführen, so können Sie zwei der zuvor genannten Parameter auch in den Dateieigenschaften einer Skriptdatei setzen. Denn auf der Registerkarte SKRIPT können Sie zum einen den Timeout für die Skriptaussführung festlegen und zum anderen auch die Anzeige des Banners unterdrücken, falls das Skript in der Eingabeaufforderung ausgeführt wird. Um diese Einstellungen vornehmen zu können, markieren Sie im WINDOWS EXPLORER die entsprechende Skriptdatei, rufen dann aus dem Kontextmenü heraus die EIGENSCHAFTEN auf und wechseln auf die Registerkarte SKRIPT. Dort können Sie die Einstellung vornehmen und mit einem Klick auf OK anschließend übernehmen.

Mit dem Übernehmen der Skripteigenschaften wird zeitgleich eine *.wsh-Datei angelegt, die den gleichen Dateinamen verwendet wie die Skriptdatei selbst. Diese *.wsh-Datei ist eine Verlinkung zu der Skriptdatei und enthält die Skripteigenschaften.

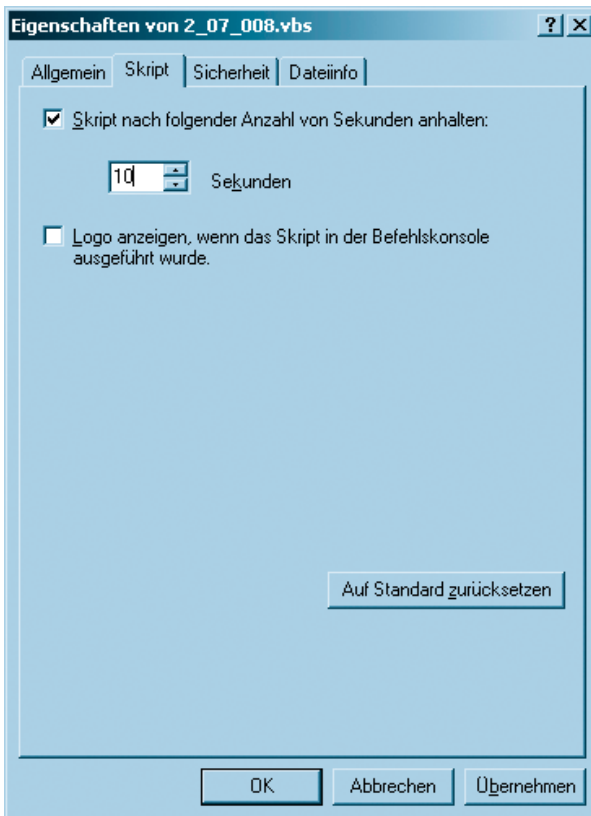


Abbildung 36: Die Skripteigenschaften festlegen

WMI

Windows Management Instrumentation, kurz *WMI* genannt, stellt die Verwaltungsschnittstelle des Betriebssystems dar, auf die Sie zum einen mit der WMI-Konsole der Eingabeaufforderung (siehe *Eingabeaufforderung* in Teil I des Buches) zugreifen können, zum anderen aber mit entsprechenden Visual Basic- und Java-Skripten. Der WMI-Zugriff erlaubt Ihnen das Abfragen unendlicher Informationen zum Betriebssystem wie auch zu den verwendeten Hard- und Software-Komponenten. Des Weiteren können Sie die WMI-Schnittstelle dazu verwenden, um eine ganze Reihe von Konfigurationsänderungen vorzunehmen. Die WMI-Schnittstelle ist daher gerade für Administratoren und Supporter von großem Interesse, da sich mit Hilfe entsprechender Skripte viele Informationen auslesen und insbesondere auch Aufgaben automatisieren lassen, ohne dass hierfür zusätzliche Software-Produkte erforderlich wären.

Scriptomatic: WMI-Skripting leicht gemacht

Microsoft bietet auf seiner Internetseite (www.microsoft.com) ein kostenloses Tool zum Download an, womit Sie per simplem Mausklick fertige WMI-Skripte erzeugen und diese auf dem lokalen Computersystem wie auch auf Remote-Systemen direkt ausführen können: *Scriptomatic*.

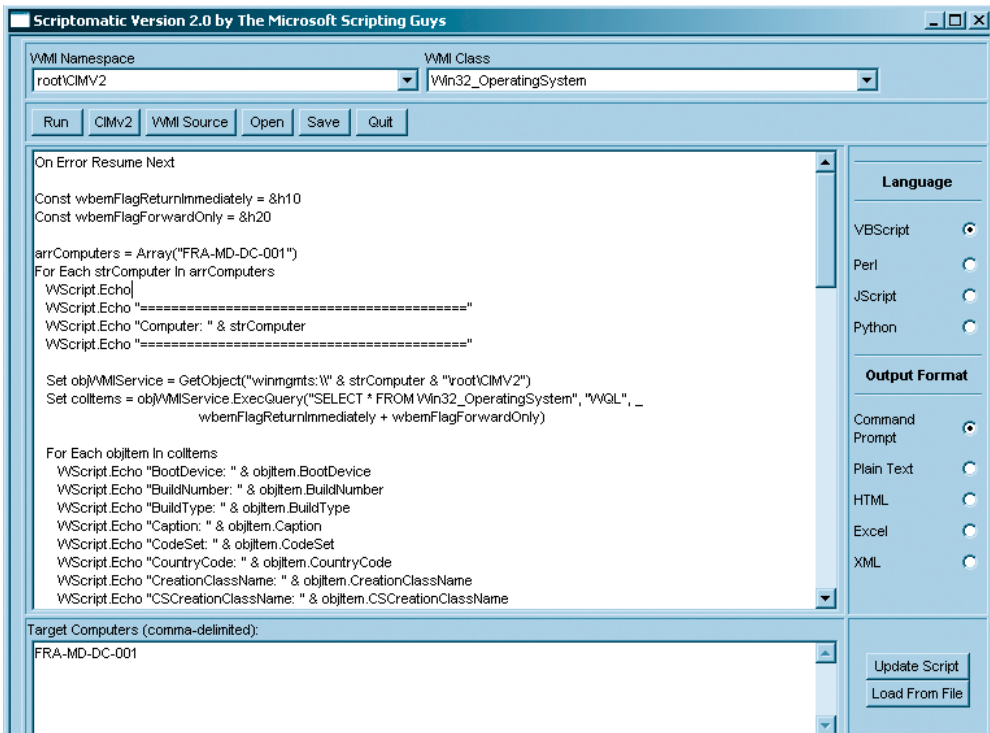


Abbildung 37: Ein mit Scriptomatic erzeugtes Visual Basic-Skript

Bei *Scriptomatic* handelt es sich um eine HTML-Anwendung, die eine ganze Reihe fertiger Listings für Visual Basic- und Java-Skripte (aber auch für Python und Perl) enthält. Das Erzeugen eines fertigen Skripts setzt lediglich die Auswahl des WMI-Namespace und der WMI-Klasse wie auch die Angabe des Computernamens voraus. Daraufhin wird ein Listing erzeugt, das Sie per Mausklick ausführen und womit Sie einen Großteil der Eigenschaften, die durch die jeweilige WMI-Klasse bereitgestellt werden, abfragen und am Bildschirm ausgeben lassen können. Die Skripte können gespeichert und für künftige Abfragen verwendet, aber auch kopiert und damit in eigene Skriptdateien integriert werden.

Ein WMI-Skript erzeugen

Zugegeben, *Scriptomatic* bietet eine sehr einfache und effiziente Möglichkeit, Systeminformationen mit Hilfe eines Skriptes, das die WMI-Schnittstelle nutzt, abzufragen. Denn die Skripte müssen nicht von Hand erstellt werden. Der Nachteil hierbei ist jedoch, dass zum einen nicht alle verfügbaren WMI-Namespace und auch nicht alle WMI-Klassen berücksichtigt werden und dass die mit *Scriptomatic* erzeugten Skripte nicht zur Automatisierung administrativer Aufgaben genutzt werden können, da lediglich die Werte der Eigenschaften abgefragt werden. Wenn Sie also Informationen abfragen möchten, für die *Scriptomatic* keine Bausteine liefert, oder wenn Sie Änderungen an der Systemkonfiguration vornehmen wollen, so bleibt nur das Erstellen eigener WMI-Skripte.

Um eigene WMI-Skripte zu stellen, müssen Sie wissen, welche WMI-Namespace (logische Einheit zur Gruppierung von Klassen) und welche Klassen (verwaltbare Basiseinheiten) zur Verfügung stehen und welche Objekte, Eigenschaften und Methoden jeweils bereitgestellt werden. Um Sie nun nicht mit seitenlanger Theorie über das WMI-Objektmodell und mit der CIM-Hierarchie zu langweilen (und zu erschrecken!), sei an dieser Stelle auf die WMI-Referenz verwiesen, worüber Sie sich zumindest einen rudimentären Überblick über die wichtigsten Namespaces, Klassen, Objekte, Methode und Eigenschaften verschaffen können und die Sie insbesondere für die alltägliche Arbeit mit VBScript als Nachschlagereferenz verwenden können: Sie finden die WMI-Referenz auf der Internetseite http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/wmi_reference.asp.

Das Erstellen eines WMI-Skriptes gestaltet sich – zumindest grob betrachtet – folgendermaßen: Sie stellen eine Verbindung zum jeweiligen WMI-Namespace (z.B. `root\cimv2`) her und binden diesen an eine Objektvariable. Anschließend stellen Sie eine Verbindung zur darunter liegenden Klasse (z.B. `Win32_LogicalDisk`) her und binden diese ebenfalls an eine Objektvariable (diese beiden Schritte lassen sich natürlich auch miteinander verbinden und in einer einzigen Codezeile unterbringen). Dann wenden Sie entweder eine Methode an, um eine neue Instanz (am Beispiel der Klasse `Win32_LogicalDisk` wäre eine Instanz ein Laufwerk) zu erzeugen, zu verändern oder zu löschen, oder aber Sie fragen die Werte der Eigenschaften eines Objektes ab.

Der erste Schritt besteht also darin, eine Verbindung zu einem WMI-Namespace und damit zur obersten Ebene herzustellen. Der mit Abstand am häufigsten benötigte

WMI-Namespace ist `root\cimv2`. Um nun eine Verbindung zu diesem Namespace herzustellen und den Pfad an die Objektvariable `oWMI` zu binden, verwenden Sie folgende Codezeile:

```
Set oWMI = GetObject("winmgmts:\root\cimv2")
```

Damit wird die Verbindung zum WMI-Namespace `root\cimv2` des lokalen Computersystems hergestellt. Wenn Sie die Verbindung jedoch nicht zum lokalen Computersystem, sondern zu einem Remotesystem herstellen wollen, so müssen Sie den Hostnamen (oder wahlweise auch die IP-Adresse) des Remotesystems angeben. Und damit die Codezeile überschaubar bleibt und leichter angepasst werden kann, übergeben Sie den Hostnamen (oder die IP-Adresse) im Idealfall als Variable, die Sie zu Beginn des Skriptes aufführen und der Sie einfach den entsprechenden Wert zuweisen (das lokale Computersystem kann mit `."` bezeichnet werden; z.B. `strComputer = "."`):

```
strComputer = "FRA-MD-DC-001"
Set oWMI = GetObject("winmgmts:" & strComputer & _
    "\root\cimv2")
```

Was noch fehlt, ist die Personifizierung (*Impersonate*), denn nicht jede WMI-Anwendung darf so ohne weiteres auf die WMI-Schnittstelle eines Remotesystems zugreifen. Mit der Personifizierung legen Sie fest, wie viele Privilegien ein Client dem Server zugesteht. Die Personifizierung wird dabei in Form eines Levels angegeben. Folgende Levels sind möglich:

Level	Name	Beschreibung
1	Anonymous	Dieser Level legt fest, dass die Identität des Aufrufenden unbekannt ist.
2	Identify	Dieser Level legt fest, dass die Identität und auch die Privilegien für Anfragen verwendet werden dürfen.
3	Impersonate	Dieser Level legt fest, dass die Identität und auch die Privilegien des aufrufenden Prozesses verwendet werden. Dies ist der am häufigsten verwendete Personifizierungslevel.
4	Delegate	Dieser Level legt fest, dass die Identität und auch die Privilegien des aufrufenden Prozesses an andere Objekte weitergereicht werden dürfen.

Tabelle 13: Die Impersonation-Level und ihre Bedeutung

Die nachfolgende Codezeile zeigt, wie der Impersonation-Level *Impersonate* übergeben wird:

```
Set oWMI = GetObject("winmgmts:" & _
    "{impersonationLevel=impersonate}!/" & strComputer & _
    "\root\cimv2")
```

Damit haben Sie bereits die erste Hürde genommen. Was noch fehlt, ist das Herstellen einer Verbindung zur WMI-Klasse (welche Klassen es gibt, können Sie der WMI-Referenz entnehmen). Um beispielsweise eine Verbindung zur WMI-Klasse *Win32_LogicalDisk* herzustellen, fügen Sie dem WMI-Namespace zunächst einen Doppelpunkt an und übergeben dann den Namen der Klasse:

```
Set oWMI = GetObject("winmgmts:" & _
    "{impersonationLevel=impersonate}!/" & strComputer & _
    "\root\cimv2:Win32_LogicalDisk")
```

Das erste Konstrukt, das als Bestandteil vieler Skripte, die in diesem Buch vorgestellt werden, verwendet wird, ist damit erstellt.

Nun ist es so, dass viele Klassen (z.B. *Win32_LogicalDisk*) mehrere Instanzen beinhalten. Unter einer Instanz ist dabei ein einzelnes Objekt zu verstehen, dessen Objekttyp durch die Klasse beschrieben wird. Dies lässt sich sehr gut am Beispiel der WMI-Klasse *Win32_LogicalDisk* verdeutlichen: Die WMI-Klasse *LogicalDisk* besitzt Informationen zu den einzelnen Laufwerken, die sich im Computersystem befinden. Jedes einzelne Laufwerk stellt dabei eine so genannte Instanz dar, die jeweils abgefragt und auch modifiziert werden kann.

Wenn Sie also eine Verbindung zur WMI-Klasse *Win32_LogicalDisk* hergestellt haben, müssen Sie die betreffenden Instanzen selektieren, um sie abfragen oder modifizieren zu können. Hierfür können Sie eine WQL-Abfrage (*WMI Query Language*) verwenden, die einer SQL-Abfrage gleicht. Um beispielsweise das Laufwerk mit dem Laufwerkbuchstaben *E:* zu selektieren und um das Objekt an eine Objektvariable zu binden, nutzen Sie die folgende beiden Codezeile:

```
Set colDrives = oWMI.ExecQuery ("SELECT * FROM _
    Win32_LogicalDisk WHERE Name = 'E:'")
```

Daraufhin werden alle Laufwerke, deren Laufwerkbuchstaben gleich *E:* ist, in Form einer Kollektion zurückliefert. Und auf die Inhalte dieses Arrays können Sie dann mit Hilfe einer *For Each*-Schleife zugreifen und im weiteren Verlauf eine Methode anwenden oder Eigenschaften auslesen. Wie einzelne Methode angewendet und Eigenschaften abgefragt werden, wird in *Teil II* des Buches noch ausführlich beschrieben und auch angewendet, weshalb an dieser Stelle nicht näher darauf eingegangen wird.

WSH deaktivieren und wieder aktivieren

Gerade in der produktiven Serverumgebung besteht oftmals der Bedarf, den WSH aus Sicherheitsgründen zu deaktivieren, um die Ausführung von Visual Basic- und Java-Skripten unterbinden und um möglichen Sabotage-Attacken durch Malware oder fehlerhafte Skripte vorbeugen zu können.

Wenn Sie keine Skripte zur Automatisierung von Aufgaben verwenden und z.B. ausschließlich die Eingabeaufforderung nutzen, so können Sie mit Hilfe einer kleinen Änderung innerhalb der Registrierung den WSH deaktivieren und damit die Skriptausführung (wahlweise für alle oder nur für bestimmte Benutzer) unterbinden. Wenn Sie den WSH beispielsweise vollständig deaktivieren möchten und damit keinem

Benutzer die Ausführung von Visual Basic- oder Java-Skripten gestatten möchten, so legen Sie im Schlüssel HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS SCRIPT HOST\SETTINGS einen neuen DWORD-Wert mit der Bezeichnung *Enabled* an und weisen diesem DWORD-Wert den Wert 0 zu.

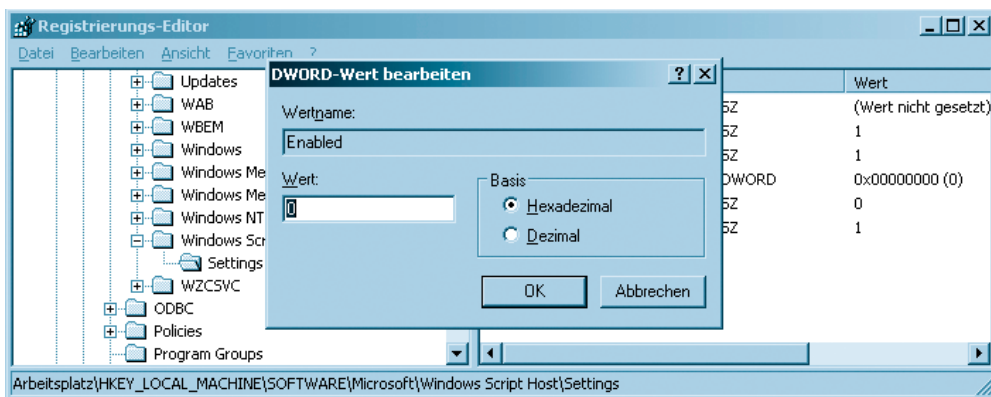


Abbildung 38: Den WSH vollständig deaktivieren

Wenn Sie diese Einstellung vorgenommen haben und nun jemand versucht, ein Skript mit Hilfe des Windows Script Hosts ausführen zu lassen, so erhält der Benutzer automatisch eine Fehlermeldung, dass der Windows Script Host deaktiviert wurde. Die Ausführung von Visual Basic- und Java-Skripten ist erst dann wieder möglich, wenn Sie entweder den soeben hinzugefügten Wert wieder aus der Registrierung entfernen oder dem DWORD-Wert *Enabled* den Wert 1 zuweisen.

Die Einstellung müssen Sie nicht unbedingt systemweit vornehmen. Denn Sie können den DWORD-Wert *Enabled* auch in den benutzerspezifischen Einstellungen hinterlegen. Wenn Sie den Windows Script Host beispielsweise für den gerade angemeldeten Benutzer deaktivieren möchten, so öffnen Sie hierzu den Registrierungs-Editor und wechseln in den Schlüssel HKEY_CURRENT_USER\SOFTWARE\MICROSOFT\WINDOWS SCRIPT HOST\SETTINGS und legen dort einen neuen DWORD-Wert mit der Bezeichnung *Enabled* an, dem Sie den Wert 0 zuweisen. Alternativ können Sie die Einstellung auch in den jeweiligen Benutzerprofilen hinterlegen, die im Hauptzweig HKEY_USERS abgelegt sind (die Identifizierung der Benutzer erfolgt über die SID, siehe hierzu *Benutzer und Gruppen* in Teil II des Buches).

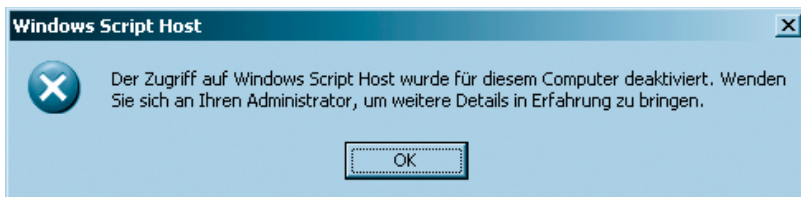
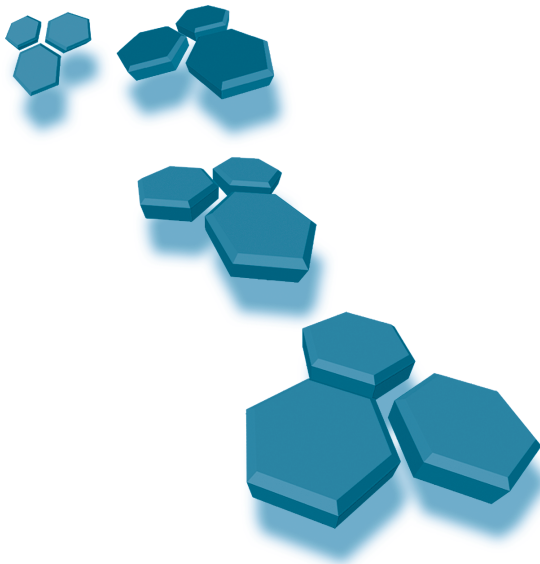


Abbildung 39: Der Windows Scripting Host wurde deaktiviert.

Teil II Rezepte



Benutzer und Gruppen

Dateien und Ordner

Datenträgerverwaltung

Netzwerk

Registrierung

Systemmanagement

Server-Rollen

Automatisierung

Benutzer und Gruppen

Die Benutzer- und Gruppenverwaltung zählt mit zu den wichtigsten Verwaltungsaufgaben im Windows Server 2003-Umfeld. Denn bevor ein Benutzer auf die Ressourcen eines einzelnen Servers oder auf die einer Domäne zugreifen kann, muss der Administrator zunächst ein entsprechendes Benutzerkonto einrichten und die Anmeldeinformationen (Benutzername und Kennwort) hinterlegen wie auch die Gruppenmitgliedschaften zur Vergabe der Benutzerrechte definieren. Erst dann kann sich der Benutzer durch Eingabe des Benutzernamens und des Kennwortes (oder durch Verwendung einer SmartCard) dem System gegenüber als gültiger Benutzer authentifizieren und – je nach Berechtigung – auf die bereitgestellten Ressourcen zugreifen.

Die Verwaltung der Benutzer- und Gruppenkonten beschränkt sich jedoch nicht nur auf das Anlegen neuer Konten, sondern umfasst auch das Ändern und Löschen bestehender Konten, das vorübergehende Deaktivieren von Benutzerkonten, das Auslesen der Eigenschaften von Benutzer- und Gruppenkonten, das Zurücksetzen von Kennwörtern wie auch das Abfragen bestimmter Ereignisse (wann hat sich ein bestimmter Benutzer zuletzt angemeldet, welcher Benutzer ist gerade angemeldet, welche SID hat der Benutzer oder die Gruppe etc.).

Die meisten der soeben genannten Verwaltungsaufgaben lassen sich mit der Benutzerverwaltung von Windows Server 2003 auf recht einfache Art und Weise durchführen. Für viele Aufgaben müssen Sie jedoch auf die Werkzeuge der Eingabeaufforderung zurückgreifen oder entsprechende Skripte einsetzen, da die Benutzerverwaltung für einige dieser Aufgaben keine Funktionen bereitstellt. Dies gilt insbesondere dann, wenn Sie spezielle Benutzer- und Gruppeninformationen auslesen oder Aufgaben automatisieren möchten.

Die nachfolgenden Rezepte zeigen daher, wie Sie die gängigsten Aufgaben aus der Benutzer- und Gruppenverwaltung unter grafischer Oberfläche, in der Eingabeaufforderung und auch mit VBScript durchführen und wie Sie viele Aufgaben ein Stück weit vereinfachen und natürlich auch automatisieren können.

Exkurs

Lokale Benutzer und Gruppen vs. Domänenbenutzer und -gruppen

In der Benutzer- und Gruppenverwaltung von Windows Server 2003 gilt es, zwischen lokalen Benutzern und Gruppen und Domänenbenutzern und -gruppen zu unterscheiden. Lokale Konten werden direkt auf einem Stand-alone-Server in einer Arbeitsgruppe oder auf einem Member-Server in einer Domäne angelegt und ermöglichen die Anmeldung des Benutzers lediglich auf dem Computersystem, auf dem das Konto angelegt wurde. Die Kontoinformationen werden in der lokalen Sicherheitsdatenbank (SAM = Security Accounts Manager) abgelegt und nicht per Replikation an andere Computersysteme übertragen.

Domänenkonten dagegen werden zentral auf einem Domänen-Controller eingerichtet und ermöglichen damit die Anmeldung von jedem Computersystem aus, das Mitglied der Domäne ist. Die Kontoinformationen werden in der zentralen Active Directory-Datenbank hinterlegt und per Replikation an andere Domänen-Controller (falls vorhanden) übertragen.

1 Existiert der Benutzer bereits?

Bevor Sie einen neuen Benutzer anlegen, sollten Sie zunächst überprüfen, ob der betreffende Benutzer nicht schon existiert. Denn Benutzer müssen auf lokalen Computersystemen und auch innerhalb einer Domäne eindeutig sein und können daher nicht mehrfach angelegt werden.

Um unter grafischer Oberfläche das Vorhandensein eines lokalen Benutzers zu überprüfen, rufen Sie auf einem Standalone- oder Member-Server die lokale Benutzerverwaltung (LOKALE BENUTZER UND GRUPPEN) auf und schauen einfach nach, ob der gewünschte Benutzer bereits existiert. Möchten Sie beispielsweise wissen, ob bereits ein Benutzer mit dem Benutzernamen *TestUser1* erstellt wurde, so öffnen Sie hierzu das STARTMENÜ, klicken auf AUSFÜHREN und geben `lusrmgr.msc` ein. Sie gelangen daraufhin in das Dialogfenster LOKALE BENUTZER UND GRUPPEN. Markieren Sie in der linken Fensterhälfte den Ordner BENUTZER, und prüfen Sie in der rechten Fensterhälfte anhand der Einträge, ob der Benutzer schon vorhanden ist.

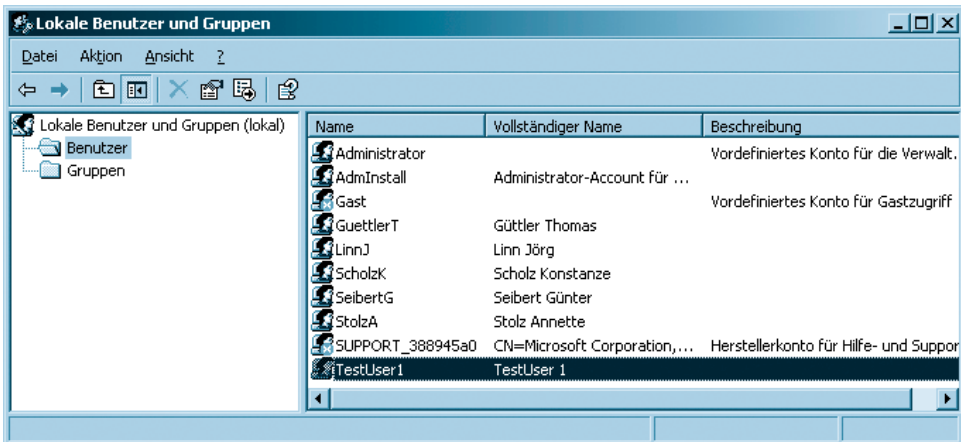


Abbildung 40: Der Benutzer »Testuser1« existiert bereits.

Möchten Sie dagegen das Vorhandensein eines Domänenbenutzers (z.B. *TestUser1*) überprüfen, so rufen Sie auf dem Domänen-Controller das Verwaltungswerkzeug ACTIVE DIRECTORY-BENUTZER UND -COMPUTER auf. Dieses Verwaltungswerkzeug finden Sie im STARTMENÜ unter VERWALTUNG. Sie können ACTIVE DIRECTORY-BENUTZER UND -COMPUTER aber auch direkt aufrufen, wenn Sie das STARTMENÜ öffnen, auf AUSFÜHREN klicken und `Dsa.msc` eingeben.

Sofern Sie keine OU-Struktur eingerichtet haben und alle Benutzerkonten im Container USERS anlegen, so brauchen Sie lediglich den zuvor genannten Container zu markieren. Die Benutzerkonten werden daraufhin in der rechten Fensterhälfte angezeigt. Ist der gewünschte Benutzer (z.B. *TestUser1*) in der Liste der Benutzer enthalten, so existiert der Benutzer bereits.

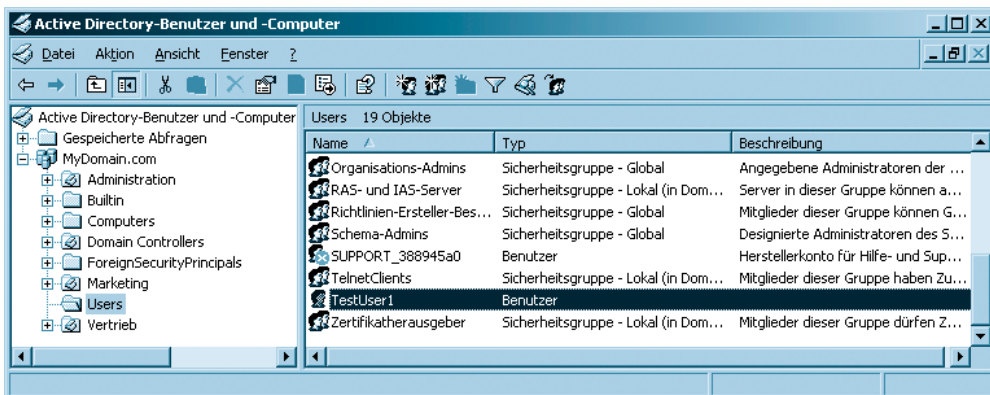


Abbildung 41: Der Domänenbenutzer »TestUser1« ist bereits vorhanden.

Wenn Sie dagegen mehrere Organisationseinheiten eingerichtet und die Benutzer direkt in der OU abgelegt haben, so ist es meist einfacher, sich die Liste aller Benutzer über die Such-Funktion anzeigen zu lassen. Denn wenn Sie die genaue OU nicht kennen, in der das Benutzerkonto abgelegt wurde, so müssten Sie sämtliche Organisationseinheiten durchsehen. Wählen Sie der Einfachheit halber im Menü AKTION den Eintrag SUCHEN aus, und suchen Sie innerhalb der Domäne nach allen Benutzern, Gruppen und Kontakten. Wenn der gewünschte Benutzer nicht in der Liste erscheint, so existiert das Benutzerkonto noch nicht.

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus das Vorhandensein eines lokalen Benutzers überprüfen möchten, so können Sie hierfür den `Net User`-Befehl einsetzen, dem Sie den Namen des betreffenden Benutzerkontos als Parameter übergeben. Der `Net User`-Befehl versucht daraufhin, eine Verbindung zu dem angegebenen Benutzerkonto herzustellen und dessen Eigenschaften am Bildschirm auszugeben. Kann das angegebene Benutzerkonto allerdings nicht gefunden oder zumindest nicht darauf zugegriffen werden, so gibt der `Net User`-Befehl stattdessen die Meldung `Der Benutzername konnte nicht gefunden werden aus.`

```
@Echo off
Net User TestUser5
```

Listing 24: 2_01_001a.bat

128 >> Existiert der Benutzer bereits?

Ohne Angabe weiterer Parameter werden die Benutzerkonten des Computersystems abgefragt, auf dem der Befehl ausgeführt wird. Wenn Sie von einem Client oder von einem Member-Server aus ein Domänenbenutzerkonto abfragen möchten, so müssen Sie den Befehl mit Hilfe des Parameters `/Domain` explizit gegen die Domäne ausführen, da anderenfalls die lokalen Benutzerkonten überprüft werden.

```
@Echo off
Net User TestUser5 /Domain
```

Listing 25: 2_01_001b.bat

Eine weitere Möglichkeit, ein Benutzerkonto zu überprüfen, bietet der WMIC-Befehl unter Verwendung des Alias `UserAccount`, womit Sie von der Eingabeaufforderung aus eine Verbindung zur WMI-Klasse `Win32_UserAccount` herstellen und die Eigenschaften der Benutzerobjekte auslesen können.

Um beispielsweise zu ermitteln, ob der Benutzer `TestUser5` bereits existiert, wenden Sie den WMIC `UserAccount`-Befehl an und selektieren mit Hilfe der `Where`-Klausel, der Sie den Namen des gesuchten Benutzers übergeben, das entsprechende Benutzerkonto. Anschließend fragen Sie mit `Get` den Wert der Eigenschaft `Name` ab. Der WMIC-Befehl versucht daraufhin, eine Verbindung zu dem angegebenen Benutzerkonto herzustellen und den Wert von `Name` abzufragen. Existiert das Benutzerkonto, so wird der Name des Benutzers am Bildschirm ausgegeben. Anderenfalls – wenn das Benutzerkonto nicht existiert – gibt der WMIC-Befehl die Meldung `Keine Instanzen verfügbar` aus.

```
@Echo off
WMIC UserAccount Where "Name='TestUser'" Get Name
```

Listing 26: 2_01_001c.bat

Die Lösung mit VBScript

Benutzerkonten lassen sich natürlich auch mit VBScript abfragen. Wenn Sie z.B. herausfinden möchten, ob auf einem bestimmten Computersystem ein lokales Benutzerkonto für `TestUser5` existiert, so stellen Sie mit der `GetObject`-Methode eine Verbindung zum ADSI-Provider `WinNT` des betreffenden Computersystems her und setzen die `Filter`-Eigenschaft mit dem Filterkriterium `User` ein, um eine Kollektion der Benutzerkonten zu erhalten.

Mit einer `FOR EACH`-Schleife können Sie dann für jedes Benutzerkonto prüfen, ob der Wert der Eigenschaft `Name` gleich `TestUser5` ist. Trifft diese Bedingung zu, dann können Sie die `For Each`-Schleife mit `Exit For` vorzeitig verlassen, da aufgrund der Eindeutigkeit der Benutzerkonten keine weiteren Ergebnisse geliefert werden.

>> Benutzer und Gruppen

```

Option Explicit

Dim oComputer
Dim strComputer
Dim strUser
Dim strUserName

strComputer = "."
strUserName = "TestUser5"

Set oComputer = GetObject("WinNT://" & strComputer)
oComputer.Filter = Array("User")

For Each strUser in oComputer
    If strUser.Name = strUserName Then
        WScript.Echo "Das Benutzerkonto existiert ..."
        Exit For
    End If
Next

Set oComputer = Nothing

```

Listing 27: 2_01_001a.vbs

Die Domänenbenutzer lassen sich dagegen per LDAP abfragen, wobei Sie die Möglichkeit haben, die Suche auch auf bestimmte Organisationseinheiten oder Container zu beschränken. Wenn Sie beispielsweise herausfinden möchten, ob in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* ein Domänenbenutzer namens *TestUser5* existiert, dann stellen Sie mit der *GetObject*-Methode eine Verbindung zum ADSI-Provider LDAP der Domäne her und setzen die *Filter*-Eigenschaft mit dem Filterkriterium *User* ein. Sie erhalten daraufhin eine Kollektion der Benutzerkonten zurück und können mit Hilfe einer *For Each*-Schleife für jedes Benutzerkonto überprüfen, ob der Wert der Eigenschaft *SamAccountName* gleich *TestUser5* ist. Und wenn die Bedingung zutrifft, verlassen Sie die Schleife vorzeitig mit *Exit For*.

```

Option Explicit

Dim oDomain
Dim strDomain
Dim strUser
Dim strUserName

strDomain = "OU=Vertrieb,DC=MyDomain,DC=Com"
strUserName = "TestUser5"

```

Listing 28: 2_01_001b.vbs

130 >> Benutzer auflisten

```

Set oDomain = GetObject ("LDAP://" & strDomain)
oDomain.Filter = Array("User")

For Each strUser in oDomain
    If strUser.SamAccountName = strUserName Then
        WScript.Echo "Benutzerkonto existiert ..."
    Exit For
End If
Next

Set oDomain = Nothing

```

Listing 28: 2_01_001b.vbs (Forts.)

2 Benutzer auflisten

Um unter grafischer Oberfläche eine Aufstellung der bereits eingerichteten Benutzer zu erstellen, rufen Sie lediglich die Benutzerverwaltung auf (siehe Existiert der Benutzer bereits? auf Seite 126) und lassen sich die Benutzerkonten entweder nur anzeigen oder exportieren sie in eine Datei. Wie Sie dabei vorgehen, hängt davon ab, ob Sie die lokalen Benutzer eines Standalone- oder Member-Servers auslesen möchten oder aber die Domänenbenutzer, die auf dem Domänen-Controller eingerichtet sind.

Zum Auslesen der lokalen Benutzer rufen Sie auf dem Standalone- oder Member-Server **LOKALE BENUTZER UND GRUPPEN** auf und markieren den Ordner **BENUTZER**, woraufhin alle eingerichteten lokalen Benutzerkonten angezeigt werden.

Sie können die Aufstellung der lokalen Benutzerkonten natürlich auch in eine Datei exportieren. Öffnen Sie hierzu das Kontextmenü des Ordners **BENUTZER**, wählen Sie **LISTE EXPORTIEREN...** aus, und geben Sie den Dateinamen wie auch das gewünschte Dateiformat (Text- oder Unicode-Format) an.

Wenn Sie stattdessen eine Aufstellung der Domänenbenutzer benötigen, so rufen Sie das Verwaltungswerkzeug **ACTIVE DIRECTORY-BENUTZER UND -COMPUTER** auf (siehe Existiert der Benutzer bereits? auf Seite 126). Für den Fall, dass Sie alle Benutzer ausschließlich im Container **USERS** ablegen, so reicht es aus, wenn Sie den Container markieren und die Ansicht nach *Typ* sortieren. Die Benutzerkonten werden daraufhin in der rechten Fensterhälfte angezeigt.

Haben Sie jedoch mehrere Organisationseinheiten eingerichtet und die Benutzerkonten auch in den einzelnen Organisationseinheiten abgelegt, so müssen Sie eine Abfrage erstellen, um eine Aufstellung aller Benutzer zu erhalten. Hierzu öffnen Sie das Kontextmenü von **GESPEICHERTE ABFRAGEN**, wählen **NEU** und anschließend **ABFRAGE** aus, um eine neue Abfrage zu erstellen. Geben Sie im Feld **NAME** eine kurze Bezeichnung (z.B. *Benutzerliste*) ein, legen Sie den Startpunkt der Abfrage (z.B. Root der Domäne) fest, und klicken Sie anschließend auf **FESTLEGEN**, um die eigentliche Abfrage zu definieren.

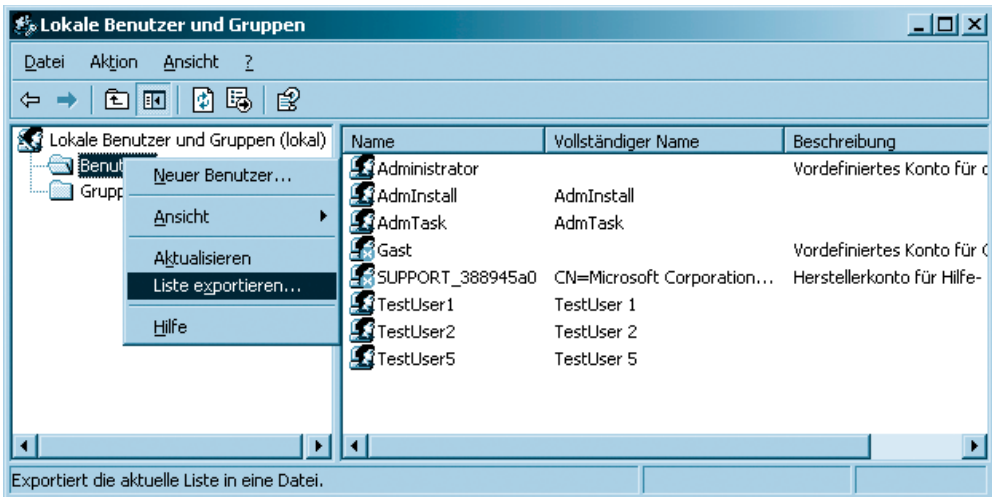


Abbildung 42: Die lokalen Benutzer anzeigen lassen und die Aufstellung exportieren

Wechseln Sie auf die Registerkarte ERWEITERT, wählen Sie im Feld SUCHEN den Eintrag BENUTZER, KONTAKTE UND GRUPPEN aus, und legen Sie unter FELD erst BENUTZER und dann NAME fest. Als Bedingung wählen Sie VORHANDEN aus. Klicken Sie auf HINZUFÜGEN, und schließen Sie das Abfragefenster wieder. Die Abfrage wird nun ausgeführt, woraufhin die bereits eingerichteten Benutzer in der rechten Fensterhälfte angezeigt werden.

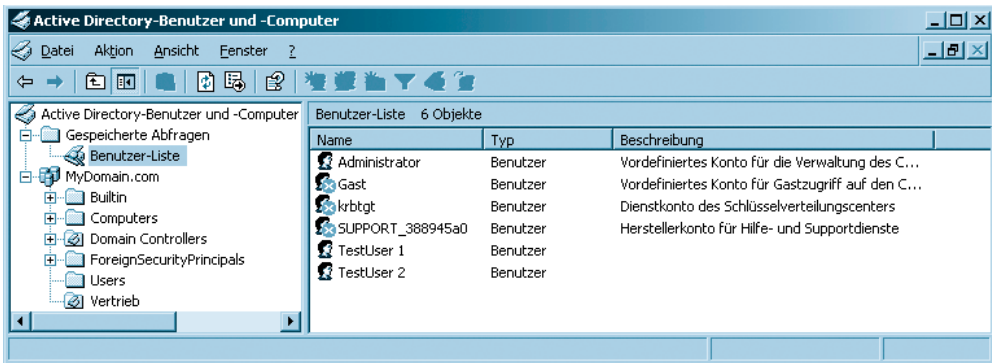


Abbildung 43: Die Auflistung der Benutzer als Abfrageergebnis

Die Lösung per Eingabeaufforderung

Wenn Sie alle lokalen Benutzer eines Standalone- oder Member-Servers von der Eingabeaufforderung aus auflisten möchten, so führen Sie im einfachsten Fall den Befehl `Net User` aus, woraufhin alle Benutzerkonten am Bildschirm ausgegeben wer-

132 >> Benutzer auflisten

den. Statt das Ergebnis am Bildschirm auszugeben, können Sie den Befehl auch um eine Ausgabeumleitung erweitern und damit die Auflistung der Benutzer in eine Datei ausgeben.

```
@Echo off
Net User > C:\Temp\UserList.txt
```

Listing 29: 2_01_002a.bat

Möchten Sie von einem Client oder von einem Member-Server aus die Domänenbenutzerkonten auslesen, so müssen Sie den `Net User`-Befehl gegen die Domäne ausführen und damit den Parameter `/Domain` anfügen:

```
@Echo off
Net User /Domain > C:\Temp\UserList.txt
```

Listing 30: 2_01_002b.bat

Die `Net User`-Abfrage können Sie sowohl zum Auslesen der lokalen Benutzer verwenden wie auch für die Auflistung der Domänenbenutzer. Letzteres lässt sich aber auch mit Hilfe des `Dsquery User`-Befehls realisieren, womit Sie die Suche noch etwas genauer spezifizieren können, da Sie sowohl den Startpunkt wie auch den Suchbereich explizit festlegen können.

- ▶ **Startpunkt:** Der Startpunkt legt fest, von welchem Punkt aus die Suche nach Benutzerkonten erfolgen soll. Geben Sie `DomainRoot` an, wenn Sie innerhalb der Domäne suchen möchten. Um stattdessen im gesamten Forest zu suchen, legen Sie `ForestRoot` fest. Des Weiteren können Sie auch einen definierten Namen als Startpunkt festlegen. Wenn Sie z.B. nur in der Organisationseinheit mit der Bezeichnung *Vertrieb* suchen möchten, so geben Sie als Startpunkt den definierten Namen der OU Vertrieb (z.B. `OU=Vertrieb,DC=MyDomain,DC=Com`) an. Ohne Angabe dieses Parameters wird `DomainRoot` verwendet.
- ▶ **Suchbereich:** Der Suchbereich gibt an, welche Ebenen nach Benutzerkonten durchsucht werden sollen. Geben Sie `-Scope Subtree` an, wenn Sie alle untergeordneten Ebenen des angegebenen Startpunktes durchsuchen möchten. Legen Sie `-Scope One-Level` fest, wenn Sie nur die erste untergeordnete Ebene des Startpunktes durchsuchen wollen, und verwenden Sie `-Scope Base`, um ausschließlich im Startpunkt selbst nach Benutzerkonten zu suchen. Ohne Angabe des Scope-Parameters wird `-Scope Subtree` verwendet.

Das nachfolgende Code-Beispiel führt eine `Dsquery User`-Abfrage gegen die Domäne (`DomainRoot`) durch und durchsucht dabei alle untergeordneten Teilstrukturen (`-Scope Subtree`). Das Ergebnis wird im Unicode-Format (`-uco`) per Ausgabeumleitung in die Datei `C:\Temp\Userlist.txt` geschrieben.


```
@Echo off
Dsquery User DomainRoot -Scope Subtree -uco > C:\Temp\UserList.txt
```

Listing 31: 2_01_002c.bat

Die Lösung mit VBScript

Um per VBScript alle lokalen Benutzer eines Standalone- oder Member-Servers auszulesen, nutzen Sie wieder den WinNT-Provider und erzeugen mit der Filter-Eigenschaft eine Kollektion aller Benutzerkonten. Anschließend fragen Sie jeweils den Wert der Eigenschaft Name ab und geben den Wert mit der WScript.Echo-Anweisung am Bildschirm aus.

```
Option Explicit

Dim oComputer
Dim strComputer
Dim strUser

strComputer = "."
strFile = "C:\Temp\UserList.txt"

Set oComputer = GetObject("WinNT://" & strComputer)
oComputer.Filter = Array("User")

For Each strUser in oComputer
    WScript.Echo strUser.Name
Next

Set oComputer = Nothing
```

Listing 32: 2_01_002a.vbs

Die Domänenbenutzer lesen Sie mit Hilfe einer LDAP-Abfrage aus, wobei Sie die Abfrage auch hier wieder auf bestimmte Container oder Organisationseinheiten beschränken können. Wenn Sie beispielsweise nur die Benutzerkonten der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* auslesen möchten, so stellen Sie mit der GetObject-Methode eine Verbindung zum ADSI-Provider LDAP her, übergeben den definierten Namen der Organisationseinheit und wenden die Filter-Eigenschaft an, um eine Kollektion der Benutzerkonten zu erstellen. Anschließend setzen Sie eine For Each-Schleife ein und geben für jedes Element der Kollektion den Wert der Eigenschaft SamAccountName aus.

134 >> Neue Benutzer anlegen

```
Option Explicit
```

```
Dim oDomain
```

```
Dim strDomain
```

```
Dim strUser
```

```
strDomain = "OU=Vertrieb,DC=MyDomain,DC=Com"
```

```
Set oDomain = GetObject ("LDAP://" & strDomain)
```

```
oDomain.Filter = Array("User")
```

```
For Each strUser in oDomain
```

```
    WScript.Echo strUser.SamAccountName
```

```
Next
```

```
Set oDomain = Nothing
```

Listing 33: 2_01_002b.vbs

3 Neue Benutzer anlegen

Dank der komfortablen Benutzerverwaltung von Windows Server 2003 sind neue Benutzer schnell erstellt. Um beispielsweise ein lokales Benutzerkonto für den Benutzer *Christine Nack* zu erstellen, ein Startkennwort zu vergeben und die Kennwortänderung nach der ersten Anmeldung zu erzwingen, rufen Sie zunächst **LOKALE BENUTZER UND GRUPPEN** auf (siehe Existiert der Benutzer bereits? auf Seite 126). Anschließend öffnen Sie das Kontextmenü des Ordners **BENUTZER** und wählen **NEUER BENUTZER...** aus, woraufhin ein Dialogfenster zum Anlegen eines neuen Benutzerkontos geöffnet wird.

Legen Sie im Feld **BENUTZERNAME** den künftigen Anmeldenamen (max. 20 Zeichen) des neuen Benutzers fest, womit sich der Benutzer dem Computersystem gegenüber später authentifizieren wird. Des Weiteren vergeben Sie im Feld **KENNWORT** ein Startkennwort, das der Benutzer bei seiner ersten Anmeldung (ebenfalls der Authentifizierung wegen) eingeben muss. Dieses Kennwort müssen Sie im Feld **KENNWORT BESTÄTIGEN** noch einmal eingeben, um mögliche Eingabefehler bei der Kennwortvergabe zu vermeiden. Und zu guter Letzt aktivieren Sie die Option **BENUTZER MUSS KENNWORT BEI DER NÄCHSTEN ANMELDUNG ÄNDERN**, um die Kennwortänderung seitens des Benutzers zu erzwingen.

Klicken Sie anschließend auf **ERSTELLEN**, um das neue Benutzerkonto anzulegen. Der neue Benutzer ist damit automatisch Mitglied der lokalen Gruppe **BENUTZER** und kann sich von nun an dem Computersystem gegenüber als gültiger Benutzer authentifizieren.

Zum Anlegen eines neuen Domänenbenutzers dagegen rufen Sie das Verwaltungswerkzeug **ACTIVE DIRECTORY-BENUTZER UND -COMPUTER** auf. Je nachdem, ob Sie das neue Benutzerkonto im Container **USERS** oder direkt in einer bestimmten Organisa-

tionseinheit anlegen möchten, öffnen Sie das Kontextmenü des Containers oder der OU und wählen erst NEU und anschließend BENUTZER aus. Sie gelangen daraufhin in ein Dialogfenster, worüber Sie die Namensinformationen eingeben können.

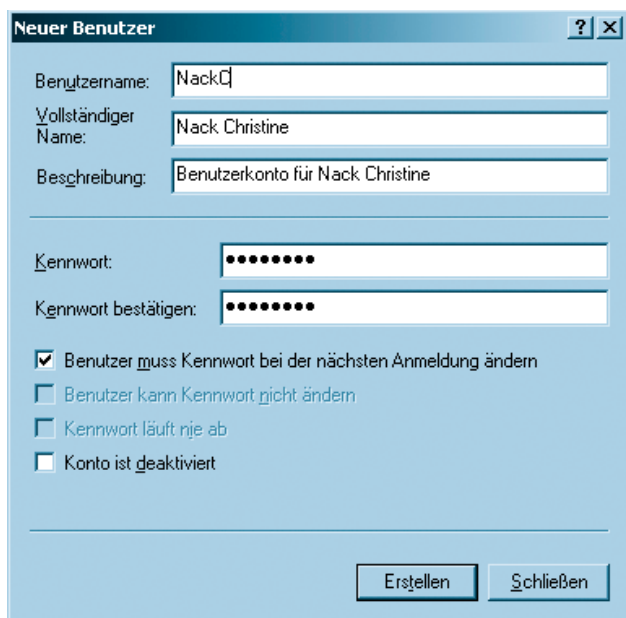


Abbildung 44: Die Anmelde- und Kennwort-Optionen für den neuen Benutzer

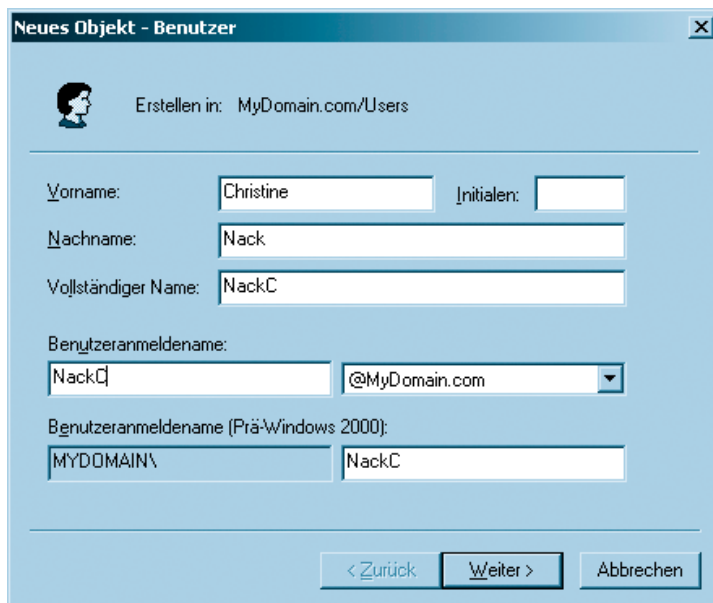


Abbildung 45: Die Namensinformationen für den neuen Benutzer anlegen

136 >> Neue Benutzer anlegen

Legen Sie zumindest den Vor- und Nachnamen des Benutzers wie auch den Benutzeranmeldenamen (z.B. *NackC*) fest, und klicken Sie auf WEITER, um das Startkennwort einzugeben, das Sie noch einmal bestätigen müssen. Des Weiteren aktivieren Sie die Option BENUTZER MUSS KENNWORT BEI DER NÄCHSTEN ANMELDUNG ÄNDERN, womit der Benutzer gezwungen wird, das Startkennwort während der ersten Anmeldung durch ein persönliches Kennwort zu ersetzen.

Klicken Sie anschließend auf WEITER und dann auf FERTIG STELLEN, um das Benutzerkonto anzulegen. Der Benutzer ist nun automatisch Mitglied der globalen Sicherheitsgruppe DOMÄNENBENUTZER und kann sich ab sofort der Domäne gegenüber als gültiger Benutzer authentifizieren.

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung lassen sich lokale Benutzer mit dem Befehl `Net User /Add` erstellen, dem Sie zumindest den Benutzernamen als Parameter übergeben müssen. Wenn Sie z.B. ein lokales Benutzerkonto für den Benutzer *Christine Nack* unter Angabe des Anmeldenamens (*NackC*), des Startkennwortes (*12345678*), des vollständigen Namens (*Nack Christine*) und einer kurzen Beschreibung zum Benutzerkonto (Benutzerkonto für *Nack Christine*) anlegen möchten, so verwenden Sie hierfür folgende Befehlszeile:

```
@Echo off
Net User NackC 12345678 /Add /FullName:"Nack Christine" _
/Comment:"Benutzerkonto für Nack Christine"
```

Listing 34: 2_01_003a.bat

Das Erstellen eines Domänenbenutzers dagegen erfolgt mit Hilfe des `Dsadd User`-Befehls, der weitaus mehr Möglichkeiten bietet als der `Net User /Add`-Befehl. Wenn Sie z.B. in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* ein neues Benutzerkonto für *Christine Nack* erstellen und mit der ersten Anmeldung eine Kennwortänderung erzwingen möchten, so verwenden Sie folgende Befehlszeile:

```
@Echo off
Dsadd User CN=NackC,OU=Vertrieb,DC=MyDomain,DC=Com _
-samid NackC -pwd 123QWEasd -mustchpwd yes
```

Listing 35: 2_01_003b.bat

Wenn Sie beim Anlegen des neuen Domänenbenutzers auch den Vor- und Nachnamen wie auch den Anzeigenamen des Benutzerkontos festlegen möchten, so verwenden Sie stattdessen die nachfolgende Befehlszeile:

>> Benutzer und Gruppen

```
@Echo off
Dsadd User CN=NackC,OU=Vertrieb,DC=MyDomain,DC=Com _
-samid NackC -fn Christine -ln Nack -display NackC _
-pwd 123QWEasd -mustchpwd yes
```

Listing 36: 2_01_003c.bat

Die Lösung mit VBScript

Wenn Sie mit VBScript einen lokalen Benutzer anlegen möchten, so stellen Sie zunächst eine Verbindung zum WinNT-Provider des betreffenden Computersystems her und wenden dann die `Create`-Methode unter Angabe des gewünschten Objekttyps (User) und des anzulegenden Benutzerkontos (`strUserName`) an. Die `Create`-Methode erzeugt dabei eine neue Instanz des User-Objektes, das Sie dann mit der `SetInfo`-Methode zurückschreiben und speichern können.

Das nachfolgende Code-Beispiel legt auf dem aktuellen Computersystem (`strComputer = "."`) ein lokales Benutzerkonto für *Christine Nack* an und setzt dabei die Eigenschaften für den Anmeldenamen (`strUserName`), den vollständigen Namen (`strFullName`), das Startkennwort (`strPassword`) und erzwingt auch die Kennwortänderung (`PasswordExpired`) bei der ersten Anmeldung durch den Benutzer:

Option Explicit

```
Dim oComputer
Dim oUser
Dim strComputer
Dim strUserName
Dim strFullName
Dim strPassword
```

```
strComputer = "."
strUserName = "NackC"
strFullName = "Nack Christine"
strPassword = "12345678"
```

```
Set oComputer = GetObject("WinNT://" & strComputer)
Set oUser = oComputer.Create("User", strUserName)
oUser.FullName = strFullName
oUser.SetPassword strPassword
oUser.Put "PasswordExpired", 1
oUser.SetInfo
```

```
Set oComputer = Nothing
Set oUser = Nothing
```

Listing 37: 2_01_003a.vbs

138 >> Neue Benutzer anlegen

Wenn Sie stattdessen im Active Directory (z.B. in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com*) einen neuen Domänenbenutzer anlegen möchten, so stellen Sie erst eine Verbindung zum LDAP-Provider her und übergeben den definierten Namen der OU, in der das Benutzerkonto angelegt werden soll. Dann erzeugen Sie mit der `Create`-Methode eine neue Instanz des `User`-Objektes, übergeben die Bezeichnung des neuen Benutzerkontos und setzen die `Put`-Methode ein, um die Eigenschaften des Benutzerkontos zu setzen, die Sie im Anschluss daran mit der `SetInfo`-Methode zurückschreiben.

Danach übergeben Sie mit der `SetPassword`-Methode das Startkennwort und aktivieren das Benutzerkonto. Letzteres ist erforderlich, da ein per VBScript erzeugtes Benutzerkonto standardmäßig deaktiviert ist. Und zu guter Letzt wenden Sie die `SetInfo`-Methode an, um die Informationen vom Cache in das Active Directory zu übertragen und alle Einstellungen damit dauerhaft zu speichern.

Option Explicit

```
Dim oOU
Dim oUser
Dim strOU
Dim strUserName
Dim strDisplayName
Dim strPassword

strOU = "OU=Vertrieb,DC=MyDomain,DC=Com"
strUserName = "NackC"
strDisplayName = "Nack Christine"
strPassword = "123QWEasd"

Set oOU = GetObject("LDAP://" & strOU)
Set oUser = oOU.Create("User", "CN=" & strUserName)
oUser.Put "SAMAccountName", strUserName
oUser.Put "DisplayName", strDisplayName
oUser.SetInfo

oUser.SetPassword strPassword
oUser.AccountDisabled=False
oUser.SetInfo

Set oOU = Nothing
Set oUser = Nothing
```

Listing 38: 2_01_003b.vbs

4 Mehrere Benutzer auf einmal anlegen

Der Windows Server 2003 stellt unter grafischer Oberfläche leider keine Möglichkeit bereit, womit Sie mehrere Benutzerkonten auf einmal anlegen können. Dies geht nur in der Eingabeaufforderung oder mit Hilfe eines Skriptes.

Wenn Sie jedoch im Active Directory mehrere Benutzerkonten anlegen müssen, dann können Sie sich die Arbeit etwas vereinfachen, indem Sie ein entsprechendes Template (Vorlage) erzeugen, die Eigenschaften festlegen und die neuen Benutzerkonten dann durch Kopieren des Templates erstellen. Denn Sie brauchen dann nur noch die Namens- und Anmeldeinformationen einzugeben, alle übertragbaren Eigenschaften werden aus dem Template übernommen und müssen somit nicht noch einmal konfiguriert werden.

Die Standardeinstellung von Windows Server 2003 sieht allerdings nicht vor, dass alle Eigenschaften eines Benutzerkontos kopiert werden, sondern nur die notwendigen (z.B. Kennwort-Attribut, Anmeldezeiten, Adress- und Organisationsinformationen ...). Sie können den Windows Server 2003 aber dazu bewegen, noch weitere Eigenschaften zu kopieren. Hierzu benötigen Sie die Managementkonsole mit der Bezeichnung ACTIVE DIRECTORY-SCHEMA, die Sie zunächst registrieren müssen, bevor Sie die Konsole das erste Mal aufrufen können: Öffnen Sie hierzu das START-MENÜ, klicken Sie auf AUSFÜHREN, und geben Sie `Regsvr32 Schmmgmt.dll` ein, um ACTIVE DIRECTORY-SCHEMA zu registrieren. Anschließend öffnen Sie eine leere MICROSOFT MANAGEMENTKONSOLE und fügen der Konsole das Snap-In ACTIVE DIRECTORY-SCHEMA hinzu. Markieren Sie den Ordner KLASSEN und unterhalb davon USER. Für alle der hier angezeigten Eigenschaften können Sie nun im Ordner ATTRIBUTE das Flag `ATTRIBUT WIRD KOPIERT, WENN EIN BENUTZER DUPLIZIERT WIRD` setzen, indem Sie die Eigenschaften des Attributs öffnen und die zuvor genannte Option anklicken (siehe Abbildung 46).

Um nun ein Template zu erstellen, starten Sie zunächst ACTIVE DIRECTORY-BENUTZER UND -COMPUTER aus dem Verwaltungsmenü heraus oder führen den Befehl `Dsa.msc` aus. Anschließend markieren Sie den Container Users und wählen im Kontextmenü NEU und dann BENUTZER aus, um in das Dialogfenster NEUES OBJEKT – BENUTZER zu gelangen. Geben Sie zumindest einen Anmeldenamen und mit einem Klick auf WEITER ein Kennwort ein, um das Template anzulegen. Danach rufen Sie die Eigenschaften des Benutzerkontos auf und legen alle Eigenschaften fest, die später mit dem Kopieren des Templates übertragen werden sollen (siehe Abbildung 47).

Wenn Sie das Template Ihren Anforderungen entsprechend angelegt haben, dann rufen Sie im Kontextmenü des Templates den Eintrag KOPIEREN... auf, woraufhin Sie wieder in das Dialogfenster NEUES OBJEKT – BENUTZER gelangen. Dort geben Sie dann die Namens- und Anmeldeinformationen für den neuen Benutzer ein und klicken auf FERTIG STELLEN. Das neue Benutzerkonto verfügt nun über sämtliche Eigenschaften, die Sie auch im Template hinterlegt haben.

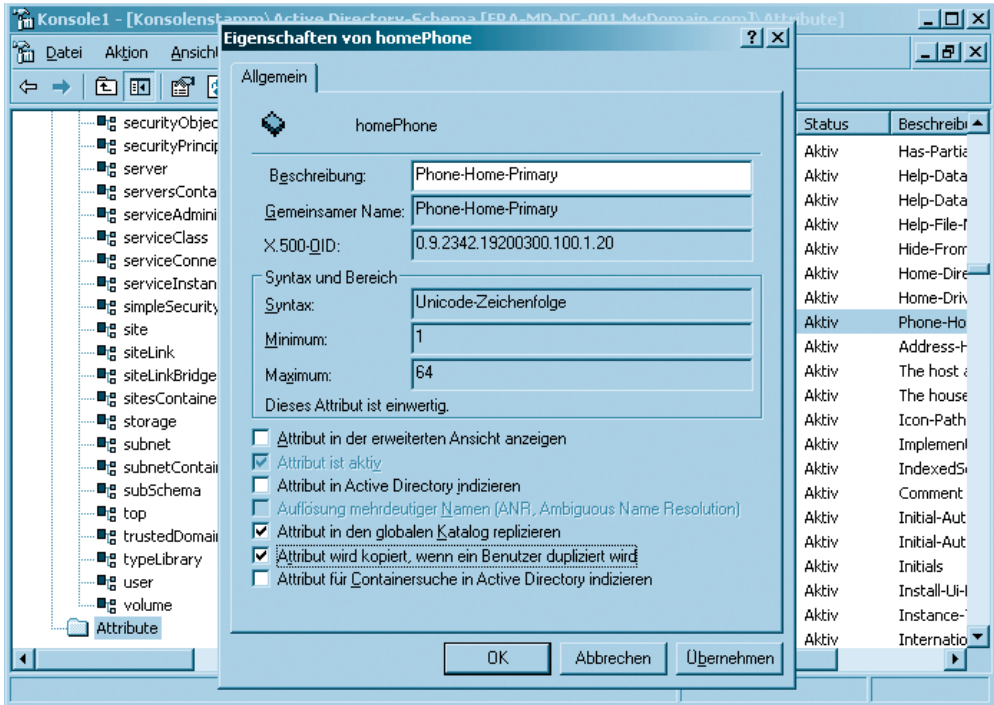


Abbildung 46: Das Attribut einer Eigenschaft ändern

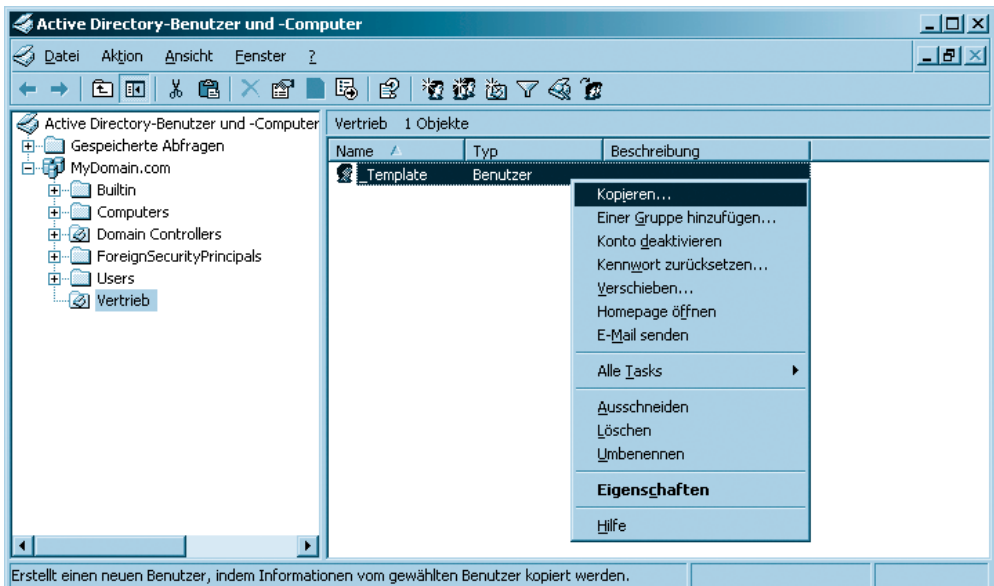


Abbildung 47: Das Template kopieren und damit einen neuen Benutzer erzeugen

Die Lösung per Eingabeaufforderung

Um von der Eingabeaufforderung aus mehrere Benutzerkonten auf einmal anzulegen, gehen Sie ähnlich vor wie beim Erstellen eines einzelnen Benutzerkontos. Sprich: Sie nutzen den `Net User-` und `Dsadd User-`Befehl und übergeben den Benutzernamen und auch das Startkennwort als Parameter. Nur mit dem Unterschied, dass Sie die Parameter diesmal nicht direkt in der `Net User-` bzw. `Dsadd User-`Befehlszeile angeben, sondern aus einer Textdatei auslesen, in der die zum Anlegen des Benutzerkontos erforderlichen Informationen hinterlegt sind.

Wenn Sie z.B. auf einem Standalone- oder Member-Server fünf lokale Benutzerkonten anlegen möchten, so wenden Sie hierfür den `Net User-`Befehl an, der den Benutzernamen und auch das dazugehörige Startkennwort (oder `*` als Aufforderung zur Eingabe des Kennwortes) als Parameter erwartet. Statt die Parameter nun mit dem Befehlsaufruf zu übergeben, hinterlegen Sie die Benutzernamen und Kennwörter der anzulegenden Benutzerkonten in einer Textdatei und führen den `Net User-`Befehl innerhalb einer `For /F`-Schleife aus, womit Sie die in der Textdatei hinterlegten Parameter zeilenweise auslesen, in eine Variable übertragen und dann an den `Net User-`Befehl übergeben können.

Die nachfolgende `For`-Befehlszeile zeigt ein solches Beispiel. Mit der Anweisung `For /F "Tokens=*" %%Var in (User.txt)` wird zunächst der Inhalt einer Zeile aus *User.txt* in die Variable `%%Var` (außerhalb einer Batch-Datei wird nur ein `%`-Zeichen verwendet) übertragen und dann an den Befehl `Net User /Add` übergeben, der daraufhin mit `Do @Net User %%Var /Add` ausgeführt wird. Die Schleife wird dabei so oft durchlaufen, wie Zeilen in *User.txt* enthalten sind.

```
@Echo off
For /F "Tokens=*" %%Var in (User.txt) Do @Net User %%Var /Add
```

Listing 39: 2_01_004a.bat

In der Datei *User.txt* sind die anzulegenden Benutzerkonten unter Angabe des Benutzernamens und auch des Kennwortes hinterlegt, wobei für jedes anzulegende Benutzerkonto eine Zeile verwendet wird.

```
TestUser1 123QWEasd
TestUser2 123QWEasd
TestUser3 123QWEasd
...
```

Auf die gleiche Art und Weise lassen sich natürlich auch mehrere Domänenbenutzer anlegen. Wenn Sie z.B. in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* fünf neue Benutzerkonten erstellen möchten, so legen Sie im ersten Schritt eine Textdatei an und hinterlegen darin zumindest die definierten Namen der neuen Benutzer wie auch die jeweiligen SAM-Account-Namen und die Startkennwörter.

142 >> Mehrere Benutzer auf einmal anlegen

```
CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com -samid TestUser1 -pwd 123QWEasd
CN=TestUser2,OU=Vertrieb,DC=MyDomain,DC=Com -samid TestUser2 -pwd 123QWEasd
CN=TestUser3,OU=Vertrieb,DC=MyDomain,DC=Com -samid TestUser3 -pwd 123QWEasd
...
```

Anschließend setzen Sie den `For /F`-Befehl ein, um die Zeilen aus der soeben erstellten Textdatei in die Variable `%Var` zu übertragen, diese an den `Dsadd User`-Befehl zu übergeben und den `Dsadd User`-Befehl mit den übergebenen Parametern daraufhin auszuführen.

```
@Echo off
For /F "Tokens=*" %%Var in (User.txt) Do @Dsadd User %%Var
```

Listing 40: 2_01_004b.bat

Die Lösung mit VBScript

Auch mit VBScript lassen sich auf recht einfache Art und Weise mehrere Benutzerkonten auf einmal anlegen. Hierzu erstellen Sie im ersten Schritt wieder eine Textdatei und nehmen darin die Benutzernamen auf, für die Sie neue Benutzerkonten anlegen wollen. Dann lesen Sie per VBScript den Inhalt dieser Textdatei zeilenweise ein und erzeugen mit der `Create`-Methode des `WinNT`- oder `LDAP`-Providers das entsprechende Benutzerkonto.

Das folgende Listing zeigt eine solche Möglichkeit und legt auf einem Standalone- oder Member-Server die neuen lokalen Benutzerkonten *TestUser1*, *TestUser2*, *TestUser3*, *TestUser4* und *TestUser5* an, setzt jeweils das Startkennwort auf *123QWEasd* und erzwingt die Änderung des Kennwortes mit der ersten Anmeldung durch den Benutzer.

Zunächst wird mit der `OpenTextFile`-Methode die Textdatei *CreateUser.txt*, in der pro Zeile ein Benutzername und (gefolgt von einem Leerzeichen) auch das Kennwort hinterlegt sind, im Lesemodus (`Read = 1`) geöffnet und per `GetObject` eine Verbindung zum `WinNT`-Provider des Servers hergestellt. Innerhalb einer `Do Loop`-Schleife wird dann der Inhalt der ersten Zeile (Benutzername und Kennwort) eingelesen. Mit Hilfe der `Instr`-Funktion lässt sich dabei die Position des ersten Leerzeichens ermitteln, womit eine Trennung des Benutzernamens und des Kennwortes vorgenommen werden kann. Die bis zum Auftreten des ersten Leerzeichens vorhandenen Zeichen werden der Variablen `strUserName` übergeben, die Zeichen hinter dem ersten Leerzeichen dagegen der Variablen `strPassword`. Der Inhalt der Variablen `strUserName` wird anschließend der `Create`-Methode des `WinNT`-Providers übergeben, womit das neue Benutzerkonto angelegt wird.

Mit der `SetPassword`-Methode wird anschließend der Eigenschaft `SetPassword` der Inhalt von `strPassword` zugewiesen, womit das Kennwort festgelegt wird. Damit der Benutzer während seiner ersten Anmeldung zur Änderung des Kennwortes aufgefordert wird, muss der Eigenschaft `PasswordExpired` der Wert `1` übergeben werden. Und zu guter Letzt werden die neuen Einstellungen mit der `SetInfo`-Methode gespeichert. Der Inhalt der Schleife wird dabei so lange wiederholt, bis das Ende der Datei *CreateUser.txt* erreicht ist.

>> Benutzer und Gruppen

Option Explicit

```
Dim oFS
Dim oFile
Dim oComputer
Dim oUser
Dim strComputer
Dim strFile
Dim strUserName
Dim strPassword
Dim strTmp
Dim intPosition1
Dim intPosition2
```

```
strComputer = "."
strFile = "CreateUser.txt"
Const Read = 1
```

```
Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFile = oFS.OpenTextFile(strFile, Read)
Set oComputer = GetObject("WinNT://" & strComputer)
```

```
Do While Not (oFile.AtEndOfStream)
    On Error Resume Next
    strTmp = oFile.ReadLine
    intPosition1 = Instr(1, strTmp, " ", 1)
    strUserName = Left(strTmp, intPosition1 - 1)
    intPosition2 = Instr(intPosition1, strTmp, " ", 1)
    strPassword = Mid(strTmp, intPosition1 + 1, intPosition2 - 1)
    Set oUser = oComputer.Create("User", strUserName)
    oUser.SetPassword strPassword
    oUser.Put "PasswordExpired", 1
    oUser.SetInfo
Loop
```

```
Set oFS = Nothing
Set oFile = Nothing
Set oComputer = Nothing
Set oUser = Nothing
```

Listing 41: 2_01_004a.vbs

Wenn Sie die neuen Benutzer nicht auf einem Standalone- oder Member-Server anlegen wollen, sondern im Active Directory, so öffnen Sie mit der `OpenTextFile`-Methode erst die Textdatei mit den Benutzernamen, stellen dann eine Verbindung zum LDAP-Provider der Domäne (oder der untergeordneten Organisationseinheit) her und wenden die `Create`-Methode unter Angabe des definierten Benutzernamens an, um das Benutzerkonto anzulegen.

144 >> Mehrere Benutzer auf einmal anlegen

Anschließend können Sie mit der `Put`-Methode weitere Eigenschaften für das Benutzerkonto festlegen. Das nachfolgende Listing legt beispielsweise noch den SAM-Namen für das Benutzerkonto ("`SAMAccountName`", `strUserName`) fest, erzwingt die Kennwortänderung mit der ersten Anmeldung ("`pwdLastSet`", 0), bestimmt das Startkennwort (`SetPassword strPassword`) und aktiviert zu guter Letzt das neue Benutzerkonto (`AccountDisabled=False`), da neue Benutzerkonten, die per VBScript erstellt werden, per Default deaktiviert sind.

Option Explicit

```
Dim oFS
Dim oFile
Dim oOU
Dim oUser
Dim strOU
Dim strFile
Dim strLine
Dim strUserName
Dim strPassword
Dim strTmp
Dim intPosition1
Dim intPosition2
```

```
strOU = "OU=Vertrieb,DC=MyDomain,DC=Com"
strFile = "CreateUser.txt"
Const Read = 1
```

```
Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFile = oFS.OpenTextFile(strFile, Read)
Set oOU = GetObject("LDAP://" & strOU)
```

```
Do While Not (oFile.AtEndOfStream)
    On Error Resume Next
    strTmp = oFile.ReadLine
    intPosition1 = Instr(1, strTmp, " ", 1)
    strUserName = Left(strTmp, intPosition1 - 1)
    intPosition2 = Instr(intPosition1, strTmp, " ", 1)
    strPassword = Mid(strTmp, intPosition1 + 1, intPosition2 - 1)
    Set oUser = oOU.Create("User", "CN=" & strUserName)
    oUser.Put "SAMAccountName", strUserName
    oUser.SetInfo
    oUser.Put "pwdLastSet", 0
    oUser.SetPassword strPassword
    oUser.AccountDisabled=False
    oUser.SetInfo
Loop
```

Listing 42: 2_01_004b.vbs

```

Set oFS = Nothing
Set oFile = Nothing
Set oOU = Nothing
Set oUser = Nothing

```

Listing 42: 2_01_004b.vbs (Forts.)

5 Eigenschaften eines Benutzers auslesen

Wenn Sie unter grafischer Oberfläche herausfinden möchten, welche Eigenschaften ein lokales Benutzerkonto zur Verfügung stellt und welche Werte jeweils hinterlegt sind, starten Sie auf dem Standalone- oder Member-Server zunächst die lokale Benutzerverwaltung (siehe Existiert der Benutzer bereits? auf Seite 126) und markieren in der linken Fensterhälfte den Ordner BENUTZER. Anschließend wählen Sie in der rechten Fensterhälfte den Benutzer aus, dessen Eigenschaften Sie einsehen möchten, und wählen im Kontextmenü den Eintrag EIGENSCHAFTEN aus.

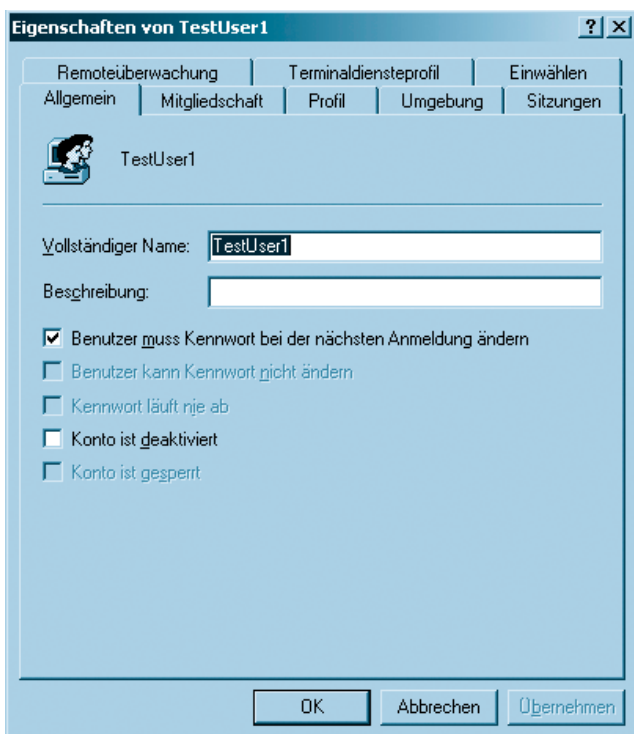


Abbildung 48: Die Eigenschaften eines lokalen Benutzers

Auf den acht Registerkarten des Dialogfensters EIGENSCHAFTEN VON BENUTZERNAME finden Sie nun alle verfügbaren Eigenschaften vor und können sich die jeweiligen

Konfigurationseinstellungen anzeigen lassen. Eine Zusammenfassung, welche Eigenschaften auf den jeweiligen Registerkarten bereitgestellt werden, zeigt die nachfolgende Tabelle.

Registerkarte	Eigenschaften
Allgemein	Auf der Registerkarte <i>Allgemein</i> sind der Benutzername und auch eine kurze Beschreibung zum Benutzerkonto zu finden. Des Weiteren werden auf dieser Registerkarte die Kennwort-Optionen bereitgestellt.
Mitgliedschaft	Die Registerkarte <i>Mitgliedschaft</i> enthält eine Auflistung aller Gruppen, in denen der Benutzer Mitglied ist.
Profil	Auf der Registerkarte <i>Profil</i> sind die Pfade für das Benutzerprofil, für das Anmeldeskript und auch für das Home-Laufwerk zu finden.
Umgebung	Die Registerkarte <i>Umgebung</i> enthält Konfigurationseinstellungen zu Terminalserver-Verbindungen, z.B. den automatischen Programmstart nach dem Herstellen einer Terminalserver-Verbindung, wie auch die Laufwerks- und Druckerumleitung.
Sitzungen	Die Registerkarte <i>Sitzungen</i> stellt ebenfalls Eigenschaften für Terminalserver-Verbindungen bereit. Über diese Registerkarte werden die Timeout-Einstellungen und die Verhaltensweisen beim Trennen einer Terminalserver-Verbindung bestimmt.
Remoteüberwachung	Auf der Registerkarte <i>Remoteüberwachung</i> sind die Eigenschaften zur Überwachung einer Terminalserver-Verbindung (Überwachung aktivieren, Benutzerberechtigung anfordern und Steuerungsebene) hinterlegt.
Terminaldienstprofil	Auf der Registerkarte <i>Terminaldienstprofil</i> sind die Eigenschaften für das Benutzerprofil während einer Terminalserver-Verbindung hinterlegt (z.B. Profilpfad, Anmeldeberechtigung).
Einwählen	Auf der Registerkarte <i>Einwählen</i> sind die Eigenschaften für die Remote-Einwahl des Benutzers zu finden. Hierüber ist festgelegt, ob sich der Benutzer per RAS einwählen darf, ob er einen Rückruf vom RAS-Server erhält und welche IP-Adresse und ggf. Route verwendet werden soll.

Tabelle 14: Die Registerkarten der Eigenschaften eines lokalen Benutzerkontos

Domänenbenutzerkonten verfügen über weitaus mehr Eigenschaften als lokale Benutzerkonten. Wenn Sie unter grafischer Oberfläche herausfinden möchten, welche Eigenschaften im Einzelnen zur Verfügung gestellt werden und welche Werte für ein bestimmtes Benutzerkonto hinterlegt sind, dann starten Sie auf dem Domänen-Controller (oder auf einem anderen Computersystem, auf dem die Verwaltungstools installiert sind) das Verwaltungswerkzeug ACTIVE DIRECTORY-BENUTZER UND -COMPUTER und rufen die Eigenschaften des jeweiligen Benutzerkontos auf. Anschließend markieren Sie den Benutzer, dessen Eigenschaften Sie einsehen möchten, und wählen im Kontextmenü des Benutzerkontos den Eintrag EIGENSCHAFTEN aus. Sie gelangen daraufhin in das Dialogfenster EIGENSCHAFTEN VON BENUTZERNAME und finden nun alle Eigenschaften des Benutzerkontos vor.

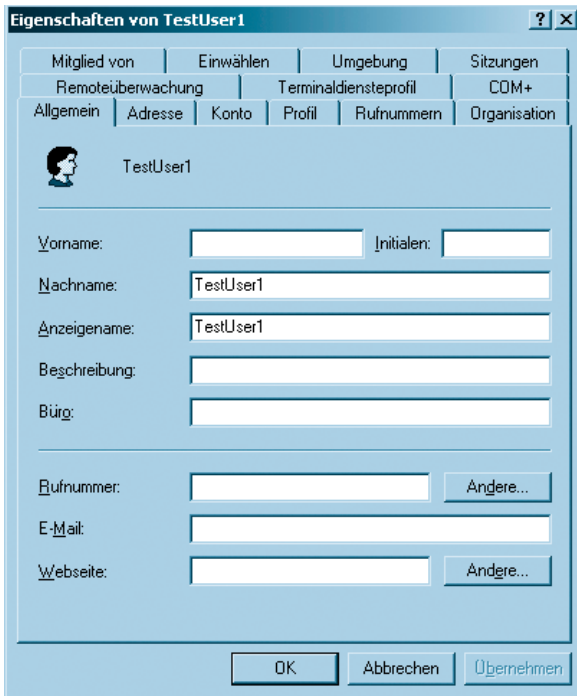


Abbildung 49: Die Eigenschaften eines Domänenbenutzerkontos

Die nachfolgende Tabelle zeigt, welche Eigenschaften auf den einzelnen Registerkarten zu finden sind:

Registerkarte	Eigenschaften
Allgemein	Auf der Registerkarte <i>Allgemein</i> sind die Eigenschaften zum Benutzernamen (Vor- und Nachname, Anzeigename) wie auch die Kontobeschreibung und Kontaktinformationen enthalten.
Adresse	Auf der Registerkarte <i>Adresse</i> sind die Eigenschaften zur Anschrift (Straße, Postfach, Ort, Bundesland, PLZ und Land) zu finden.
Konto	Die Registerkarte <i>Konto</i> enthält die Eigenschaften zum Anmeldenamen, zu den Kennwort-Optionen, zu den Anmeldezeiten und -systemen wie auch zum Ablaufdatum des Kontos.
Profil	Auf der Registerkarte <i>Profil</i> sind die Pfade für das Benutzerprofil, für das Anmeldeskript und auch für das Home-Laufwerk zu finden.
Rufnummern	Die Registerkarte <i>Rufnummern</i> stellt die Eigenschaften für Telefon-, Fax-, Mobil-, Pager- und IP-Telefonkontakte zur Verfügung.
Organisation	Auf der Registerkarte <i>Organisation</i> sind die Eigenschaften zur Position, zur Abteilung, zur Firma, zum Vorgesetzten und auch für die Mitarbeiter zu finden.

Tabelle 15: Die Registerkarten der Eigenschaften eines Domänenbenutzerkontos

Registerkarte	Eigenschaften
Remoteüberwachung	Auf der Registerkarte <i>Remoteüberwachung</i> sind die Eigenschaften zur Überwachung einer Terminalserver-Verbindung (Überwachung aktivieren, Benutzerberechtigung anfordern und Steuerungsebene) hinterlegt.
Terminaldiensteprofil	Auf der Registerkarte <i>Terminaldiensteprofil</i> sind die Eigenschaften für das Benutzerprofil während einer Terminalserver-Verbindung hinterlegt (z.B. Profilpfad, Anmeldeberechtigung).
COM+	Die Registerkarte <i>COM+</i> enthält die Eigenschaften zur Partitionsgruppe.
Mitgliedschaft	Die Registerkarte <i>Mitgliedschaft</i> enthält alle Gruppen, in denen der Benutzer Mitglied ist.
Einwählen	Auf der Registerkarte <i>Einwählen</i> sind die Eigenschaften für die Remote-Einwahl des Benutzers zu finden. Hierüber ist festgelegt, ob sich der Benutzer per RAS einwählen darf, ob er einen Rückruf vom RAS-Server erhält und welche IP-Adresse und ggf. Route verwendet werden soll.
Umgebung	Die Registerkarte <i>Umgebung</i> enthält Konfigurationseinstellungen zu Terminalserver-Verbindungen, z.B. den automatischen Programmstart nach dem Herstellen einer Terminalserver-Verbindung, wie auch die Laufwerks- und Druckerumleitung.
Sitzungen	Die Registerkarte <i>Sitzungen</i> stellt ebenfalls Eigenschaften für Terminalserver-Verbindungen bereit. Über diese Registerkarte werden die Timeout-Einstellungen und die Verhaltensweisen beim Trennen einer Terminalserver-Verbindung bestimmt.

Tabelle 15: Die Registerkarten der Eigenschaften eines Domänenbenutzerkontos

Möglicherweise sehen Sie noch zusätzliche Registerkarten für die Eigenschaften, die ein Exchange-Server bzw. die ein Zertifikatsserver verwendet, falls die entsprechenden Managementmodule installiert sind.

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus die Eigenschaften eines Benutzerkontos auslesen möchten, so verwenden Sie hierfür die Befehle `Net User` und `Dsget User`, wobei Sie den `Dsget User`-Befehl aber nur innerhalb einer Domäne einsetzen können.

Wenn Sie z.B. auf einem Standalone- oder Member-Server die Eigenschaften des lokalen Benutzerkontos für `TestUser1` auslesen möchten, so wenden Sie den `Net User`-Befehl gegen das lokale Computersystem an und übergeben den Benutzernamen als Parameter. Der `Net User`-Befehl versucht daraufhin, eine Verbindung zu dem angegebenen Benutzerkonto herzustellen und dessen Eigenschaften am Bildschirm auszugeben. Mit Hilfe einer Ausgabeumleitung lässt sich das Ergebnis aber auch in eine Datei schreiben (oder an ein sonstiges Ausgabegerät senden):


```
@Echo off
Net User TestUser1 > TestUser1.txt
```

Listing 43: 2_01_005a.bat

Die nachfolgende Tabelle zeigt, welche Eigenschaften durch eine `Net User`-Abfrage ausgegeben werden:

Eigenschaft	Beschreibung
Benutzername	Anmeldename des Benutzers
Vollständiger Name	Vollständiger Name des Benutzers (z.B. Vor- und Nachname)
Beschreibung	Beschreibung zum Benutzerkonto
Benutzerbeschreibung	Beschreibung zum Benutzer
Ländereinstellung	Ländercode, der verwendet wird
Konto aktiv	Status des Benutzerkontos (aktiviert oder deaktiviert)
Konto abgelaufen	Datum und Uhrzeit, wann das Benutzerkonto zuletzt abgelaufen ist
Letztes Setzen des Kennwortes	Datum und Uhrzeit der letzten Kennwortänderung
Kennwort läuft ab	Datum und Uhrzeit, wann das Kennwort seine Gültigkeit verliert
Kennwort änderbar	Benutzer darf das Kennwort ändern (Ja) oder nicht (Nein)
Kennwort erforderlich	Benutzer muss ein Kennwort vergeben (Ja) oder nicht (Nein)
Benutzer kann Kennwort ändern	Benutzer darf das Kennwort ändern (Ja) oder nicht (Nein)
Erlaubte Arbeitsstationen	Auflistung der Computersysteme, die eine Anmeldung erlauben
Anmeldeskript	Pfad des Anmeldeskriptes
Benutzerprofil	Pfad des Benutzerprofils
Basisverzeichnis	Pfad des Home-Laufwerkes
Letzte Anmeldung	Datum und Uhrzeit der letzten Anmeldung
Erlaubte Anmeldezeiten	Auflistung der erlaubten Anmeldezeiten
Lokale Gruppenmitgliedschaften	Lokale Gruppen, in denen der Benutzer Mitglied ist
Globale Gruppenmitgliedschaften	Globale Gruppen, in denen der Benutzer Mitglied ist

Tabelle 16: Die Eigenschaften, die mit dem Net User-Befehl ausgelesen werden können

150 >> Eigenschaften eines Benutzers auslesen

Möchten Sie die Eigenschaften eines Domänenbenutzerkontos auslesen, so können Sie ebenfalls den `Net User`-Befehl einsetzen, allerdings müssen Sie die `Net User`-Abfrage mit dem zusätzlichen Parameter `/Domain` explizit gegen die Domäne richten. Denn standardmäßig wird die Abfrage gegen die lokale Benutzerdatenbank des Computersystems gerichtet, auf dem der Befehl ausgeführt wird.

```
@Echo off
Net User TestUser1 /Domain > TestUser1.txt
```

Listing 44: 2_01_005b.bat

Die Eigenschaften eines Domänenbenutzerkontos können Sie alternativ dazu auch mit dem `Dsget User`-Befehl auslesen, wobei Sie in diesem Fall aber die jeweiligen Eigenschaften, die ausgelesen werden sollen, explizit angeben müssen. Die nachfolgende Tabelle zeigt die wichtigsten Eigenschaften, eine vollständige Übersicht finden Sie in *Teil III* dieses Buches.

Parameter	Beschreibung
-dn	Definierter Name des Benutzers
-samid	Anmeldename des Benutzers
-sid	SID des Benutzers
-fn	Vorname des Benutzers
-ln	Nachname des Benutzers
-display	Anzeigename des Benutzers
-desc	Beschreibung zum Benutzerkonto
-hmdir	Ordner des Home-Laufwerks
-hmdrv	Laufwerk des Home-Laufwerks
-profile	Pfad des Benutzerprofils
-loscr	Pfad zum Anmeldeskript
-mustchpwd	Benutzer muss das Kennwort bei der nächsten Anmeldung ändern (Ja) oder nicht (Nein)
-canchpwd	Benutzer kann das Kennwort ändern (Ja) oder nicht (Nein)
-pwdneverexpires	Gibt an, ob das Kennwort nie abläuft (Ja)
-disabled	Gibt an, ob das Benutzerkonto aktiviert (Nein) oder deaktiviert (Ja) ist
-acctexpires	Datum und Uhrzeit, wann das Benutzerkonto seine Gültigkeit verliert
-reversiblepwd	Gibt an, ob das Kennwort mit umkehrbarer Verschlüsselung gespeichert wird (Ja) oder nicht (Nein)
-memberof	Gruppenmitgliedschaften
-expand	Rekursive Abfrage der Gruppenmitgliedschaften

Tabelle 17: Die Eigenschaften, die mit dem `Dsget`-Befehl ausgelesen werden können

Um beispielsweise die Namensinformationen (Benutzername, Vor- und Nachname) für den Domänenbenutzer *TestUser1* abzufragen, wenden Sie den `Dsget User`-Befehl an und übergeben den definierten Namen des Benutzerkontos wie auch die Parameter `-samid` (Benutzername), `-fn` (Vorname) und `-ln` (Nachname). Und der besseren Übersicht wegen fügen Sie noch den Parameter `-L` hinzu, damit die Ausgabe in Form einer Liste und nicht im Tabellenformat erfolgt:

```
@Echo off
Dsget User CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com _
-samid -fn -ln -L
```

Listing 45: 2_01_005c.bat

Die Lösung mit VBScript

Auch mit VBScript lassen sich die Eigenschaften eines Benutzerkontos auf recht einfache Art und Weise abfragen. Denn Sie müssen im Grunde genommen nur eine Verbindung zum WinNT- oder LDAP-Provider des betreffenden Computersystems oder zu der Domäne herstellen, das entsprechende User-Objekt referenzieren und die Werte der gewünschten Eigenschaften ermitteln.

Die nachfolgende Tabelle enthält eine Aufstellung der wichtigsten Eigenschaften unter Angabe einer kurzen Beschreibung und der jeweiligen Unterstützung durch den WinNT- und LDAP-Provider. Eine vollständige Aufstellung der Eigenschaften finden Sie in *Teil III* dieses Buches.

Eigenschaft	WinNT	LDAP	Beschreibung
AccountDisabled	•	•	Legt fest, ob das angegebene Benutzerkonto aktiviert (<i>no</i>) oder deaktiviert (<i>yes</i>) ist
AccountExpirationDate	•	•	Legt das Datum fest, an dem das Benutzerkonto seine Gültigkeit verliert
Description	•	•	Legt eine kurze Beschreibung zum Benutzerkonto fest
FirstName		•	Legt den Vornamen des Benutzers fest
FullName	•	•	Legt den vollständigen Namen des Benutzers fest (z.B. Vor- und Nachname)
Groups	•	•	Legt die Gruppenmitgliedschaften des Benutzers fest
HomeDirectory	•	•	Legt den Pfad für das Home-Laufwerk fest
LastFailedLogin		•	Liefert das Datum und auch die Uhrzeit der letzten fehlerhaften Anmeldung zurück (read-only)

Tabelle 18: Die wichtigsten Eigenschaften eines ADSI-User-Objektes

Eigenschaft	WinNT	LDAP	Beschreibung
LastLogin	•	•	Liefert das Datum und auch die Uhrzeit der letzten erfolgreichen Anmeldung zurück (read-only)
LastLogoff	•	•	Liefert das Datum und auch die Uhrzeit der letzten Abmeldung zurück (read-only)
LastName		•	Legt den Nachnamen des Benutzers fest
LoginHours	•	•	Legt fest, zu welchen Uhrzeiten eine Anmeldung erlaubt ist
LoginScript	•	•	Legt den Pfad zum Anmeldeskript fest
LoginWorkstations	•	•	Legt fest, von welchen Computersystemen aus eine Anmeldung erlaubt ist
PasswordExpirationDate	•	•	Legt das Datum und die Uhrzeit fest, wann das Kennwort seine Gültigkeit verliert
PasswordLastChanged		•	Liefert das Datum und auch die Uhrzeit der letzten Kennwort-Änderung zurück (read-only)
Profile	•	•	Legt den Pfad zum Benutzerprofil fest
SetPassword	•	•	Legt das Kennwort für den Benutzer fest

Tabelle 18: Die wichtigsten Eigenschaften eines ADSI-User-Objektes (Forts.)

Wenn Sie z.B. auf einem Standalone- oder Member-Server den Anmeldennamen und auch den vollständigen Namen des lokalen Benutzers *TestUser1* am Bildschirm ausgeben möchten, so stellen Sie zunächst eine Verbindung zum WinNT-Provider des entsprechenden Computersystems her und binden das betreffende User-Objekt an eine Objektvariable. Anschließend fragen Sie den Wert der Eigenschaft *Name* (Anmeldename) und auch der Eigenschaft *FullName* (vollständiger Name) ab und geben die Informationen mit der *WScript.Echo*-Anweisung am Bildschirm aus.

```
Option Explicit
```

```
Dim oUser
Dim strComputer
Dim strUserName
```

```
strComputer = "."
strUserName = "TestUser1"
```

```
Set oUser = GetObject("WinNT://" & strComputer & "/" & strUserName)
```

```
On Error Resume Next
WScript.Echo "Anmeldename:" & oUser.Name
```

Listing 46: 2_01_005a.vbs

>> Benutzer und Gruppen

```
WScript.Echo "Vollständiger Name: " & oUser.FullName
```

```
Set oUser = Nothing
```

Listing 46: 2_01_005a.vbs (Forts.)

Wenn Sie stattdessen die Eigenschaften eines Domänenbenutzers ermitteln möchten, so nutzen Sie anstelle des WinNT-Providers den LDAP-Provider und fragen die Werte der gewünschten Eigenschaften ab.

Das nachfolgende Code-Beispiel liest z.B. den SAM-Account-Namen wie auch den Vor- und Nachnamen des Benutzers *TestUser1* aus, der in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* abgelegt ist.

```
Option Explicit
```

```
Dim oUser
```

```
Dim strComputer
```

```
Dim strUserName
```

```
strComputer = "."
```

```
strUserName = "CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com"
```

```
Set oUser = GetObject("LDAP://" & strUserName)
```

```
On Error Resume Next
```

```
WScript.Echo "Anmeldename: " & oUser.SamAccountName
```

```
WScript.Echo "Vorname: " & oUser.FirstName
```

```
WScript.Echo "Nachname: " & oUser.LastName
```

```
Set oUser = Nothing
```

Listing 47: 2_01_005b.vbs

6 Benutzer ändern

Ein vorhandenes Benutzerkonto lässt sich mit Hilfe der Benutzerverwaltung von Windows Server 2003 jederzeit ändern. Hierzu rufen Sie lediglich die Eigenschaften des Benutzerkontos auf und verändern die entsprechenden Eigenschaften (siehe auch Eigenschaften eines Benutzers auslesen auf Seite 145).

Wie Sie dabei vorgehen und welche Eigenschaften Sie ändern können, hängt nun davon ab, ob Sie ein lokales Benutzerkonto auf einem Standalone- oder Member-Server oder aber ein Domänenbenutzerkonto im Active Directory ändern möchten.

Um ein lokales Benutzerkonto zu bearbeiten und z.B. ein Anmeldeskript zu hinterlegen, wechseln Sie in die lokale Benutzerverwaltung des betreffenden Servers. Anschließend

markieren Sie den Ordner BENUTZER und rufen aus dem Kontextmenü des betreffenden Benutzerkontos heraus die EIGENSCHAFTEN auf. Auf der Registerkarte PROFIL können Sie im Feld ANMELDESKRIPT den Pfad zur Skriptdatei hinterlegen. Falls das Skript in einer Freigabe auf einem anderen Computersystem abgelegt ist, so müssen Sie sicherstellen, dass der betreffende Benutzer zumindest eine Leseberechtigung auf die Freigabe hat, da das Skript sonst nicht ausgeführt werden kann.

Wenn Sie den Pfad zum Skript hinterlegt und gegebenenfalls noch weitere Eigenschaften geändert oder hinzugefügt haben, dann schließen Sie das Dialogfenster mit einem Klick auf ÜBERNEHMEN, um die Änderung zu speichern.

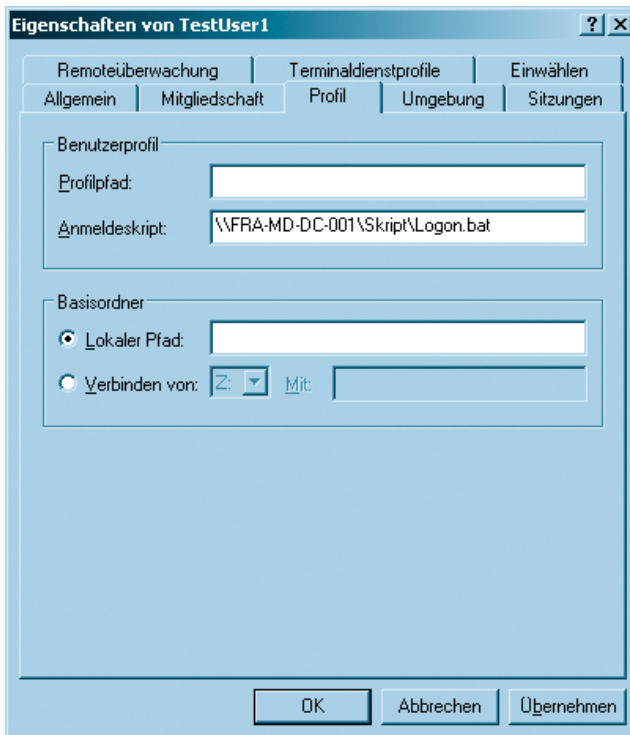


Abbildung 50: Das Anmeldeskript für den Benutzer bestimmen

Ein Domänenbenutzerkonto lässt sich auf ähnliche Art und Weise ändern: Öffnen Sie auf dem Domänen-Controller das Verwaltungsmenü, und starten Sie ACTIVE DIRECTORY-BENUTZER UND -COMPUTER. Anschließend rufen Sie aus dem Kontextmenü des Benutzerkontos die EIGENSCHAFTEN auf und wechseln auf die Registerkarte PROFIL. Dort können Sie im Feld ANMELDESKRIPT den Pfad zur Skriptdatei hinterlegen und auch über die weiteren Registerkarten natürlich noch zusätzliche Eigenschaften setzen oder entfernen. Wenn Sie alle Änderungen durchgeführt haben, schließen Sie das Dialogfenster mit einem Klick auf ÜBERNEHMEN, um die Änderungen der Eigenschaften zu speichern.

Die Lösung per Eingabeaufforderung

Der Windows Server 2003 hält entsprechende Befehle bereit, womit Sie bestehende Benutzerkonten auch von der Eingabeaufforderung aus bearbeiten können. Wenn Sie z.B. ein lokales Benutzerkonto per Befehlszeile oder aus einer Batch-Datei heraus modifizieren möchten, so können Sie hierfür den `Net User`-Befehl einsetzen, dem Sie den Benutzernamen wie auch die zu ändernde Eigenschaft als Parameter übergeben. Welche Eigenschaften verändert werden können und welche Parameter hierfür eingesetzt werden müssen, können Sie der nachfolgenden Tabelle entnehmen:

Parameter	Beschreibung
<code>/active:</code>	Legt fest, ob das angegebene Benutzerkonto aktiviert (<code>/active:yes</code>) oder deaktiviert (<code>/active:no</code>) werden soll
<code>/comment:</code>	Legt eine Beschreibung für das angegebene Benutzerkonto fest
<code>/countrycode:</code>	Legt den Ländercode für das angegebene Benutzerkonto fest. Mit dem Parameter <code>/countrycode:0</code> wird die Systemeinstellung verwendet.
<code>/expires:</code>	Legt das Datum fest, an dem das Benutzerkonto seine Gültigkeit verliert
<code>/fullname:</code>	Legt den vollständigen Namen des Benutzers fest. Enthält der vollständige Name ein oder mehrere Leerzeichen, so muss der Name in Anführungszeichen gesetzt werden.
<code>/homedir:</code>	Legt das zu verwendende Home-Laufwerk des Benutzers fest
<code>/passwordchg:</code>	Legt fest, ob der Benutzer bei der nächsten Anmeldung das Kennwort ändern (<code>/passwordchg:yes</code>) muss oder nicht (<code>/passwordchg:no</code>)
<code>/passwordreq:</code>	Legt fest, ob ein Kennwort erforderlich ist (<code>/passwordreq:yes</code>) oder nicht (<code>/passwordreq:no</code>)
<code>/profilepath:</code>	Legt den Pfad für das Benutzerprofil fest
<code>/scriptpath:</code>	Legt den Pfad für das Anmeldeskript fest
<code>/times:</code>	Legt die erlaubten Anmeldezeiten für das angegebene Benutzerkonto fest
<code>/usercomment:</code>	Legt eine Beschreibung für das Benutzerkonto fest
<code>/workstations:</code>	Legt fest, an welchen Computersystemen sich der Benutzer anmelden darf

Tabelle 19: Die Eigenschaften, die mit dem Net User-Befehl geändert werden können

Wenn Sie auf einem Standalone- oder Member-Server das lokale Benutzerkonto `TestUser1` bearbeiten und beispielsweise ein Anmeldeskript zuweisen möchten, so wenden Sie den `Net User`-Befehl unter Angabe des Benutzerkontos und des Parameters `/Scriptpath:` an, dem Sie den Pfad zur Skriptdatei übergeben:

```
@Echo off
Net User TestUser1 /Scriptpath:\\FRA-MD-DC-001\Skript\Logon.bat
```

Listing 48: 2_01_006a.bat

156 >> Benutzer ändern

Mit der gleichen Befehlszeile können Sie natürlich auch ein Domänenbenutzerkonto bearbeiten, wobei Sie den Befehl aber mit Hilfe des Parameters `/Domain` explizit gegen die Domäne ausführen müssen, sofern Sie den Befehl nicht direkt auf dem Domänen-Controller aufrufen.

```
@Echo off
Net User TestUser1 /Scriptpath:\\FRA-MD-DC-001\Skript\Logon.bat /Domain
```

Listing 49: 2_01_006b.bat

In einer Domänen-Umgebung können Sie ein Benutzerkonto auch mit dem `Dsmod User`-Befehl bearbeiten, der im Gegensatz zum `Net User`-Befehl weitaus mehr Möglichkeiten bietet. Die nachfolgende Tabelle zeigt, welche Parameter erlaubt sind und welche Eigenschaften damit verändert werden können:

Parameter	Beschreibung
-upn	Legt den Benutzerprinzipalnamen fest
-fn	Legt den Vornamen des Benutzers fest
-mi	Legt die Initialen des zweiten Vornamens fest
-ln	Legt den Nachnamen des Benutzers fest
-display	Legt den Anzeigenamen des Benutzers fest
-empid	Legt die Personalnummer des Benutzers fest
-pwd	Gibt das Kennwort des Benutzers an
-desc	Gibt die Beschreibung zum Benutzerkonto an
-office	Gibt den (oder die) Standort(e) des Büros an
-tel	Legt die primäre Telefonnummer fest
-email	Gibt die E-Mail-Adresse des Benutzers an
-hometel	Legt die private Telefonnummer des Benutzers fest
-pager	Legt die Pager-Nummer des Benutzers fest
-mobile	Legt die Mobilfunknummer des Benutzers fest
-fax	Gibt die Faxnummer des Benutzers an
-iptel	Gibt die IP-Telefonnummer des Benutzers an
-webpg	Gibt die Homepage des Benutzers an
-title	Legt den Titel des Benutzers fest
-dept	Gibt die Abteilung des Benutzers an
-company	Legt den Firmennamen fest
-mgr	Gibt den Namen des Managers an

Tabelle 20: Die Eigenschaften, die mit dem Dsmod User-Befehl geändert werden können

Parameter	Beschreibung
-hmdir	Legt den Ordner für das Home-Laufwerk fest
-hmdrv	Legt das Laufwerk für das Home-Laufwerk fest
-profile	Legt den Pfad für das Benutzerprofil fest
-loscr	Legt den Pfad für das Anmeldeskript fest
-mustchpwd	Legt fest, ob der Benutzer bei der nächsten Anmeldung das Kennwort ändern muss (-mustchpwd yes) oder nicht (-mustchpwd no)
-canchpwd	Legt fest, ob der Benutzer das Kennwort ändern kann (-canchpwd yes) oder nicht (canchpwd no)
-reversiblepwd	Legt fest, ob das Kennwort mit umkehrbarer Verschlüsselung (-reversiblepwd yes) gespeichert werden soll oder nicht (-reversiblepwd no)
-pwdneverexpires	Legt fest, ob das Kennwort nie abläuft (-pwdneverexpires yes) oder gemäß der Richtlinie nach einer gewissen Zeit abläuft (-pwdneverexpires no)
-acctexpires	Legt das Datum fest, an dem das Benutzerkonto seine Gültigkeit verliert
-disabled	Legt fest, ob das Benutzerkonto aktiviert (-disabled no) oder deaktiviert (-disabled yes) werden soll

Tabelle 20: Die Eigenschaften, die mit dem Dsmod User-Befehl geändert werden können (Forts.)

Wenn Sie z.B. dem Benutzer *TestUser1*, der in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* abgelegt ist, ein Anmeldeskript zuweisen möchten, so wenden Sie den `Dsmod User`-Befehl gegen das entsprechende Benutzerkonto an und fügen den Parameter `-loscr` an, um das Anmeldeskript zu übergeben:

```
@Echo off
Dsmod User CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com _
    -loscr \\FRA-MD-DC-001\Skript\Logon.bat
```

Listing 50: 2_01_006c.bat

Die Lösung mit VBScript

Bestehende Benutzerkonten lassen sich auch per VBScript ohne großen Aufwand bearbeiten. Denn Sie müssen im Grunde genommen nur eine Verbindung zu dem betreffenden Benutzerkonto herstellen und der entsprechenden Eigenschaft einen neuen Wert zuweisen, um eine Benutzerkonto-Eigenschaft zu ändern (eine Aufstellung der Eigenschaften finden Sie im Rezept Eigenschaften eines Benutzers auslesen ab Seite 145).

Wenn Sie z.B. auf einem lokalen Computersystem (Standalone- oder Member-Server) das lokale Benutzerkonto für *TestUser1* bearbeiten und ein Anmeldeskript zuweisen

158 >> Benutzer ändern

möchten, so stellen Sie mit der `GetObject`-Methode eine Verbindung zu dem betreffenden Benutzerkonto her, weisen der Eigenschaft `LoginScript` den Pfad des Anmeldeskriptes (`strScriptPath`) zu und schreiben das `User`-Objekt mit der `SetInfo`-Methode zurück, um die Änderung zu speichern.

```
Option Explicit

Dim oUser
Dim strComputer
Dim strUserName
Dim strScriptPath

strComputer = "."
strUserName = "TestUser1"
strScriptPath = "\\FRA-MD-DC-001\Skript\Logon.bat"

Set oUser = GetObject("WinNT://" & strComputer & "/" & _
    & strUserName)
oUser.LoginScript = strScriptPath
oUser.SetInfo

Set oUser = Nothing
```

Listing 51: 2_01_006a.vbs

Gleiches lässt sich natürlich auch im Active Directory durchführen. Wenn Sie ein Domänenbenutzerkonto (z.B. *TestUser1*) bearbeiten und ein Anmeldeskript zuweisen möchten, so nutzen Sie den ADSI-Provider LDAP, stellen eine Verbindung zum Benutzerobjekt her und weisen der Eigenschaft `LoginScript` den Pfad zum Anmeldeskript zu. Anschließend schreiben Sie die Änderung mit der `SetInfo`-Methode zurück, damit die Änderung gespeichert wird.

```
Option Explicit

Dim oUser
Dim strComputer
Dim strUserName
Dim strScriptPath

strComputer = "."
strUserName = "CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com"
strScriptPath = "\\FRA-MD-DC-001\Skript\Logon.bat"

Set oUser = GetObject("LDAP://" & strUserName)
oUser.LoginScript = strScriptPath
```

Listing 52: 2_01_006b.vbs

```
oUser.SetInfo
```

```
Set oUser = Nothing
```

Listing 52: 2_01_006b.vbs (Forts.)

7 Benutzer deaktivieren

Um ein lokales Benutzerkonto zu deaktivieren, starten Sie zunächst die lokale Benutzerverwaltung. Anschließend markieren Sie das betreffende Benutzerkonto, das Sie deaktivieren möchten, und rufen aus dem Kontextmenü des Benutzerkontos heraus die EIGENSCHAFTEN auf. Auf der Registerkarte ALLGEMEIN aktivieren Sie die Option KONTO IST DEAKTIVIERT und klicken anschließend auf ÜBERNEHMEN, um die Änderung zu speichern.

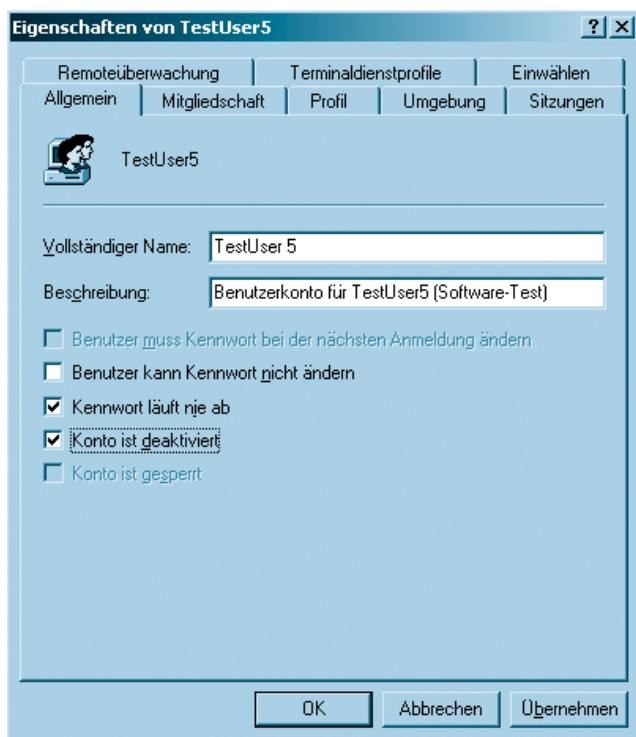


Abbildung 51: Ein lokales Benutzerkonto deaktivieren

Wenn Sie dagegen ein Domänenbenutzerkonto deaktivieren möchten, dann rufen Sie das Verwaltungswerkzeug ACTIVE DIRECTORY-BENUTZER UND -COMPUTER auf, markieren das entsprechende Benutzerkonto und wählen im Kontextmenü des Benutzerkontos den Eintrag EIGENSCHAFTEN aus. Anschließend wechseln Sie auf die Registerkarte

160 >> Benutzer deaktivieren

KONTO und aktivieren die Option KONTO IST DEAKTIVIERT. Übernehmen Sie die neue Einstellung, und schließen Sie ACTIVE DIRECTORY-BENUTZER UND -COMPUTER wieder.

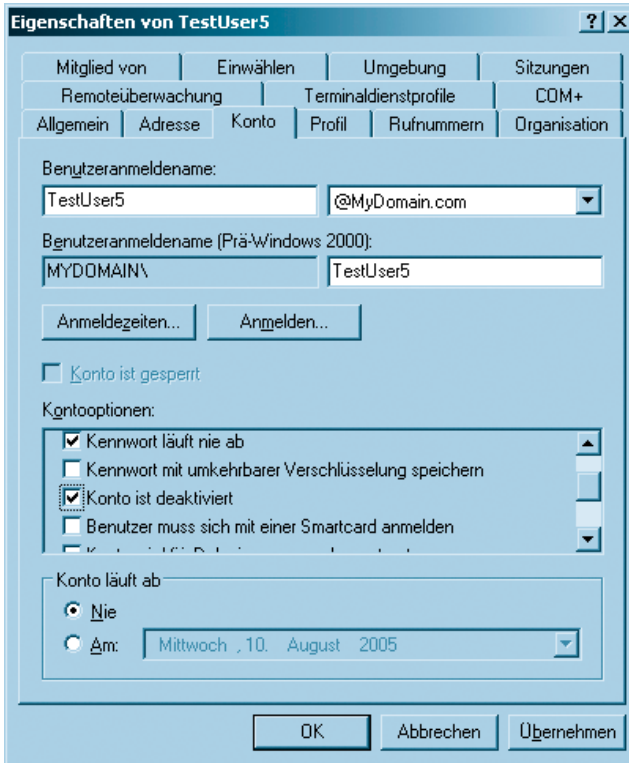


Abbildung 52: Ein Domänenbenutzerkonto deaktivieren

Möchten Sie die soeben deaktivierten Konten zu einem späteren Zeitpunkt wieder aktivieren, so brauchen Sie nur die Option KONTO IST DEAKTIVIERT wieder zu deaktivieren und die Änderung mit einem Klick auf ÜBERNEHMEN zu speichern.

Die Lösung per Eingabeaufforderung

Um ein lokales Benutzerkonto von der Eingabeaufforderung aus zu deaktivieren, verwenden Sie den `Net User`-Befehl, dem Sie das zu deaktivierende Benutzerkonto wie auch die Option `/active:no` als Parameter übergeben. Der `Net User`-Befehl versucht daraufhin, eine Verbindung zu dem angegebenen Benutzerkonto herzustellen und das Benutzerkonto zu deaktivieren.

```
@Echo off
Net User TestUser5 /active:no
```

Listing 53: 2_01_007a.bat

Wenn Sie das soeben deaktivierte Benutzerkonto zu einem späteren Zeitpunkt wieder aktivieren möchten, dann ändern Sie lediglich die Option von `/active:no` auf `/active:yes`:

```
@Echo off
Net User TestUser5 /active:yes
```

Listing 54: 2_01_007b.bat

Um ein Domänenbenutzerkonto zu deaktivieren, können Sie natürlich ebenfalls den `Net User`-Befehl einsetzen. Alternativ können Sie aber auch den `Dsmod User`-Befehl verwenden, der die gleichen Möglichkeiten bietet wie der `Net User`-Befehl, sich aber nur für Domänenbenutzerkonten einsetzen lässt. Möchten Sie beispielsweise mit Hilfe des `Dsmod User`-Befehls das Benutzerkonto *TestUser5* deaktivieren, dann übergeben Sie dem Befehl den definierten Namen des Benutzerkontos und die Option `-disabled yes` als Parameter:

```
@Echo off
Dsmod User "CN=TestUser5,OU=Vertrieb,DC=MyDomain,DC=Com" -disabled yes
```

Listing 55: 2_01_007c.bat

Wenn Sie das Benutzerkonto später wieder aktivieren möchten, dann ändern Sie auch hier nur die Option von `-disabled yes` auf `-disabled no`:

```
@Echo off
Dsmod User "CN=TestUser5,OU=Vertrieb,DC=MyDomain,DC=Com" -disabled no
```

Listing 56: 2_01_007d.bat

Die Lösung mit VBScript

Auch per VBScript ist das Deaktivieren von Benutzerkonten kein Problem. Möchten Sie beispielsweise auf einem Standalone- oder Member-Server das lokale Benutzerkonto *TestUser5* deaktivieren, so stellen Sie mit der `GetObject`-Methode eine Verbindung zum ADSI-Provider `WinNT` des betreffenden Computersystems her und übergeben zeitgleich auch den Benutzernamen, um eine Bindung an das `User`-Objekt zu erzeugen. Anschließend setzen Sie die `AccountDisabled`-Eigenschaft des `User`-Objektes auf `True` und schreiben die Änderung mit der `SetInfo`-Methode zurück, um das Benutzerkonto mit sofortiger Wirkung zu deaktivieren.

162 >> Benutzer deaktivieren

```
Option Explicit
```

```
Dim oUser
Dim strComputer
Dim strUserName
```

```
strComputer = "."
strUserName = "TestUser5"
```

```
Set oUser = GetObject("WinNT://" & strComputer & "/" & strUserName)
oUser.AccountDisabled = True
oUser.SetInfo
```

```
Set oUser = Nothing
```

Listing 57: 2_01_007a.vbs

Wenn Sie dagegen ein Domänenbenutzerkonto deaktivieren möchten, so stellen Sie stattdessen eine Verbindung zum ADSI-Provider LDAP her und übergeben den definierten Namen des Benutzerkontos. Anschließend setzen Sie die `AccountDisabled`-Eigenschaft auf `True` und schreiben die Änderung der Eigenschaft mit der `SetInfo`-Methode zurück, woraufhin das Domänenkonto unmittelbar deaktiviert wird.

```
Option Explicit
```

```
Dim oUser
Dim strOU
Dim strUserName
```

```
strOU = "OU=Vertrieb,DC=MyDomain,DC=Com"
strUserName = "CN=TestUser5"
```

```
Set oUser = GetObject("LDAP://" & strUserName & "," & strOU)
oUser.AccountDisabled = True
oUser.SetInfo
```

```
Set oUser = Nothing
```

Listing 58: 2_01_007b.vbs

Wenn Sie die soeben deaktivierten Benutzerkonten zu einem späteren Zeitpunkt wieder aktivieren möchten, so verwenden Sie die gleichen Code-Zeilen und setzen lediglich die `AccountDisabled`-Eigenschaft wieder auf `False`.

8 Benutzer umbenennen

Die Benutzerverwaltung von Windows Server 2003 hält eine entsprechende Funktion bereit, womit Sie ein Benutzerkonto jederzeit umbenennen können. Wenn Sie also einen falschen Benutzernamen eingegeben oder sich versehentlich verschrieben haben, dann lässt sich dies sehr schnell korrigieren, indem Sie das Benutzerkonto einfach umbenennen.

Um ein lokales Benutzerkonto umzubenennen, rufen Sie zunächst die lokale Benutzerverwaltung (LOKALE BENUTZER UND GRUPPEN) auf. Anschließend markieren Sie den Ordner BENUTZER, woraufhin alle eingerichteten Benutzerkonten angezeigt werden. Markieren Sie das Benutzerkonto, das Sie umbenennen möchten, und wählen Sie aus dem Kontextmenü heraus UMBENENNEN aus. Daraufhin wird der Bearbeitungsmodus für das Feld BENUTZERNAME geöffnet, und Sie können den Benutzernamen durch einen neuen Benutzernamen (mit max. 20 Zeichen) überschreiben.

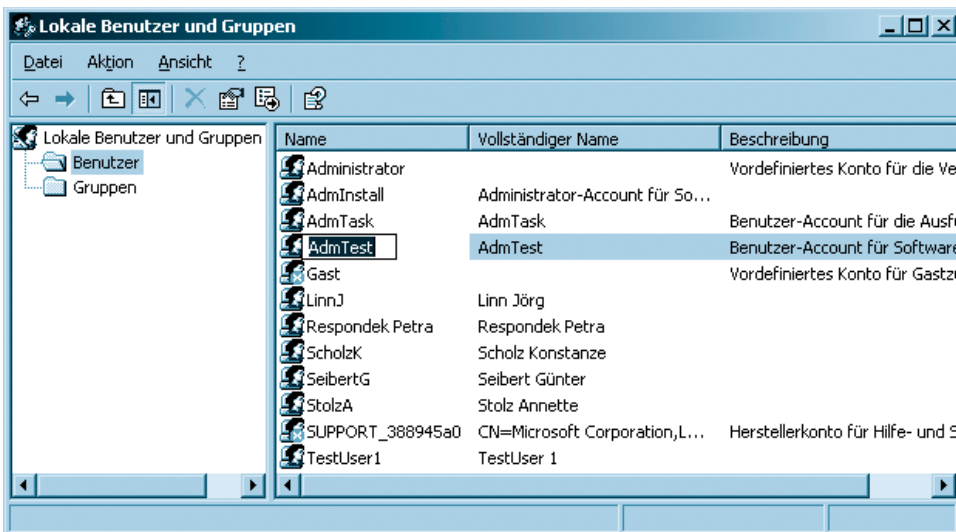


Abbildung 53: Das AdmTest-Benutzerkonto umbenennen

Ein Domänen-Benutzerkonto lässt sich auf ähnlichem Wege umbenennen. Hierzu öffnen Sie ACTIVE DIRECTORY-BENUTZER UND -COMPUTER und markieren das Benutzerkonto, das Sie umbenennen möchten. Anschließend öffnen Sie das Kontextmenü des Benutzerkontos und wählen UMBENENNEN aus. Daraufhin wird der Bearbeitungsmodus geöffnet, und Sie können den Benutzernamen durch einen neuen Benutzernamen überschreiben.

Die Lösung per Eingabeaufforderung

Wenn Sie einzelne Benutzerkonten von der Eingabeaufforderung (z.B. aus einer Batch-Datei heraus) umbenennen möchten, so können Sie zum einen den WMIC-Befehl mit dem Alias `UserAccount` einsetzen, zum anderen aber auch den `Dsmove-`

164 >> Benutzer umbenennen

Befehl. Der `Dsmove`-Befehl setzt allerdings eine Domänenumgebung voraus und lässt sich daher nicht zum Umbenennen lokaler Benutzerkonten verwenden.

Um beispielsweise das lokale Benutzerkonto *TestUser1* nach *TestUser10* umzubenennen, wenden Sie den `WMIC UserAccount`-Befehl an, selektieren mit der `Where`-Klausel das entsprechende Benutzerkonto und wenden mit `Call` die `Rename`-Methode gegen das Benutzerobjekt an, der Sie den neuen Benutzernamen als Parameter übergeben.

```
@Echo off
WMIC UserAccount Where "Name='TestUser1'" Call Rename TestUser10
```

Listing 59: 2_01_008a.bat

Ein Domänenbenutzerkonto lässt sich alternativ auch mit dem `Dsmove`-Befehl umbenennen, dem Sie zunächst den definierten Namen des umzubenennenden Benutzerkontos übergeben und dann mit dem Parameter `-NewName` den neuen Benutzernamen zuweisen.

```
@Echo off
Dsmove CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com _
-NewName CN=TestUser10,OU=Vertrieb,DC=MyDomain,DC=Com
```

Listing 60: 2_01_008b.bat

Die Lösung mit VBScript

Das Umbenennen eines Benutzerkontos per VBScript nehmen Sie im einfachsten Fall mit der `MoveHere`-Methode des `WinNT`- und des `LDAP`-Providers vor, der Sie den ADS-Pfad des `User`-Objektes wie auch den neuen Benutzernamen als Parameter übergeben.

Wenn Sie z.B. das lokale Benutzerkonto *TestUser1* nach *TestUser10* umbenennen möchten, so stellen Sie mit `GetObject` zunächst eine Verbindung zum `WinNT`-Provider des betreffenden Computersystems her und binden dann das jeweilige `User`-Objekt an eine Objektvariable. Anschließend wenden Sie die `MoveHere`-Methode an und übergeben den ADS-Pfad des `User`-Objektes und auch den neuen Benutzernamen als Argument.

```
Option Explicit

Dim oComputer
Dim oUser
Dim oNewUser
Dim strOldName
Dim strNewName
```

Listing 61: 2_01_008a.vbs

>> Benutzer und Gruppen

```

Dim strComputer

strComputer = "."
strOldName = "TestUser1"
strNewName = "TestUser10"

Set oComputer = GetObject("WinNT://" & strComputer)
Set oUser = GetObject("WinNT://" & strComputer & "/" & strOldName & ",User")
Set oNewUser = oComputer.MoveHere(oUser.ADsPath, strNewName)

Set oComputer = Nothing
Set oUser = Nothing
Set oNewUser = Nothing

```

Listing 61: 2_01_008a.vbs (Forts.)

Auf dem gleichen Weg benennen Sie auch ein Domänenbenutzerkonto um, nur dass Sie in diesem Fall nicht den WinNT-Provider verwenden, sondern den LDAP-Provider.

```

Option Explicit

Dim oOU
Dim oUser
Dim strOU
Dim strOldName
Dim strNewName

strOldName = "CN=TestUser1"
strNewName = "CN=TestUser10"
strOU = "OU=Vertrieb,DC=MyDomain,DC=Com"

Set oOU = GetObject("LDAP://" & strOU)
Set oUser = oOU.MoveHere("LDAP://" & strOldName & "," & strOU, strNewName)

Set oOU = Nothing
Set oUser = Nothing

```

Listing 62: 2_01_008b.vbs

9 Benutzer löschen

Benutzerkonten lassen sich natürlich jederzeit löschen, wobei Sie vorher aber sicherstellen müssen, dass das betreffende Benutzerkonto auch wirklich nicht mehr benötigt wird. Denn wenn Sie ein Benutzerkonto löschen und selbiges später neu erstellen, dann wird dem neuen Benutzerkonto eine neue SID (siehe Welche SID hat

der Benutzer? auf Seite 180) zugewiesen, womit sämtliche Berechtigungen neu vergeben werden müssen. Das Wiederherstellen eines gelöschten Benutzerkontos ist nicht möglich.

Um ein lokales Benutzerkonto zu löschen, rufen Sie auf dem Standalone- oder Member-Server die lokale Benutzerverwaltung auf. Anschließend markieren Sie im Ordner BENUTZER das Benutzerkonto, das Sie löschen möchten, und wählen aus dem Kontextmenü heraus den Eintrag LÖSCHEN aus. Sie erhalten daraufhin eine Sicherheitswarnung, dass gelöschte Benutzerkonten nicht wiederhergestellt werden können. Klicken Sie auf JA, um das Benutzerkonto endgültig zu löschen.

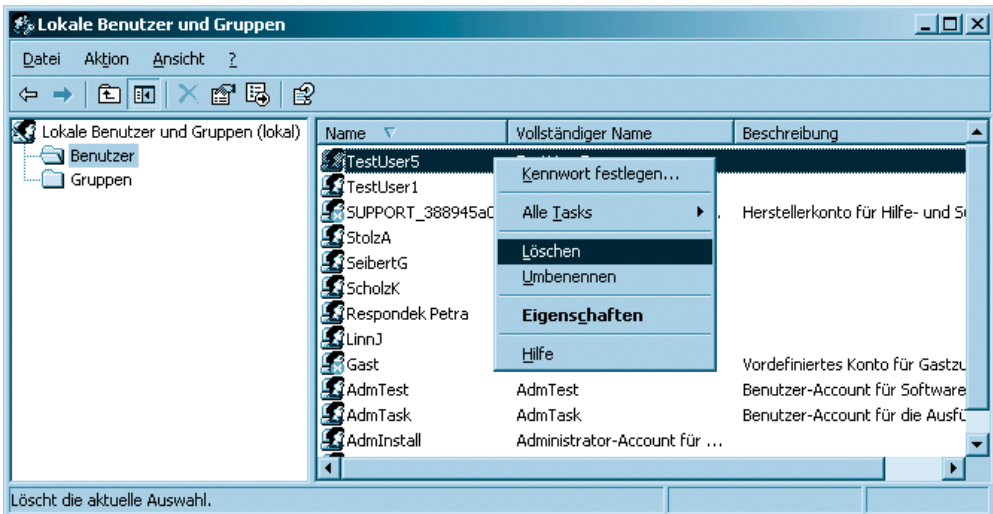


Abbildung 54: Ein lokales Benutzerkonto löschen

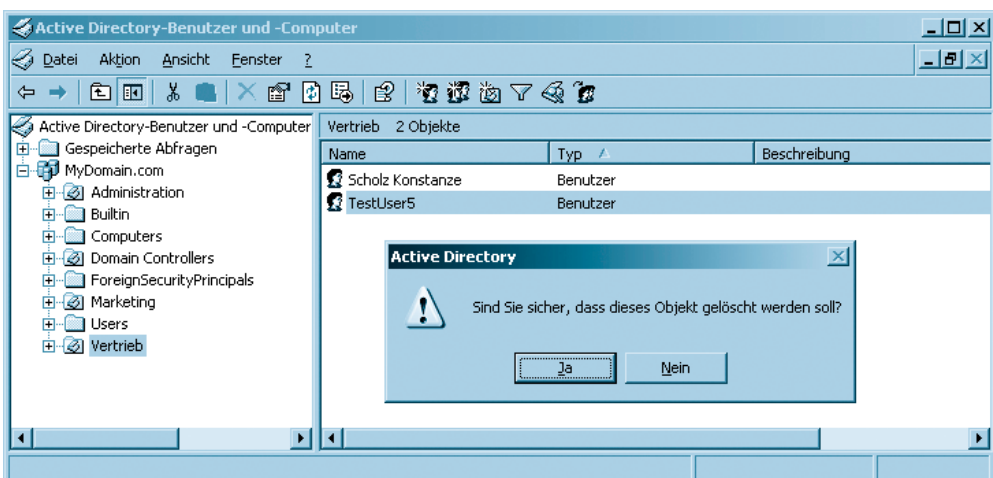


Abbildung 55: Ein Domänenbenutzerkonto löschen

Wenn Sie dagegen ein Domänenbenutzerkonto löschen möchten, dann rufen Sie im STARTMENÜ, genauer gesagt unter VERWALTUNG, das Verwaltungswerkzeug ACTIVE DIRECTORY-BENUTZER UND -COMPUTER auf, markieren das betreffende Benutzerkonto und wählen im Kontextmenü des Benutzerkontos den Eintrag LÖSCHEN aus. Sie werden daraufhin noch einmal gefragt, ob Sie das Benutzerkonto auch wirklich löschen möchten. Bestätigen Sie diese Rückfrage mit einem Klick auf JA.

Die Lösung per Eingabeaufforderung

Um ein lokales Benutzerkonto von der Eingabeaufforderung aus zu löschen, verwenden Sie wieder den Befehl `Net User`, dem Sie das zu löschende Benutzerkonto wie auch die Option `/Delete` als Parameter übergeben. Der `Net User`-Befehl versucht daraufhin, eine Verbindung zu dem angegebenen Benutzerkonto herzustellen und das Benutzerkonto ohne jegliche Rückfrage zu löschen.

```
@Echo off
Net User TestUser5 /Delete
```

Listing 63: 2_01_009a.bat

Den gleichen Befehl können Sie auch zum Löschen eines Domänenbenutzerkontos verwenden, indem Sie den Befehl mit dem zusätzlichen Parameter `/Domain` explizit gegen die Domäne richten. Alternativ dazu lassen sich Domänenbenutzerkonten aber auch mit dem `Dsrm`-Befehl löschen, dem Sie zumindest den definierten Namen des Benutzerkontos als Parameter übergeben müssen. Um das Löschen des Benutzerkontos zu erzwingen, fügen Sie der Befehlszeile den zusätzlichen Parameter `-noprompt` an.

```
@Echo off
Dsrm CN=TestUser5,OU=Vertrieb,DC=MyDomain,DC=Com -noprompt
```

Listing 64: 2_01_009b.bat

Die Lösung mit VBScript

Um ein vorhandenes Benutzerkonto per VBScript zu löschen, sind nur wenige Code-Zeilen erforderlich. Wenn Sie z.B. das lokale Benutzerkonto für *TestUser5* auf dem Standalone- oder Member-Server *FRA-MD-FPS-001* löschen möchten, so stellen Sie mit der `GetObject`-Methode eine Verbindung zum ADSI-Provider `WinNT` des betreffenden Computersystems her und wenden die `Delete`-Methode gegen das betreffende User-Objekt an:

168 >> Welcher Benutzer ist gerade angemeldet?

```
Option Explicit
```

```
Dim oComputer
Dim strComputer
Dim strUserName
```

```
strComputer = "FRA-MD-FPS-001"
strUserName = "TestUser5"
```

```
Set oComputer = GetObject("WinNT://" & strComputer)
oComputer.Delete "User", strUserName
```

```
Set oComputer = Nothing
```

Listing 65: 2_01_009a.vbs

Möchten Sie stattdessen ein Benutzerkonto im Active Directory löschen, so stellen Sie die Verbindung zum ADSI-Provider LDAP her und setzen die Delete-Methode gegen das betreffende Benutzerkonto ein.

Das nachfolgende Code-Beispiel löscht das Benutzerkonto für *TestUser5*, das in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* abgelegt ist:

```
Option Explicit
```

```
Dim oOU
Dim strOU
Dim strUserName
```

```
strOU = "OU=Vertrieb,DC=MyDomain,DC=Com"
strUserName = "CN=TestUser1"
Set oOU = GetObject("LDAP://" & strOU)
oOU.Delete "User", strUserName
```

```
Set oOU = Nothing
```

Listing 66: 2_01_009b.vbs

10 Welcher Benutzer ist gerade angemeldet?

Wenn Sie auf die Schnelle herausfinden müssen, welcher Benutzer gerade an Windows Server 2003 angemeldet ist, so brauchen Sie nur aus dem Kontextmenü der TASK-LEISTE den TASK-MANAGER zu öffnen und auf die Registerkarte BENUTZER zu wechseln, denn dort werden alle aktiven Anmeldungen angezeigt.

Auf dieser Registerkarte werden natürlich auch Windows-Sitzungen angezeigt, die via Terminal-Services von einem anderen Computersystem aus hergestellt wurden.

Wenn also gerade eine aktive Remotedesktop-Verbindung zu diesem Server existiert, so werden diese Anmeldungen ebenfalls angezeigt.

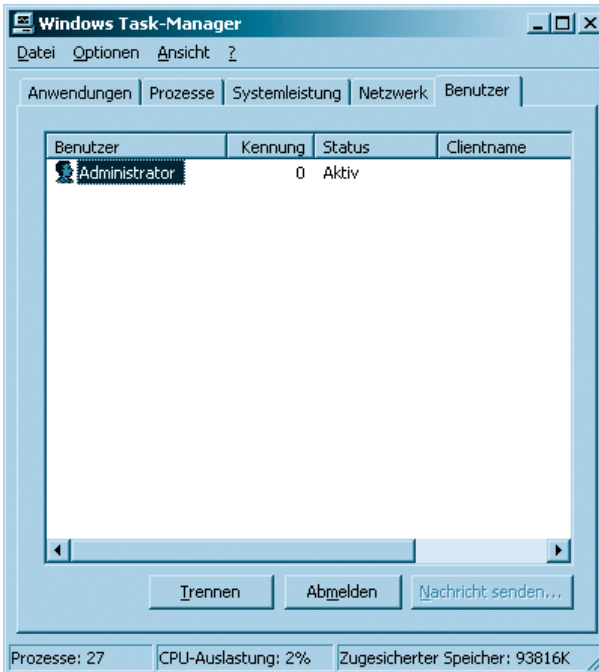


Abbildung 56: Die aktiven Benutzeranmeldungen anzeigen lassen

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung können Sie die aktiven Anmeldungen auf unterschiedlichen Wegen überprüfen. Wenn Sie z.B. herausfinden möchten, mit welchem Benutzernamen Sie gerade angemeldet sind, dann brauchen Sie in der Eingabeaufforderung nur den Befehl `whoami` einzugeben. Daraufhin wird der aktuelle Benutzername im Format `Computername\Benutzername` angezeigt.

```
@Echo off
whoami
```

Listing 67: 2_01_010a.bat

Wenn Sie stattdessen alle aktiven Windows-Sitzungen abfragen möchten (also auch die der Terminalserver-Verbindungen), dann führen Sie stattdessen den Befehl `Query Session` aus. Sie erhalten daraufhin eine Aufstellung aller aktiven Anmeldungen.

170 >> Benutzer abmelden

```
@Echo off
Query Session
```

Listing 68: 2_01_010b.bat

Die Lösung mit VBScript

Auch VBScript erfordert nur wenige Code-Zeilen, um den aktuellen Benutzer auszu-lesen. Denn Sie müssen im Grunde genommen nichts weiter tun, als ein neues Net-work-Objekt zu erzeugen und den Wert der Eigenschaft `UserName` abzufragen.

```
Option Explicit

Dim oNetwork

Set oNetwork = WScript.CreateObject("Wscript.Network")
WScript.Echo oNetwork.UserName

Set oNetwork = Nothing
```

Listing 69: 2_01_010.vbs

11 Benutzer abmelden

Für den Fall, dass Sie gerade aktiv an der Windows Server 2003-Konsole angemeldet sind und eine Abmeldung durchführen möchten, so öffnen Sie das STARTMENÜ, klicken auf ABMELDEN und führen im Dialogfenster WINDOWS-ABMELDUNG, das daraufhin eingeblendet wird, einen Klick auf die Schaltfläche ABMELDEN aus. Die aktive Windows-Sitzung wird damit beendet, woraufhin benutzerspezifische Einstellungen in das Benutzerprofil geschrieben werden.

Möchten Sie dagegen einen Benutzer abmelden, der gerade per Remotedesktop mit dem Windows Server 2003 verbunden ist, dann rufen Sie aus dem Kontextmenü der TASK-LEISTE den TASK-MANAGER auf und wechseln auf die Registerkarte BENUTZER. Anschließend markieren Sie den Benutzer, den Sie abmelden möchten, und klicken auf ABMELDEN.

Achtung

Da die Abmeldung eines per Remotedesktop verbundenen Benutzers sofort und nicht erst nach Ablauf einer gewissen Wartezeit geschieht, sollten Sie den entsprechenden Benutzer mit Hilfe einer Netzwerkmeldung über den bevorstehenden Abmeldevorgang informieren. Denn wenn gerade Daten transferiert oder Anwendungen ausgeführt werden, so kann dies zu einem Datenverlust führen, da mit der Abmeldung auch sämtliche Prozesse des Benutzers beendet werden.

Die Lösung per Eingabeaufforderung

Der Windows Server 2003 stellt den Befehl `Logoff.exe` bereit, womit Sie einen Benutzer auch von der Eingabeaufforderung aus abmelden können. Ohne Angabe weiterer Parameter wird dabei die aktive Windows-Sitzung des Benutzers abgemeldet, der den Befehl gerade ausführt.

```
@Echo off
Logoff
```

Listing 70: 2_01_011a.bat

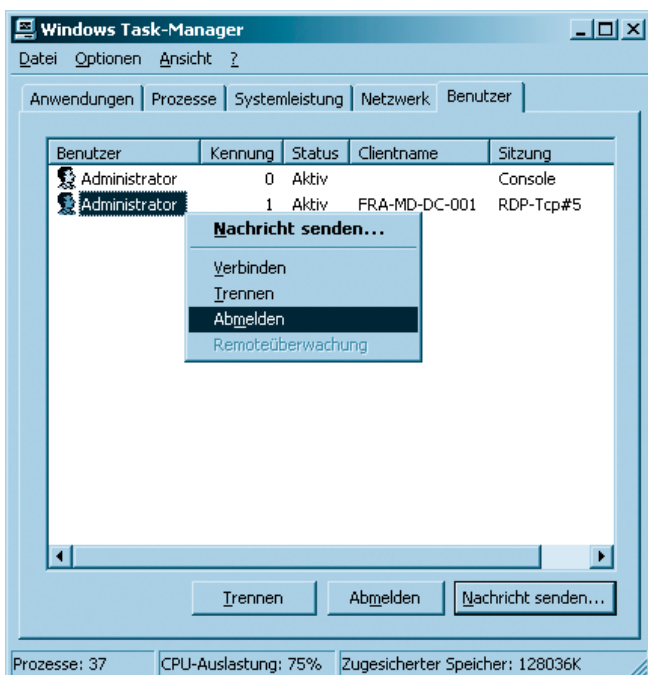


Abbildung 57: Einen Benutzer abmelden

Sie können den `Logoff`-Befehl aber auch für die Abmeldung anderer Windows-Sitzungen verwenden. Wenn beispielsweise gerade einige Administratoren per Remotedesktop mit dem Server verbunden sind, so können Sie sich die einzelnen Windows-Sitzungen mit dem Befehl `Query Session` anzeigen lassen und durch Angabe des Sitzungsnamens (z.B. `rdp-tcp#1`) eine spezielle Windows-Sitzung abmelden.

```
@Echo off
Logoff rdp-tcp#1
```

Listing 71: 2_01_011b.bat

172 >> Benutzer abmelden

Eine weitere Möglichkeit, von der Eingabeaufforderung aus eine Abmeldung durchzuführen, bietet der WMIC-Befehl in Verbindung mit dem Alias OS, womit Sie die Win32ShutDown-Methode der WMI-Klasse Win32_OperatingSystem aufrufen und eine Windows-Sitzung abmelden können. Die Win32ShutDown-Methode erwartet dabei die Angabe der ShutDown-Option, die Sie mit dem Methodenaufruf übergeben müssen. Der Wert 0 führt hierbei zu einer gewöhnlichen Abmeldung, der Wert 4 dagegen erzwingt die Abmeldung.

```
@Echo off
WMIC OS Where "Status='OK'" Call Win32ShutDown 4
```

Listing 72: 2_01_011c.bat

Die Lösung mit VBScript

Mit der Win32ShutDown-Methode der WMI-Klasse Win32_OperatingSystem lässt sich eine Windows-Abmeldung auch per VBScript herbeiführen. Hierzu führen Sie eine ExecQuery-Abfrage gegen die Win32_OperatingSystem-Klasse durch, um das OperatingSystem-Objekt zu selektieren, und wenden dann – innerhalb einer For Each-Schleife – die Win32ShutDown-Methode gegen das OperatingSystem-Objekt an, der Sie den Wert der ShutDown-Option (0 = Abmelden, 4 = Abmelden erzwingen) als Parameter übergeben.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colOS
Dim OS

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate,(ShutDown)}!\\" & strComputer & "\root\cimv2")
Set colOS = oWMI.ExecQuery ("Select * from _
    Win32_OperatingSystem")

For Each OS in colOS
    OS.Win32ShutDown(4)
Next

Set oWMI = Nothing
```

Listing 73: 2_01_011.vbs

12 Eine sekundäre Anmeldung durchführen

Für den Fall, dass Sie gerade als gewöhnlicher Benutzer oder vielleicht als Hauptbenutzer lokal angemeldet sind, zur Ausführung einer bestimmten Aufgabe aber administrative Berechtigungen benötigen, so müssen Sie sich nicht ab- und wieder anmelden, um administrative Berechtigungen zu erhalten. Denn mit der AUSFÜHREN ALS-Funktion können Sie während einer bestehenden Windows-Sitzung eine sekundäre Anmeldung durchführen und die gewünschte Aufgabe auch unter Verwendung eines anderen Benutzernamens ausführen. Dies setzt allerdings voraus, dass der Dienst SEKUNDÄRE ANMELDUNG gestartet ist.

Wenn Sie z.B. gerade als Benutzer mit eingeschränkten Berechtigungen angemeldet sind und Änderungen in der Benutzerverwaltung vornehmen möchten, so rufen Sie die Benutzerverwaltung einfach unter Verwendung eines anderen Benutzernamens auf: Öffnen Sie das STARTMENÜ, wechseln Sie in das Untermenü VERWALTUNG, und rufen Sie das Kontextmenü von COMPUTERVERWALTUNG auf. Im Kontextmenü wählen Sie AUSFÜHREN ALS aus, woraufhin ein Dialogfenster geöffnet wird, worüber Sie den Benutzernamen und auch das dazugehörige Kennwort des Benutzers eingeben müssen, unter dessen Anmeldung die COMPUTERVERWALTUNG ausgeführt werden soll.

Geben Sie die Anmeldeinformationen (Benutzername und Kennwort) ein, und klicken Sie anschließend auf OK. Die COMPUTERVERWALTUNG wird nun unter einem anderen Benutzernamen (z.B. *Administrator*) ausgeführt, und Sie haben – trotz der Anmeldung als gewöhnlicher Benutzer oder Hauptbenutzer – nun administrative Rechte. Diese administrativen Rechte gelten aber nur für die COMPUTERVERWALTUNG und auch nur so lange, wie Sie die COMPUTERVERWALTUNG unter der sekundären Anmeldung ausführen. Wenn Sie die COMPUTERVERWALTUNG schließen, verliert die sekundäre Anmeldung ihre Gültigkeit.

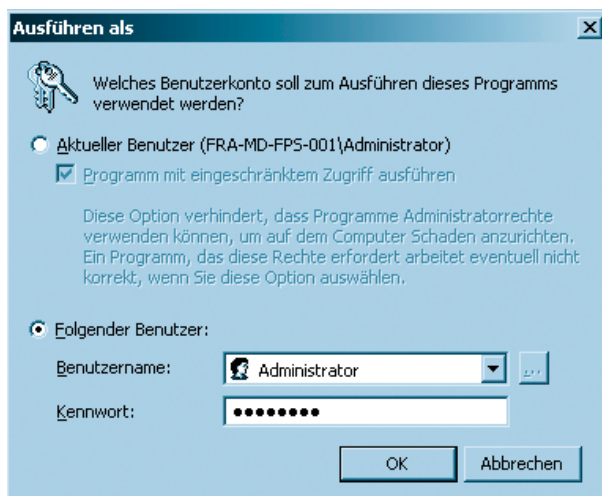


Abbildung 58: Eine Aufgabe unter anderem Benutzernamen ausführen

174 >> Wann hat sich ein bestimmter Benutzer zuletzt angemeldet?

Die AUSFÜHREN ALS-Funktion können Sie nicht nur aus dem Kontextmenü der START-MENÜ-Einträge heraus aufrufen, sondern auch aus den Kontextmenüs sämtlicher Programmdateien. Wenn Sie beispielsweise die lokale Benutzerverwaltung LOKALE BENUTZER UND GRUPPEN unter einem anderen Benutzernamen ausführen möchten, so wählen Sie AUSFÜHREN ALS aus dem Kontextmenü der Datei *Lusrmgr.msc*, die sich im Ordner *C:\Windows\System32* befindet.

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung verwenden Sie den *Runas*-Befehl, um eine sekundäre Anmeldung durchzuführen. Wenn Sie z.B. gerade als *TestUser1* angemeldet und Mitglied der lokalen Gruppe *Hauptbenutzer* sind, für eine bestimmte Aufgabe aber administrative Rechte benötigen, so können Sie die Aufgabe auch unter Verwendung des Administrator-Accounts ausführen. Sie müssen der Befehlszeile nur den *Runas*-Befehl mit dem entsprechenden Benutzernamen voranstellen.

```
@Echo off
Runas /user:FRA-MD-FPS-001\Administrator _
Net User TestUser5 /Delete
```

Listing 74: 2_01_012.bat

Sie werden daraufhin nach dem Kennwort für das Administratorkonto gefragt. Geben Sie das Kennwort ein, um die Befehlszeile *Net User TestUser5 /Delete* auszuführen.

Die Lösung mit VBScript

Mit dem *Runas*-Befehl lässt sich natürlich auch ein Skript unter Verwendung eines anderen Benutzers ausführen. Wenn Sie z.B. gerade als gewöhnlicher Benutzer angemeldet sind und ein Skript ausführen möchten, das administrative Berechtigungen erfordert, so führen Sie der Einfachheit halber vor dem eigentlichen *Cscript*-Aufruf den *Runas*-Befehl aus und übergeben den Benutzernamen wie auch das auszuführende Skript als Parameter.

Das nachfolgende *Runas*-Beispiel führt das Skript *MyScript.vbs* unter Verwendung des Benutzers *Administrator* aus und erwartet daraufhin die Eingabe des Kennwortes für den *Administrator*-Account:

```
Runas /User:FRA-MD-FPS-001\Administrator CScript MyScript.vbs
```

13 Wann hat sich ein bestimmter Benutzer zuletzt angemeldet?

Wenn Sie das Sicherheitsprotokoll der Ereignisanzeige regelmäßig sichern und damit auch noch auf ältere Ereignisse zugreifen können, so können Sie den letzten Anmeldezeitpunkt eines Benutzers dem Sicherheitsprotokoll entnehmen. Denn der Windows

Server 2003 zeichnet per Default sämtliche An- und Abmeldevorgänge der Benutzer auf und erzeugt im Sicherheitsprotokoll entsprechende Ereignisse mit der Ereignis-ID 528 und dem jeweiligen Benutzernamen, womit Sie die An- und Abmeldevorgänge auswerten können.

Um herauszufinden, wann sich der *Administrator* zuletzt angemeldet hat, öffnen Sie zunächst die EREIGNISANZEIGE und wechseln in das Sicherheitsprotokoll. Hierzu öffnen Sie das STARTMENÜ, wechseln in das Untermenü VERWALTUNG und klicken auf EREIGNISANZEIGE oder wählen im STARTMENÜ den Eintrag AUSFÜHREN und geben Eventvwr.msc ein. Mit einem Klick auf SICHERHEIT gelangen Sie in die Übersicht des Sicherheitsprotokolls und finden alle sicherheitsrelevanten Ereignisse vor.

Um ein bestimmtes Ereignis zu suchen (z.B. die An- und Abmeldevorgänge des *Administrators*), nehmen Sie die Filter-Funktion zu Hilfe und lassen sich nur die Ereignisse anzeigen, die einem bestimmten Suchkriterium entsprechen. Öffnen Sie dazu das Kontextmenü von SICHERHEIT, und wählen Sie erst ANSICHT, dann FILTER aus. Sie gelangen nun auf die Registerkarte FILTER des Dialogfensters EIGENSCHAFTEN VON SICHERHEIT und können die Suchkriterien festlegen. Geben Sie im Feld EREIGNISKENNUNG den Wert 528 und im Feld BENUTZERNAMEN den Namen des Benutzers (in diesem Fall *Administrator*) ein, und übernehmen Sie die Filtereinstellung mit einem Klick auf OK.

Sie erhalten nun eine Übersicht aller Ereignisse mit der ID 528 und dem Benutzernamen *Administrator* und damit eine Aufstellung der An- und Abmeldevorgänge unter Angabe des Datums und der Uhrzeit. Wenn Sie die einzelnen Ereignisse per Doppelklick öffnen, können Sie dem Ereignis weitere Informationen (*Anmeldetyp, Arbeitsstation, Domäne ...*) entnehmen.

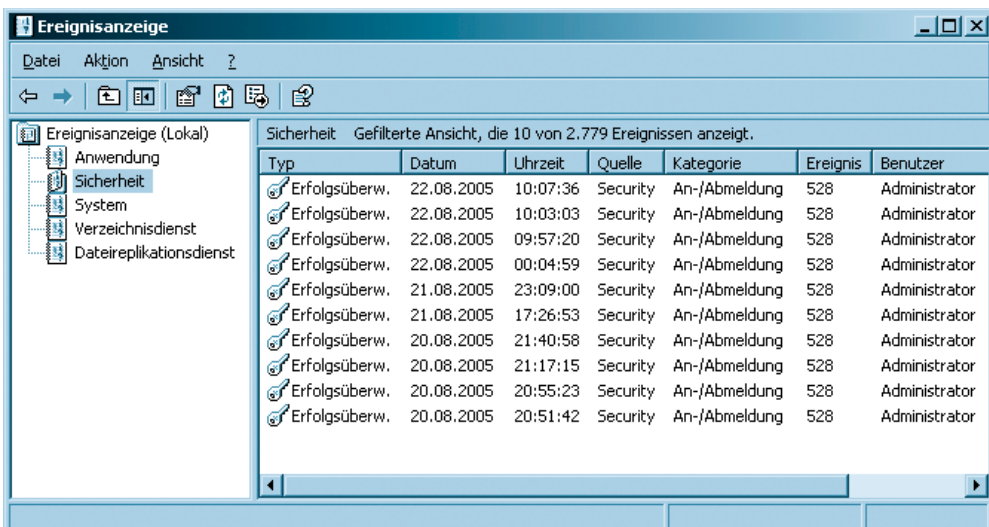


Abbildung 59: Die letzten Anmeldeereignisse des Administrators

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus ermitteln möchten, wann sich ein bestimmter Benutzer zuletzt an einem Computersystem oder an der Domäne angemeldet hat, dann verwenden Sie hierfür den `Net User`-Befehl, dem Sie lediglich das zu überprüfende Benutzerkonto als Parameter übergeben. Der `Net User`-Befehl versucht daraufhin, dessen Eigenschaften auszulesen, womit auch der letzte Anmeldezeitpunkt ausgegeben wird.

Um beispielsweise herauszufinden, wann sich der Benutzer *TestUser1* zuletzt auf dem lokalen Computersystem angemeldet hat, wenden Sie den `Net User`-Befehl unter Angabe des Benutzerkontos an und lassen sich die Eigenschaften des Benutzerkontos entweder am Bildschirm anzeigen oder leiten die Ausgabe der Eigenschaften in eine Datei um. Sofern sich der Benutzer schon einmal angemeldet hat, können Sie das Datum und auch die Uhrzeit der Eigenschaft `Letzte Anmeldung` entnehmen. Anderenfalls (wenn sich der Benutzer noch nicht angemeldet hat) wird die letzte Anmeldung mit `Nie` ausgewiesen.

Wenn Sie das Ergebnis der `Net User`-Abfrage in eine Datei umleiten und eine `Findstr`-Suche mit dem Kriterium `/C:"Letzte Anmeldung"` durchführen, so können Sie den letzten Anmeldezeitpunkt aus den Eigenschaften herausfiltern und damit nur das Datum und auch die Uhrzeit der letzten Anmeldung ausgeben lassen.

```
@Echo off
Net User TestUser1 > User.txt
Findstr /C:"Letzte Anmeldung" User.txt
```

Listing 75: 2_01_013a.bat

Die `Net User`-Abfrage wird standardmäßig immer gegen das Computersystem ausgeführt, auf dem auch der Befehl aufgerufen wird. Wenn Sie die `Net User`-Abfrage gegen die Domäne richten möchten, so wenden Sie den zusätzlichen Parameter `/Domain` an.

```
@Echo off
Net User TestUser1 /Domain > User.txt
Findstr /C:"Letzte Anmeldung" User.txt
```

Listing 76: 2_01_013b.bat

Für den Fall, dass Sie mehrere Domänen-Controller einsetzen, so liefert diese Abfrage jedoch kein exaktes Ergebnis, da die Anmeldezeit nicht zwischen den Domänen-Controllern repliziert wird. In einer Domänenumgebung mit mehreren Domänen-Controllern empfiehlt sich daher der Einsatz des externen Freeware-Tools *GetUser-Info* (www.joeware.net/win/free/tools/getuserinfo.htm), das die exakte Anmeldezeit zurückliefert. Um die gleiche Abfrage mit *GetUser-Info* durchzuführen, können Sie folgende Befehlszeile verwenden:

```
@Echo off
GetUserInfo MyDomain\TestUser1 > User.txt
Findstr /C:"Last login" User.txt
```

Listing 77: 2_01_013c.bat

Die Lösung mit VBScript

Um den letzten Anmeldezeitpunkt eines bestimmten Benutzers per VBScript zu ermitteln, stellen Sie lediglich eine Verbindung zu dem betreffenden Benutzerkonto her und ermitteln den Wert der Eigenschaft LastLogin.

```
Option Explicit

Dim oUser
Dim strComputer
Dim strUserName

strComputer = "."
strUserName = "TestUser1"

Set oUser = GetObject("WinNT://" & strComputer & _
    "/" & strUserName)
WScript.Echo oUser.LastLogin

Set oUser = Nothing
```

Listing 78: 2_01_013a.vbs

Wenn Sie die Abfrage nicht gegen das lokale Computersystem, sondern gegen die Domäne richten wollen, dann nutzen Sie stattdessen den ADSI-Provider LDAP, übergeben den definierten Namen des jeweiligen Benutzerkontos und fragen die Last-Login-Eigenschaft ab.

```
Option Explicit

Dim oUser
Dim strComputer
Dim strUserName

strComputer = "."
strUserName = "CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com"

Set oUser = GetObject("LDAP://" & strUserName)
```

Listing 79: 2_01_013b.vbs

178 >> Einzelne Benutzer in andere Organisationseinheiten verschieben

```
WScript.Echo oUser.LastLogin
```

```
Set oUser = Nothing
```

Listing 79: 2_01_013b.vbs (Forts.)

14 Einzelne Benutzer in andere Organisationseinheiten verschieben

Ein Domänen-Benutzerkonto lässt sich innerhalb einer Domäne problemlos verschieben. Vorausgesetzt natürlich, Sie verfügen über die erforderliche Berechtigung. Denn Sie müssen entweder als *Administrator* angemeldet oder zumindest Mitglied der Gruppe *Konten-Operatoren* oder *Organisations-Admins* sein, um ein Objekt im Active Directory verschieben zu können. Oder aber die Berechtigungen wurden entsprechend an Sie delegiert.

Wenn Sie beispielsweise im Container *USERS* der Domäne *MyDomain.com* mehrere Benutzerkonten angelegt haben und diese nun die Organisationseinheit *Vertrieb* verschieben möchten, so rufen Sie im Verwaltungsmenü *ACTIVE DIRECTORY-BENUTZER UND -COMPUTER* auf, markieren die entsprechenden Benutzerkonten (eine Mehrfachauswahl ist mit gedrückter **[Strg]**-Taste möglich) und ziehen sie per *Drag & Drop* in die gewünschte Organisationseinheit.

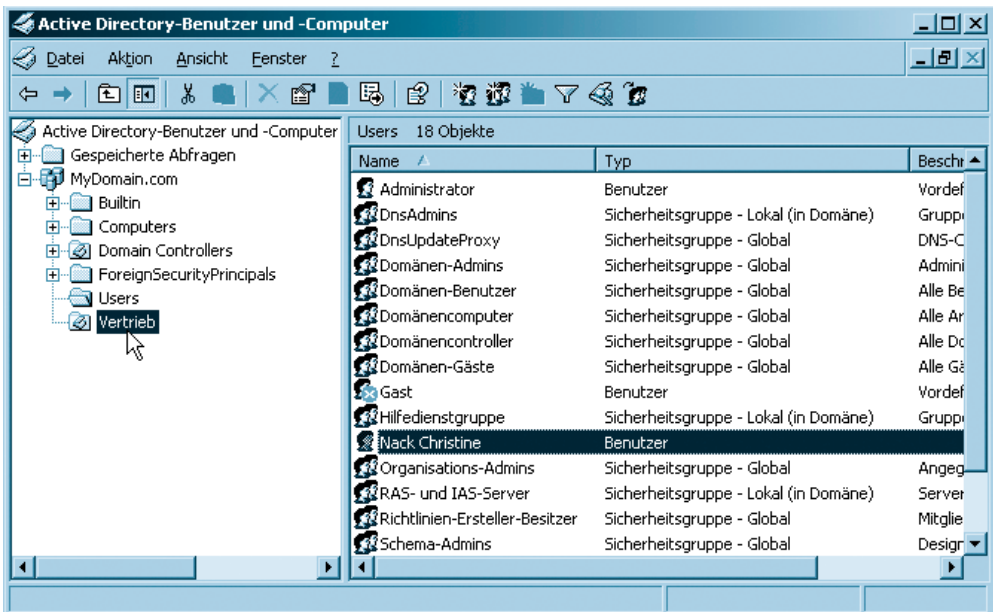


Abbildung 60: Ein Benutzerkonto per Drag & Drop verschieben

Die Lösung per Eingabeaufforderung

Ein vorhandenes Domänen-Benutzerkonto lässt sich mit dem `Dsmove`-Befehl auch von der Eingabeaufforderung aus verschieben. Wenn Sie z.B. das Benutzerkonto *Nack Christine*, das sich derzeit im Container *Users* der Domäne *MyDomain.com* befindet, in die Organisationseinheit *Vertrieb* verschieben wollen, dann wenden Sie den `Dsmove`-Befehl an, dem Sie den definierten Namen des Benutzerkontos und – gefolgt von dem Parameter `-newparent` – auch den definierten Namen des Zielknotens übergeben.

```
@Echo off
Dsmove "CN=Nack Christine,CN=Users,DC=MyDomain,DC=Com" _
    -newparent OU=Vertrieb,DC=MyDomain,DC=Com
```

Listing 80: 2_01_014.bat

Die Lösung mit VBScript

Selbst mit VBScript ist das Verschieben eines Benutzerkontos in eine andere Organisationseinheit kein Problem. Das ADSI-Interface `IADSCONTAINER` stellt hierfür die `MoveHere`-Methode bereit, womit Sie einzelne Objekte (Benutzer, Gruppen, Computer ...) in einen anderen Container verschieben und wahlweise auch umbenennen können. Die `MoveHere`-Methode erwartet dabei die Angabe des Objektes, das verschoben und/oder umbenannt werden soll, wie auch den neuen ADS-Pfad und die künftige Bezeichnung des Objektes als Parameter.

Das nachfolgende Code-Beispiel zeigt, wie Sie mit Hilfe der `MoveHere`-Methode ein Benutzerkonto aus dem *Users*-Container in die Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* verschieben können. Dabei wird der Zielpfad an eine Objektvariable (`oOU`) gebunden und die `MoveHere`-Methode eingesetzt, wobei das zu verschiebende Objekt (`strUserName`), der neue Pfad (`strSourcePath`) und auch die künftige Bezeichnung (`strUserName`) übergeben werden. Letzteres bleibt in diesem Beispiel aber unverändert, da das Benutzerkonto nicht umbenannt, sondern nur verschoben werden soll.

```
Option Explicit

Dim oOU
Dim oUser
Dim strSourcePath
Dim strDestinationPath
Dim strUserName

strSourcePath = "CN=Users,DC=MyDomain,DC=Com"
```

Listing 81: 2_01_014.vbs

180 >> Welche SID hat der Benutzer?

```
strDestinationPath = "OU=Vertrieb,DC=MyDomain,DC=Com"
strUserName = "CN=NackC"
```

```
Set oOU = GetObject("LDAP://" & strDestinationPath)
Set oUser = oOU.MoveHere("LDAP://" & strUserName _
    & "," & strSourcePath, strUserName)
```

```
Set oOU = Nothing
Set oUser = Nothing
```

Listing 81: 2_01_014.vbs (Forts.)

15 Welche SID hat der Benutzer?

Jedem Benutzerkonto wird mit dem Anlegen des Kontos eine so genannte SID (Security Identifier) zugeordnet, die seitens des Betriebssystems zur eindeutigen Referenzierung des Benutzerkontos verwendet wird. Diese SID stellt einen Binärwert variabler Länge dar, der sich aus den folgenden drei Hauptkomponenten zusammensetzt:

- ▶ **Revision-Level:** Der Revision-Level gibt Auskunft über die Version der SID-Struktur. Alle SIDs, die mit Windows Server 2003 oder älteren Windows-Betriebssystemversionen (bis Windows NT, da ältere Windows-Versionen keine SIDs verwenden) erstellt wurden, verwenden den Revision-Level 1.
- ▶ **Identifier Authority:** Die Komponente *Identifier Authority* legt den höchsten Autoritätslevel fest, womit eine SID erzeugt werden kann. Die gängigsten Autoritätslevel sind 0 (*SECURITY_NULL_SID_AUTHORITY*), 1 (*SECURITY_WORLD_SID_AUTHORITY*), 2 (*SECURITY_LOCAL_SID_AUTHORITY*), 3 (*SECURITY_CREATOR_SID_AUTHORITY*) und 5 (*SECURITY_NT_AUTHORITY*).
- ▶ **Sub-Authorities:** Die Komponente *Sub-Authorities* enthält einen oder auch mehrere Werte, womit die untergeordneten Autoritätslevel bezeichnet werden. Der untergeordnete Autoritätslevel setzt sich dabei aus einer (meist aus mehreren Werten bestehenden) Domänen-ID und einer Benutzer- oder Gruppen-ID (auch *RID* = *Relative Identifier* genannt) zusammen, wobei der letzte Wert immer die Benutzer- oder Gruppen-ID abbildet und die Werte davor die Domänen-ID repräsentieren.

Die in eine Zeichenfolge konvertierte SID des lokalen Administrators kann beispielsweise folgendermaßen aussehen:

```
S-1-5-21-1659004503-113007714-854245398-500
```

Das Zeichen S gibt Auskunft darüber, dass es sich hierbei um eine SID handelt. Der darauf folgende Wert 1 stellt den Revisionslevel dar, der Wert 5 repräsentiert den übergeordneten Autoritätslevel (*NT-Authority*), die Werte 21-1659004503-113007714-

>> Benutzer und Gruppen

854245398 bilden die Domänen-ID, und der Wert 500 repräsentiert den Benutzer *Administrator*. Wenn Sie diese SID nun mit der SID des lokalen Administrators auf einem anderen Computersystem vergleichen, so werden Sie feststellen, dass mit Ausnahme der Domänen-ID alle Werte gleich sind.

Mit dieser Kombination lassen sich auf einzelnen Computersystemen und auch innerhalb der Domäne bzw. der Gesamtstruktur eindeutige SIDs erzeugen, die – sofern sie einmal erstellt wurden – kein zweites Mal verwendet werden. Wenn Sie also einen Benutzer löschen und den Benutzer neu anlegen, so können Sie im Frontend zwar den gleichen Benutzernamen verwenden, systemintern wird jedoch eine andere SID verwendet, weshalb der neue Benutzer nicht die gleichen Berechtigungen erhält wie der zuvor gelöschte Benutzer.

Wenn Sie nun die SIDs der einzelnen Benutzerkonten ermitteln möchten, weil Sie vielleicht die Objekte im Ordner *Recycler* (in diesem Ordner erfolgt keine SID-Auflösung) zuordnen wollen oder weil Sie eine SID-Übersetzungstabelle für den Import von Sicherheitsobjekten (Benutzer-, Gruppen-, Computerkonten) fertigen müssen, so bleibt nur der Einsatz der Eingabeaufforderung oder eines Skriptes, da der Windows Server 2003 unter grafischer Oberfläche leider keine Möglichkeiten bietet, womit Sie die SIDs der Benutzerkonten so ohne weiteres abfragen können.

Sie können zwar den Registrierungs-Editor zu Hilfe nehmen, in den Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\PROFILELIST wechseln und eine Zuordnung zwischen SID und Benutzer-Profilpfad vornehmen, allerdings werden in diesem Schlüssel nur die SIDs der Benutzer angezeigt, für die bereits ein Benutzerprofil angelegt wurde. Wenn Sie Benutzerkonten angelegt haben und sich die Benutzer bis dato noch nicht an Windows Server 2003 angemeldet haben, so werden die jeweiligen SIDs dort nicht abgebildet. Letzteres erfordert also den Einsatz der Eingabeaufforderung oder den Einsatz eines Skriptes, um die SID auszulesen.

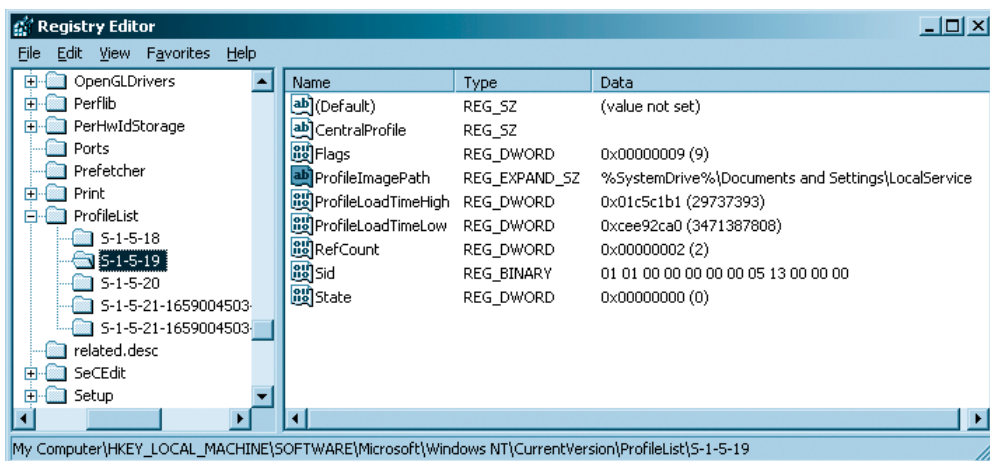


Abbildung 61: Der Schlüssel »ProfileList« zeigt Zuordnungen zwischen Benutzerprofil und SID.

Die Lösung per Eingabeaufforderung

Wenn Sie die SID des Benutzers ermitteln möchten, der gerade an der Konsole angemeldet ist, so brauchen Sie in der Eingabeaufforderung nur den `Whoami`-Befehl mit dem Parameter `/user` anzuwenden, da dieser Befehl den Benutzernamen und auch die SID der aktuellen Benutzeranmeldung zurückgibt:

```
Whoami /user
```

Um stattdessen die SID eines anderen Benutzerkontos zu ermitteln, führen Sie mit dem `WMIC`-Befehl (unter Angabe des Alias `UserAccount`) eine WMI-Abfrage gegen die Klasse `Win32_UserAccount` durch und fragen den Wert der Eigenschaft `SID` ab. Wenn Sie z.B. die SID von *TestUser1* ermitteln möchten, so wenden Sie den `WMIC UserAccount`-Befehl an, selektieren das Benutzerobjekt mit Hilfe der `Where`-Klausel und geben mit `Get SID` nur die SID des angegebenen Benutzers aus.

```
@Echo off
WMIC UserAccount Where "Name='TestUser1'" Get SID
```

Listing 82: 2_01_015a.bat

In einer Domänenumgebung lässt sich die SID eines bestimmten Benutzerkontos auch mit dem `Dsget`-Befehl ermitteln. Wenn Sie beispielsweise die SID des Benutzers *TestUser5* herausfinden müssen, der in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* abgelegt ist, dann wenden Sie den `Dsget`-Befehl mit der Option `User` an, übergeben den definierten Namen des zu überprüfenden Benutzerkontos und legen mit dem Parameter `-sid` fest, dass lediglich die Sicherheitskennung ausgegeben werden soll.

```
@Echo off
Dsget User CN=TestUser5,OU=Vertrieb,DC=MyDomain,DC=Com -sid
```

Listing 83: 2_01_015b.bat

Die Lösung mit VBScript

Mit VBScript lässt sich die SID eines bestimmten Benutzerkontos am einfachsten mit WMI ermitteln, da die WMI-Klasse `Win32_UserAccount` (WMI-Namespace `root\cimv2`) die Eigenschaft `SID` besitzt, deren Wert Sie nur abzufragen brauchen.

Das nachfolgende Code-Beispiel zeigt eine solche Lösung: Zunächst wird eine Abfrage gegen die WMI-Klasse `Win32_UserAccount` durchgeführt, um alle Benutzerkonten zu selektieren, deren Name gleich dem gesuchten Benutzerkonto (`strUserName`) entspricht. Das Ergebnis dieser Abfrage wird daraufhin in Form einer Kollektion zurückgeliefert. Mit einer `For Each`-Schleife können Sie dann auf jedes Element der Kollektion zugreifen und den Wert der Eigenschaft `SID` abfragen.

>> Benutzer und Gruppen

```
Option Explicit
```

```
Dim oWMI
Dim strComputer
Dim strUserName
Dim colUser
Dim strUser
```

```
strComputer = "."
strUserName = "TestUser5"
```

```
Set oWMI = GetObject("winmgmts:{impersonationlevel=
    impersonate}!\" & strComputer & "\root\cimv2")
Set colUser = oWMI.ExecQuery("SELECT * FROM _
    Win32_UserAccount WHERE Name ='" & strUserName & "'")
```

```
For Each strUser in colUser
    WScript.Echo strUser.Name & ": " & strUser.SID
Next
```

```
Set oWMI = Nothing
```

Listing 84: 2_01_015.vbs

16 Kennwörter zurücksetzen

Hin und wieder kommt es vor, dass sich Benutzer nicht mehr anmelden können, weil sie ihr Kennwort vergessen haben oder weil das Kennwort (aus welchen Gründen auch immer) während der Anmeldung einfach nicht akzeptiert wird. Sie können nun – sofern Sie als *Administrator* angemeldet sind – das Kennwort des Benutzers zurücksetzen, indem Sie das aktuelle Kennwort durch ein neues Kennwort überschreiben.

Wenn Sie das Kennwort für ein lokales Benutzerkonto zurücksetzen müssen, das auf einem Standalone- oder Member-Server eingerichtet ist, so melden Sie sich als *Administrator* an und öffnen die lokale Benutzerverwaltung. Anschließend markieren Sie das betreffende Benutzerkonto, öffnen dessen Kontextmenü und wählen den Eintrag KENNWORT FESTLEGEN... aus. Sie gelangen daraufhin in ein Dialogfenster, worüber Sie das neue Kennwort vergeben können.

Wenn Sie das Kennwort eingegeben und auch bestätigt haben, dann schließen Sie das Dialogfenster wieder. Sie sollten nun das Kennwort-Attribut BENUTZER MUSS KENNWORT BEI DER NÄCHSTEN ANMELDUNG ÄNDERN setzen, das Sie in den EIGENSCHAFTEN des Benutzerkontos festlegen können. Damit erzwingen Sie die Kennwortänderung durch den Benutzer und vermeiden, dass der Benutzer weiterhin das Startkennwort nutzt.

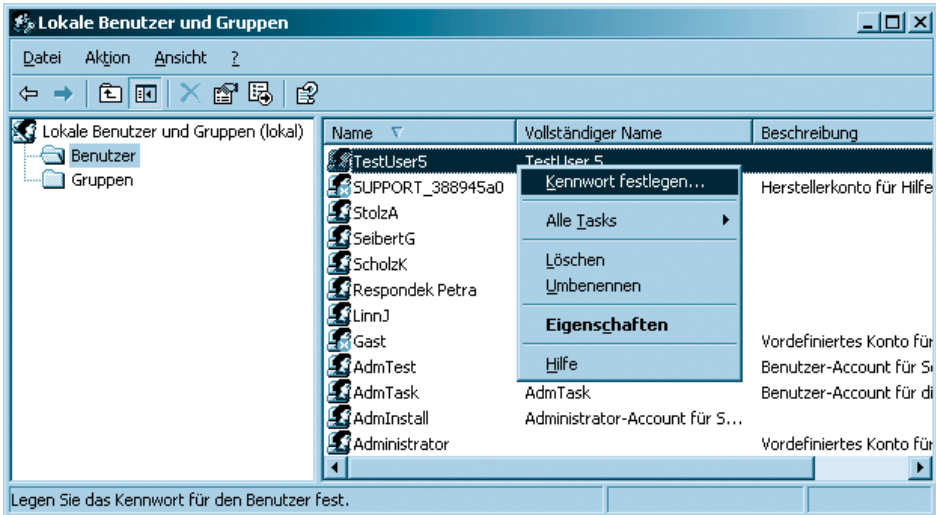


Abbildung 62: Ein neues Kennwort festlegen

Um das Kennwort eines Domänenbenutzerkontos zurückzusetzen, öffnen Sie statt der lokalen Benutzerverwaltung das Verwaltungswerkzeug **ACTIVE DIRECTORY-BENUTZER UND -COMPUTER**. Anschließend markieren Sie das betreffende Benutzerkonto, öffnen dessen Kontextmenü und wählen den Eintrag **KENNWORT ZURÜCKSETZEN...** aus. Sie gelangen daraufhin in das Dialogfenster **KENNWORT ZURÜCKSETZEN** und können das neue Startkennwort vergeben.

Auch in diesem Fall sollten Sie wieder das Kennwort-Attribut **BENUTZER MS KENNWORT BEI DER NÄCHSTEN ANMELDUNG ÄNDERN SETZEN**, um die Kennwort-Änderung durch den Benutzer zu erzwingen. Das Kennwort-Attribut finden Sie in den **EIGENSCHAFTEN** des Benutzerkontos, genauer gesagt auf der Registerkarte **KONTO**.

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus das Kennwort eines lokalen Benutzerkontos zurücksetzen müssen, dann können Sie hierfür den **Net User**-Befehl einsetzen, dem Sie lediglich den Namen des Benutzerkontos wie auch das neue Kennwort als Parameter übergeben:

```
@Echo off
Net User TestUser5 QWEasyxc
```

Listing 85: 2_01_016a.bat

Der **Net User**-Befehl versucht daraufhin, eine Verbindung zu dem angegebenen Benutzerkonto herzustellen und das Kennwort zu überschreiben. Hierbei muss jedoch erwähnt werden, dass der **Net User**-Befehl ohne Angabe weiterer Parameter immer auf die lokale Benutzerverwaltung des Computersystems zugreift, auf dem der Befehl

>> Benutzer und Gruppen

gerade ausgeführt wird. Wenn Sie also von einem Client oder Member-Server aus das Kennwort ein Domänenbenutzerkontos ändern müssen (was durchaus möglich ist), dann müssen Sie den Befehl gegen die Domäne ausführen, was Sie mit dem zusätzlichen Parameter `/Domain` festlegen können.

```
@Echo off
Net User TestUser5 QWEasydxc /Domain
```

Listing 86: 2_01_016b.bat

Wenn es darum geht, ein Domänenbenutzerkonto zu modifizieren, so hält der Windows Server 2003 noch ein weiteres Tool bereit: `Dsmod User`. Mit diesem Befehl können Sie – ähnlich wie mit dem `Net User`-Befehl – ein vorhandenes Kennwort durch ein neues Kennwort überschreiben, indem Sie dem `Dsmod User`-Befehl den definierten Namen des betreffenden Benutzerkontos übergeben und mit dem Parameter `-pwd` das neue Kennwort festlegen:

```
@Echo off
Dsmod User CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com _
-pwd 123QWEasd
```

Listing 87: 2_01_016c.bat

Die Lösung mit VBScript

Um per VBScript ein Kennwort zurückzusetzen, stellen Sie lediglich eine Verbindung zu dem betreffenden User-Objekt her, wenden die `SetPassword`-Methode gegen das User-Objekt an und übergeben das neue Kennwort.

Wenn Sie beispielsweise auf einem Standalone- oder Member-Server das Kennwort des lokalen Benutzerkontos `TestUser5` auf `123QWEasd` zurücksetzen möchten, dann binden Sie das Benutzerkonto des betreffenden Computersystems mit der `GetObject`-Methode an eine Objektvariable und wenden die `SetPassword`-Methode gegen die Objektvariable an, der Sie das neue Kennwort als Parameter übergeben.

```
Option Explicit

Dim oUser
Dim strComputer
Dim strUserName
Dim strPassword

strComputer = "."
strUserName = "TestUser5"
```

Listing 88: 2_01_016a.vbs

186 >> Existiert die Gruppe bereits?

```
strPassword = "123QWEasd"

Set oUser = GetObject("WinNT://" & strComputer & "/" & _
    & strUserName)
oUser.SetPassword strPassword

Set oUser = Nothing
```

Listing 88: 2_01_016a.vbs (Forts.)

Wenn Sie stattdessen das Kennwort eines Domänenbenutzerkontos zurücksetzen müssen, dann nutzen Sie den ADSI-Provider LDAP und binden den definierten Namen des Benutzerkontos an eine Objektvariable (oUser). Anschließend wenden Sie auch hier wieder die SetPassword-Methode gegen die Objektvariable an und übergeben das neue Kennwort.

```
Option Explicit

Dim oUser
Dim strUserName
Dim strPassword

strUserName = "CN=TestUser5,OU=Vertrieb,DC=MyDomain,DC=Com"
strPassword = "123QWEasd"

Set oUser = GetObject("LDAP://" & strUserName)
oUser.SetPassword strPassword

Set oUser = Nothing
```

Listing 89: 2_01_016b.vbs**17 Existiert die Gruppe bereits?**

Bevor Sie eine neue Gruppe anlegen, sollten Sie sich zunächst vergewissern, dass diese Gruppe noch nicht existiert. Denn Gruppen müssen eindeutig sein und können daher nicht mehrfach angelegt werden.

Um das Vorhandensein einer Gruppe in der lokalen Benutzerverwaltung zu überprüfen, rufen Sie auf dem Standalone- oder Member-Server das Verwaltungswerkzeug **LOKALE BENUTZER UND GRUPPEN** auf. Anschließend markieren Sie den Ordner **GRUPPEN** und prüfen anhand der Einträge, ob die gewünschte Gruppe bereits vorhanden ist.

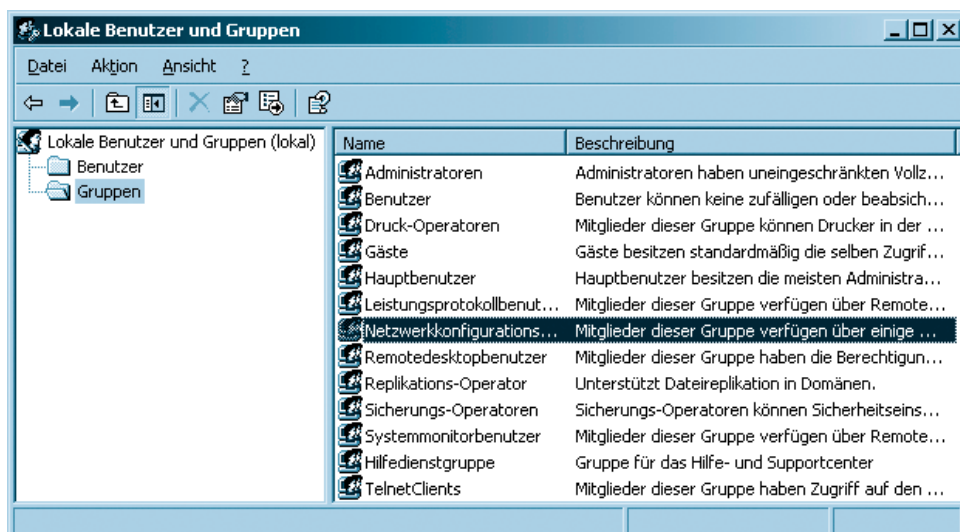


Abbildung 63: Die lokalen Gruppen überprüfen

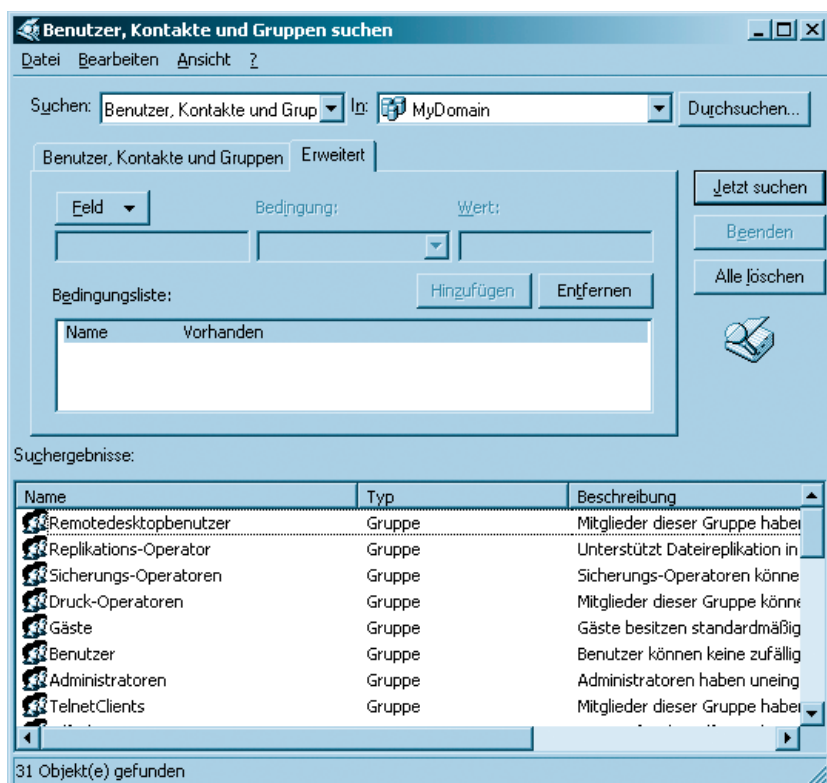


Abbildung 64: Die Anzeige aller Gruppen als Suchergebnis

188 >> Existiert die Gruppe bereits?

Und um das Vorhandensein einer Domänengruppe zu überprüfen, nutzen Sie stattdessen das Verwaltungswerkzeug ACTIVE DIRECTORY-BENUTZER UND -COMPUTER. Wenn Sie lediglich eine der vordefinierten Standardgruppen überprüfen möchten, so brauchen Sie sich nur den Inhalt der Container BUILTIN und USERS anzusehen. Haben Sie dagegen eigene Gruppen angelegt, die sich in unterschiedlichen Organisationseinheiten befinden, so können Sie die Gruppen leichter überprüfen, wenn Sie die Suchfunktion zu Hilfe nehmen.

Um nach einer Domänengruppe zu suchen, öffnen Sie das Kontextmenü der Domäne oder wahlweise auch eines darunter liegenden Zweiges und wählen den Eintrag SUCHEN... aus. Im Dialogfenster BENUTZER, KONTAKTE UND GRUPPEN SUCHEN aktivieren Sie die Registerkarte ERWEITERT, worüber Sie nach bestimmten Feldinhalten suchen können. Klicken Sie auf FELD, dann auf GRUPPE und anschließend auf NAME, um die zuvor markierte Ebene nach Gruppen zu durchsuchen. Als Bedingung wählen Sie VORHANDEN aus und klicken anschließend auf HINZUFÜGEN, um die Einstellung zu übernehmen. Klicken Sie auf JETZT SUCHEN, um eine Auflistung aller Gruppen zu erzeugen (siehe Abbildung 64).

Die Lösung per Eingabeaufforderung

Das Vorhandensein einer Gruppe lässt sich natürlich auch von der Eingabeaufforderung aus überprüfen. Wenn Sie z.B. herausfinden möchten, ob bereits eine lokale Gruppe mit der Bezeichnung *TestBenutzer* existiert, dann wenden Sie hierfür den Befehl `Net LocalGroup` an und übergeben den zu überprüfenden Gruppennamen als Parameter. Der `Net LocalGroup`-Befehl versucht daraufhin, eine Verbindung zu dem Gruppenkonto herzustellen und dessen Eigenschaften auszulesen, welche standardmäßig am Bildschirm ausgegeben werden. Ist die Gruppe jedoch nicht vorhanden, so gibt der `Net LocalGroup`-Befehl die Meldung *Die angegebene lokale Gruppe ist nicht vorhanden* aus.

```
@Echo off
Net LocalGroup TestBenutzer
```

Listing 90: 2_01_017a.bat

Ähnlich gehen Sie vor, um das Vorhandensein einer globalen Gruppe zu überprüfen, nur dass Sie nicht den `Net LocalGroup`-Befehl verwenden, sondern eine `Net Group`-Abfrage gegen die entsprechende Gruppe durchführen.

```
@Echo off
Net Group TestBenutzer
```

Listing 91: 2_01_017b.bat

Standardmäßig liest der `Net LocalGroup-/Net Group`-Befehl die Gruppen des lokalen Computersystems aus, von wo aus die Abfrage gestartet wird. Wenn Sie mit Hilfe des `Net Group`-Befehls die globalen Gruppen einer Domäne von einem Client oder Member-Server aus abfragen möchten, so fügen Sie der Befehlszeile den Parameter `/Domain an`, um die Abfrage nicht gegen das lokale Computersystem, sondern gegen die Domäne zu richten.

```
Net Group TestBenutzer /Domain
```

Innerhalb einer Domäne lassen sich Gruppen auch mit dem `Dsquery Group`-Befehl abfragen. Wenn Sie beispielsweise herausfinden möchten, ob in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* eine Gruppe mit der Bezeichnung *Test-Benutzer* existiert, dann führen Sie eine `Dsquery Group`-Abfrage gegen die Domäne durch und übergeben den definierten Namen der gesuchten Gruppe. Der `Dsquery Group`-Befehl versucht daraufhin, eine Verbindung zu der angegebenen Gruppe herzustellen, und gibt den definierten Namen der Gruppe zurück, wenn die angegebene Gruppe gefunden wurde. Anderenfalls gibt der `Dsquery Group`-Befehl die Meldung `Dsquery ist fehlgeschlagen:Verzeichnisobjekt nicht gefunden aus`.

```
@Echo off
Dsquery Group CN=TestBenutzer,OU=Vertrieb,DC=MyDomain,_,
DC=Com > NUL
```

Listing 92: 2_01_017c.bat

Die Lösung mit VBScript

Um das Vorhandensein einer bestimmten Gruppe per VBScript zu überprüfen, stellen Sie zunächst eine Verbindung zu dem betreffenden Computersystem oder zu der Domäne her und setzen die `Filter`-Eigenschaft mit dem Filterkriterium `Group` ein, woraufhin eine Kollektion mit den vorhandenen Gruppen zurückgeliefert wird. Mit Hilfe einer `For Each`-Schleife prüfen Sie dann für jedes Element der Kollektion, ob der Wert der Eigenschaft `Name` dem gesuchten Gruppennamen entspricht. Und wenn diese Bedingung zutrifft, so wurde die gesuchte Gruppe gefunden, und die Schleife kann mit `Exit For` vorzeitig verlassen werden, da weitere Ergebnisse aufgrund der Eindeutigkeit der Gruppennamen nicht geliefert werden können.

Wenn Sie die lokalen Gruppen eines Standalone- oder Member-Servers per VBScript auf ihre Existenz hin überprüfen möchten, so stellen Sie mit der `GetObject`-Methode eine Verbindung zum ADSI-Provider `WinNT` des betreffenden Computersystems her und wenden die `Filter`-Eigenschaft mit dem Kriterium `Group` an, um eine Kollektion der vorhandenen Gruppen zu erhalten und dann jedes Element auf seinen Wert hin zu überprüfen:

190 >> Existiert die Gruppe bereits?

```
Option Explicit
```

```
Dim oComputer
Dim strComputer
Dim strGroup
Dim strGroupName
```

```
strComputer = "."
strGroupName = "TestBenutzer"
```

```
Set oComputer = GetObject("WinNT://" & strComputer)
oComputer.Filter = Array("Group")
```

```
For Each strGroup in oComputer
    If strGroup.Name = strGroupName Then
        WScript.Echo "Die Gruppe existiert ..."
        Exit For
    End If
Next
```

```
Set oComputer = Nothing
```

Listing 93: 2_01_017a.vbs

Auf gleiche Art und Weise lassen sich natürlich auch die Gruppen der Domäne überprüfen. Einziger Unterschied hierbei: Sie stellen die Verbindung nicht zu einem lokalen Computersystem her, sondern mittels LDAP zur Domäne oder wahlweise auch zu einer Organisationseinheit und prüfen den Wert der Eigenschaft `SamAccountName`:

```
Option Explicit
```

```
Dim oOU
Dim strOU
Dim strGroup
Dim strGroupName
```

```
strOU = "OU=Vertrieb,DC=MyDomain,DC=Com"
strGroupName = "TestBenutzer"
```

```
Set oOU = GetObject ("LDAP://" & strOU)
oOU.Filter = Array("Group")
```

```
For Each strGroup in oOU
    If strGroup.SamAccountName = strGroupName Then
        WScript.Echo "Die Gruppe existiert ..."
    End If
Next
```

Listing 94: 2_01_017b.vbs

```

Exit For
End If
Next

Set oOU = Nothing

```

Listing 94: 2_01_017b.vbs (Forts.)

18 Gruppen auflisten

Für den Fall, daß Sie eine Auflistung der bereits eingerichteten Gruppen benötigen, so können Sie sich die Gruppen unter grafischer Oberfläche mit Hilfe der Benutzerverwaltung anzeigen oder mit der Export-Funktion auch in eine Datei ausgeben lassen. Wie Sie dabei vorgehen, hängt lediglich davon ab, ob Sie nur die lokalen Gruppen eines Standalone- oder Member-Servers oder aber die lokalen, globalen und universalen Gruppen eines Domänen-Controllers auflisten möchten.

Um beispielsweise alle lokalen Gruppen, die auf einem Standalone- oder Member-Server eingerichtet wurden, am Bildschirm anzeigen zu lassen, öffnen Sie zunächst die lokale Benutzerverwaltung (LOKALE BENUTZER UND GRUPPEN) des betreffenden Servers. Anschließend markieren Sie den Ordner GRUPPEN, woraufhin alle eingerichteten Gruppen in der rechten Fensterhälfte angezeigt werden.

Möchten Sie die Gruppen nicht nur am Bildschirm ausgeben, sondern in eine Datei exportieren, so markieren Sie den Ordner GRUPPEN, wählen im Kontextmenü des Ordners LISTE EXPORTIEREN... aus und geben den Dateinamen wie auch das gewünschte Format (Text- oder Unicode-Format mit Tabulatoren- oder Kommatrennung) an. Die Gruppen werden daraufhin in die angegebene Datei exportiert.

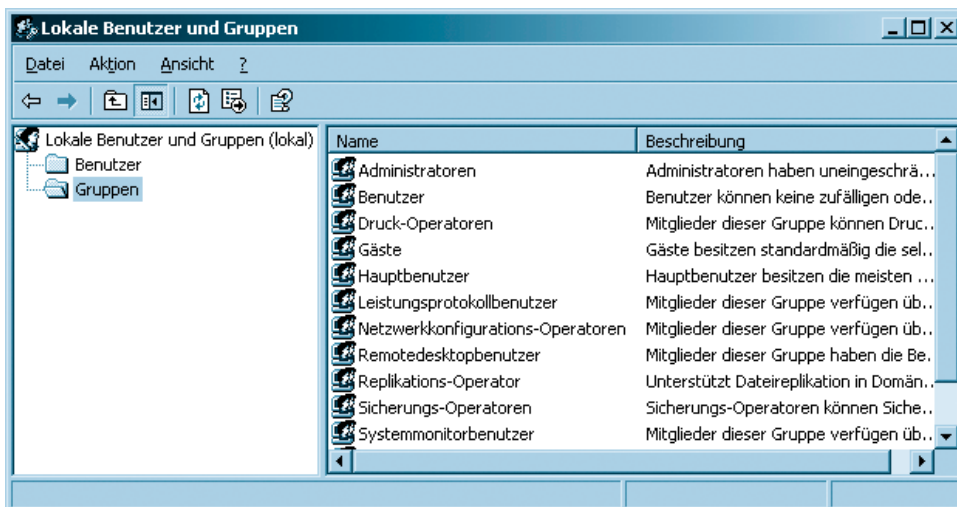


Abbildung 65: Die Auflistung der lokalen Gruppen

Wenn Sie dagegen alle lokalen, globalen und universalen Gruppen eines Domänen-Controllers auflisten möchten, so starten Sie das Verwaltungswerkzeug ACTIVE DIRECTORY-BENUTZER UND -COMPUTER und nehmen entweder die Such-Funktion zu Hilfe oder führen eine Abfrage durch. Denn die Gruppen sind in der Regel in unterschiedlichen Containern und – je nach Active Directory-Design – auch in unterschiedlichen Organisationseinheiten abgelegt, womit sich die Auflistung der Gruppen nicht ganz so einfach gestaltet wie die der lokalen Gruppen auf einem Standalone- oder Member-Server.

Starten Sie also aus dem Verwaltungsmenü heraus ACTIVE DIRECTORY-BENUTZER UND -COMPUTER, und wählen Sie im Kontextmenü von GESPEICHERTE ABFRAGEN erst NEU und dann ABFRAGE aus, um eine neue Abfrage zu definieren. Geben Sie im Feld NAME eine kurze Bezeichnung für die Abfrage (z.B. *Gruppen-Liste*) ein, wählen Sie den ABFRAGESTAMM aus, und klicken Sie anschließend auf FESTLEGEN, um die eigentliche Abfrage zu definieren.

Im Dialogfenster ALLGEMEINE ABFRAGEN SUCHEN wechseln Sie auf die Registerkarte GRUPPEN und wählen im Feld NAME den Eintrag HAT EINEN WERT aus. Klicken Sie auf HINZUFÜGEN, um die Abfragedefinition zu übernehmen, und schließen Sie das Dialogfenster wieder. Die Abfrage wird daraufhin ausgeführt, und sämtliche Gruppen des zuvor festgelegten Abfragestammes werden in der rechten Fensterhälfte angezeigt.

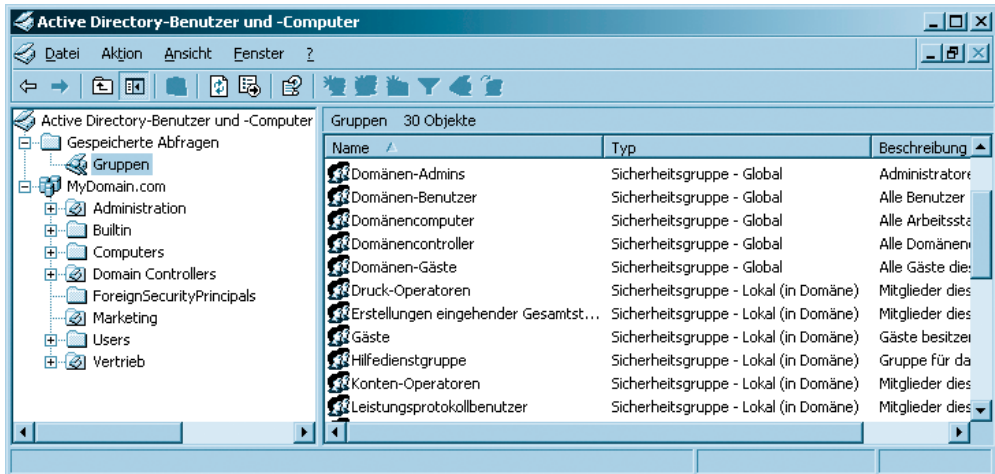


Abbildung 66: Die Auflistung aller Gruppen eines Domänen-Controllers

Auch in diesem Fall können Sie die Auflistung der Gruppen wieder in eine Datei exportieren, indem Sie das Kontextmenü von GESPEICHERTE ABFRAGEN öffnen und LISTE EXPORTIEREN ... auswählen.

Die Lösung per Eingabeaufforderung

Mit Hilfe der Befehle `Net LocalGroup` und `Net Group` lassen sich die Gruppen auch per Eingabeaufforderung auslesen, beispielsweise aus einer Batch-Datei heraus. Wenn Sie z.B. eine Liste aller lokalen Gruppen eines Standalone- oder Member-Servers in eine Datei namens *GroupList.txt* exportieren möchten, so wenden Sie den Befehl `Net LocalGroup` unter Angabe der gewünschten Ausgabeumleitung (`> GroupList.txt`) an:

```
@Echo off
Net LocalGroup > GroupList.txt
```

Listing 95: 2_01_018a.bat

Möchten Sie dagegen die globale Gruppe der Domäne auslesen, so verwenden Sie hierfür den `Net Group`-Befehl, der keine weiteren Parameter erwartet, sofern er direkt auf dem Domänen-Controller ausgeführt wird:

```
@Echo off
Net Group > GroupList.txt
```

Listing 96: 2_01_018b.bat

Falls Sie den `Net Group`-Befehl von einem Client oder von einem Member-Server aus aufrufen, so fügen Sie den Parameter `/Domain` an, damit die Abfrage gegen die Domäne und nicht gegen das lokale Computersystem ausgeführt wird. Denn Letzteres würde auf einem Client und auch auf einem Standalone-Server zu einem Laufzeitfehler führen.

```
@Echo off
Net Group /Domain > GroupList.txt
```

Listing 97: 2_01_018c.bat

Innerhalb einer Domäne lassen sich die Gruppen auch mit dem `Dsquery Group`-Befehl auslesen, der im Gegensatz zum `Net LocalGroup`- und `Net Group`-Befehl weitaus mehr Möglichkeiten bietet. Denn mit dem `Dsquery Group`-Befehl können Sie zum einen den Startpunkt der Suche bestimmen und zum anderen auch den Suchbereich als solches festlegen.

- ▶ **Startpunkt:** Legt den Startknoten der Suche fest. Geben Sie `DomainRoot` an, wenn die Suche innerhalb der aktuellen Domäne erfolgen soll, oder `ForestRoot`, um die Suche im gesamten Forest durchzuführen. Alternativ können Sie auch den definierten Namen einer bestimmten Organisationseinheit oder eines Containers angeben, wenn Sie die Suche auf eine bestimmte OU oder auf einen Container einschränken möchten. Ohne Angabe dieses Parameters wird `DomainRoot` verwendet.

194 >> Gruppen auflisten

- **Suchbereich:** Legt den Suchbereich der Dsquery-Abfrage fest. Geben Sie `-Scope Subtree` an, wenn alle untergeordneten Teilstrukturen des angegebenen Startpunktes durchsucht werden sollen. Verwenden Sie stattdessen `-Scope OneLevel`, um nur die erste Ebene der untergeordneten Teilstruktur zu durchsuchen, und `-Scope Base`, um ausschließlich im angegebenen Startpunkt zu suchen. Ohne Angabe dieses Parameters wird `-Scope Subtree` verwendet.

Um beispielsweise alle Gruppen der aktuellen Domäne unter Angabe ihres definierten Namens in eine Datei mit der Bezeichnung *GroupList.txt* auszugeben, verwenden Sie folgende Befehlszeile:

```
@Echo off
Dsquery Group DomainRoot > GroupList.txt
```

Listing 98: 2_01_018d.bat

Die Lösung mit VBScript

Um die lokalen Gruppen eines Standalone- oder Member-Servers per VBScript auszulesen, stellen Sie mit der `GetObject`-Methode eine Verbindung zum ADSI-Provider WinNT des betreffenden Computersystems her und wenden anschließend die `Filter`-Eigenschaft mit dem Kriterium `Group` an, woraufhin eine Kollektion der vorhandenen Gruppen zurückgeliefert wird.

Mit Hilfe einer `For Each`-Schleife können Sie dann für jedes Element den Wert der Eigenschaft `Name` auslesen und mit der `WScript.Echo`-Anweisung am Bildschirm ausgeben.

```
Option Explicit

Dim oComputer
Dim strComputer
Dim strGroup
strComputer = "."

Set oComputer = GetObject("WinNT://" & strComputer)
oComputer.Filter = Array("Group")

For Each strGroup in oComputer
    WScript.Echo strGroup.Name
Next

Set oComputer = Nothing
```

Listing 99: 2_01_018a.vbs

Das Auslesen der Domänengruppen nehmen Sie über eine Verbindung zum ADSI-Provider LDAP vor. Wenn Sie beispielsweise alle Gruppen des Containers *Users* abfragen und in eine Datei namens *GroupList.txt* ausgeben möchten, so stellen Sie eine Verbindung zum ADSI-Provider LDAP her und übergeben den definierten Namen des Containers, um eine entsprechende Objektvariable zu referenzieren. Anschließend wenden Sie die *Filter*-Eigenschaft mit dem Kriterium *Group* an, um eine Kollektion der Gruppen zu erhalten. Mit Hilfe einer *For Each*-Schleife können Sie dann auf jedes Gruppenelement zugreifen und den Wert der Eigenschaft *SamAccountName* abfragen.

```
Option Explicit

Dim oDomain
Dim strDomain
Dim strGroup
strDomain = "CN=Users,DC=MyDomain,DC=Com"

Set oDomain = GetObject ("LDAP://" & strDomain)
oDomain.Filter = Array("Group")

For Each strGroup in oDomain
    WScript.Echo strGroup.SamAccountName
Next

Set oDomain = Nothing
```

Listing 100: 2_01_018b.vbs

19 Neue Gruppen anlegen

Neue Gruppen sind dank der komfortablen Benutzerverwaltung von Windows Server 2003 schnell erstellt. Wenn Sie z.B. auf einem Standalone- oder Member-Server eine lokale Gruppe anlegen möchten, so melden Sie sich zunächst als Administrator oder als Hauptbenutzer an dem betreffenden Computersystem an und öffnen dann die lokale Benutzerverwaltung.

Innerhalb der Benutzerverwaltung markieren Sie den Ordner GRUPPEN und wählen aus dem Kontextmenü heraus NEUE GRUPPE... aus, woraufhin ein Dialogfenster zum Anlegen einer neuen Gruppe geöffnet wird. Legen Sie im Feld GRUPPENNAME den Namen der neuen Gruppe (max. 256 Zeichen) fest, und geben Sie im Feld BESCHREIBUNG wahlweise eine kurze Erläuterung zu Sinn und Zweck der Gruppe ein. Klicken Sie anschließend auf ERSTELLEN, um die neue Gruppe anzulegen.

Um dagegen eine neue Gruppe im Active Directory anzulegen, melden Sie sich als Administrator oder zumindest als Mitglied der Gruppen KONTEN-OPERATOREN oder ORGANISATIONS-ADMINS an und rufen aus dem Verwaltungsmenü heraus ACTIVE DIRECTORY-BENUTZER UND -COMPUTER auf.



Abbildung 67: Eine lokale Gruppe anlegen

Markieren Sie anschließend den betreffenden Container oder die Organisationseinheit, und wählen Sie aus dem Kontextmenü heraus den Eintrag NEU und anschließend GRUPPE aus, woraufhin das Dialogfenster NEUES OBJEKT – GRUPPE geöffnet wird. Geben Sie im Feld GRUPPENAMEN einen eindeutigen und aussagekräftigen Namen für die neue Gruppe ein, und legen Sie anschließend den Gruppenbereich und Gruppentyp fest.

- ▶ **Gruppenbereich:** Legen Sie den Gruppenbereich LOKAL (IN DOMÄNE) fest, um Zugriffe auf Ressourcen oder Objekte innerhalb der Domäne zu steuern. Im gemischten Modus kann eine lokale Gruppe Benutzerkonten und globale Gruppen aus sämtlichen Domänen der Gesamtstruktur enthalten. Im einheitlichen Modus dagegen kann eine lokale Gruppe weitere lokale Gruppen wie auch globale und universale Gruppen enthalten. Verwenden Sie den Gruppentyp GLOBAL, um eine globale Gruppe für die Zugriffssteuerung innerhalb der Domäne zu erstellen, in der die Gruppe erstellt wurde. Im gemischten Modus kann eine globale Gruppe nur Benutzerkonten der eigenen Domäne enthalten, im einheitlichen Modus dagegen kann eine globale Gruppe weitere globalen Gruppen der Domäne enthalten, in der sie erstellt wurde. Sofern Sie den einheitlichen Modus verwenden, können Sie auch den Gruppenbereich UNIVERSAL auswählen. Eine universale Gruppe kann für die Zugriffssteuerung der Ressourcen und Objekte in der Gesamtstruktur verwendet werden und sowohl Benutzerkonten wie auch globale und universale Gruppen sämtlicher Domänen innerhalb der Gesamtstruktur enthalten.
- ▶ **Gruppentyp:** Legen Sie den Gruppentyp SICHERHEIT fest, wenn Sie die Gruppe in einer ACL aufnehmen möchten, um Zugriffe für Ressourcen und Objekte zu steuern. Legen Sie dagegen VERTEILUNG fest, wenn Sie lediglich einen E-Mail-Verteiler erstellen möchten. Eine Verteilergruppe können Sie nicht in der ACL einer Ressource oder eines Objektes aufführen, um Berechtigungen zu steuern.

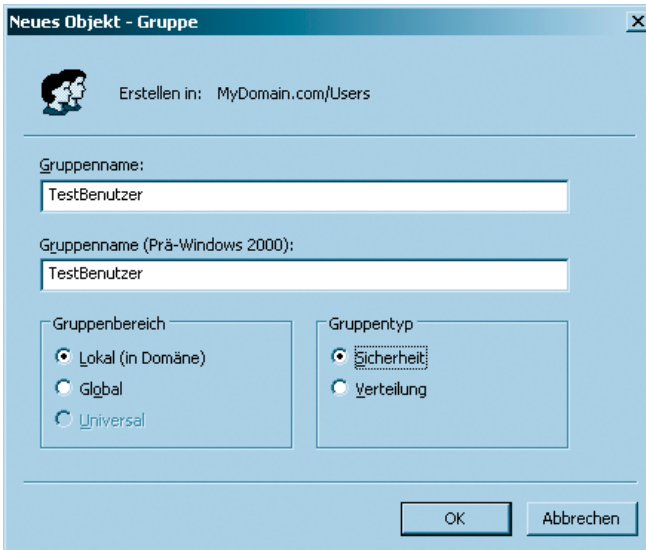


Abbildung 68: Eine lokale Sicherheitsgruppe anlegen

Wenn Sie den Gruppennamen wie auch den Gruppenbereich und -typ festgelegt haben, klicken Sie anschließend auf OK, um die neue Gruppe im Active Directory anzulegen.

Die Lösung per Eingabeaufforderung

Um von der Eingabeaufforderung aus auf einem Standalone- oder Member-Server eine lokale Gruppe anzulegen, setzen Sie den `Net LocalGroup`-Befehl ein. Wenn Sie z.B. eine neue lokale Gruppe mit der Bezeichnung *TestBenutzer* anlegen möchten, so wenden Sie den `Net LocalGroup`-Befehl gegen das betreffende Computersystem an und übergeben den Namen der Gruppe wie auch die Option `/Add` als Parameter.

```
@Echo off
Net LocalGroup TestBenutzer /Add
```

Listing 101: 2_01_019a.bat

Wenn Sie dagegen im Active Directory der aktuellen Domäne eine globale Sicherheitsgruppe anlegen möchten, so wenden Sie stattdessen den `Net Group`-Befehl gegen die Domäne an und übergeben den Gruppennamen wie auch die Optionen `/Add` und `/Domain` als Parameter:

```
@Echo off
Net Group TestBenutzer /Add /Domain
```

Listing 102: 2_01_019b.bat

198 >> Neue Gruppen anlegen

Zum Anlegen neuer Gruppen im Active Directory können Sie auch den `Dsadd Group`-Befehl verwenden, der im Gegensatz zum `Net LocalGroup`- und `Net Group`-Befehl weitaus mehr Möglichkeiten bietet. Denn mit dem `Dsadd Group`-Befehl können Sie – ähnlich wie unter grafischer Oberfläche – den Gruppenbereich und auch den Gruppentyp festlegen, was mit Hilfe des `Net LocalGroup`- und `Net Group`-Befehls leider nicht der Fall ist.

Wenn Sie z.B. in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* eine globale Sicherheitsgruppe anlegen möchten, so setzen Sie den `Dsadd Group`-Befehl gegen die aktuelle Domäne ein und übergeben zunächst den definierten Namen der neuen Gruppe, um den Knoten zu bestimmen. Dann legen Sie mit dem Parameter `-secgrp` fest, ob es sich um eine Sicherheitsgruppe (`yes`) oder um eine Verteilergruppe (`no`) handelt. Mit dem Parameter `-scope` legen Sie den Gültigkeitsbereich der neuen Gruppe fest. Verwenden Sie `-scope l` für eine lokale Gruppe, `-scope g` für eine globale Gruppe und `-scope u` für eine universale Gruppe. Und zu guter Letzt übergeben Sie mit dem Parameter `-samid` den Namen der Gruppe.

```
@Echo off
Dsadd Group CN=TestBenutzer,OU=Vertrieb,DC=MyDomain,_
DC=Com -secgrp yes -scope g -samid TestBenutzer
```

Listing 103: 2_01_019c.bat

Die Lösung mit VBScript

Eine neue Gruppe lässt sich natürlich auch per VBScript anlegen. Wenn Sie beispielsweise auf einem Standalone- oder Member-Server eine lokale Gruppe mit der Bezeichnung *TestBenutzer* erstellen möchten, so stellen Sie im einfachsten Fall eine Verbindung zum ADSI-Provider `WinNT` des betreffenden Computersystems her und wenden die `Create`-Methode an, der Sie den Objekttyp (`Group`) und auch den Namen der Gruppe (`strGroupName`) übergeben. Wenn Sie zeitgleich auch eine Beschreibung festlegen möchten, so weisen Sie der Eigenschaft `Description` den entsprechenden Wert zu und speichern die neue Gruppe anschließend mit der `SetInfo`-Methode ab.

```
Option Explicit

Dim oComputer
Dim oGroup
Dim strComputer
Dim strGroupName
Dim strDescription

strComputer = "."
strGroupName = "TestBenutzer"
```

Listing 104: 2_01_019a.vbs

```
strDescription = "Temporäre TestBenutzer"

Set oComputer = GetObject("WinNT://" & strComputer)
Set oGroup = oComputer.Create("Group", strGroupName)
oGroup.Put "Description", strDescription
oGroup.SetInfo

Set oComputer = Nothing
Set oGroup = Nothing
```

Listing 104: 2_01_019a.vbs (Forts.)

Auf ähnliche Art und Weise legen Sie eine neue Gruppe im Active Directory an. Um beispielsweise in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* eine globale Sicherheitsgruppe mit der Bezeichnung *TestBenutzer* anzulegen, stellen Sie mit der *GetObject*-Methode zunächst eine Verbindung zum ADSI-Provider *LDAP* her und übergeben den definierten Namen des Startknotens. Anschließend wenden Sie die *Create*-Methode gegen das *OU*-Objekt an und übergeben den Objekttyp (*Group*) wie auch den definierten Namen der neuen Gruppe (*strGroupName*).

Um nun den Gruppentyp und auch den Gruppenbereich zu bestimmen, weisen Sie der *GroupType*-Eigenschaft den hexadezimalen Wert des Gruppentyps und des Gruppenbereichs zu. Folgende Werte sind möglich:

Gruppen-Typ	Wert
ADS_GROUP_TYPE_GLOBAL_GROUP	&H2
ADS_GROUP_TYPE_LOCAL_GROUP	&H4
ADS_GROUP_TYPE_DOMAIN_LOCAL_GROUP	&H4
ADS_GROUP_TYPE_UNIVERSAL_GROUP	&H8
ADS_GROUP_TYPE_SECURITY_ENABLED	&H80000000

Tabelle 21: Die hexadezimalen Werte der Gruppentypen und des Gruppenbereichs

Einer globalen Sicherheitsgruppe weisen Sie demnach den Wert der Konstanten *ADS_GROUP_TYPE_GLOBAL_GROUP* und *ADS_GROUP_TYPE_SECURITY_ENABLED* zu, während Sie einer globalen Verteilerguppe nur den Wert von *ADS_GROUP_TYPE_GLOBAL_GROUP* zuweisen.

```
Option Explicit

Dim oOU
Dim oGroup
Dim strOU
```

Listing 105: 2_01_019b.vbs

200 >> Eigenschaften einer Gruppe auslesen

```
Dim strGroupName

strOU = "OU=Vertrieb,DC=MyDomain,DC=Com"
strGroupName = "CN=TestBenutzer"
Const GlobalGroup = &H2
Const SecurityGroup = &H80000000

Set oOU = GetObject("LDAP://" & strOU)
Set oGroup = oOU.Create("Group", strGroupName)
oGroup.Put "SamAccountName", "TestBenutzer"
oGroup.Put "GroupType", GlobalGroup OR SecurityGroup
oGroup.SetInfo

Set oOU = Nothing
Set oGroup = Nothing
```

Listing 105: 2_01_019b.vbs (Forts.)

20 Eigenschaften einer Gruppe auslesen

Auch Gruppen verfügen über gewisse Eigenschaften, die Sie in der Regel beim Anlegen des Gruppenkontos konfigurieren, die sich aber auch während der laufenden Administration verändern können. Denken Sie beispielsweise an die Gruppenmitgliedschaften: Wenn Sie einen Benutzer zu einer Gruppe hinzufügen und eine neue Mitgliedschaft definieren, so ändert sich in diesem Fall nicht nur die Eigenschaft MITGLIEDSCHAFT des Benutzerkontos, sondern auch die Eigenschaft MITGLIEDER des Gruppenkontos.

Um nun unter grafischer Oberfläche herauszufinden, welche Eigenschaften ein bestimmtes Gruppenkonto besitzt, starten Sie lediglich die Benutzerverwaltung und rufen aus dem Kontextmenü der entsprechenden Gruppe die Eigenschaften auf.

Auf einem Standalone- oder Member-Server starten Sie hierfür die lokale Benutzerverwaltung, die Sie in der COMPUTERVERWALTUNG unter LOKALE BENUTZER UND GRUPPEN finden oder mit `Lusrmgr.msc` auch direkt aufrufen können. Im Ordner GRUPPEN markieren Sie die betreffende Gruppe und öffnen aus dem Kontextmenü heraus die EIGENSCHAFTEN.

Wenn Sie dagegen die Eigenschaften einer Domänengruppe auslesen möchten, so rufen Sie im Verwaltungsmenü ACTIVE DIRECTORY-BENUTZER UND -COMPUTER auf oder öffnen das STARTMENÜ, klicken auf AUSFÜHREN und geben `Dsa.msc` ein. Anschließend öffnen Sie das Kontextmenü der betreffenden Gruppe und rufen die Eigenschaften auf. Auf den Registerkarten ALLGEMEIN, MITGLIEDER, MITGLIED VON und VERWALTET VON finden Sie nun alle Eigenschaften der Domänengruppe.



Abbildung 69: Die Eigenschaften einer Gruppe

Die Lösung per Eingabeaufforderung

Die Eigenschaften einer Gruppe lassen sich natürlich auch per Eingabeaufforderung auslesen. Hierfür stellt der Windows Server 2003 die Befehle `Net LocalGroup`, `Net Group` und `Dsget Group` zur Verfügung, wobei Sie den `Net LocalGroup` zum Auslesen der Eigenschaften einer lokalen Gruppe und die Befehle `Net Group` und `Dsget Group` zur Auflistung der Eigenschaften einer Domänengruppe verwenden.

Wenn Sie z.B. auf einem Standalone- oder Member-Server die Eigenschaften der lokalen Gruppe *TestBenutzer* ausgeben möchten, so wenden Sie den `Net LocalGroup`-Befehl an, dem Sie lediglich den Namen der Gruppe als Parameter übergeben. Der `Net LocalGroup`-Befehl versucht daraufhin, eine Verbindung zu dem angegebenen Gruppenkonto herzustellen und dessen Eigenschaften (*Name*, *Beschreibung*, *Mitglieder*) am Bildschirm auszugeben.

```
@Echo off
Net LocalGroup TestBenutzer
```

Listing 106: 2_01_020a.bat

Um dagegen die Eigenschaften einer Domänengruppe auszulesen, können Sie zum einen den `Net Group`-Befehl anwenden, zum anderen aber auch den `Dsget Group`-

202 >> Eigenschaften einer Gruppe auslesen

Befehl. Der `Net Group`-Befehl verwendet die gleiche Syntax wie der `Net LocalGroup`-Befehl und erfordert daher nur die Angabe des Gruppennamens, den Sie als Parameter übergeben müssen. Sofern die angegebene Gruppe existiert, gibt der `Net Group`-Befehl daraufhin den Namen der Gruppe, die Beschreibung zum Gruppenkonto wie auch die einzelnen Mitglieder am Bildschirm aus.

```
@Echo off
Net Group TestBenutzer
```

Listing 107: 2_01_020b.bat

Für den Fall, dass Sie den `Net Group`-Befehl nicht direkt auf dem Domänen-Controller ausführen, so müssen Sie den zusätzlichen Parameter `/Domain` anwenden, um die Abfrage explizit gegen die Domäne zu richten. Denn anderenfalls würde ein Laufzeitfehler auftreten, da der `Net Group`-Befehl die Abfrage standardmäßig gegen das Computersystem richtet, auf dem der Befehl ausgeführt wird.

Etwas komfortabler gestaltet sich das Auslesen der Gruppeneigenschaften mit Hilfe des `Dsget Group`-Befehls, da Sie weitaus mehr Informationen auslesen können als mit dem `Net Group`-Befehl. Allerdings müssen Sie dem `Dsget Group`-Befehl per Parameter genau mitteilen, welche Eigenschaften ausgelesen werden sollen (siehe nachfolgende Tabelle).

Parameter	Beschreibung
-dn	Definierter Name der Gruppe
-samid	Name des Gruppenkontos
-sid	SID der Gruppe
-desc	Beschreibung des Gruppenkontos
-secgrp	Sicherheitsgruppe (Ja) oder nicht (Nein)
-scope	Gruppenbereich (Lokal, Global oder Universal)
-memberof	Gruppen, in denen die angegebene Gruppe selbst Mitglied ist
-members	Mitglieder der angegebenen Gruppe
-expand	Führt eine rekursive Abfrage aus und ist nur mit den Parametern -memberof und -members zu verwenden

Tabelle 22: Die Parameter zum Auslesen der Eigenschaften

Wenn Sie z.B. die SID, den Gruppentyp und auch Gruppenbereich einer bestimmten Domänengruppe (z.B. *TestBenutzer*) auslesen möchten, so wenden Sie auf dem Domänen-Controller den `Dsget Group`-Befehl unter Angabe des definierten Namens der abzufragenden Gruppe und auch der erforderlichen Parameter an:

```
@Echo off
Dsget Group CN=TestBenutzer,OU=Vertrieb,DC=MyDomain,DC=Com _
-sid -secgrp -scope
```

Listing 108: 2_01_020c.bat

Die Lösung mit VBScript

Um die Eigenschaften einer Gruppe per VBScript auszulesen, stellen Sie im einfachsten Fall eine Verbindung zum WinNT- oder LDAP-Provider des Computersystems oder der Domäne her und binden das Group-Objekt an eine Objektvariable. Damit haben Sie Zugriff auf die Eigenschaften und können deren Werte ermitteln.

Die nachfolgende Tabelle zeigt, welche Eigenschaften durch eine WinNT- oder LDAP-Verbindung ausgelesen werden können:

Eigenschaft	WinNT	LDAP	Beschreibung
Name	•	•	Name der Gruppe
Description	•	•	Beschreibung zum Gruppenkonto
Member		•	Mitglieder der Gruppe
Members	•		Mitglieder der Gruppe
SamAccountName		•	SAM-Kontoname der Gruppe

Tabelle 23: Die Eigenschaften eines Group-Objektes

Um beispielsweise auf einem Standalone- oder Member-Server die Eigenschaften einer lokalen Gruppe auszulesen, stellen Sie mit der GetObject-Methode eine Verbindung zum WinNT-Provider des betreffenden Computersystems her und binden das entsprechende Group-Objekt an eine Objektvariable (oGroup). Anschließend ermitteln Sie die Werte der zuvor genannten Eigenschaften und geben das Ergebnis mit der WScript.Echo-Anweisung am Bildschirm aus.

Option Explicit

```
Dim oGroup
Dim strComputer
Dim strGroupName
Dim strMember
```

```
strComputer = "."
strGroupName = "TestBenutzer"
Set oGroup = GetObject("WinNT://" & strComputer & "/" & strGroupName)
```

Listing 109: 2_01_020a.vbs

204 >> Eigenschaften einer Gruppe auslesen

```

WScript.Echo "Name: " & oGroup.Name
WScript.Echo "Beschreibung: " & oGroup.Description & vbCRLF
WScript.Echo "Mitglieder: "

For Each strMember in oGroup.Members
    WScript.Echo strMember.Name
Next

Set oGroup = Nothing

```

Listing 109: 2_01_020a.vbs (Forts.)

Ähnlich gehen Sie vor, wenn Sie die Eigenschaften einer Domänengruppe auslesen möchten. Wenn Sie z.B. den Kontonamen, die Beschreibung und auch die Mitglieder der globalen Gruppe *TestBenutzer* ermitteln und am Bildschirm ausgeben möchten, dann stellen Sie mittels LDAP eine Verbindung zu der betreffenden Gruppe her, lesen die Werte der jeweiligen Eigenschaften aus und geben die zurückgewonnene Information mit Hilfe der *Wscript.Echo*-Anweisung am Bildschirm aus:

```

Option Explicit

Dim oGroup
Dim strGroupName
Dim strMember

strGroupName = "CN=TestBenutzer,OU=Vertrieb,DC=MyDomain,DC=Com"
Set oGroup = GetObject("LDAP://" & strGroupName)

WScript.Echo "Name: " & oGroup.Name
WScript.Echo "SAM-Name: " & oGroup.SAMAccountName
WScript.Echo "Beschreibung: " & oGroup.Description & vbCRLF
WScript.Echo "Mitglieder: "

For Each strMember in oGroup.Member
    WScript.Echo strMember
Next

Set oGroup = Nothing

```

Listing 110: 2_01_020b.vbs

21 Benutzer in Gruppen aufnehmen

Wenn Sie eine neue Gruppe angelegt haben, so gilt es nun, diese Gruppe mit Leben, sprich: mit Mitgliedern, zu füllen. Auch wenn Sie die vordefinierten Gruppen, die mit der Installation des Betriebssystems eingerichtet werden, erweitern möchten, fügen Sie die Mitglieder der Gruppe hinzu.

Um in einer lokalen Gruppe auf einem Standalone- oder Member-Server ein oder auch mehrere Mitglieder (dies können einzelne Benutzerkonten wie auch Gruppen sein) aufzunehmen, melden Sie sich zunächst als Administrator an, rufen die lokale Benutzerverwaltung (LOKALE BENUTZER UND GRUPPEN) auf und markieren im Ordner GRUPPEN die betreffende Gruppe, der Sie weitere Mitglieder hinzufügen möchten. Wählen Sie im Kontextmenü der Gruppe den Eintrag ZUR GRUPPE HINZUFÜGEN... aus, oder rufen Sie die EIGENSCHAFTEN auf, um Mitglieder aufzunehmen.

Klicken Sie auf HINZUFÜGEN, dann auf ERWEITERT und anschließend auf JETZT SUCHEN, um eine Liste der Benutzerkonten und Gruppen zu erhalten, und markieren Sie die Benutzerkonten und Gruppen, die Sie der Gruppe hinzufügen möchten. Wenn Sie die betreffenden Mitglieder ausgewählt haben, so schließen Sie die Dialogfenster mit einem Klick auf OK, womit die neue Einstellung übernommen wird.

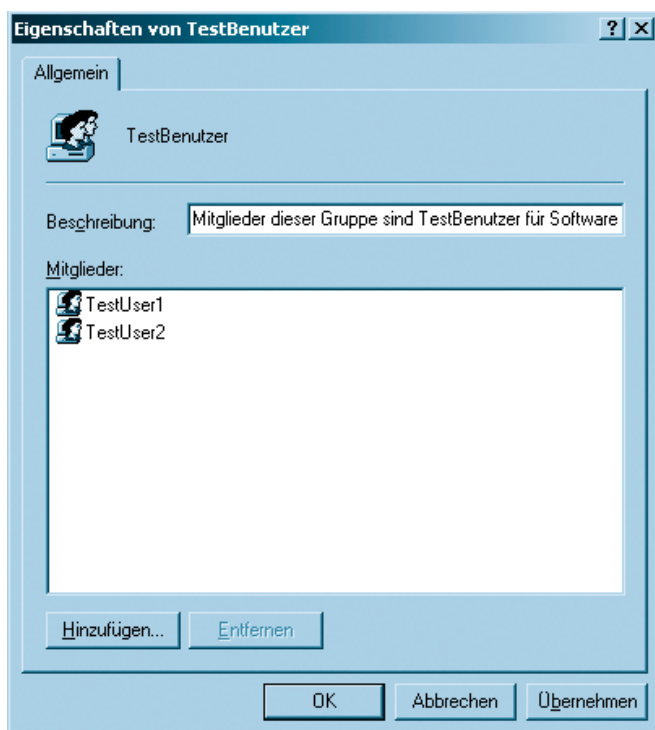


Abbildung 70: Neue Mitglieder aufnehmen

206 >> Benutzer in Gruppen aufnehmen

Auf ähnliche Art und Weise können Sie auch lokale, globale und universale Gruppen um weitere Mitglieder erweitern. Rufen Sie zunächst das Verwaltungswerkzeug ACTIVE DIRECTORY-BENUTZER UND -COMPUTER auf, und markieren Sie die betreffende Gruppe, die Sie um weitere Mitglieder erweitern möchten. Öffnen Sie aus dem Kontextmenü heraus die EIGENSCHAFTEN, und wechseln Sie auf die Registerkarte MITGLIEDER, um neue Benutzerkonten – und wahlweise auch weitere Gruppen – aufzunehmen.

Jede Gruppe kann natürlich auch weitere Gruppen enthalten. Welche Gruppentypen (*Lokal*, *Global* oder *Universal*) Sie dabei aufnehmen können, hängt nun von dem Typ der Gruppe ab, der Sie weitere Mitglieder hinzufügen möchten:

- ▶ **Lokale Gruppen:** Im gemischten Modus kann eine lokale Gruppe Benutzerkonten und auch globale Gruppen aller Domänen der Gesamtstruktur enthalten. Im einheitlichen Modus können auch weitere lokale Gruppen und universale Gruppen aufgenommen werden.
- ▶ **Globale Gruppen:** Im gemischten Modus kann eine globale Gruppe nur Benutzerkonten der Domäne aufnehmen, in der die Gruppe erstellt wurde. Im einheitlichen Modus dagegen können auch andere globale Gruppen der gleichen Domäne aufgenommen werden.
- ▶ **Universale Gruppen:** Eine universale Gruppe kann Benutzerkonten, globale und auch universale Gruppen sämtlicher Domänen der Gesamtstruktur enthalten.

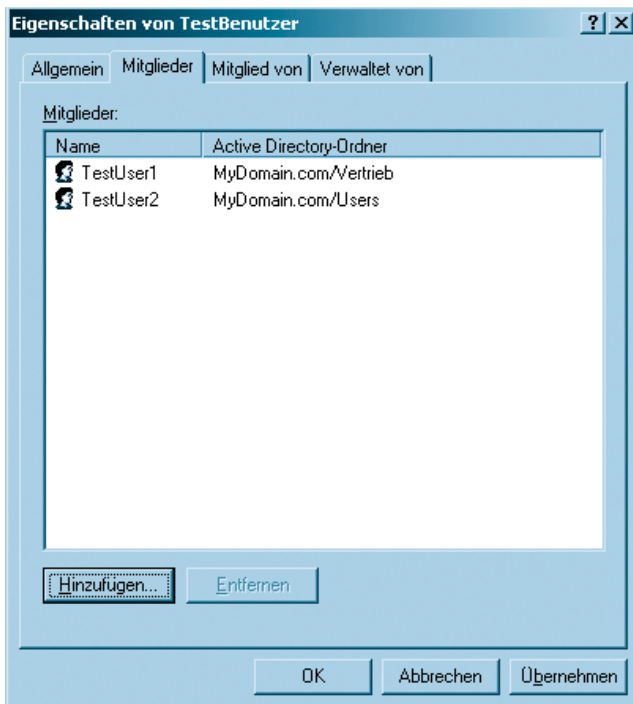


Abbildung 71: Die neuen Mitglieder hinzufügen

Wenn Sie auf HINZUFÜGEN, anschließend auf ERWEITERT und dann auf JETZT SUCHEN klicken, erhalten Sie eine Liste der Objekte, die Sie der Gruppe hinzufügen können. Markieren Sie die betreffenden Benutzerkonten (und wahlweise auch die Gruppen), und klicken Sie auf OK, um die ausgewählten Objekte als Mitglieder aufzunehmen (siehe Abbildung 71).

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus einer lokalen Gruppe neue Mitglieder hinzufügen möchten, so verwenden Sie hierfür den `Net LocalGroup`-Befehl, dem Sie folgende Parameter übergeben:

- ▶ Gruppe: Der Name der Gruppe, die um weitere Mitglieder ergänzt werden soll.
- ▶ Mitglieder: Der Name der Benutzerkonten, die der Gruppe als Mitglieder hinzugefügt werden sollen. Sie können hierbei auch mehrere Benutzerkonten angeben, die Sie jeweils durch ein Leerzeichen getrennt hintereinander angeben.
- ▶ Aktion: Legen Sie die Aktion `/Add` fest, um neue Mitglieder aufzunehmen.

Um in der lokalen Gruppe *TestBenutzer* die beiden Benutzerkonten *TestUser1* und *TestUser2* aufzunehmen, führen Sie folgende Befehlszeile aus:

```
@Echo off
Net LocalGroup TestBenutzer TestUser1 TestUser2 /Add
```

Listing 111: 2_01_021a.bat

Die zuvor genannte Befehlszeile können Sie verwenden, um auf einem Standalone- oder Member-Server eine lokale Gruppe mit weiteren Mitgliedern zu versehen, wie auch auf einem Domänen-Controller, um eine lokale Sicherheits- oder Verteilergruppe zu erweitern. Letzteres verlangt jedoch den zusätzlichen Parameter `/Domain`, wenn die Befehlszeile nicht direkt auf dem Domänen-Controller ausgeführt wird, sondern von einem anderen Computersystem aus aufgerufen wird.

Möchten Sie innerhalb der Domäne eine globale oder universale Gruppe mit weiteren Mitgliedern versehen, so verwenden Sie stattdessen den Befehl `Net Group`, der die gleiche Syntax verwendet wie der `Net LocalGroup`-Befehl:

```
@Echo off
Net Group TestBenutzer TestUser1 TestUser2 /Add /Domain
```

Listing 112: 2_01_021b.bat

Eine weitere Möglichkeit, um Domänengruppen mit Mitgliedern zu versehen, bietet der `Dsmod Group`-Befehl, dem Sie den definierten Namen der Gruppe wie auch den Parameter `-addmbr` unter Angabe der aufzunehmenden Mitglieder übergeben:

208 >> Benutzer in Gruppen aufnehmen

```
@Echo off
Dsmmod Group CN=TestBenutzer,OU=Vertrieb,DC=MyDomain,DC=Com _
-addmbr CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com _
CN=TestUser2,OU=Vertrieb,DC=MyDomain,DC=Com
```

Listing 113: 2_01_021c.bat

Die Lösung mit VBScript

Die Gruppenmitgliedschaften lassen sich natürlich auch per VBScript pflegen. Wenn Sie z.B. der lokalen Gruppe *TestBenutzer* eines Standalone- oder Member-Servers die vorhandenen Benutzerkonten *TestUser1* und *TestUser2* hinzufügen möchten, so stellen Sie zunächst eine Verbindung zum ADSI-Provider WinNT her und erzeugen eine Referenz zu der betreffenden Gruppe. Anschließend wenden Sie die *Add*-Methode an und übergeben dabei den Namen des Benutzerkontos, das als neues Mitglied aufgenommen werden soll.

```
Option Explicit

Dim oGroup
Dim strComputer
Dim strGroupName
Dim strUserName1
Dim strUserName2

strComputer = "FRA-MD-FPS-001"
strGroupName = "TestBenutzer"
strUserName1 = "TestUser1"
strUserName2 = "TestUser2"

Set oGroup = GetObject("WinNT://" & strComputer & "/" & _
    strGroupName & ".group")
oGroup.Add "WinNT://" & strComputer & "/" & strUserName1
oGroup.Add "WinNT://" & strComputer & "/" & strUserName2

Set oGroup = Nothing
```

Listing 114: 2_01_021a.vbs

Wenn Sie dagegen neue Mitglieder in einer Gruppe aufnehmen möchten, die im Active Directory abgelegt ist, so erzeugen Sie zunächst eine Referenz zu der Gruppe, die um weitere Mitglieder ergänzt werden soll, und wenden dann die *PutEx*-Methode gegen das *Group*-Objekt (*oGroup*) an, um die Eigenschaft *Member* zu modifizieren und die Mitglieder hinzuzufügen. Die *PutEx*-Methode erwartet dabei drei Parameter, die Sie mit dem Methodenaufruf übergeben müssen:

- ▶ **Modus:** Der Modus legt fest, ob eine Objekteigenschaften hinzugefügt (Append), ersetzt (Replace), entfernt (Remove) oder gänzlich gelöscht (Delete) werden soll.
- ▶ **Name der Eigenschaft:** Bestimmt die Objekteigenschaft, die modifiziert werden soll. In diesem Beispiel übergeben Sie den Parameter *Member*.
- ▶ **Wert der Eigenschaft:** Mit dem letzten Parameter übergeben Sie den eigentlichen Wert der Eigenschaft.

Das nachfolgende Code-Beispiel zeigt, wie Sie die Gruppe *TestBenutzer*, die in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* abgelegt ist, um die Mitglieder *TestUser1* und *TestUser2* erweitern können:

Option Explicit

```
Dim oGroup
Dim strGroupName
Dim strUserName1
Dim strUserName2
```

```
strGroupName = "CN=TestBenutzer,OU=Vertrieb,_"
               DC=MyDomain,DC=Com"
strUserName1 = "CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com"
strUserName2 = "CN=TestUser2,OU=Vertrieb,DC=MyDomain,DC=Com"
Const Append = 3
```

```
Set oGroup = GetObject("LDAP://" & strGroupName)
oGroup.PutEx Append, "Member", strUserName1
oGroup.PutEx Append, "Member", strUserName2
oGroup.SetInfo
```

```
Set oGroup = Nothing
```

Listing 115: 2_01_021b.vbs

22 Welchen Gruppen gehört der Benutzer an?

Wenn Sie herausfinden müssen, welchen Gruppen ein bestimmter Benutzer angehört, so brauchen Sie nur die Eigenschaften des Benutzerkontos zu öffnen und sich auf der Registerkarte MITGLIEDSCHAFT die einzelnen Gruppenmitgliedschaften anzeigen zu lassen.

Um beispielsweise auf einem Standalone- oder Member-Server herauszufinden, welchen lokalen Gruppen der Benutzer *TestUser1* angehört, so rufen Sie zunächst die lokale Benutzerverwaltung auf. Anschließend markieren Sie im Ordner BENUTZER das Benutzerkonto, dessen Gruppenmitgliedschaften Sie überprüfen möchten, und öffnen aus dem Kontextmenü heraus die EIGENSCHAFTEN. Wechseln Sie auf die Registerkarte

210 >> Welchen Gruppen gehört der Benutzer an?

MITGLIEDSCHAFT, Sie erhalten nun eine Aufstellung aller Gruppenmitgliedschaften des entsprechenden Benutzers.

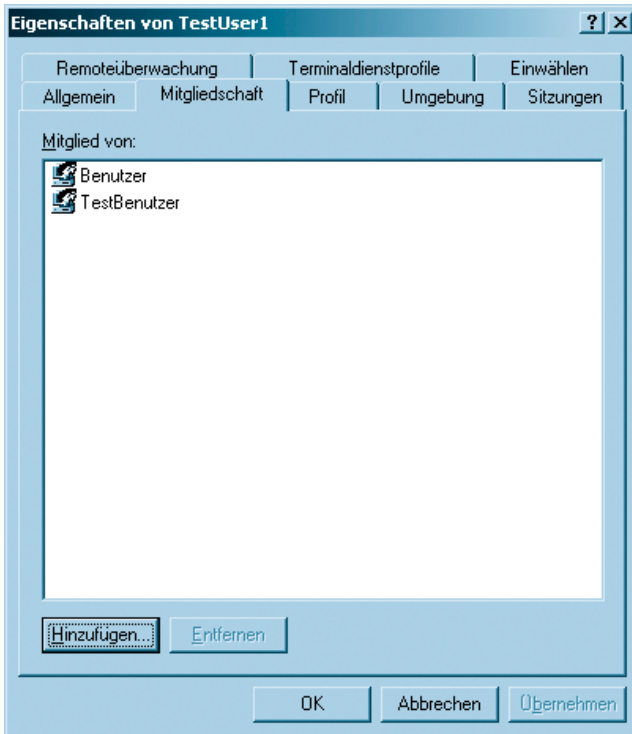


Abbildung 72: Die Gruppenmitgliedschaften des Benutzerkontos

Wenn Sie dagegen die Mitgliedschaften eines Domänenbenutzerkontos ermitteln möchten, so rufen Sie stattdessen **ACTIVE DIRECTORY-BENUTZER UND -COMPUTER** auf, markieren das betreffende Benutzerkonto und rufen die **EIGENSCHAFTEN** aus dem Kontextmenü heraus auf. Anschließend wechseln Sie auf die Registerkarte **MITGLIED VON**, dort werden alle Gruppenmitgliedschaften angezeigt.

Die Lösung per Eingabeaufforderung

Um von der Eingabeaufforderung aus die Gruppenmitgliedschaften eines Benutzerkontos zu ermitteln, können Sie zum einen den `Net User`-Befehl verwenden, zum anderen aber auch den `Dsget User`-Befehl. Der wesentliche Unterschied der beiden Befehle besteht darin, dass Sie den `Dsget User`-Befehl im Gegensatz zum `Net User`-Befehl nur in einer Domänenumgebung einsetzen können, dafür aber wesentlich mehr Abfragemöglichkeiten haben.

Wenn Sie z.B. herausfinden müssen, in welchen lokalen und globalen Gruppen das lokale Benutzerkonto mit der Bezeichnung `TestUser1` Mitglied ist, dann wenden Sie den `Net User`-Befehl gegen das betreffende Computersystem oder gegen die Domäne

(/Domain) an und übergeben den Namen des Benutzerkontos. Der *Net User*-Befehl versucht daraufhin, eine Verbindung zu dem angegebenen Benutzerkonto herzustellen und die Eigenschaften auszulesen. Bestandteil dieser Eigenschaften sind unter anderem auch die Mitgliedschaften der lokalen und globalen Gruppen.

```
@Echo off
Net User TestUser1
```

Listing 116: 2_01_022a.bat

Eine weitere Möglichkeit, die Gruppenmitgliedschaften eines Benutzers zu ermitteln, bietet der *Dsget User*-Befehl, der sich im Gegensatz zum *Net User*-Befehl aber nur innerhalb einer Domäne einsetzen lässt. Wenn Sie z.B. herausfinden müssen, in welchen Gruppen der Benutzer *TestUser1* Mitglied ist, so führen Sie eine *Dsget User*-Abfrage gegen das betreffende Benutzerkonto durch und übergeben den definierten Namen des Benutzerkontos wie auch den Parameter *-memberof*:

```
@Echo off
Dsget user CN=Testuser1,OU=Vertrieb,DC=MyDomain,DC=Com _
-mmemberof
```

Listing 117: 2_01_022b.bat

Jede Gruppe kann natürlich selbst Mitglied weiterer Gruppen sein. Um die effektiven Gruppenmitgliedschaften zu ermitteln, fügen Sie der Befehlszeile den zusätzlichen Parameter *-expand* an:

```
@Echo off
Dsget user CN=Testuser1,OU=Vertrieb,DC=MyDomain,DC=Com _
-mmemberof -expand
```

Listing 118: 2_01_022c.bat

Die Lösung mit VBScript

Um per VBScript herauszufinden, in welchen Gruppen ein bestimmter Benutzer Mitglied ist, so brauchen Sie nur die Werte der Eigenschaft *Groups* des betreffenden Benutzerkontos auszulesen, da in dieser Eigenschaft alle Gruppenmitgliedschaften hinterlegt sind. Wenn Sie z.B. auf einem Standalone- oder Member-Server die Gruppenmitgliedschaften des lokalen Benutzers *TestUser1* abfragen möchten, so stellen Sie im ersten Schritt mit *GetObject* eine Verbindung zum WinNT-Provider des entsprechenden Computersystems her und binden das *User*-Objekt an eine Objektvariable. Anschließend fragen Sie die Werte der Eigenschaft *Groups* ab, wobei Sie hier eine *For Each*-Schleife einsetzen müssen, da die *Group*-Eigenschaft mehrere Werte enthalten kann.

212 >> Welchen Gruppen gehört der Benutzer an?

Das nachfolgende Code-Beispiel liest die Gruppenmitgliedschaften des lokalen Benutzers *TestUser1* aus und gibt das Ergebnis, sprich: die einzelnen Mitgliedschaften, per WScript.Echo am Bildschirm aus.

```
Option Explicit

Dim oUser
Dim strComputer
Dim strUserName
Dim strGroup

strComputer = "FRA-MD-FPS-001"
strUserName = "TestUser1"

Set oUser = GetObject("WinNT://" & strComputer & "/" & strUserName)

For Each strGroup in oUser.Groups
    WScript.Echo strGroup.Name
Next

Set oUser = Nothing
```

Listing 119: 2_01_022a.vbs

Wenn Sie dagegen die Gruppenmitgliedschaften eines Domänenbenutzers auslesen möchten, so stellen Sie erst eine Verbindung zum LDAP-Provider her und binden das User-Objekt an eine Objektvariable. Anschließend setzen Sie eine For Each-Schleife ein, fragen die Werte der Eigenschaft *Groups* ab und geben mit der WScript.Echo-Anweisung für jedes ausgelesene Gruppenelement den Wert der Eigenschaft *SAM-AccountName* am Bildschirm aus.

Im Gegensatz zu lokalen Benutzerkonten verfügen Domänenbenutzerkonten zusätzlich über die Mitgliedschaft einer primären Gruppe (standardmäßig wird *Domänenbenutzer* verwendet), wobei diese Mitgliedschaft aber nicht in der Eigenschaft *Groups* hinterlegt ist und daher auch nicht mit ausgegeben wird. Wenn Sie also nicht nur die gewöhnlichen Mitgliedschaften abfragen möchten, sondern auch die primäre Gruppe ausgeben wollen, so lesen Sie zunächst den Wert der Eigenschaft *PrimaryGroupID* aus, womit Sie die Kennung der primären Gruppe zurückerhalten. Anschließend stellen Sie mit Hilfe des ADSI DB OLE-Providers eine Verbindung zur Active Directory-Datenbank her, selektieren innerhalb der angegebenen Domäne alle Group-Objekte und vergleichen den Wert der Eigenschaft *PrimaryGroupID* mit dem Wert der Eigenschaft *PrimaryGroupToken*. Wenn eine Übereinstimmung gefunden wurde, fragen Sie den Wert der Eigenschaft *Name* ab und geben den Namen per WScript.Echo-Anweisung am Bildschirm aus.

Option Explicit

```

Dim oUser
Dim strComputer
Dim strUserName
Dim strGroup
Dim oConnection
Dim oCommand
Dim oRecordSet
Dim strDomain
Dim strFilter
Dim strProperties
Dim strScope
Dim strPrimaryGroup

strComputer = "."
strUserName = "CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com"

Set oUser = GetObject("LDAP://" & strUserName)
For Each strGroup in oUser.Groups
    WScript.Echo strGroup.SamAccountName
Next

strDomain = "<LDAP://MyDomain.com>;"
strFilter = "(&(ObjectCategory=Group)" & strFilter & ");"
strProperties = "Name,PrimaryGroupToken,cn;"
strScope = "Subtree;"

Set oConnection = CreateObject("ADODB.Connection")
oConnection.Provider = "ADsDSOObject"
oConnection.Open "Active Directory Provider"

Set oCommand = CreateObject("ADODB.Command")
Set oCommand.ActiveConnection = oConnection
oCommand.CommandText = strDomain & strFilter & strProperties & strScope

Set oRecordSet = oCommand.Execute
strPrimaryGroup = ""

Do While ( (not oRecordSet.EOF) and (strPrimaryGroup = "") )
    If (oUser.PrimaryGroupID = oRecordSet.Fields("primaryGroupToken")_
        .Value) Then
        strPrimaryGroup = oRecordSet.Fields("Name").Value
    End If
    oRecordSet.MoveNext
Loop

```

Listing 120: 2_01_022b.vbs

214 >> Welche Benutzer sind in einer bestimmten Gruppe?

```
WScript.Echo strPrimaryGroup & " (PrimaryGroup)"
oConnection.Close
```

```
Set oUser = Nothing
Set oConnection = Nothing
Set oCommand = Nothing
Set oRecordSet = Nothing
```

Listing 120: 2_01_022b.vbs (Forts.)

23 Welche Benutzer sind in einer bestimmten Gruppe?

Ähnlich wie Sie die Gruppenmitgliedschaften eines bestimmten Benutzerkontos überprüfen können, so können Sie sich auch die Mitglieder der jeweiligen Gruppen anzeigen lassen. Um beispielsweise herauszufinden, welche lokalen Benutzerkonten in der Gruppe *TestBenutzer* enthalten sind, rufen Sie zunächst die lokale Benutzerverwaltung auf. Anschließend markieren Sie im Ordner GRUPPEN die Gruppe, deren Mitglieder Sie ermitteln möchten. Wählen Sie im Kontextmenü EIGENSCHAFTEN aus, Sie gelangen daraufhin in ein Dialogfenster, worüber alle Mitglieder angezeigt werden.

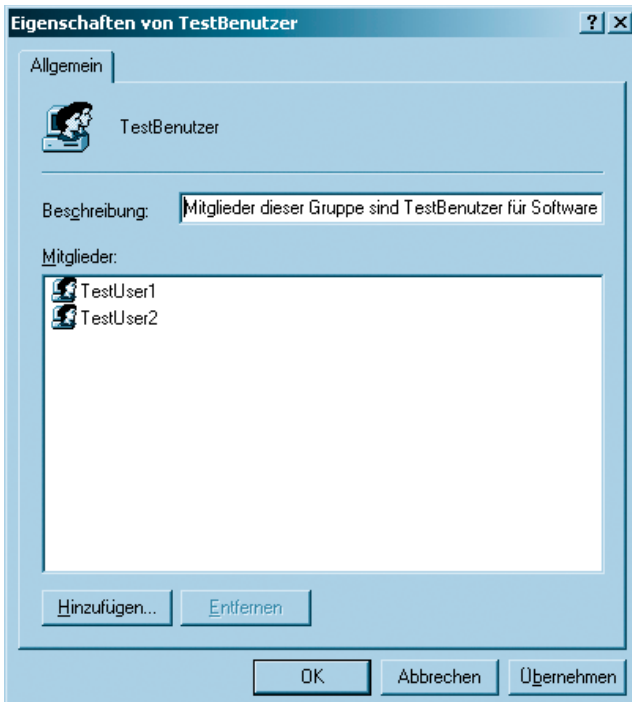


Abbildung 73: Die Mitglieder der Gruppe anzeigen lassen

Ähnlich gehen Sie auch vor, wenn Sie die Mitglieder einer Domänengruppe ermitteln möchten. Starten Sie hierzu ACTIVE DIRECTORY-BENUTZER UND -COMPUTER, und rufen Sie aus dem Kontextmenü der Gruppe die EIGENSCHAFTEN auf. Anschließend wechseln Sie auf die Registerkarte MITGLIEDER, worüber alle Benutzerkonten und Gruppen angezeigt werden, die Mitglied dieser Gruppe sind.

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus die Mitglieder einer lokalen Gruppe auslesen möchten, so wenden Sie den `Net LocalGroup`-Befehl an und übergeben den entsprechenden Gruppennamen als Parameter. Der `Net LocalGroup`-Befehl stellt daraufhin eine Verbindung zu der angegebenen Gruppe her und gibt – sofern die Gruppe existiert – dessen Mitglieder am Bildschirm aus.

Die nachfolgende `Net LocalGroup`-Befehlszeile liest beispielsweise die Mitglieder der lokalen Gruppe *TestBenutzer* aus und leitet das Ergebnis in die Datei *Member.txt* um:

```
@Echo off
Net LocalGroup TestBenutzer > Member.txt
```

Listing 121: 2_01_023a.bat

Um dagegen die Mitglieder einer Domänengruppe auszugeben, können Sie den `Net Group`-Befehl einsetzen, der die gleiche Syntax wie der `Net LocalGroup`-Befehl verwendet:

```
@Echo off
Net Group TestBenutzer /Domain > Member.txt
```

Listing 122: 2_01_023b.bat

In einer Domänenumgebung können Sie alternativ dazu auch den `Dsget Group`-Befehl anwenden, dem Sie lediglich den definierten Namen der auszulesenden Gruppe und auch den Parameter `-members` übergeben müssen:

```
@Echo off
Dsget Group CN=TestBenutzer,OU=Vertrieb,DC=MyDomain,DC=Com _
-members > Member.txt
```

Listing 123: 2_01_023c.bat

216 >> Welche Benutzer sind in einer bestimmten Gruppe?

Die Lösung mit VBScript

Mit Hilfe von ADSI lassen sich die Mitglieder einer Gruppe auch per VBScript problemlos ermitteln. Denn Sie müssen im Grunde genommen nur die Werte der Eigenschaft `Members` des betreffenden `Group`-Objektes auslesen, um eine Auflistung der Mitglieder zu erhalten.

Wenn Sie z.B. die Mitglieder der lokalen Gruppe *TestBenutzer* auslesen möchten, so binden Sie das entsprechende Gruppenobjekt mit `GetObject` an eine Objektvariable und lesen dann innerhalb einer `For Each`-Schleife die einzelnen Werte der Eigenschaft `Members` aus.

```
Option Explicit
```

```
Dim oGroup
Dim strComputer
Dim strGroupName
Dim strMember
```

```
strComputer = "."
strGroupName = "TestBenutzer"
```

```
Set oGroup = GetObject("WinNT://" & strComputer & "/" _
    & strGroupName)
For Each strMember in oGroup.Members
    WScript.Echo strMember.Name
Next
```

```
Set oGroup = Nothing
```

Listing 124: 2_01_023a.vbs

Auf die gleiche Art und Weise lassen sich auch die Mitglieder einer Domänengruppe auslesen, wobei Sie in diesem Fall aber nicht den `WinNT`-Provider verwenden, sondern den `LDAP`-Provider.

Das nachfolgende Code-Beispiel zeigt, wie Sie die Mitglieder der Domänengruppe *TestBenutzer*, die in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* abgelegt ist, auslesen und mit der `WScript.Echo`-Anweisung am Bildschirm ausgeben können.

```
Option Explicit
```

```
Dim oGroup
Dim strComputer
Dim strGroupName
Dim strMember
```

Listing 125: 2_01_023b.vbs

>> Benutzer und Gruppen

```

strComputer = "."
strGroupName = "CN=TestBenutzer,OU=Vertrieb,_"
              DC=MyDomain,DC=Com"

Set oGroup = GetObject("LDAP://" & strGroupName)
For Each strMember in oGroup.Members
    WScript.Echo strMember.SamAccountName
Next

Set oGroup = Nothing

```

Listing 125: 2_01_023b.vbs (Forts.)

24 Benutzer aus Gruppen entfernen

Einzelne Benutzer lassen sich jederzeit mit nur wenigen Klicks wieder aus den Gruppen entfernen. Um beispielsweise auf einem Standalone- oder Member-Server die lokalen Benutzerkonten *TestUser1* und *TestUser2* aus der Gruppe *TestBenutzer* zu entfernen, rufen Sie zunächst die lokale Benutzerverwaltung auf. Anschließend markieren Sie im Ordner GRUPPEN die Gruppe *TestBenutzer* und rufen aus dem Kontextmenü heraus die EIGENSCHAFTEN auf. Markieren Sie die beiden Benutzer *TestUser1* und *TestUser2*, und klicken Sie auf ENTFERNEN.

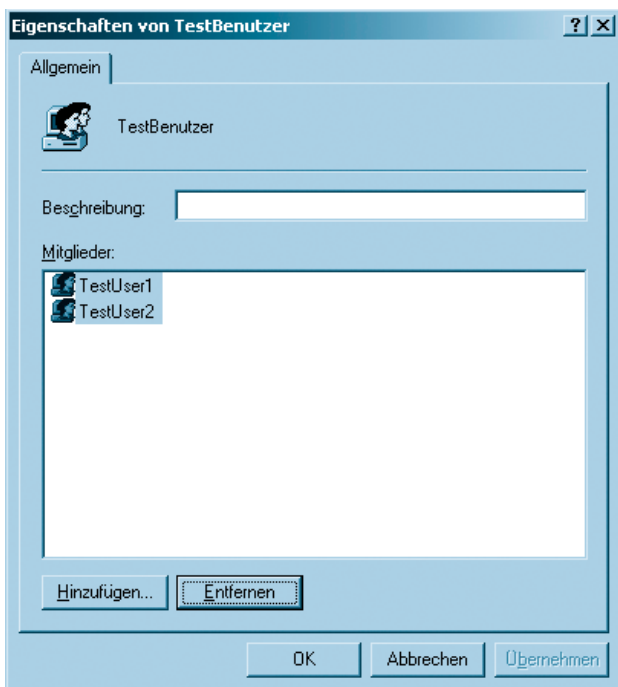


Abbildung 74: Mitglieder aus einer Gruppe entfernen

218 >> Benutzer aus Gruppen entfernen

Wenn Sie dagegen im Active Directory einzelne Mitglieder aus einer Domänengruppe entfernen möchten, so rufen Sie stattdessen das Verwaltungswerkzeug ACTIVE DIRECTORY-BENUTZER UND -COMPUTER auf, öffnen die EIGENSCHAFTEN der betreffenden Gruppe und wechseln auf die Registerkarte MITGLIEDER. Dort markieren Sie die Benutzerkonten (oder Gruppen), deren Mitgliedschaft Sie aufheben wollen, und klicken auf ENTFERNEN.

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus einzelne Mitglieder aus lokalen, globalen oder auch universalen Gruppen entfernen möchten, so können Sie hierfür die Befehle `Net LocalGroup`, `Net Group` und auch `Dsmod Group` verwenden, die zum festen Befehlsumfang von Windows Server 2003 gehören.

Um ein Mitglied aus einer lokalen Gruppe zu entfernen, wenden Sie den `Net LocalGroup`-Befehl mit der Option `/Delete` an und übergeben sowohl den Namen der Gruppe wie auch den Namen des Benutzers, der aus der Gruppe herausgelöscht werden soll.

Die nachfolgende `Net LocalGroup`-Befehlszeile löscht beispielsweise den Benutzer *TestUser1* aus der lokalen Gruppe *Testbenutzer*:

```
@Echo off
Net LocalGroup TestBenutzer TestUser1 /Delete
```

Listing 126: 2_01_024a.bat

Wenn Sie dagegen den Benutzer *TestUser1* aus einer Domänengruppe herauslöschen möchten, so wenden Sie stattdessen den `Net Group`-Befehl mit der Option `/Delete` an, dem Sie wieder den Namen der Gruppe wie auch den Namen des Benutzers als Parameter übergeben. Und für den Fall, dass Sie den Befehl nicht direkt auf dem Domänen-Controller ausführen, sondern auf einem Member-Server oder von einem Client aus, so richten Sie den `Net Group`-Befehl mit dem Parameter `/Domain` explizit gegen die Domäne.

```
@Echo off
Net Group TestBenutzer TestUser1 /Domain /Delete
```

Listing 127: 2_01_024b.bat

Innerhalb des Active Directory können Sie auch den `Dsmod Group`-Befehl verwenden, um ein Mitglied aus einer Domänengruppe zu entfernen. Wenn Sie z.B. den Benutzer *TestUser1*, der in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* abgelegt ist, aus der Gruppe *TestBenutzer* (ebenfalls in der OU *Vertrieb* von *MyDomain.com*) entfernen möchten, so wenden Sie den `Dsmod Group`-Befehl an und übergeben zunächst den definierten Namen der Gruppe und mit dem Parameter `-rmmbr` den definierten Namen des Benutzers, der aus der Gruppe entfernt werden soll.

>> Benutzer und Gruppen

```
@Echo off
Dismod Group CN=TestBenutzer,OU=Vertrieb,DC=MyDomain,DC=Com _
-rmmbr CN=TestUser1,OU=Vertrieb,DC=MyDomain,DC=Com
```

Listing 128: 2_01_024c.bat

Die Lösung mit VBScript

Wenn Sie per VBScript einen Benutzer aus einer Gruppe entfernen möchten, so wenden Sie im einfachsten Fall die `Remove-Methode` (WinNT-Provider) bzw. die `PutEx-Methode` mit dem Parameter `ADS_PROPERTY_DELETE` (LDAP-Provider) gegen das entsprechende `Group-Objekt` an und übergeben jeweils den Namen des Benutzers, den Sie aus der Gruppe löschen möchten.

Um beispielsweise auf einem Standalone-Server den Benutzer *TestUser1* aus der lokalen Gruppe *TestBenutzer* zu löschen, stellen Sie mit `GetObject` zunächst eine Verbindung zum WinNT-Provider des betreffenden Computersystems her und binden das `Group-Objekt` an eine Objektvariable. Anschließend wenden Sie die `Remove-Methode` gegen die Objektvariable an und übergeben den ADS-Pfad des `User-Objektes` als Argument, woraufhin der Benutzer unmittelbar aus der angegebenen Gruppe entfernt wird.

```
Option Explicit

Dim oGroup
Dim strComputer
Dim strGroupName
Dim strUserName

strComputer = "FRA-MD-FPS-001"
strGroupName = "TestBenutzer"
strUserName = "TestUser1"

Set oGroup = GetObject("WinNT://" & strComputer & "/" & _
    strGroupName & ",group")
oGroup.Remove "WinNT://" & strComputer & "/" & strUserName

Set oGroup = Nothing
```

Listing 129: 2_01_024a.vbs

Auf ähnlichem Wege löschen Sie auch ein Domänenbenutzerkonto aus einer lokalen, globalen oder universalen Gruppe im Active Directory; nur dass Sie diesmal das `Group-Objekt` per `LDAP` an eine Objektvariable binden und statt der `Remove-Methode` die `PutEx-Methode` anwenden, der Sie folgende Parameter übergeben:

220 >> Benutzer aus Gruppen entfernen

- ▶ Option: Mit dem ersten Parameter legen Sie den gewünschten Bearbeitungsmodus fest (siehe nachfolgende Tabelle).
- ▶ Eigenschaft: Der zweite Parameter enthält den Namen der Eigenschaft, die mit der PutEx-Methode geändert werden soll.
- ▶ Wert: Mit dem dritten Parameter übergeben Sie das Array, das den neuen Wert als Element enthält.

Konstante	Wert	Beschreibung
ADS_PROPERTY_CLEAR	1	Entfernt alle vorhandenen Werte
ADS_PROPERTY_UPDATE	2	Ersetzt die Werte, indem die vorhandenen Werte gelöscht und die angegebenen Werte hinzugefügt werden
ADS_PROPERTY_APPEND	3	Fügt die angegebenen Werte hinzu
ADS_PROPERTY_DELETE	4	Löscht die angegebenen Werte

Tabelle 24: Die Optionen der PutEx-Methode

Das nachfolgende Listing entfernt den Benutzer *TestUser1*, der im Container *Users* der Domäne *MyDomain.com* abgelegt ist, aus der Gruppe *TestBenutzer*. Hierzu wird mit `GetObject` eine Verbindung zum LDAP-Provider hergestellt und das `Group`-Objekt an eine Objektvariable (`oGroup`) gebunden. Anschließend wird die `PutEx`-Methode gegen die Objektvariable `oGroup` angewendet, und die Option `ADS_PROPERTY_DELETE` (Wert = 1), die Eigenschaft (`Member`) und auch das Array werden mit dem Wert (`strUserName`) als Argument übergeben. Um die Änderung zu übernehmen, werden die neuen Einstellungen mit der `SetInfo`-Methode zurückgeschrieben (dies ist bei Einsatz der `Remove`-Methode nicht erforderlich).

Option Explicit

```
Dim oGroup
Dim strGroupName
Dim strUserName
```

```
strGroupName = "CN=TestBenutzer,CN=Users,DC=MyDomain,DC=Com"
strUserName = "CN=TestUser1,CN=Users,DC=MyDomain,DC=Com"
Const ADS_PROPERTY_DELETE = 4
```

```
Set oGroup = GetObject("LDAP://" & strGroupName1)
oGroup.PutEx ADS_PROPERTY_DELETE, "Member", Array(strUserName)
oGroup.SetInfo
```

```
Set oGroup = Nothing
```

Listing 130: 2_01_024b.vbs

25 Gruppen löschen

Einzelne Gruppen lassen sich natürlich jederzeit wieder löschen, wobei aber auch hier wieder gilt, dass gelöschte Gruppen nicht mehr wiederhergestellt werden können. Wenn Sie z.B. aus Versehen eine Gruppe löschen und sie zu einem späteren Zeitpunkt mit dem gleichen Namen wieder neu erstellen, so müssen Sie sämtliche Berechtigungen neu vergeben, da die neue Gruppe eine andere SID erhält.

Um nun auf einem Standalone- oder Member-Server eine lokale Gruppe löschen zu können, müssen Sie entweder als *Administrator* angemeldet sein, um alle Gruppen löschen zu können, oder aber als *Hauptbenutzer*, wenn Sie eine selbst erstellte Gruppe vom Computersystem verbannen möchten. Und wenn Sie über diese Berechtigung verfügen, so brauchen Sie nur die lokale Benutzerverwaltung (LOKALE BENUTZER UND GRUPPEN) zu öffnen, die entsprechende Gruppe zu markieren und aus dem Kontextmenü heraus den Eintrag LÖSCHEN zu wählen.

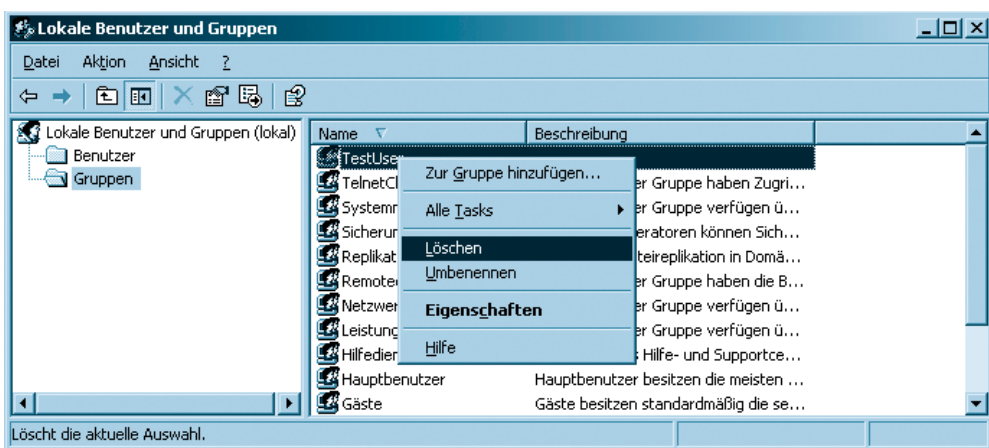


Abbildung 75: Eine lokale Gruppe löschen

Möchten Sie dagegen eine lokale oder globale Gruppe auf einem Domänen-Controller löschen, so müssen Sie entweder als *Administrator* angemeldet oder zumindest Mitglied der Gruppen *Konten-Operatoren* oder *Organisations-Admins* sein, um diesen Vorgang ausführen zu können.

Wenn Sie über die ausreichende Berechtigung zum Löschen einer Gruppe verfügen, dann öffnen Sie aus dem Verwaltungsmenü heraus ACTIVE DIRECTORY-BENUTZER UND -COMPUTER, markieren die betreffende Gruppe und wählen im Kontextmenü den Eintrag LÖSCHEN aus oder drücken die `[Entf]`-Taste.

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus bestehende Gruppen löschen müssen, dann verwenden Sie hierfür den `Net LocalGroup-/Net Group-Befehl` oder setzen den `Dsrmdir`-Befehl ein:

222 >> Gruppen löschen

- ▶ **Net LocalGroup:** Der `Net LocalGroup`-Befehl erlaubt das Löschen einer lokalen Gruppe und kann sowohl auf einem Standalone- und Member-Server wie auch auf einem Domänen-Controller eingesetzt werden. Ohne Angabe weiterer Parameter wird die angegebene Gruppe des lokalen Computersystems gelöscht, auf dem der Befehl ausgeführt wird.
- ▶ **Net Group:** Der `Net Group`-Befehl ermöglicht das Löschen einer lokalen oder globalen Domänengruppe und löscht ohne Angabe weiterer Parameter die Gruppe auf dem Computersystem, auf dem der Befehl aufgerufen wird. Um von einem Client oder Member-Server aus eine Domänengruppe zu löschen, fügen Sie den Parameter `/Domain` an, woraufhin der Befehl gegen die Domäne und nicht gegen das lokale Computersystem ausgeführt wird.
- ▶ **Dsrm:** Der `Dsrm`-Befehl erlaubt das Löschen einer lokalen, globalen und auch einer universalen Gruppe, wobei dieser Befehl aber nur in einer Domänenumgebung eingesetzt werden kann. Im Gegensatz zum `Net LocalGroup`- und `NetGroup`-Befehl müssen Sie beim Einsatz des `Dsrm`-Befehls den definierten Namen der zu löschenden Gruppe angeben.

Um beispielsweise auf einem bestimmten Computersystem die lokale Gruppe mit der Bezeichnung *TestBenutzer* zu löschen, setzen Sie im einfachsten Fall den Befehl `Net LocalGroup` ein, dem Sie den Namen der zu löschenden Gruppe (*TestBenutzer*) wie auch die gewünschte Aktion (`/Delete`) als Parameter übergeben.

```
@Echo off
Net LocalGroup TestBenutzer /Delete
```

Listing 131: 2_01_025a.bat

Eine Domänengruppe können Sie dagegen mit dem `Net Group`-Befehl löschen, dem Sie ebenfalls den Namen der Gruppe wie auch die Aktion als Parameter übergeben. Da dieser Befehl standardmäßig gegen das lokale Computersystem ausgeführt wird, müssen Sie – sofern Sie den Befehl nicht direkt auf dem Domänen-Controller aufrufen – den Parameter `/Domain` anfügen.

```
@Echo off
Net Group TestBenutzer /Delete /Domain
```

Listing 132: 2_01_025b.bat

Eine weitere Möglichkeit zum Löschen von Gruppen bietet der `Dsrm`-Befehl, der sich allerdings nur in einer Domänenumgebung einsetzen lässt. Wenn Sie z.B. die Gruppe *TestBenutzer*, die sich in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* befindet, ohne Rückfrage löschen möchten, so führen Sie den `Dsrm`-Befehl aus und übergeben den definierten Namen der zu löschenden Gruppe wie auch den Parameter

-noprompt. Letzteres dient dem Erzwingen des Löschens, da Sie das Löschen des Objektes anderenfalls von Hand bestätigen müssten.

```
@Echo off
Dsrm CN=TestBenutzer,OU=Vertrieb,DC=MyDomain,DC=Com -noprompt
```

Listing 133: 2_01_025c.bat

Die Lösung mit VBScript

Um per VBScript eine vorhandene Gruppe zu löschen, sind nur wenige Code-Zeilen erforderlich. Denn Sie müssen nur eine Referenz zu dem betreffenden Computersystem oder zu der Domäne herstellen und die `Delete`-Methode gegen die zu löschende Gruppe anwenden.

Wenn Sie z.B. auf einem Standalone- oder Member-Server die lokale Gruppe mit der Bezeichnung *TestBenutzer* löschen möchten, so stellen Sie mit der `GetObject`-Methode eine Verbindung zum ADSI-Provider `WinNT` des betreffenden Computersystems her, wenden die `Delete`-Methode gegen den Objekttyp `Group` an und übergeben dabei den Namen der zu löschenden Gruppe.

```
Option Explicit

Dim oComputer
Dim strComputer
Dim strGroupName

strComputer = "."
strGroupName = "TestBenutzer"

Set oComputer = GetObject("WinNT://" & strComputer)
oComputer.Delete "Group", strGroupName

Set oComputer = Nothing
```

Listing 134: 2_01_025a.vbs

Um eine Domänengruppe im Active Directory zu löschen, verwenden Sie stattdessen den ADSI-Provider `LDAP`. Wenn Sie z.B. die globale Gruppe *TestBenutzer*, die sich in der Organisationseinheit *Vertrieb* der Domäne *MyDomain.com* befindet, löschen möchten, so binden Sie den ADSI-Pfad der betreffenden Organisationseinheit an eine Objektvariable und führen dann die `Delete`-Methode gegen die entsprechende Objektvariable aus, wobei Sie auch hier wieder den Objekttyp (`Group`) und auch den Namen der zu löschenden Gruppe (`strGroupName`) übergeben.

224 >> Welche SID hat die Gruppe?

```
Option Explicit
```

```
Dim oOU
```

```
Dim strOU
```

```
Dim strGroupName
```

```
strOU = "OU=Vertrieb,DC=MyDomain,DC=Com"
```

```
strGroupName = "CN=TestBenutzer"
```

```
Set oOU = GetObject("LDAP://" & strOU)
```

```
oOU.Delete "Group", strGroupName
```

```
Set oOU = Nothing
```

Listing 135: 2_01_025b.vbs

26 Welche SID hat die Gruppe?

Der Windows Server 2003 stellt leider keine Möglichkeit bereit, womit Sie unter grafischer Oberfläche die SID (siehe Welche SID hat der Benutzer? auf Seite 180) einer Gruppe ermitteln können. Sie können jedoch den WMIC- und Dsget-Befehl der Eingabeaufforderung oder auch ein VBScript dazu nutzen, um die SID einer bestimmten Gruppe herauszufinden. Die nachfolgenden Abschnitte zeigen die entsprechenden Lösungsmöglichkeiten.

Die Lösung per Eingabeaufforderung

Um von der Eingabeaufforderung aus die SID einer bestimmten Gruppe zu ermitteln, können Sie den WMIC-Befehl unter Angabe des Alias `Group` verwenden, da Sie mit diesem Befehl eine WMI-Abfrage gegen die Klasse `Win32_Group` durchführen und die Eigenschaften der Gruppenobjekte auslesen können. Bestandteil der Eigenschaften ist dabei auch die SID, die Sie mit `Get` explizit abfragen können.

Wenn Sie z.B. die SID der lokalen Gruppe *DNSAdmins* herausfinden möchten, so führen Sie den `WMIC Group`-Befehl aus, selektieren das Gruppenobjekt mit Hilfe der `Where`-Klausel, der Sie den Namen übergeben, und legen mit `Get SID` fest, dass Sie nur den Wert der Eigenschaft `SID` auslesen möchten.

```
@Echo off
```

```
WMIC Group Where "Name='DNSAdmins'" Get SID
```

Listing 136: 2_01_026a.bat

Mit der gleichen Befehlszeile können Sie natürlich auch die SIDs der globalen und universalen Gruppen ermitteln. Die SIDs der Domänengruppen lassen sich alternativ dazu aber auch mit dem `Dsget Group`-Befehl herausfinden, wenn Sie den definierten Namen der Gruppe und auch den Parameter `-sid` übergeben:

```
@Echo off
Dsget Group CN=TestBenutzer,OU=Vertrieb,DC=MyDomain,DC=Com -sid
```

Listing 137: 2_01_026b.bat

Die Lösung mit VBScript

Um per VBScript die SID einer bestimmten Gruppe zu ermitteln, gehen Sie ähnlich vor wie beim Auslesen der SID für ein Benutzerkonto (siehe Welche SID hat der Benutzer? auf Seite 182); nur mit dem Unterschied, dass Sie nicht die WMI-Klasse `Win32_UserAccount` verwenden, sondern die Klasse `Win32_Group`. Denn diese Klasse enthält Informationen zu den Gruppenkonten und verfügt unter anderem über die Eigenschaft `SID`, deren Wert Sie jederzeit abfragen können.

Wenn Sie z.B. die SID der Gruppe `TestBenutzer` herausfinden müssen, so führen Sie eine Abfrage gegen die WMI-Klasse `Win32_Group` durch und selektieren alle Gruppenkonten, deren Name gleich dem gesuchten Gruppennamen (`strGroupName`) ist. Das Ergebnis wird daraufhin in Form einer Kollektion zurückgeliefert. Mit Hilfe einer `For Each`-Schleife können Sie dann auf das Element zugreifen und den Wert der Eigenschaft `SID` abfragen.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim strGroupName
Dim colGroups
Dim strGroup

strComputer = "."
strGroupName = "TestBenutzer"

Set oWMI = GetObject("winmgmts:{impersonationlevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colGroups = oWMI.ExecQuery("SELECT * FROM _
    Win32_Group WHERE Name='" & strGroupName & "'")

For Each strGroup in colGroups
    WScript.Echo strGroup.Name & ": " & strGroup.SID
Next

Set oWMI = Nothing
```

Listing 138: 2_01_026.vbs

Dateien und Ordner

Die Arbeit mit Dateien und Ordnern gehört zum Alltag eines jeden, der mit Computersystemen zu tun hat. Und gerade die Verwaltung der Dateien und Ordnern nimmt oftmals einen Großteil der administrativen Zeit in Anspruch, denn Dateien und Ordner müssen zugriffsgeschützt, regelmäßig gesichert, hin und wieder kopiert, verschoben oder gelöscht, manchmal auch verschlüsselt oder komprimiert, ab und zu gesplittet und wieder zusammengefügt und nicht selten auch inhaltlich aufbereitet werden.

Und gerade die Aufbereitung von Dateien ist es, die so manch einem den letzten Nerv raubt. Denn wer schon einmal in die Verlegenheit kam, Protokoll- oder Datendateien nachbearbeiten zu müssen, und beispielsweise den Inhalt sortieren, Leerzeilen entfernen, Feldbegrenzungszeichen austauschen, doppelte Einträge löschen oder einfach nur eine Zeilennummerierung einfügen wollte, der wird mit Sicherheit bestätigen können, dass Windows Server 2003 hierfür nur wenige Werkzeuge bereitstellt und dass derartige Aufgaben immens umständlich und vor allen Dingen auch sehr zeitaufwändig sein können.

Die nachfolgenden Rezepte zeigen daher, wie Sie die wichtigsten und gängigsten Aufgaben aus dem Datei- und Ordner-Management unter grafischer Oberfläche lösen und wie Sie diese Aufgaben durch Einsatz der Eingabeaufforderung und durch entsprechende Skripte auch automatisieren können.

27 Dateien und Ordner suchen

Wenn Sie bestimmte Dateien oder auch Ordner suchen, weil Sie deren genauen Pfad nicht kennen oder eine Selektion aller Objekte benötigen, die einem bestimmten Kriterium entsprechen, so können Sie unter grafischer Oberfläche die Windows-eigene Such-Funktion nutzen, um das Dateisystem nach der Datei (oder wahlweise auch nach dem Ordner) zu durchsuchen.

Die Such-Funktion von Windows Server 2003 erlaubt das Festlegen unterschiedlicher Such-Optionen, womit Sie die Datenträger nach folgenden Kriterien durchsuchen können:

- ▶ **Name:** Geben Sie den vollständigen Namen (oder zumindest Teile davon) an, wenn Sie nach einer Datei oder einem Ordner mit einem bestimmten Namen suchen.
- ▶ **Inhalt:** Legen Sie den gesuchten Inhalt fest, wenn Sie nach einer Datei oder einem Ordner mit einem bestimmten Inhalt suchen.
- ▶ **Bereich:** Geben Sie den Suchbereich (z.B. Netzwerk, lokales Computersystem, Laufwerk, Ordner ...) an, wenn Sie die Suche auf bestimmte Bereiche einschränken möchten.

- ▶ **Änderungsdatum:** Geben Sie ein Datum ein, um nach allen Dateien oder Ordnern zu suchen, die seit dem angegebenen Datum geändert wurden.
- ▶ **Größe:** Legen Sie eine Größe fest, um nach allen Dateien und/oder Ordnern zu suchen, deren Größe auf dem Dateisystem entweder kleiner, gleich oder größer als der angegebene Wert ist.
- ▶ **Dateityp:** Legen Sie den zu suchenden Dateityp (z.B. Bitmap-Datei, Adressbuch-Datei, Skriptdateien etc.) fest, um nach allen Dateien des angegebenen Typs zu suchen.

Wenn Sie beispielsweise alle Dateien und Ordner der lokalen Festplatte auffinden wollen, deren Name *Setup* enthält, so rufen Sie mit der Taste **[F3]** (oder im START-MENÜ mit einem Klick auf **SUCHEN**) die Such-Funktion von Windows Server 2003 auf und geben im Feld **GESAMTER ODER TEIL DES DATEINAMENS** den Suchbegriff *Setup* ein. In der Auswahlliste **SUCHEN IN** wählen Sie den entsprechenden Datenträger aus (z.B. **LOKALE FESTPLATTEN (C:)**). Klicken Sie anschließend auf **SUCHEN**, um mit der Suche zu beginnen. Sobald die Suche abgeschlossen ist, werden in der rechten Fensterhälfte alle Dateien und Ordner angezeigt, deren Name *Setup* enthält.

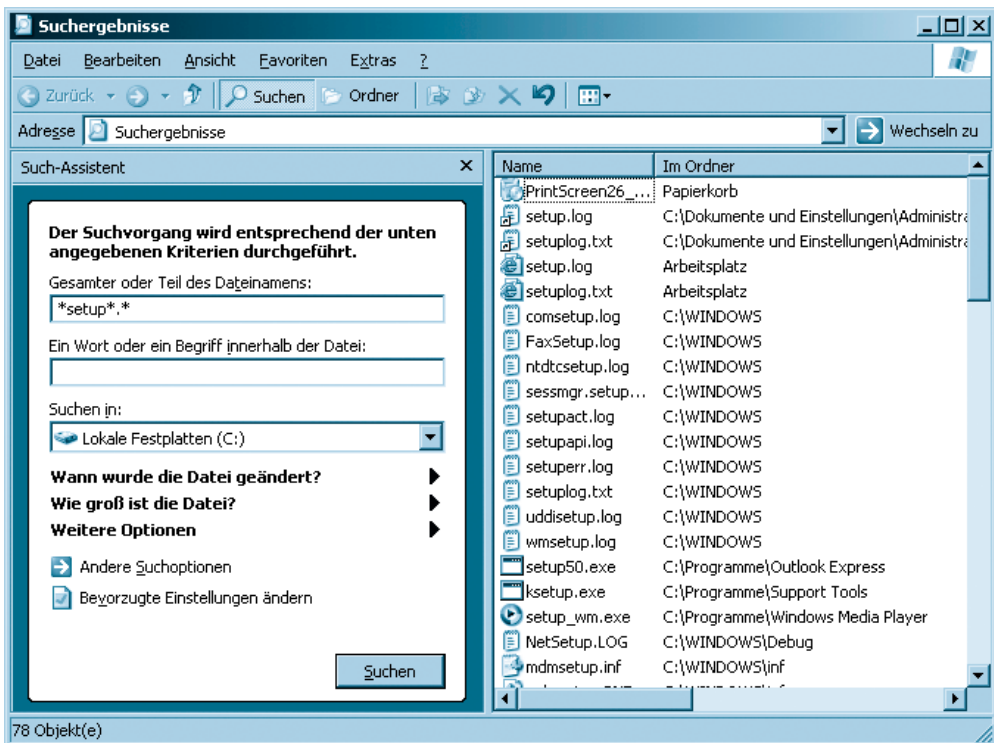


Abbildung 76: Eine Datei suchen

Um eine Aufstellung aller Dateien und Ordner zu erhalten, die ab dem 27.07.2005 geändert wurden, rufen Sie mit `F3` die Such-Funktion auf und klicken auf WANN WURDE DIE DATEI GEÄNDERT?. Sie sehen nun weitere Suchoptionen. Aktivieren Sie die Option DATUMSANGABE und geben Sie das Datum 27.07.2005 ein. Mit einem Klick auf SUCHEN wird nach allen Dateien und Ordnern gesucht, die seit dem 27.07.2005 geändert wurden.

Die Lösung per Eingabeaufforderung

Ähnliche Suchmöglichkeiten haben Sie natürlich auch in der Eingabeaufforderung. Wenn Sie z.B. den gesamten Datenträger nach Dateien durchsuchen möchten, deren Dateiname die Zeichenfolge *Setup* enthält, so wenden Sie im Root des Datenträgers (z.B. C:\) den `Dir`-Befehl an, übergeben den Suchbegriff (Format `Dateiname.Suffix`) und legen mit dem Parameter `/s` fest, dass die Suche rekursiv, sprich: auch in den untergeordneten Ordnern, erfolgen soll. Sie können hierbei natürlich auch so genannte Wildcards (`?` oder `*`) verwenden, um die Position der gesuchten Zeichenfolge zu bestimmen oder um nur nach bestimmten Dateitypen (z.B. **.txt*, **.log* ...) zu suchen.

```
@Echo off
CD \
Dir *Setup*. * /s
```

Listing 139: 2_02_001a.bat

Wenn Sie nicht nach Dateien, sondern nach Ordnern suchen möchten, so verwenden Sie die gleiche Befehlszeile und fügen den Parameter `/a:d` hinzu, womit Sie nach allen Objekten mit dem Attribut *Directory* suchen können.

```
@Echo off
CD \
Dir *Setup* /s/a:d
```

Listing 140: 2_02_001b.bat

Der `Dir`-Befehl erlaubt allerdings nur wenige Selektionsmöglichkeiten. Denn Sie können im Grunde genommen nur nach Datei- und Ordnernamen suchen, nach Dateitypen, wie auch nach bestimmten Datei-/Ordner-Attributen (*Archiv*, *Schreibschutz*, *Versteckt*, *System*).

Wenn Sie nun – ähnlich wie unter grafischer Oberfläche – nach weiteren Eigenschaften (z.B. Erstell- oder Änderungsdatum, Besitzer ...) selektieren möchten, so können Sie hierfür eine WMI-Abfrage durchführen und Dateien wie auch Ordner nach bestimmten Eigenschaften filtern. Der WMI-Alias `DataFile` erlaubt dabei den Zugriff auf die WMI-Klasse `CIM_DataFile` und damit auf die Eigenschaften einer Datei und der Alias `FSDir` dagegen auf die WMI-Klasse `Win32_Directory` und die Eigen-

schaften eines Ordners. Beide Aliase stellen die gleichen Eigenschaften bereit (siehe Teil III dieses Buches), die Sie jederzeit abfragen und per `Get`-Anweisung auch auslesen können. Die nachfolgende Tabelle enthält die wichtigsten Eigenschaften der beiden Klassen:

Eigenschaft	Beschreibung
Archive	Enthält den Wert <i>True</i> , wenn das Archiv-Attribut gesetzt ist, anderenfalls ist der Wert gleich <i>False</i> .
Caption	Kurze Beschreibung zur Datei oder zum Ordner
Compressed	Komprimierungsattribut. Diese Eigenschaft enthält den Wert <i>True</i> , wenn die Datei oder der Ordner komprimiert ist, anderenfalls ist der Wert <i>False</i> .
CreationDate	Datum und Uhrzeit des Erstellungszeitpunktes
Description	Beschreibung zur Datei oder zum Ordner
Drive	Laufwerkbuchstabe des Pfades
EightDotThreeFileName	Datei- bzw. Ordnername nach der 8.3-Konvention
Encrypted	Verschlüsselungsattribut. Die Eigenschaft besitzt den Wert <i>True</i> , wenn die Datei oder der Ordner verschlüsselt ist, anderenfalls ist der Wert gleich <i>False</i> .
Extension	Suffix der Datei
FileName	Name der Datei
FileSize	Die Größe einer Datei
FileType	Typ der Datei
Hidden	Hidden-Attribut. Diese Eigenschaft besitzt den Wert <i>True</i> , wenn die Datei oder der Ordner versteckt ist, ansonsten ist der Wert gleich <i>False</i> .
LastAccessed	Datum und Uhrzeit des letzten Zugriffs
LastModified	Datum und Uhrzeit des letzten Änderungszeitpunktes
Path	Der Pfad der Datei bzw. des Ordners

Tabelle 25: Die Eigenschaften des *DataFile*- und *FSDir*-Alias

Wenn Sie z.B. eine Aufstellung aller Textdateien des Laufwerks C erstellen möchten, die zuletzt am 25.07.2005 oder später geändert wurden, so führen Sie eine WMIC-Abfrage unter Verwendung des Alias *DataFile* aus und übergeben den Wert der Eigenschaften *Drive*, *Extension* und *LastModified* (erfordert die Angabe des Zeitpunkts im UTC-Format, siehe nachfolgenden Kasten) als Parameter. Und da ohne Angabe weiterer Parameter alle Eigenschaften der gefundenen Dateien ausgegeben werden, beschränken Sie die Ausgabe des Ergebnisses mit Hilfe der `Get`-Anweisung auf die Ausgabe des Dateinamens (*Name*) und des letzten Änderungszeitpunktes (*LastModified*).

```
@Echo off
WMIC DataFile WHERE "Drive='C:' AND Extension='txt' _
AND LastModified>'20050725000000.000000+120'" _
GET Name, LastModified
```

Listing 141: 2_02_001c.bat

Möchten Sie stattdessen eine Aufstellung aller Ordner des Laufwerks C erzeugen, die seit dem 25.07.2005 zuletzt geändert wurden, so verwenden Sie stattdessen den WMIC-Alias `FSDir` und fragen den Pfad wie auch den gewünschten Änderungszeitpunkt ab. Und mit der `Get`-Anweisung beschränken Sie die Ausgabe des Ergebnisses wieder auf die Eigenschaften `Name` und `LastModified`, da anderenfalls alle Eigenschaften ausgegeben werden.

```
@Echo off
WMIC FSDir WHERE "Drive='C:' AND _
LastModified>'20050725000000.000000+120'" _
GET Name, LastModified
```

Listing 142: 2_02_001d.bat

Exkurs

UTC – die koordinierte Weltzeit

Die UTC (**U**niversal **T**ime **C**oordinated) ist als Nachfolgerin der GMT (**G**reenwich **M**ean **T**ime) zu verstehen, der eine vom Nullmeridian ausgehende Zeitzone zugeordnet wird. Die mitteleuropäische Zeit (MEZ) 12:00 Uhr entspricht damit der UTC 11:00 +1:00, wobei dies jedoch nur während der Winterzeit gilt. Denn während der Sommerzeit entspricht die MEZ 12:00 Uhr der UTC 11:00 +2:00, da die Uhren während der Sommerzeit eine Stunde vorgestellt werden.

Die UTC kann dabei in unterschiedlichen Formaten dargestellt werden. Wenn Sie z.B. eine WMI-Abfrage durchführen und eine UTC als Ergebnis dargestellt bekommen, so wird die UTC im Format `YYYYMMDDHHMMSS.000000+120` (Sommerzeit) oder in `YYYYMMDDHHMMSS.000000+060` (Winterzeit) dargestellt, wobei nach den letzten sechs Stellen der Millisekunde die positive (+) oder auch negative (–) Zeitdifferenz in Minuten ausgegeben wird.

Wenn Sie selbst eine UTC angeben müssen und nicht sicher sind, welche Zeitzone Sie angeben müssen, so können Sie das Systemsteuerungsobjekt **DATUM UND UHRZEIT** öffnen und auf der Registerkarte **ZEITZONE** nachsehen, welche Zeitzone in den einzelnen Ländern und Städten gültig ist.

Die Lösung mit VBScript

Wenn Sie per VBScript nach bestimmten Dateien und Ordnern suchen möchten, so gehen Sie ähnlich vor wie in der Eingabeaufforderung. Sprich: Sie nutzen eine WMI-Abfrage und selektieren alle Dateien und Ordner, die dem angegebenen Suchkriterium entsprechen.

Um beispielsweise nach allen Textdateien (*.txt) zu suchen, die seit dem 25.07.2005 zuletzt geändert wurden, führen Sie eine Abfrage gegen die CIM_DataFile-Klasse durch und verwenden die Eigenschaften Extension und LastModified als Suchkriterium. Hierfür wenden Sie die ExecQuery-Methode unter Angabe des Suchkriteriums an, woraufhin die Dateien, die dem Suchkriterium entsprechen, in Form einer Kollektion zurückgeliefert werden.

Mit Hilfe einer For Each-Schleife können Sie dann auf jedes Element der Kollektion zugreifen und die gewünschte Eigenschaft (z.B. Name und Änderungsdatum) am Bildschirm ausgeben oder auch an ein sonstiges Ausgabegerät übergeben.

Option Explicit

```
Dim oWMI
Dim colFile
Dim objItem
Dim strComputer
```

```
strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colFile = oWMI.ExecQuery ("SELECT * FROM CIM_DataFile _
    WHERE Extension='txt' AND _
    LastModified>'20050725000000.000000+120'")
```

```
For Each objItem in colFile
    Wscript.Echo objItem.Name & ": " & objItem.LastModified
Next
```

```
Set oWMI = Nothing
```

Listing 143: 2_02_001a.vbs

Wenn Sie das Dateisystem stattdessen nach bestimmten Ordnern durchsuchen möchten, so richten Sie die Abfrage nicht gegen die CIM_DataFile-Klasse des betreffenden Computersystems, sondern gegen die Win32_Directory-Klasse:

```
Option Explicit

Dim oWMI
Dim colFolder
Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colFolder = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Directory WHERE _
    LastModified>'20050725000000.000000+120'")

For Each objItem in colFolder
    Wscript.Echo objItem.Name & ": " & objItem.LastModified
Next

Set oWMI = Nothing
```

Listing 144: 2_02_001b.vbs

28 Welche Berechtigung ist auf die Datei oder auf den Ordner vergeben?

Mit dem Einsatz des Dateisystems *NTFS* werden die Dateien und Ordner, die auf dem NTFS-Laufwerk abgelegt werden, automatisch mit einer Sicherheitsbeschreibung (engl. *Security Descriptor*) versehen. Diese Sicherheitsbeschreibung setzt sich aus den folgenden vier Komponenten zusammen:

- ▶ **DACL:** Die Komponente *DACL (Discretionary Access Control List)* stellt die eigentliche Zugriffskontrollliste (engl. *Access Control List*) dar und enthält die erforderlichen Einträge (*ACE = Access Control List*) der Benutzer und Gruppen, für die der Zugriff auf die Datei oder auf den Ordner zugelassen oder auch verweigert wird. Über *DACL* werden also die einzelnen Zugriffe für das Datei- oder Ordner-Objekt gesteuert.
- ▶ **SACL:** Die Komponente *SACL (System Access Control List)* dagegen ermöglicht das Aufzeichnen bzw. Protokollieren von Objektzugriffen und dient der Überwachung (engl. *Auditing*) der Objektzugriffe. Die Einträge der *SACL* bestimmen demnach, ob die Objektzugriffe generell überwacht werden sollen und – wenn ja – welche erfolgreichen bzw. fehlgeschlagenen Objektzugriffe im Einzelnen im Sicherheitsprotokoll aufgezeichnet werden.

- ▶ **Owner:** Die Komponente *Owner* beinhaltet die SID (*Security Identifier*) des Datei- oder Ordnerbesitzers. Der Besitzer ist in der Regel der Benutzer (dies kann auch eine Gruppe sein), der das Objekt angelegt hat. Das Besitz-recht kann auch von einem ursprünglichen Besitzer übertragen oder durch die Vererbung der Berechtigungen weitervererbt worden sein.
- ▶ **Primary Group:** Die Komponente *Primary Group* enthält entweder den Namen der primären Gruppe des Benutzers, der das Datei- oder Ordner-Objekt erstellt hat, oder – falls keine primäre Gruppe definiert wurde – den Namen der Standard-Primary-Group (Domänenbenutzer). Diese Komponente ist jedoch nur innerhalb des Active Directory relevant und auch nur dann, wenn POSIX-kompatible Anwendungen genutzt und/oder von Macintosh-Systemen aus eine Anmeldung im Netzwerk erfolgt.

Wichtig für die Zugriffssteuerung von Dateien und Ordnern ist die Komponente *DACL* (nachfolgend nur noch *ACL* genannt), da die Einträge dieser Zugriffskontrollliste darüber entscheiden, ob ein Benutzer oder eine Gruppe auf ein Datei- oder Ordner-Objekt zugreifen darf oder nicht. Jede Zugriffskontrollliste verfügt dabei über so genannte generische Berechtigungen, die sich aus mehreren Einzelberechtigungen zusammensetzen. Diese generischen Berechtigungen, die für alle Dateien bzw. für alle Ordner gleich sind, lassen sich einzelnen Benutzern und auch den Mitgliedern einer Gruppe per simplem Klick Lese-, Änderungs-, Schreib- und auch Vollzugriffsrechte einräumen oder wahlweise auch verweigern (hierzu jedoch im nächsten Rezept mehr).

Die nachfolgende Tabelle zeigt, welche generischen Berechtigungen in der Zugriffskontrollliste einer Datei enthalten sind und welche Aktionen mit den jeweiligen Berechtigungen ausgeführt werden können:

Berechtigung	Beschreibung
Vollzugriff	Der Benutzer bzw. die Gruppe hat volle Zugriffsrechte auf das Ordner-Objekt und darf den Ordner durchsuchen, den Inhalt auflisten, Dateien erstellen, Ordner erstellen, Unterordner und Dateien löschen, dessen Attribute lesen und schreiben, dessen erweiterte Attribute lesen und schreiben, Berechtigungen lesen und auch ändern sowie die Besitzrechte übernehmen.
Ändern	Der Benutzer bzw. die Gruppe darf den Ordner durchsuchen, den Inhalt auflisten, Dateien erstellen, Ordner erstellen, dessen Attribute lesen und schreiben, dessen erweiterte Attribute lesen und schreiben sowie Berechtigungen lesen.
Lesen, Ausführen	Der Benutzer bzw. die Gruppe darf den Ordner durchsuchen, den Inhalt auflisten, dessen Attribute lesen, dessen erweiterte Attribute lesen sowie Berechtigungen lesen.

Tabelle 26: Die generischen Berechtigungen

Berechtigung	Beschreibung
Ordnerinhalt auflisten	Der Benutzer bzw. die Gruppe darf den Ordner durchsuchen, den Inhalt auflisten, dessen Attribute lesen sowie Berechtigungen lesen.
Lesen	Der Benutzer bzw. die Gruppe darf den Inhalt auflisten, dessen Attribute lesen, dessen erweiterte Attribute lesen sowie Berechtigungen lesen.
Schreiben	Der Benutzer bzw. die Gruppe darf Dateien erstellen/Daten schreiben, Ordner erstellen/Daten anhängen, dessen Attribute schreiben sowie dessen erweiterte Attribute schreiben.

Tabelle 26: Die generischen Berechtigungen (Forts.)

Wie bereits erwähnt, setzen sich die generischen Berechtigungen für Dateien und Ordner aus mehreren Einzelberechtigungen (von nun an *spezielle Berechtigungen* genannt) zusammen. Die folgende Tabelle enthält eine Übersicht dieser speziellen Berechtigungen unter Angabe einer kurzen Beschreibung:

Berechtigung	Beschreibung
Vollzugriff	Das Recht <i>Vollzugriff</i> setzt sich aus den nachfolgenden Einzelberechtigungen zusammen und beinhaltet damit alle Rechte, die zugelassen oder verweigert werden können.
Ordner durchsuchen/ Datei ausführen	Das Recht <i>Ordner durchsuchen</i> berechtigt einen Benutzer oder eine Gruppe, den Inhalt des Ordners inklusive der darin enthaltenen Objekte zu durchsuchen, auch wenn der Benutzer oder die Gruppe keine Berechtigungen auf die untergeordneten Objekte hat. Das Recht <i>Datei ausführen</i> berechtigt den Benutzer oder die Gruppe, die Datei auszuführen.
Ordner auflisten/ Datei lesen	Das Recht <i>Ordner auflisten</i> berechtigt einen Benutzer oder eine Gruppe, den Inhalt eines Ordners (Unterordner und Dateien) aufzulisten. Das Recht <i>Datei lesen</i> berechtigt einen Benutzer oder eine Gruppe, eine Datei zu öffnen und deren Inhalt einzusehen.
Attribute lesen	Das Recht <i>Attribute lesen</i> berechtigt einen Benutzer oder eine Gruppe, die Datei- oder Ordner-Attribute (Read-Only, Hidden, System und Archive) zu lesen.
Erweiterte Attribute lesen	Das Recht <i>Erweiterte Attribute lesen</i> berechtigt einen Benutzer oder eine Gruppe, die erweiterten Attribute, die seitens einer Applikation bereitgestellt werden, zu lesen.
Dateien erstellen/ Daten schreiben	Das Recht <i>Dateien erstellen</i> berechtigt einen Benutzer oder eine Gruppe, im Ordner neue Dateien zu erstellen. Das Recht <i>Daten schreiben</i> berechtigt einen Benutzer oder eine Gruppe, die Datei zu öffnen und deren Inhalt zu verändern.

Tabelle 27: Die speziellen Berechtigungen für Dateien und Ordner

Berechtigung	Beschreibung
Ordner erstellen/ Daten anhängen	Das Recht <i>Ordner erstellen</i> berechtigt einen Benutzer oder eine Gruppe, im Ordner weitere Unterordner anzulegen. Das Recht <i>Daten anhängen</i> berechtigt einen Benutzer oder eine Gruppe, der Datei weitere Daten anzufügen, ohne dass der bestehende Inhalt verändert werden kann.
Attribute schreiben	Das Recht <i>Attribute schreiben</i> berechtigt einen Benutzer oder eine Gruppe, die Datei- oder Ordner-Attribute (Read-Only, Hidden, System und Archive) zu verändern.
Erweiterte Attribute schreiben	Das Recht <i>Erweiterte Attribute schreiben</i> berechtigt einen Benutzer oder eine Gruppe, die erweiterten Attribute, die seitens einer Applikation bereitgestellt werden, zu schreiben und damit auch zu ändern.
Löschen	Das Recht <i>Löschen</i> berechtigt einen Benutzer oder eine Gruppe, die betreffende Datei oder den Ordner vom Dateisystem zu löschen.
Berechtigungen lesen	Das Recht <i>Berechtigungen lesen</i> berechtigt einen Benutzer oder eine Gruppe, die Einträge der Zugriffskontrollliste zu lesen.
Berechtigungen ändern	Das Recht <i>Berechtigungen ändern</i> berechtigt einen Benutzer oder eine Gruppe, Änderungen an der Zugriffskontrollliste der Datei oder des Ordners vorzunehmen und Berechtigungen für sich selbst oder für andere Benutzer/Gruppen neu einzurichten, zu ändern oder auch zu entfernen.
Besitzrechte übernehmen	Das Recht <i>Besitzrechte übernehmen</i> berechtigt einen Benutzer oder eine Gruppe, die Besitzrechte einer Datei oder eines Ordners zu übernehmen.

Tabelle 27: Die speziellen Berechtigungen für Dateien und Ordner (Forts.)

Nun genug der Theorie über Zugriffskontrolllisten, generische und spezielle Berechtigungen. Wenn Sie unter grafischer Oberfläche die Berechtigungen für eine Datei oder für einen Ordner ermitteln möchten, so rufen Sie zunächst den WINDOWS EXPLORER auf und markieren die Datei bzw. den Ordner, dessen Berechtigungen Sie einsehen möchten. Anschließend wählen Sie im Kontextmenü der Datei oder des Ordners den Eintrag EIGENSCHAFTEN aus und wechseln im Dialogfenster EIGENSCHAFTEN VON auf die Registerkarte SICHERHEIT.

Im Feld GRUPPEN- ODER BENUTZERNAME finden Sie nun alle Benutzer und Gruppen vor, für die der Zugriff auf die Datei oder auf den Ordner zugelassen oder aber explizit verweigert wird. Wenn Sie einen Benutzer oder eine Gruppe markieren, so sehen Sie im unteren Bereich des Dialogfensters (im Feld BERECHTIGUNGEN FÜR) die Standardberechtigungen, die für den entsprechenden Benutzer oder für die Gruppe gelten.

Für den Fall, dass für einen Benutzer oder für eine Gruppe bestimmte Berechtigungen zugelassen und auch verweigert wurden, so haben die verweigerten Berechtigungen auf jeden Fall Vorrang. Wenn beispielsweise ein Benutzer in der Spalte ZULASSEN über volle Zugriffsrechte verfügt, in der Spalte VERWEIGERN aber das Recht SCHREIBEN angeklickt ist, so darf er – trotz der vollen Zugriffsrechte – keine Änderungen an der Datei oder am Ordnerinhalt vornehmen, da die verweigerten Berechtigungen immer vor den zugelassenen Berechtigungen Gültigkeit haben.

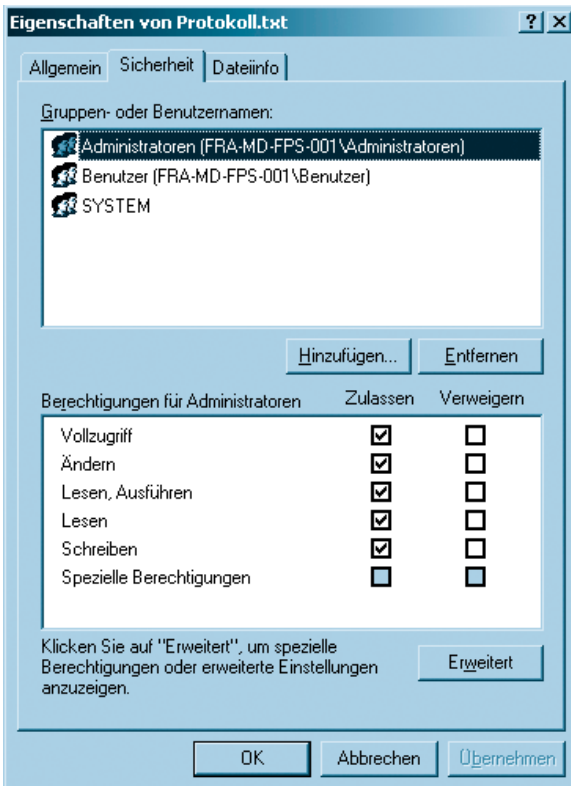


Abbildung 77: Die Berechtigungen für eine Datei anzeigen lassen

Sofern spezielle Berechtigungen vergeben oder aber verweigert wurden, so klicken Sie auf ERWEITERT, markieren dann den Benutzer oder die Gruppe erneut und führen einen Klick auf die Schaltfläche BEARBEITEN aus, woraufhin Sie in die Einstellungen der speziellen Berechtigungen gelangen und sich die einzelnen Rechte ansehen können. Grau dargestellte Berechtigungen sind vererbte Berechtigungen, die auf einer übergeordneten Ebene vergeben wurden.

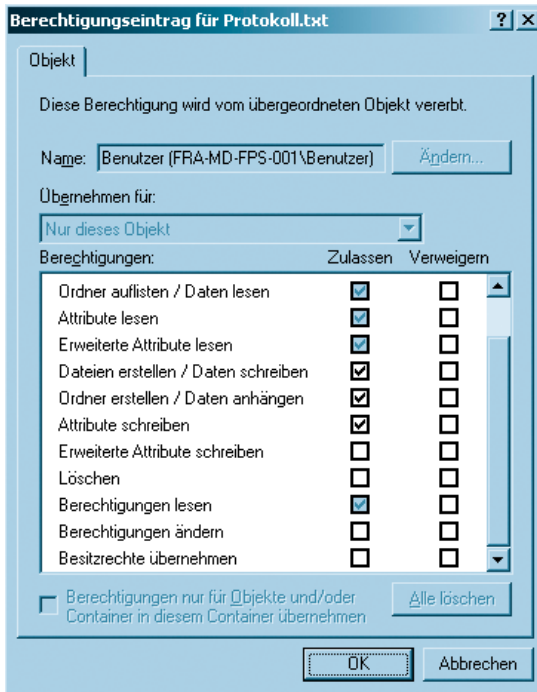


Abbildung 78: Die speziellen Berechtigungen für eine Datei anzeigen lassen

Die Lösung per Eingabeaufforderung

Wenn Sie die Berechtigungen, die auf eine Datei oder auf einen Ordner vergeben sind, dagegen von der Eingabeaufforderung aus abfragen möchten, so können Sie hierfür den Windows Server 2003-eigenen Befehl mit der Bezeichnung `Cacls.exe` einsetzen, der zumindest die Angabe der Datei oder des Ordners, dessen Zugriffs-kontrollliste Sie auslesen möchten, als Parameter erwartet.

Um herauszufinden, welche Einträge in der Zugriffskontrollliste der Datei `C:\Log-Files\Protokoll.txt` enthalten sind, so wenden Sie den `Cacls`-Befehl an und übergeben den Pfad und auch den Namen der entsprechenden Datei. Der `Cacls`-Befehl fragt daraufhin die ACL der angegebenen Datei ab und gibt das Ergebnis am Bildschirm aus.

```
@Echo off
Cacls C:\LogFiles\Protokoll.txt
```

Listing 145: 2_02_002a.bat

Das Ergebnis enthält eine Auflistung aller Benutzer- und Gruppenkonten, die in der Zugriffskontrollliste eingetragen sind, und zeigt am Ende jeder Zeile das entsprechende Berechtigungs-Flag, worüber die Berechtigung für den Benutzer oder für die Gruppe abgelesen werden kann. Folgende Flags werden verwendet:

Wert	Beschreibung
N	Keine Berechtigung
R	Lesen
W	Schreiben
C	Ändern
F	Vollzugriff

Tabelle 28: Die Werte der Berechtigungen

Der `CacIs`-Befehl liest nicht nur die Zugriffskontrolllisten einer Datei aus, sondern auch die eines Ordners. Wenn Sie also herausfinden möchten, welche Einträge die ACL des Ordners `C:\LogFiles` aufweist, so übergeben Sie dem `CacIs`-Befehl lediglich den Pfad und auch den Namen des betreffenden Ordners als Parameter.

```
@Echo off
CacIs C:\LogFiles
```

Listing 146: 2_02_002b.bat

Sie erhalten daraufhin eine ähnliche Aufstellung wie bei der Ausgabe der ACL-Einträge einer Datei. Ähnlich heißt in diesem Fall, dass zunächst wieder die Benutzer und Gruppen aufgelistet werden, dann das entsprechende Vererbungs-Flag (siehe nachfolgende Tabelle) ausgegeben und erst dann das Berechtigungs-Flag (siehe vorherige Tabelle) ausgewiesen wird.

Wert	Beschreibung
CI (Container inherit)	Der ACL-Eintrag gilt für diesen Ordner und auch für Unterordner.
OI (Object inherit)	Der ACL-Eintrag gilt für diesen Ordner und auch für die Dateien.
IO (Inherit only)	Der ACL-Eintrag gilt nicht für die aktuelle Datei bzw. den aktuellen Ordner, sondern dient nur der Vererbung.
(OI)(CI)	Der ACL-Eintrag gilt für diesen Ordner, für die Unterordner und auch für Dateien.
(OI)(CI)(IO)	Der ACL-Eintrag gilt nur für Unterordner und auch für Dateien.
(CI)(IO)	Der ACL-Eintrag gilt nur für Unterordner.
(OI)(IO)	Der ACL-Eintrag gilt nur für Dateien.
Keine Angabe	Der ACL-Eintrag gilt nur für diesen Ordner.

Tabelle 29: Die Werte der Vererbungen

Die Lösung mit VBScript

Um die Zugriffskontrolllisten einer Datei oder eines Ordners per VBScript auszulesen, verwenden Sie WMI und stellen zunächst eine Verbindung zur WMI-Klasse Win32_LogicalFileSecuritySetting her. Anschließend setzen Sie die GetSecurityDescriptor-Methode ein, um eine Instanz der Sicherheitsbeschreibung für das angegebene Datei- oder Ordner-Objekt zurückzuerhalten, und lesen dann den Inhalt des DACL-Arrays und damit die Einträge der Zugriffskontrollliste aus.

Die einzelnen Datei- und Ordnerberechtigungen sind dabei als Werte hinterlegt, die wie folgt zu interpretieren sind:

Konstante	Wert	Spezielle Berechtigung
FILE_LIST_DIRECTORY FILE_READ_DATA	&h000001	Ordner auflisten/Daten lesen
FILE_ADD_FILE FILE_WRITE_DATA	&h000002	Dateien erstellen/Daten schreiben
FILE_ADD_SUBDIRECTORY FILE_APPEND_DATA	&h000004	Ordner erstellen/Daten anhängen
FILE_READ_EA	&h000008	Erweiterte Attribute lesen
FILE_WRITE_EA	&h000010	Erweiterte Attribute schreiben
FILE_TRAVERSE FILE_EXECUTE	&h000020	Ordner durchsuchen/Datei ausführen
FILE_DELETE_CHILD	&h000040	Unterordner und Dateien löschen
FILE_READ_ATTRIBUTES	&h000080	Attribute lesen
FILE_WRITE_ATTRIBUTES	&h000100	Attribute schreiben
FILE_DELETE	&h010000	Löschen
FILE_READ_CONTROL	&h020000	Berechtigungen lesen
FILE_WRITE_DAC	&h040000	Berechtigungen ändern
FILE_WRITE_OWNER	&h080000	Besitzrechte übernehmen
FILE_SYNCHRONIZE	&h100000	Synchronisieren
FILE_ALL_ACCESS	&h1f01ff	Alle Dateirechte

Tabelle 30: Die Datei- und Ordnerberechtigungen und ihre Werte

Das nachfolgende Listing gibt beispielsweise den Inhalt der Zugriffskontrollliste für die Datei C:\LogFiles\Protokoll.txt am Bildschirm aus. Das Skript enthält zu Beginn die Deklaration der erforderlichen Variablen und Konstanten, stellt dann eine Verbindung zur WMI-Klasse Win32_LogicalFileSecuritySetting her, bindet die Instanz des Datei-Objektes an eine Objektvariable und führt dann die GetSecurityDescriptor-Methode aus, woraufhin die Sicherheitsbeschreibung (Security Descriptor) in Form einer Kollektion zurückgeliefert wird.

Innerhalb einer For Each-Schleife wird dann für jeden ACL-Eintrag der Name des Benutzers oder der Gruppe abgefragt, der Wert der Eigenschaft ACE_TYPE (ACCESS_ALLOWED_ACE_TYPE = Zulassen und ACCESS_DENIED_ACE_TYPE = Verweigern) ermittelt, die zugelassene oder verweigerte Berechtigung ausgelesen und per WScript.Echo-Anweisung am Bildschirm ausgegeben. Diese Abfragen werden so lange wiederholt, bis alle Einträge des DACL-Arrays verarbeitet wurden.

Option Explicit

```
Dim oWMI
Dim oFSS
Dim oSD
Dim strComputer
Dim strFile
Dim intVal
Dim intFlags
Dim arrACE
Dim objItem
Dim strPerm
```

```
strcomputer = "."
strFile = "C:\\LogFiles\\Protokoll.txt"
```

```
Const ACCESS_ALLOWED_ACE_TYPE = &h0
Const ACCESS_DENIED_ACE_TYPE = &h1
```

```
Const FILE_READ_DATA = &h000001
Const FILE_WRITE_DATA = &h000002
Const FILE_APPEND_DATA = &h000004
Const FILE_READ_EA = &h000008
Const FILE_WRITE_EA = &h000010
Const FILE_EXECUTE = &h000020
Const FILE_READ_ATTRIBUTES = &h000080
Const FILE_WRITE_ATTRIBUTES = &h000100
Const FILE_DELETE = &h010000
Const FILE_READ_CONTROL = &h020000
Const FILE_WRITE_DAC = &h040000
Const FILE_WRITE_OWNER = &h080000
```

```
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=impersonate}!\\_
    & strComputer & "\\root\\cimv2")
Set oFSS = oWMI.Get("Win32_LogicalFileSecuritySetting='" & strFile & "'")
intVal = oFSS.GetSecurityDescriptor(oSD)
intFlags = oSD.ControlFlags
```

```
If intFlags Then
```

Listing 147: 2_02_002.vbs

```

arrACE = oSD.DACL
For Each objItem in arrACE
    WScript.Echo objItem.Trustee.Domain & "\" & objItem.Trustee.Name

    If objItem.AceType = ACCESS_DENIED_ACE_TYPE Then
        strPerm = " (Verweigert)"
    Else
        strPerm = ""
    End If

    If objItem.AccessMask AND FILE_EXECUTE Then
        WScript.Echo VBTAB & "Ordner durchsuchen/Datei ausführen" & strPerm
    End If

    If objItem.AccessMask AND FILE_READ_DATA Then
        WScript.Echo VBTAB & "Ordner auflisten/Daten lesen" & strPerm
    End If

    If objItem.AccessMask AND FILE_READ_ATTRIBUTES Then
        WScript.Echo VBTAB & "Attribute lesen" & strPerm
    End If

    If objItem.AccessMask AND FILE_READ_EA Then
        WScript.Echo VBTAB & "Erweiterte Attribute lesen" & strPerm
    End If

    If objItem.AccessMask AND FILE_WRITE_DATA Then
        WScript.Echo VBTAB & "Dateien erstellen/Daten schreiben" & strPerm
    End If

    If objItem.AccessMask AND FILE_APPEND_DATA Then
        WScript.Echo VBTAB & "Ordner erstellen/Daten anhängen" & strPerm
    End If

    If objItem.AccessMask AND FILE_WRITE_ATTRIBUTES Then
        WScript.Echo VBTAB & "Attribute schreiben" & strPerm
    End If

    If objItem.AccessMask AND FILE_WRITE_EA Then
        WScript.Echo VBTAB & "Erweiterte Attribute schreiben" & strPerm
    End If

    If objItem.AccessMask AND FILE_DELETE Then
        WScript.Echo VBTAB & "Löschen" & strPerm
    End If

```

```

If objItem.AccessMask AND FILE_READ_CONTROL Then
    WScript.Echo VBTAB & "Berechtigungen lesen" & strPerm
End If

If objItem.AccessMask AND FILE_WRITE_DAC Then
    WScript.Echo VBTAB & "Berechtigungen ändern" & strPerm
End If

If objItem.AccessMask AND FILE_WRITE_OWNER Then
    WScript.Echo VBTAB & "Besitzrechte übernehmen" & strPerm
End If
WScript.Echo ""
Next
End If

Set oWMI = Nothing
Set oFSS = Nothing

```

Listing 147: 2_02_002.vbs (Forts.)

29 Berechtigungen für Dateien und Ordner einrichten und bearbeiten

Wenn Sie auf einem mit *NTFS* formatierten Datenträger ein Datei- oder Ordner-Objekt erzeugen, so wird die Datei bzw. der Ordner automatisch mit einer Sicherheitsbeschreibung (siehe Welche Berechtigung ist auf die Datei oder auf den Ordner vergeben? auf Seite 233) versehen und erhält damit auch eine Zugriffskontrollliste (engl. = *Access Control List*), worüber die Berechtigungen für das Datei-/Ordner-Objekt gesteuert werden können. Mit Hilfe dieser Zugriffskontrollliste können Sie den Benutzern und Gruppen Zugriffsrechte erteilen oder wahlweise auch verweigern.

Die Zugriffskontrollliste, die mit dem Anlegen eines Datei- oder Ordner-Objektes erstellt wird, ist jedoch nicht leer, sondern enthält per Default bereits entsprechende Einträge (engl. = *Access Control Entry*). Denn Windows Server 2003 verwendet einen Vererbungsmechanismus, womit die Berechtigungen eines übergeordneten Objektes automatisch auf die untergeordneten Objekte weitergegeben (vererbt) werden. Wenn Sie also innerhalb eines Ordners eine Datei oder einen zusätzlichen Ordner anlegen, so erhält das neue Objekt automatisch die Berechtigungen, die für das übergeordnete Objekt vergeben wurden. Sollte die Vererbung des übergeordneten Objektes jedoch deaktiviert sein, so enthält die neue ACL automatisch die Standardberechtigung für Datei- und Ordner-Objekte. Windows Server 2003 verwendet die folgenden Standardberechtigungen:

Benutzer/Gruppe	Berechtigung
Administratoren	Vollzugriff
Benutzer	Lesen und Ausführen
System	Vollzugriff
Ersteller-Besitzer	Vollzugriff

Tabelle 31: Die Standardberechtigungen für Datei- und Ordner-Objekte

Legen Sie beispielsweise im Root des Laufwerks C einen neuen Ordner mit der Bezeichnung *LogFiles* an, so erhält dieser Ordner erst einmal die Berechtigungen, die auf das Root des Laufwerks (C:\) vergeben wurden. Die ACL des Ordners C:\LogFiles enthält damit ausschließlich vererbte Berechtigungen, die nicht in der ACL des Ordners C:\LogFiles verändert werden können, da sie nicht explizit vergeben, sondern von einem übergeordneten Objekt weitergegeben wurden. Sie können der ACL von C:\LogFiles aber weitere ACL-Einträge hinzufügen und explizite Berechtigungen vergeben, womit diese Berechtigungen dann Vorrang haben und wiederum an alle Objekte, die unterhalb von C:\LogFiles angelegt werden, automatisch weitergegeben werden.

- ▶ **Vererbte Berechtigungen:** Eine vererbte Berechtigung wurde in einem übergeordneten Objekt erzeugt und wird automatisch an die untergeordneten Objekte weitergereicht (vererbt). Die Vererbung der Berechtigungen kann für ein Datei- oder Ordner-Objekt deaktiviert werden, womit Sie die Möglichkeit haben, die bereits vererbten Berechtigungen zu kopieren oder ganz zu entfernen und damit nur explizite Berechtigungen zu erteilen. Vererbte Berechtigungen können nur dort verändert werden, wo sie explizit vergeben wurden.
- ▶ **Explizite Berechtigungen:** Explizite Berechtigungen wurden seitens des Erstellers oder des Administrators in der ACL des entsprechenden Datei- oder Ordner-Objektes erteilt und nicht von einem übergeordneten Objekt geerbt, sondern explizit festgelegt. Eine explizite Berechtigung hat gegenüber einer geerbten Berechtigung Vorrang. Das heißt, wenn eine Zugriffsverweigerung vererbt, aber eine explizite Zugriffsberechtigung erteilt wurde, so haben die explizite Berechtigung und damit die Zugriffsberechtigung Vorrang vor der Zugriffsverweigerung.

Achtung

Einmal angenommen, Sie haben den Ordner C:\LogFiles angelegt, der per Vererbung die Berechtigungen vom Root des Laufwerks (C:\) erhalten hat. Damit haben alle Mitglieder der Gruppe *Benutzer* nun eine Lese- und Ausführungsrecht, dürfen aber keine Daten ändern, geschweige denn löschen. Nun möchten Sie dem Benutzer *TestUser1*, der ebenfalls Mitglied der Gruppe *Benutzer* ist, die Berechtigung zum Ändern des Ordners und auch dessen Inhaltes erteilen. Hierzu rufen Sie den WINDOWS EXPLORER auf, markieren den Ordner C:\LogFiles und rufen aus dem Kontextmenü des Ordners die EIGENSCHAFTEN auf. Anschließend wechseln Sie auf die Registerkarte SICHERHEIT und fügen zunächst den Benutzer *TestUser1* der ACL hinzu. Klicken Sie auf HINZUFÜGEN,

und geben Sie den Benutzernamen ein, oder betätigen Sie die Schaltfläche ERWEITERT... und anschließend JETZT SUCHEN, woraufhin alle Benutzer- und Gruppenkonten aufgelistet werden. Wählen Sie das entsprechende Benutzerkonto aus, und schließen Sie die letzten beiden Dialogfenster wieder, um zur Registerkarte SICHERHEIT zurückzukehren.

Der Benutzer *TestUser1* ist nun in der ACL enthalten, und Sie können dem Benutzer nun das generische Änderungsrecht erteilen, indem Sie den Benutzer markieren und im unteren Teil des Dialogfensters in der Spalte ZULASSEN die Option ÄNDERN markieren. Damit hat der Benutzer zwar immer noch die vererbte Berechtigung LESEN und AUSFÜHREN, doch die explizite Vergabe der Berechtigung ÄNDERN hat Vorrang, womit die soeben erteilte Berechtigung als primäre bzw. als effektive Berechtigung gilt.

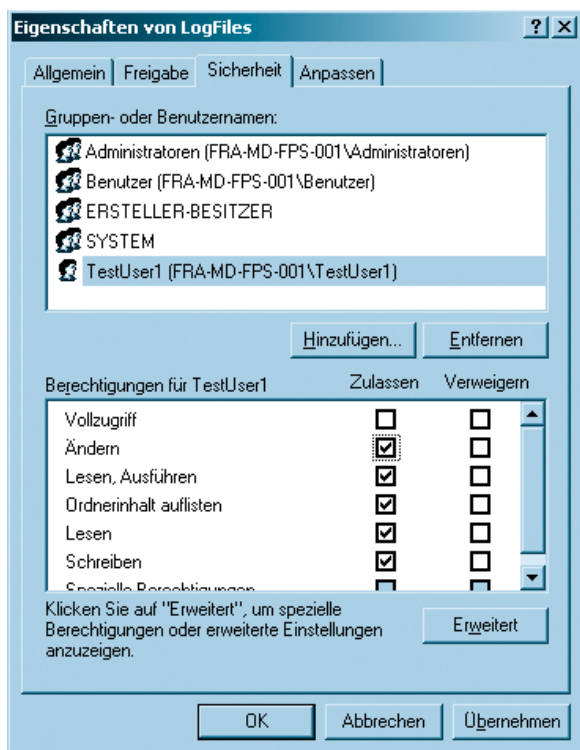


Abbildung 79: Eine zusätzliche Berechtigung für »TestUser1« einrichten

Die generischen Berechtigungen, die auf der Registerkarte SICHERHEIT angezeigt werden, stellen eine Gruppierung diverser Einzelberechtigungen dar und lassen sich natürlich auch einzeln vergeben. Wenn Sie dem *TestUser1* kein generisches Änderungsrecht vergeben möchten, sondern beispielsweise das ausschließliche Recht zum Erstellen neuer Dateien, weil Sie mit diesem Benutzerkonto vielleicht neue Protokolldateien erzeugen, so klicken Sie zunächst auf die Schaltfläche ERWEITERT, markieren auf der Registerkarte

BERECHTIGUNGEN den Benutzer *TestUser1* (wobei Sie der Spalte GEERBT VON die vererbte und explizite Berechtigung entnehmen können), klicken dann auf BEARBEITEN und erteilen dem Benutzer das entsprechende Recht.

Haben Sie beispielsweise im Ordner *C:\LogFiles* einen weiteren Ordner mit der Bezeichnung *Protokolle* angelegt und möchten Sie die Vererbung der Berechtigungen vom Ordner *C:\LogFiles* unterbinden und damit ausschließlich explizite Berechtigungen verwenden, so können Sie die Vererbung deaktivieren. Hierzu markieren Sie den Ordner *Protokolle*, rufen die EIGENSCHAFTEN auf, wechseln auf die Registerkarte SICHERHEIT und klicken auf ERWEITERT, um das Dialogfenster ERWEITERTE SICHERHEITSEINSTELLUNGEN FÜR PROTOKOLLE zu öffnen. Dort deaktivieren Sie die Option: BERECHTIGUNGEN ÜBERGEORDNETER OBJEKTE, SOFERN VERERBBAR, ÜBER ALLE UNTERGEORDNETEN OBJEKTE VERBREITEN. DIESE OBJEKTE INKLUSIVE DER HIER DEFINIERTEN EINTRÄGE MIT EINBEZIEHEN.

Sie werden daraufhin gefragt, ob Sie die bereits vererbten Berechtigungen kopieren oder entfernen möchten oder ob Sie den Vorgang abbrechen wollen. Wählen Sie KOPIEREN aus, um alle bereits vererbten Berechtigungen in explizite Berechtigungen zu konvertieren, womit die bisherigen Berechtigungen bestehen bleiben und von nun an auch geändert werden können. Wählen Sie stattdessen ENTFERNEN aus, um alle bereits vererbten Berechtigungen aus der ACL zu löschen. Klicken Sie auf *Übernehmen*, um die neuen Einstellungen wirksam werden zu lassen, und schließen Sie das Dialogfenster wieder.

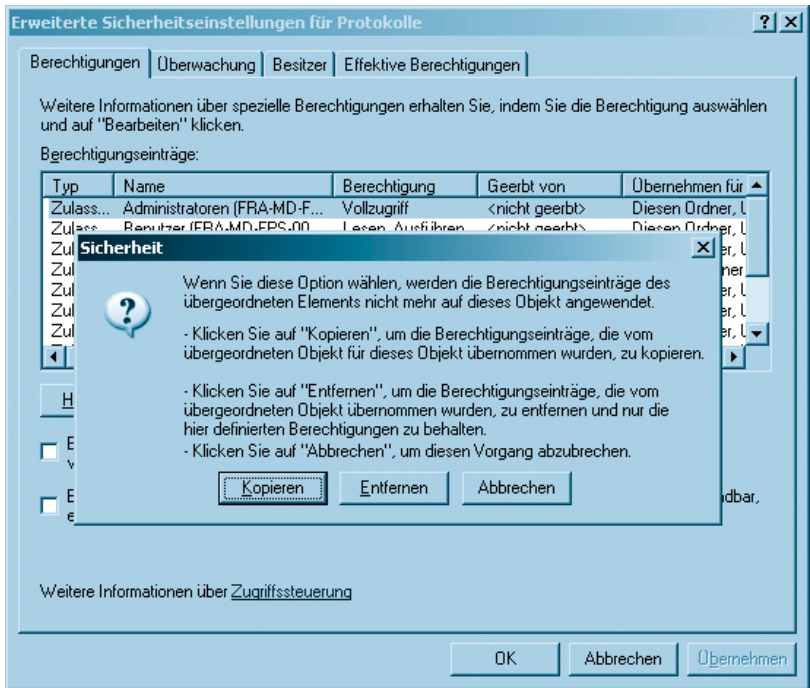


Abbildung 80: Die vererbten Berechtigungen in explizite Berechtigungen kopieren oder entfernen

Alle Datei- und Ordner-Objekte, die Sie von nun an im Ordner `C:\LogFiles\Protokolle` anlegen, erhalten lediglich die Berechtigungen, die auf den Ordner *Protokolle* vergeben wurden, da die Vererbung der Berechtigungen ausgeschaltet ist.

Tipp

Die Berechtigungen für ein Datei- oder Ordner-Objekt lassen sich nicht nur erteilen, sondern auch verweigern, wobei ein verweigertes Recht immer Vorrang vor einem erteilten Recht hat. Wenn Sie einem Benutzer oder einer Gruppe einen Vollzugriff auf eine Datei oder auf einen Ordner erlauben und das Lese-recht verweigern, so dürfen die Benutzer bzw. Gruppenmitglieder die Datei oder den Ordnerinhalt trotz des Vollzugriffes nicht lesen und auflisten. Berechtigungen werden meist dann verweigert, wenn einem Benutzer oder einer Gruppe ad hoc die Zugriffsberechtigungen genommen werden sollen, weil beispielsweise gerade Änderungen am Dateisystem bzw. an der Datei- und Ordnerstruktur vorgenommen oder Wartungsarbeiten ausgeführt werden sollen.

Die Lösung per Eingabeaufforderung

Die Berechtigungen für Dateien und Ordner lassen sich auch von der Eingabeaufforderung aus ohne großen Aufwand vergeben, da der Windows Server 2003 hierfür den `CacIs`-Befehl bereitstellt, womit Sie Datei- und Ordnerberechtigungen abfragen und auch bearbeiten können.

Um beispielsweise den Benutzer *TestUser1* in der ACL des Ordners `C:\LogFiles` aufzunehmen und diesem Benutzer das generische Recht *Ändern* zuzuweisen, wenden Sie den `CacIs`-Befehl an und übergeben zunächst den Namen des Datei- oder Ordner-Objektes (in diesem Fall `C:\LogFiles`), bestimmen mit dem Parameter `/E`, dass die ACL nicht ersetzt, sondern nur bearbeitet werden soll, und übergeben mit dem Parameter `/G` den Benutzernamen (Format *Computer- oder Domänenname\Benutzername*) und auch die gewünschte Berechtigung `C` (siehe Tabelle 28: Die Werte der Berechtigungen auf Seite 239).

```
@Echo off
CacIs C:\LogFiles /E /G FRA-MD-FPS-001\TestUser1:C
```

Listing 148: 2_02_003a.bat

Wenn Sie den Benutzer (oder wahlweise auch eine Gruppe) wieder aus der ACL herausnehmen und die soeben explizit vergabene Berechtigung aufheben möchten, so wenden Sie den Parameter `/R` an und übergeben den Namen des betreffenden Benutzers oder der Gruppe.

```
@Echo off
CacIs C:\LogFiles /R FRA-MD-FPS-001\TestUser1
```

Listing 149: 2_02_003b.bat

Mit dem `CacIs`-Befehl lässt sich natürlich auch das Anwenden der vererbten Berechtigungen eines übergeordneten Objektes deaktivieren. Wenn Sie beispielsweise im Ordner `C:\LogFiles` einen Ordner mit der Bezeichnung *Protokolle* angelegt und diesem Ordner ausschließlich explizite Berechtigungen vergeben haben und alle vererbten Berechtigungen entfernen möchten, so lassen Sie den Parameter `/E` weg und bearbeiten damit die ACL nicht, sondern ersetzen sie. Damit sind dann nur die Einträge in der ACL zu finden, die Sie mit dem Aufruf des `CacIs`-Befehls übergeben.

```
@Echo off
CacIs C:\LogFiles /G FRA-MD-FPS-001\TestUser1:F
```

Listing 150: 2_02_003c.bat

Die Lösung mit VBScript

Wenn Sie der ACL eines Datei- oder Ordner-Objektes per VBScript einen neuen Eintrag hinzufügen möchten, so binden Sie die Sicherheitsbeschreibung (engl. = *Security Descriptor*) des betreffenden Datei- oder Ordner-Objektes zunächst an eine Objektvariable und selektieren dann mit einer `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_Account` das Benutzer- oder Gruppenkonto, für das Sie die Berechtigung erteilen oder verweigern möchten.

Anschließend ermitteln Sie mit einer weiteren `ExecQuery`-Abfrage (diesmal gegen die WMI-Klasse `Win32_SID`) die SID des Benutzers oder der Gruppe, erzeugen eine neue Instanz des `Trustee`-Objektes und weisen dem `Trustee`-Objekt die Eigenschaften `Domain`, `Name`, `SID`, `SIDLength`, `SIDString` zu, die Sie aus den Eigenschaften `ReferencedDomainName`, `AccountName`, `BinaryRepresentation`, `SIDLength` und `SID` des `SID`-Objektes entnehmen. Danach erzeugen Sie eine neue Instanz des `ACE`-Objektes und weisen die Werte für die Eigenschaften `AccessMask` (Berechtigung), `Trustee` (Benutzer-/Gruppenname), `AceType` (Zulassen oder Verweigern) und `AceFlags` (Vererbungsoption) zu.

Und zu guter Letzt erzeugen Sie ein neues `Dictionary`-Objekt, nehmen alle bereits vorhandenen ACL-Einträge als Elemente auf, erzeugen im Anschluss daran den neuen ACL-Eintrag, den Sie mit der `Add`-Methode dem `Dictionary`-Objekt hinzufügen, und weisen der ACL des Datei- oder Ordner-Objektes (`oSD.DACL`) den Inhalt des `Dictionary`-Objektes zu.

Das nachfolgende Code-Beispiel fügt den Benutzer *TestUser1* (`strName`) zur ACL der Datei `C:\LogFiles\Protokoll.txt` (`strFileName`) hinzu und weist dem Benutzer das explizite Recht *Ordner auflisten/Daten lesen* (`FILE_READ_DATA`) zu, das in diesem Beispiel erlaubt (`ACCESS_ALLOWED_ACE_TYPE`) wird.

Option Explicit

```
Dim oWMI
Dim oFSS
Dim oUser
Dim oSID
Dim oTrusteeClass
Dim oTrustee
Dim oACEClass
Dim oACE
Dim oDictionary
Dim oSD
Dim colUser
Dim strComputer
Dim strName
Dim strFile
Dim intVal
Dim intCounter
Dim objItem
```

```
strcomputer = "."
strFile = "C:\\LogFiles\\Protokoll.txt"
strName = "TestUser1"
```

```
Const ACCESS_ALLOWED_ACE_TYPE = &h0
Const ACCESS_DENIED_ACE_TYPE = &h1
Const FILE_READ_DATA = &h000001
Const FILE_WRITE_DATA = &h000002
Const FILE_APPEND_DATA = &h000004
Const FILE_READ_EA = &h000008
Const FILE_WRITE_EA = &h000010
Const FILE_EXECUTE = &h000020
Const FILE_READ_ATTRIBUTES = &h000080
Const FILE_WRITE_ATTRIBUTES = &h000100
Const FILE_DELETE = &h010000
Const FILE_READ_CONTROL = &h020000
Const FILE_WRITE_DAC = &h040000
Const FILE_WRITE_OWNER = &h080000
Const OBJECT_INHERIT_ACE = &h1
Const CONTAINER_INHERIT_ACE = &h2
Const NO_PROPAGATE_INHERIT_ACE = &h4
Const INHERIT_ONLY_ACE = &h8
```

```
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=impersonate}!\" _
    & strComputer & "\root\cimv2")
Set oFSS = GetObject("winmgmts:{(Security,Restore)}!\" & strComputer _
```

Listing 151: 2_02_003.vbs

```

        & "\root\cimv2:Win32_LogicalFileSecuritySetting='" & strFile & "'")
    intVal = oFSS.GetSecurityDescriptor(oSD)
    Set colUser = oWMI.ExecQuery("SELECT * FROM WIN32_ACCOUNT WHERE Name='" & _
        & strName & "'")
    For Each objItem in ColUser
        Set oUser=objItem
    Next

    Set oSID = oWMI.Get("Win32_SID.SID='" & oUser.SID & "'")
    Set oTrusteeClass = oWMI.Get("Win32_Trustee")
    Set oTrustee = oTrusteeClass.SpawnInstance_()
    oTrustee.Domain = oSID.ReferencedDomainName
    oTrustee.Name = oSID.AccountName
    oTrustee.SID = oSID.BinaryRepresentation
    oTrustee.SidLength = oSID.SidLength
    oTrustee.SIDString = oSID.SID

    Set oACEClass = oWMI.Get("Win32_ACE")
    Set oACE = oACEClass.SpawnInstance_()

    oACE.AccessMask = FILE_READ_DATA
    oACE.Trustee = oTrustee
    oACE.AceType = ACCESS_ALLOWED_ACE_TYPE
    oACE.AceFlags = OBJECT_INHERIT_ACE

    Set oDictionary = CreateObject("Scripting.Dictionary")
    For intCounter = LBound(oSD.DACL) to UBound(oSD.DACL)
        If NOT oSD.DACL(intcounter).AceFlags AND 16 Then
            oDictionary.Add intcounter, oSD.DACL(intcounter)
        End If
    Next
    oDictionary.Add "User", oACE
    oSD.DACL = oDictionary.Items
    intVal = oFSS.SetSecurityDescriptor(oSD)

    Set oWMI = Nothing
    Set oFSS = Nothing
    Set oUser = Nothing
    Set oSID = Nothing
    Set oTrusteeClass = Nothing
    Set oTrustee = Nothing
    Set oACEClass = Nothing
    Set oACE = Nothing
    Set oDictionary = Nothing
    Set oSD = Nothing

```

Listing 151: 2_02_003.vbs (Forts.)

30 Wer ist der Besitzer der Datei oder des Ordners?

Der Besitzer einer Datei oder wahlweise auch eines Ordners lässt sich unter grafischer Oberfläche auf recht einfache Art und Weise ermitteln. Denn Sie brauchen nur die DETAIL-Ansicht des WINDOWS EXPLORERS zu öffnen und die Spalte BESITZER einzublen- den, um die gewünschte Information zu erhalten.

Starten Sie den WINDOWS EXPLORER, wechseln Sie in den entsprechenden Pfad, und wählen Sie im Menü ANSICHT den Eintrag DETAIL aus, um eine detaillierte Datei- und Ordner-Auflistung zu erhalten. Wählen Sie anschließend im Menü ANSICHT den Ein- trag DETAILS AUSWÄHLEN... aus, und aktivieren Sie die Option BESITZER, um die DETAIL- Ansicht um diese Spalte zu erweitern. Schließen Sie die Auswahl der Eigenschaften wieder. Sie können der rechten Fensterhälfte nun den Besitzer einer jeden Datei und auch eines jeden Ordners des zuvor markierten Pfades entnehmen.

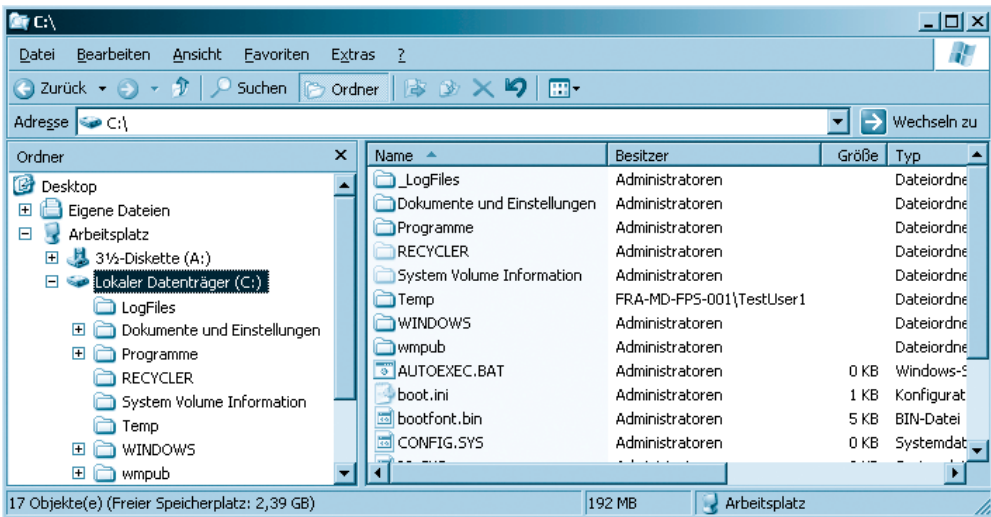


Abbildung 81: Die Anzeige des Besitzers im Windows Explorer

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung lässt sich der Besitzer einer Datei oder eines Ordners im einfachsten Fall mit dem Dir-Befehl unter Verwendung des Parameter /Q anzeigen. Allerdings stehen für die Besitzerinformation nur 23 Zeichen zur Verfügung, die restlichen Zeichen werden durch den Datei- bzw. Ordernamen überschrieben. Längere Benutzernamen werden deshalb nur unvollständig angezeigt.

```
@Echo off
Dir C:\Windows /Q
```

Listing 152: 2_02_004a.bat

Eine weitere Möglichkeit, den Besitzer einer Datei oder eines Ordners zu ermitteln, bietet der `Subinacl`-Befehl aus dem *Windows Server 2003 Resource Kit*, womit Sie sicherheitsrelevante Einstellungen für Dateien, Ordner, Freigaben, Registrierungs-schlüssel und viele weitere mehr abfragen und auch modifizieren können. Die `Subinacl`-Version, die im Resource Kit enthalten ist, ist jedoch mit einem Bug behaftet und sollte vor der ersten Verwendung durch die neuere Version, die Sie von der Internetseite www.microsoft.com herunterladen können, ausgetauscht werden.

Der `Subinacl`-Befehl erwartet zunächst den Objekttyp (in diesem Fall `/File`), den Sie abfragen oder modifizieren möchten, wie auch die Angabe des Objektes selbst. Und zum Auslesen der `Owner`-Eigenschaft übergeben Sie dem `Subinacl`-Befehl den zusätzlichen Parameter `/Display=Owner`, woraufhin die Datei (oder wahlweise auch der Ordner) unter Angabe des derzeitigen Besitzers am Bildschirm ausgegeben wird.

```
@Echo off
Subinacl /File C:\LogFiles\Protokoll.txt /Display=Owner
```

Listing 153: 2_02_004b.bat

Die Lösung mit VBScript

Der Besitzer einer Datei oder eines Ordners lässt sich natürlich auch per VBScript ermitteln. Hierzu können Sie wieder eine WMI-Abfrage einsetzen, wobei Sie in diesem Fall aber mehrere Klassen einbeziehen müssen, da es keine WMI-Klasse gibt, die Informationen über Dateien und Ordnern und zugleich auch über den Besitzer enthält.

Das nachfolgende Listing gibt beispielsweise den Besitzer der Datei `C:\LogFiles\Protokoll.txt` (`strFile`) am Bildschirm aus. Hierzu werden im ersten Schritt die Sicherheitseinstellungen des Datei-Objektes mit Hilfe einer `ExecQuery`-Abfrage selektiert und für eine weitere Abfrage an `Win32_LogicalFileOwner` übergeben. Daraufhin wird der Name (Eigenschaft `AccountName`) des Besitzers zurückgeliefert, der sich dann innerhalb einer `For Each`-Schleife abfragen und per `WScript.Echo` am Bildschirm ausgeben lässt.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim strFile
Dim colOwner
Dim objItem

strComputer = "."
strFile = "C:\LogFiles\Protokoll.txt"
```

Listing 154: 2_02_004.vbs


```

Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colOwner = oWMI.ExecQuery ("ASSOCIATORS OF _
    {Win32_LogicalFileSecuritySetting='\" & strFile & '\"}\" & \" _
    WHERE AssocClass=Win32_LogicalFileOwner ResultRole=Owner")

For Each objItem in colOwner
    Wscript.Echo objItem.AccountName
Next

Set oWMI = Nothing

```

Listing 154: 2_02_004.vbs (Forts.)

31 Besitz von Dateien und Ordner übernehmen

Mit der Voraussetzung, dass Sie entweder als *Administrator* angemeldet sind oder zumindest über das Privileg *Besitzrechte übernehmen* verfügen, so können Sie in den Sicherheitseinstellungen eines Datei- oder Ordner-Objektes problemlos den Besitz der Datei oder des Ordners übernehmen. Markieren Sie hierfür im WINDOWS EXPLORER das betreffende Objekt, und rufen Sie aus dem Kontextmenü heraus die EIGENSCHAFTEN auf. Auf der Registerkarte SICHERHEIT klicken Sie auf ERWEITERT und wechseln dann auf die Registerkarte BESITZER, um zunächst eine Übersicht des aktuellen Besitzers und auch der Benutzer und Gruppen mit dem Privileg BESITZRECHTE ÜBERNEHMEN zu erhalten.

Um das Besitzrecht zu übernehmen, markieren Sie im Feld BESITZER ÄNDERN AUF den entsprechenden Benutzer (oder die Gruppe) und klicken auf ÜBERNEHMEN. Und für den Fall, dass Sie das Besitzrecht eines Ordner-Objektes übernehmen möchten, so können Sie mit der Option BESITZER DER OBJEKTE UND UNTERGEORDNETEN CONTAINER ERSETZEN zeitgleich auch den Besitz für die darin enthaltenen Dateien und untergeordneten Ordner übernehmen und nicht nur für das Ordner-Objekt selbst (siehe Abbildung 82).

Die Lösung per Eingabeaufforderung

Selbiges lässt sich in der Eingabeaufforderung mit dem *Takeown*-Befehl realisieren, der standardmäßig im Befehlsumfang von Windows Server 2003 enthalten ist. Wenn Sie z.B. die Besitzrechte des Ordners *C:\LogFiles* inklusive der darin enthaltenen Dateien und Ordner auf die *Administratoren*-Gruppe übertragen wollen, dann wenden Sie den *Takeown*-Befehl an und übergeben mit dem Parameter */F* zunächst den Pfad des Objektes. Mit dem Parameter */A* legen Sie fest, dass die Besitzrechte auf die *Administratoren*-Gruppe übergehen sollen, und zur rekursiven Bearbeitung fügen Sie zusätzlich den Parameter */R* an.

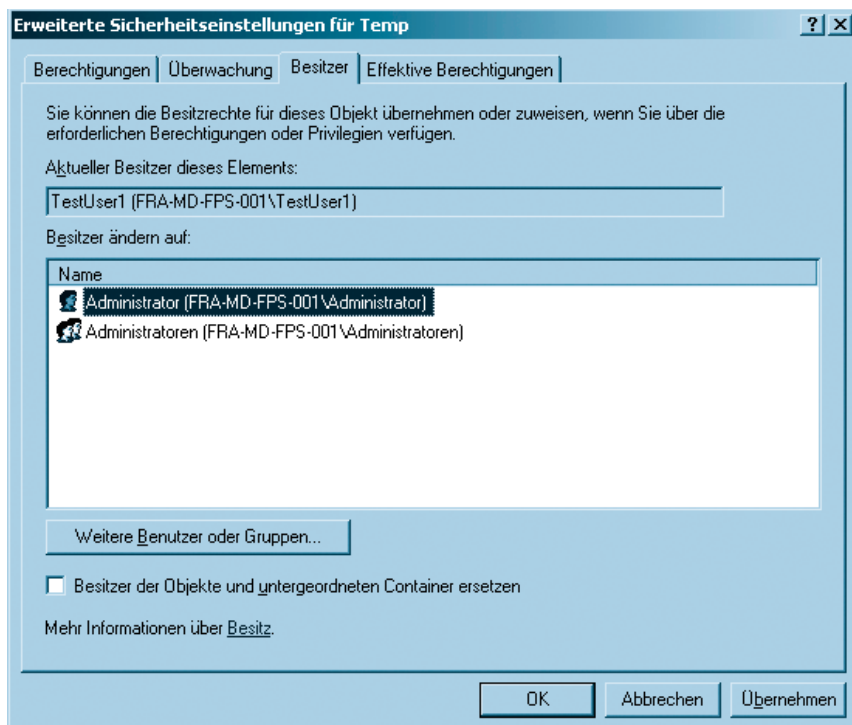


Abbildung 82: Die Besitzrechte für einen Ordner übernehmen

```
@Echo off
Takeown /F C:\LogFiles /A /R
```

Listing 155: 2_02_005a.bat

Ohne Angabe weiterer Parameter und mit der Auslassung des Parameters /A werden die Besitzrechte standardmäßig auf den Benutzer übertragen, der den Befehl gerade ausgeführt. Wenn Sie also gerade als *AdmInstall* angemeldet sind und die Besitzrechte auf den Benutzer *TestUser1* übertragen möchten, dann führen Sie den *Takeown*-Befehl einfach unter Verwendung des entsprechenden Benutzers aus, indem Sie mit dem Parameter /U den Benutzernamen und mit /P das dazugehörige Kennwort übergeben:

```
@Echo off
Takeown /F C:\LogFiles /u FRA-MD-FPS-001\TestUser1 _
/p 123QWEasd /r
```

Listing 156: 2_02_005b.bat

Die Lösung mit VBScript

Dank WMI lässt sich der Besitzer einer Datei oder eines Ordners auch mit VBScript auf sehr einfache Art und Weise ändern. Denn die WMI-Klassen `CIM_DataFile` und `Win32_Directory`, die Informationen über Dateien und Ordner besitzen, stellen die `TakeOwnership`-Methode bereit, die Sie nur gegen das betreffende Datei- oder Ordner-Objekt anwenden müssen, um den Besitz für den aktuellen Benutzer zu übernehmen.

Wenn Sie z.B. den Besitz für die Datei `C:\LogFiles\Protokoll.txt` übernehmen möchten, so wenden Sie die `ExecQuery`-Abfrage an, um die betreffende Datei unter Angabe ihres Namens zu selektieren. Das Ergebnis wird daraufhin als Kollektion zurückgeliefert, und Sie können innerhalb einer `For Each`-Schleife die `TakeOwnership`-Methode gegen das betreffende Datei-Objekt anwenden.

```
Option Explicit
```

```
Dim oWMI
Dim colFiles
Dim objItem
Dim strComputer
```

```
strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colFiles = oWMI.ExecQuery ("SELECT * FROM _
    CIM_Datafile WHERE Name = 'C:\\LogFiles\\Protokoll.txt'")
```

```
For Each objItem in colFiles
    Err = objItem.TakeOwnership
Next
```

```
Set oWMI = Nothing
```

Listing 157: 2_02_005a.vbs

Mit dem gleichen Skript können Sie natürlich auch den Besitz eines Ordners übernehmen, wobei Sie hier aber die `Win32_Directory`-Klasse verwenden und die `TakeOwnership`-Methode gegen das Ordner-Objekt richten müssen.

```
Option Explicit
```

```
Dim oWMI
Dim colFolder
Dim objItem
Dim strComputer
```

Listing 158: 2_02_005b.vbs

```

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colFolder = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Directory WHERE Name = 'C:\\LogFiles'")

For Each objItem in colFolder
    Err = objItem.TakeOwnership
Next

Set oWMI = Nothing

```

Listing 158: 2_02_005b.vbs (Forts.)

32 Einen Dump der Datei- und Ordnerberechtigungen erstellen

Wenn Sie Änderungen an den Datei- und Ordnerberechtigungen vornehmen möchten (oder bereits vorgenommen haben) und die aktuellen Einstellungen dokumentieren möchten, so können Sie einen Dump der Berechtigungen und damit ein Abbild bzw. einen Report der Einträge in den Zugriffskontrolllisten erstellen, womit sich die ursprünglichen Berechtigungen leichter wiederherstellen lassen. Ein solcher Dump ist auch für die Fehlersuche in den Sicherheitseinstellungen von Vorteil, da die einzelnen Einträge in den Zugriffskontrolllisten und auch die angewandte Vererbung von Berechtigungen meist recht übersichtlich dargestellt werden und daher sehr gut mit anderen Einstellungen verglichen werden können.

Der Windows Server 2003 hält unter grafischer Oberfläche jedoch keine Möglichkeiten bereit, womit Sie einen solchen Dump per GUI erstellen können, dies wird nur von der Eingabeaufforderung aus (siehe nächster Abschnitt) unterstützt. Wenn Sie jedoch des Öfteren einen Dump der Berechtigungen benötigen und diesen vorzugsweise unter grafischer Oberfläche erstellen wollen, so empfiehlt sich der Einsatz des Freeware-Tools *DumpSec* der Fa. Somarsoft, das Sie (nach einer kurzen Registrierung) kostenlos von der Internetseite www.somarsoft.com herunterladen und sofort einsetzen können. Denn mit diesem Tool lassen sich auf recht einfache Art und Weise die Berechtigungen für Dateien, Ordner, Registrierungseinträge, Drucker und Freigaben erstellen und so sehr gut dokumentieren.

Wenn Sie z.B. die Berechtigungen für alle Dateien und Ordner eines bestimmten Laufwerks dokumentieren möchten, so starten Sie das Tool *DumpSec* aus dem START-MENÜ heraus und wählen im Menü REPORT den Eintrag DUMP PERMISSIONS FOR FILE SYSTEM... Sie gelangen daraufhin in das Dialogfenster SELECT DIRECTORY TREE, worüber Sie das Laufwerk (oder wahlweise auch einen Ordner) auswählen können. Markieren Sie das betreffende Laufwerk, und klicken Sie auf OK, woraufhin die Berechtigungen für alle Dateien und Ordner des betreffenden Laufwerks abgefragt und anschließend im Dialogfenster SOMARSOFT DUMPSEC angezeigt werden.

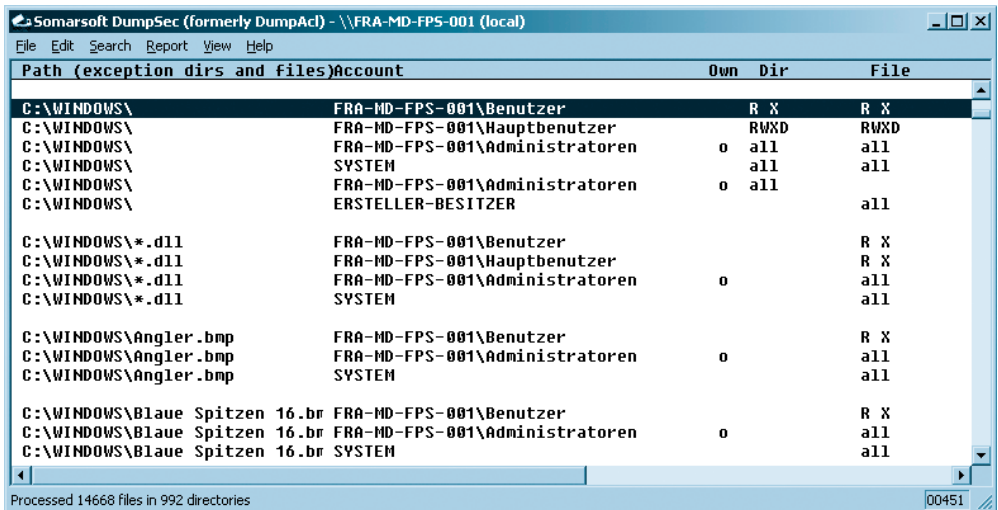


Abbildung 83: Einen Dump der Berechtigungen erstellen

Die Anzeige der einzelnen Berechtigungseinträge untergliedert sich in die folgenden fünf Spalten, die folgende Angaben enthalten:

- ▶ Path: Die erste Spalte zeigt den Pfad und auch den Namen der Datei bzw. des Ordners, für die bzw. für den die Berechtigungen angezeigt werden.
- ▶ Account: In der zweiten Spalte mit der Bezeichnung ACCOUNT werden die Benutzer- und Gruppennamen angezeigt, die in den Zugriffskontrolllisten enthalten sind.
- ▶ Own: Die dritte Spalte gibt Auskunft über den jeweiligen Besitzer des Datei- oder Ordner-Objektes, wobei der Besitzer mit einem o gekennzeichnet ist.
- ▶ Dir: In der vorletzten Spalte werden die jeweiligen Berechtigungen (siehe nachfolgende Tabelle) für einen Ordner angezeigt.
- ▶ File: Die letzte Spalte gibt Auskunft über die jeweiligen Berechtigungen einer Datei.

Die nachfolgende Tabelle zeigt, wie die einzelnen Berechtigungs-Flags zu interpretieren sind:

Flag	Beschreibung
R	Benutzer oder Gruppe kann Datei lesen bzw. den Inhalt eines Ordners auflisten.
W	Benutzer oder Gruppe kann Dateien schreiben bzw. in einem Ordner weitere Unterordner und Dateien erstellen.
X	Benutzer oder Gruppe kann die Datei ausführen bzw. die Ordnerstruktur durchlaufen.
D	Benutzer oder Gruppe kann die Datei bzw. den Ordner löschen.

Tabelle 32: Die Flags der DumpSec-Berechtigungen

Flag	Beschreibung
P	Benutzer oder Gruppe kann die Berechtigungen für die Datei bzw. für den Ordner ändern.
O	Benutzer oder Gruppe kann den Besitzer einer Datei bzw. eines Ordners ändern.
All	Benutzer oder Gruppe hat Vollzugriff auf die Datei bzw. den Ordner.
No access	Benutzer oder Gruppe hat keinerlei Zugriffsrechte auf die Datei bzw. den Ordner.

Tabelle 32: Die Flags der DumpSec-Berechtigungen (Forts.)

Ein mit *DumpSec* erzeugter Report kann jederzeit gespeichert und damit für Dokumentationszwecke weiterverwendet werden. Das Speichern des Reports nehmen Sie im Menü FILE vor, indem Sie den Eintrag SAVE REPORT AS... auswählen und die Export-Datei festlegen.

Die Lösung per Eingabeaufforderung

Wenn Sie die Berechtigungen von der Eingabeaufforderung aus abfragen möchten, so können Sie im einfachsten Fall den Befehl *CacIs* verwenden, der standardmäßig im Befehlsumfang von Windows Server 2003 enthalten ist. Der *CacIs*-Befehl erwartet dabei zumindest die Angabe der Datei (Platzhalter und Wildcards können verwendet werden), für die Sie die Berechtigungen auslesen möchten, als Parameter.

Um die Berechtigungen für alle Dateien im aktuellen Ordner abzufragen und per Ausgabeumleitung in die Datei *Result.txt* zu schreiben, können Sie die folgende *CacIs*-Befehlszeile verwenden.

```
@Echo off
CacIs *.* > Result.txt
```

Listing 159: 2_02_006a.bat

Mit der nachfolgenden *CacIs*-Befehlszeile lesen Sie dagegen die Berechtigungen für den Ordner *C:\Windows* aus und schreiben das Ergebnis der Abfrage in die Datei *Result.txt*.

```
@Echo off
CacIs C:\Windows > Result.txt
```

Listing 160: 2_02_006b.bat

Eine weitere Möglichkeit, die Berechtigungen von Dateien und Ordner abzufragen, bietet das Tool *Subinacl.exe*, das Bestandteil des *Windows Server 2003 Resource Kits* ist (siehe Seite 252).

Wenn Sie beispielsweise mit `Subinacl` die Berechtigungen für alle Dateien im aktuellen Ordner abfragen und in die Datei *Result.txt* schreiben möchten, so wenden Sie den `Subinacl`-Befehl mit dem Parameter `/Files` an und übergeben den Namen der Dateien, für die Sie die Berechtigungen abfragen möchten. Im Falle aller Dateien des aktuellen Ordners können Sie `*.*` verwenden. Des Weiteren legen Sie mit `/Display=Dacl` fest, dass die Zugriffskontrolllisten (DACL) der zuvor ausgewählten Dateien nur angezeigt werden sollen, und bestimmen zudem die zu verwendende Ausgabeumleitung.

```
@Echo off
Subinacl /Files *.* /Display=Dacl > Result.txt
```

Listing 161: 2_02_006c.bat

Möchten Sie dagegen einen vollständigen Dump aller Dateien und Ordner des Laufwerks C erzeugen, so verwenden Sie statt des Parameters `/Files` den Parameter `/Subdirectories`, da mit diesem Parameter die komplette Ordnerstruktur und auch die darin enthaltenen Dateien berücksichtigt werden.

```
@Echo off
Subinacl /Subdirectories C:\*.* /Display=Dacl > Result.txt
```

Listing 162: 2_02_006d.bat

Die Lösung mit VBScript

Auf welche Art und Weise sich die Datei- und Ordnerberechtigungen per VBScript abfragen lassen, wurde bereits in Rezept Welche Berechtigung ist auf die Datei oder auf den Ordner vergeben? auf Seite 240) demonstriert. Sicherlich lässt sich dieses Skript auch auf mehrere (oder wahlweise auch alle) Dateien und Ordner eines bestimmten Laufwerks anwenden.

Weitaus effizienter lässt sich ein Dump der Berechtigungen aber mit Hilfe der `Run`-Methode des `Shell`-Objektes erstellen, da Sie hiermit den `Caccls`- und `Subinacl`-Befehl der Eingabeaufforderung (siehe vorheriger Abschnitt) auch aus einem VBScript heraus aufrufen und mit nur fünf Code-Zeilen ebenfalls ein sehr zuverlässiges Ergebnis erzielen können.

Das nachfolgende Code-Beispiel generiert mit Hilfe des `Caccls`-Befehls einen Report (*Result.txt*) der Berechtigungen für alle Dateien, die sich im aktuellen Ordner befinden.

```
Option Explicit
Dim oShell
```

Listing 163: 2_02_006a.vbs

```
Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Cacls *.* > Result.txt"

Set oShell = Nothing
```

Listing 163: 2_02_006a.vbs (Forts.)

Mit den folgenden Code-Zeilen generieren Sie ebenfalls einen Report (*Result.txt*) der Dateiberechtigungen für den aktuellen Ordner, nur dass dieses Mal statt des `Cacls`-Befehls der `Subinacl`-Befehl verwendet wird.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Subinacl /Files *.* /Display=Dacl > Result.txt"

Set oShell = Nothing
```

Listing 164: 2_02_006b.vbs

Um einen vollständigen Report aller Dateien und Ordner des Laufwerks `C` zu generieren, können Sie das nachfolgende Listing verwenden, das den `Subinacl`-Befehl unter Angabe des Parameters `/Subdirectories` für eine rekursive Abfrage ausführt.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Subinacl /Subdirectories C:\*.* /Display=Dacl _
> Result.txt"

Set oShell = Nothing
```

Listing 165: 2_02_006c.vbs

33 Die aktuelle Dateiversion ermitteln

Hin und wieder kommt es vor, dass mit der Installation von Software-Produkten (Programme, Service Packs, Resource Kits etc.) wichtige System- und vielleicht auch Treiberdateien eingerichtet oder durch fehlerhafte Versionen ersetzt werden, womit sich bestimmte Funktionen erst gar nicht aufrufen lassen oder schlimmstenfalls das Betriebssystem an Stabilität verliert.

Bestes Beispiel dafür ist das Tool *Subinacl.exe*, das Bestandteil des *Windows Server 2003 Resource Kits* ist. Wenn Sie das Resource Kit von der Internetseite *www.microsoft.com* herunterladen und installieren, so werden Sie feststellen, dass sich mit dem *Subinacl*-Tool viele Sicherheitsinformationen partout nicht auslesen lassen. Grund dafür ist, dass die Version des *Subinacl*-Tools, das im Resource Kit enthalten ist, einen Bug (oder auch mehrere) enthält und bestimmte Abfragen erst gar nicht durchführt. Erst wenn Sie das *Subinacl*-Tool in aktualisierter Form von der Internetseite *www.microsoft.com* herunterladen und einsetzen, lassen sich alle Sicherheitsinformationen, für die das Tool entsprechende Parameter bereitstellt, problemlos abfragen.

Bei Problemen, die durch fehlerhafte Programm-, System- oder Treiberdateien verursacht werden, ist nicht selten ein aufwändiges Troubleshooting angesagt, und es gilt zunächst, die fehlerhafte Datei ausfindig zu machen und nach möglichen Fehlerursachen zu recherchieren. In solchen Fällen ist es von großem Vorteil, wenn die Datei über eine Versionsinformation verfügt, anhand derer sich die installierte Dateiversion ermitteln lässt. Denn mit Hilfe der Dateiversion ist es in der Regel weitaus einfacher, im Internet nach bekannten Bugs zu suchen und eine schnelle Lösung (z.B. durch Austausch durch eine bestimmte Dateiversion) herbeizuführen.

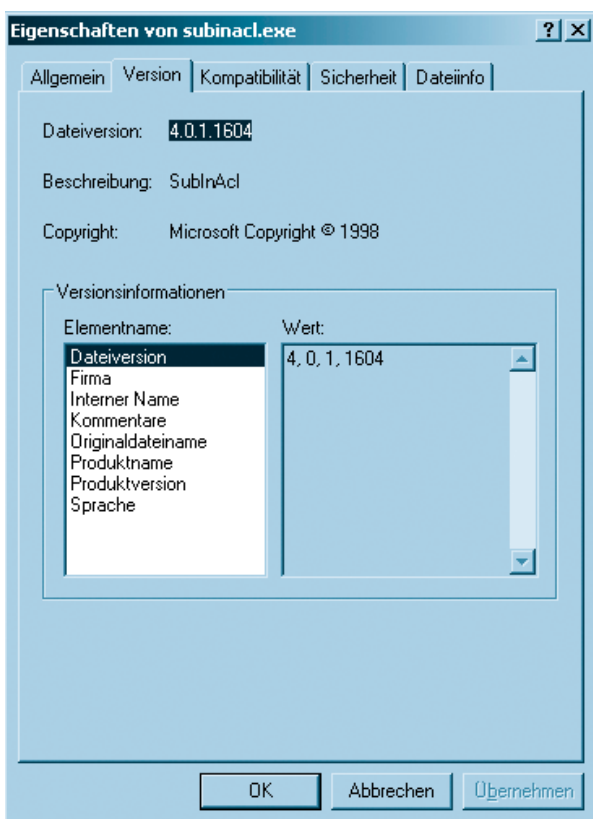


Abbildung 84: Die Version einer Datei ermitteln

Leider verfügen nicht alle Dateien über eine Versionsinformation. Die meisten Programm-, System- und Treiberdateien (insbesondere die aus dem Hause Microsoft) geben in den Datei-Eigenschaften jedoch Auskunft darüber, in welcher Version die Datei gerade vorliegt. Wenn Sie z.B. das *Windows Server 2003 Resource Kit* eingerichtet haben und die Version des *Subinac1*-Tools prüfen möchten, so starten Sie zunächst den WINDOWS EXPLORER und markieren die betreffende Datei. Anschließend rufen Sie aus dem Kontextmenü der Datei EIGENSCHAFTEN auf und wechseln auf die Registerkarte VERSION (diese Registerkarte ist nur vorhanden, wenn die Datei Versionsinformationen enthält). Dort wird die aktuelle Version angezeigt. Sollte das *Subinac1*-Tool in der Version 4.0.1.1604 vorliegen, so wird es bei der *Subinac1*-Befehlsausführung zu Problemen kommen, da diese Dateiversion fehlerhaft ist. Tauschen Sie die Datei gegen die Version 5.2.3790.1180 aus, um das Tool fehlerfrei einsetzen zu können.

Die Lösung per Eingabeaufforderung

Mit Hilfe einer WMI-Abfrage lässt sich die Version einer Datei (sofern angegeben) auch von der Eingabeaufforderung aus ermitteln. Hierzu wenden Sie den WMIC-Befehl unter Angabe des Alias *DataFile* an, womit Sie eine Verbindung zur WMI-Klasse *CIM_DataFile* herstellen und die Eigenschaften eines Datei-Objektes abfragen können, so zum Beispiel die Eigenschaft *Version*.

Um nun die Version der Datei *Subinac1.exe* abzufragen, die nach der Installation des *Windows Server 2003 Resource Kits* per Default im Ordner *C:\Programme\Windows Resource Kits\Tools* abgelegt wird, wenden Sie den WMIC-Befehl mit dem Alias *DataFile* an, selektieren mit der *Where*-Klausel die genannte Datei und lesen mit der *Get*-Anweisung den Wert der Eigenschaft *Version* aus.

```
@Echo off
WMIC DataFile Where "Name='C:\\Programme\\Windows Resource _
    Kits\\Tools\\Subinac1.exe'" Get Version
```

Listing 166: 2_02_007.bat

Die Lösung mit VBScript

Um die Version einer Datei per VBScript zu ermitteln, gehen Sie genauso vor wie in der Eingabeaufforderung. Sprich: Sie stellen eine Verbindung zur WMI-Klasse *CIM_DataFile* her, selektieren mit der *Where*-Klausel das betreffende Datei-Objekt und fragen den Wert der Eigenschaft *Version* ab.

Das nachfolgende Listing gibt die Version der Datei *Subinac1.exe*, die (nach der Installation des *Windows Server 2003 Resource Kits*) im Ordner *C:\Programme\Windows Resource Kits\Tools* abgelegt ist, mit der *WScript.Echo*-Anweisung am Bildschirm aus.

```

Option Explicit

Dim oWMI
Dim strComputer
Dim colFile
Dim objItem

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colFile = oWMI.ExecQuery ("Select * From CIM_DataFile _
    Where Name = 'C:\\Programme\\Windows Resource Kits\\Tools\\Subinac1.exe'")

For Each objItem in colFile
    Wscript.Echo "Version: " & objItem.Version
Next

Set oWMI = Nothing

```

Listing 167: 2_02_007.vbs

34 Datei- und Ordnergröße ermitteln

Anhand der Eigenschaften einer Datei oder auch eines kompletten Ordners können Sie problemlos die Größe des Objektes wie auch den verwendeten Speicherplatz auf dem Datenträger, ermitteln. Wenn Sie z.B. herausfinden müssen, welche Größe das Benutzerprofil des *Administrators* hat, dann rufen Sie den WINDOWS EXPLORER auf, markieren im Ordner *C:\Dokumente und Einstellungen\Administrator* die Datei *Ntuser.dat* und rufen aus dem Kontextmenü der Datei die EIGENSCHAFTEN auf. Auf der Registerkarte ALLGEMEIN finden Sie zum einen die Größe der Datei (die Angabe erfolgt in Bytes), zum anderen aber auch den Speicherplatz, den die Datei auf dem Datenträger verwendet.

In der Regel weichen die Werte GRÖSSE und GRÖSSE AUF DEM DATENTRÄGER voneinander ab, da eine Datei, deren Größe die Cluster-Größe des Dateisystems überschreitet, in mehrere Fragmente aufgeteilt und gespeichert wird. Hierdurch werden mehrere Cluster belegt und damit auch mehr Speicherplatz verbraucht.

Wenn Sie dagegen die Größe eines Ordners ermitteln möchten, so markieren Sie den Ordner und rufen aus dem Kontextmenü die EIGENSCHAFTEN auf. Auf der Registerkarte ALLGEMEIN finden Sie die Anzahl der Objekte, die sich in dem Ordner befinden, wie auch die Größe des Ordners selbst, dessen Unterordner und auch den verwendeten Speicherplatz auf dem Datenträger.

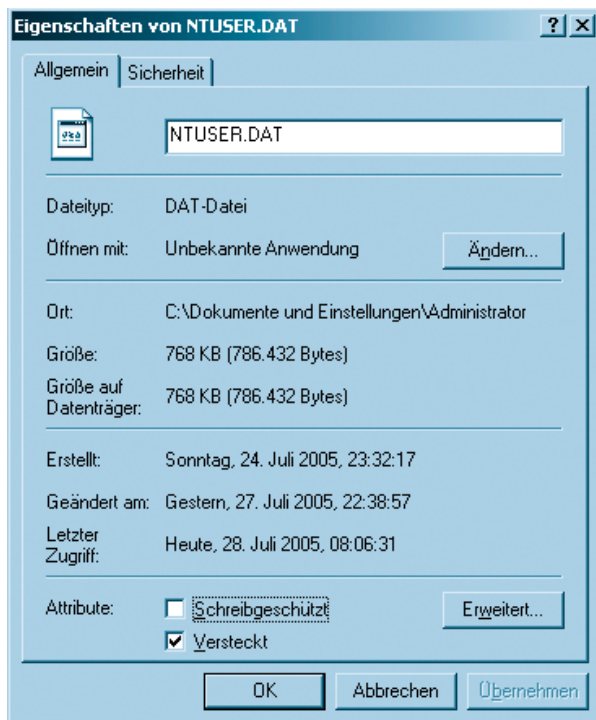


Abbildung 85: Die Größe einer Datei ermitteln

Die Lösung per Eingabeaufforderung

Die Größe einer Datei oder eines Ordners lässt sich natürlich auch von der Eingabeaufforderung aus ermitteln. Hierfür verwenden Sie im einfachsten Fall den `Dir`-Befehl, den Sie normalerweise zum Anzeigen des Inhalts eines bestimmten Pfades verwenden. Der `Dir`-Befehl zeigt aber nicht nur den Inhalt eines Ordners an, sondern gibt in der Zusammenfassung auch die Größe des Objektes wie auch den letzten Änderungszeitpunkt aus.

Um beispielsweise die Größe des Benutzerprofils für den Administrator zu ermitteln, wenden Sie den `Dir`-Befehl an und übergeben den Pfad der Datei wie auch den eigentlichen Dateinamen. Und da der `Dir`-Befehl per Default keine versteckten Dateien und Systemdateien anzeigt, erzwingen Sie die Anzeige der versteckten Datei *Ntuser.dat*, indem Sie den Parameter `/a:h` übergeben, womit auch Dateien mit dem Attribut *Versteckt* berücksichtigt werden.

```
@Echo off
```

```
Dir "C:\Dokumente und Einstellungen\Administrator\_  
Ntuser.dat" /a:h
```

Listing 168: 2_02_008a.bat

Auf gleiche Art und Weise können Sie auch die Größe eines Ordners ermitteln. Wenn Sie z.B. herausfinden müssen, wie groß der Ordner *C:\LogFiles* ist, dann setzen Sie den *Dir*-Befehl ein und übergeben den entsprechenden Ordnernamen:

```
@Echo off
Dir C:\LogFiles
```

Listing 169: 2_02_008b.bat

Der *Dir*-Befehl gibt daraufhin den Inhalt der ersten untergeordneten Ebene des angegebenen Ordners aus und zeigt in der Zusammenfassung die Anzahl der Objekte, den letzten Änderungszeitpunkt wie auch die Größe des Ordners an.

Die Lösung mit VBScript

Per VBScript lässt sich die Größe einer Datei oder eines Ordners im einfachsten Fall mit dem *FileSystem*-Objekt ermitteln, da Sie mit der *GetFile*- bzw. *GetFolder*-Methode nur die betreffende Datei oder den Ordner binden und dann den Wert der Eigenschaft *Size* auslesen müssen, um die gewünschte Information zu erhalten.

Das nachfolgende Code-Beispiel zeigt, mit welch wenigen Code-Zeilen Sie die Größe der Datei *C:\LogFiles\Protokoll.txt* ermitteln können. Zunächst wird mit der *CreateObject*-Methode eine neue Instanz des *FileSystem*-Objektes erzeugt und die betreffende Datei mit der *GetFile*-Methode an eine Objektvariable gebunden. Anschließend wird der Wert der Eigenschaft *Size* ausgelesen und mit der *WScript.Echo*-Anweisung am Bildschirm ausgegeben.

```
Option Explicit

Dim oFS
Dim oFile
Dim strFile

strFile = "C:\LogFiles\Protokoll.txt"

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFile = oFS.GetFile(strFile)
Wscript.Echo oFile.Size & " Bytes"

Set oFS = Nothing
Set oFile = Nothing
```

Listing 170: 2_02_008a.vbs

Auf die gleiche Art und Weise lässt sich auch die Größe eines Ordners ermitteln. Allerdings nehmen Sie die Bindung des Ordner-Objektes nicht mit `GetFile`, sondern mit `GetFolder` vor:

Option Explicit

Dim oFS

Dim oFolder

Dim strFolder

strFolder = "C:\LogFiles"

Set oFS = CreateObject("Scripting.FileSystemObject")

Set oFolder = oFS.GetFolder(strFolder)

Wscript.Echo oFolder.Size & " Bytes"

Set oFS = Nothing

Set oFolder = Nothing

Listing 171: 2_02_008b.vbs

35 Dateien ab einer bestimmten Größe finden

In manchen Fällen kann es sehr hilfreich sein, den Datenträger nach Dateien ab einer bestimmten Größe zu durchsuchen. Wenn der Datenträger beispielsweise sehr ausgelastet ist und Sie auf die Schnelle freien Speicherplatz schaffen müssen, so können Sie nach übergroßen Dateien suchen und durch deren Auslagerung auf einen anderen Datenträger oder auf ein Netzlaufwerk wertvollen Plattenplatz gewinnen.

Wenn Sie z.B. alle Dateien mit einer Dateigröße ab 100 MByte auflisten möchten, so rufen Sie unter grafischer Oberfläche mit F3 die Such-Funktion von Windows Server 2003 auf und klicken in der linken Fensterhälfte auf die Such-Option WIE GROSS IST DIE DATEI? Aktivieren Sie die Option GRÖSSENANGABE IN KB, und legen Sie den Wert 102.400 (100 MByte * 1024 = 102.400 KByte) fest. Im Feld SUCHEN IN wählen Sie den gewünschten Suchbereich (*Arbeitsplatz, Lokale Laufwerke ...*) aus und klicken anschließend auf *Suchen*, um mit der Suche nach den Dateien zu beginnen. Das Ergebnis der Suche wird daraufhin in der rechten Fensterhälfte unter Angabe des Dateinamens, des Pfades, der Größe, des Typs, des letzten Bearbeitungszeitpunkts und auch der Attribute angezeigt (siehe Abbildung 86).

Die Lösung per Eingabeaufforderung

Um von der Eingabeaufforderung aus nach Dateien ab einer bestimmten Größe zu suchen, verwenden Sie der Einfachheit halber den `WMIC`-Befehl mit dem Alias `Data-File`, womit Sie eine Abfrage gegen die WMI-Klasse `CIM_DataFile` durchführen, Dateien nach vielen Kriterien selektieren und Informationen zu den einzelnen Dateien abfragen können.

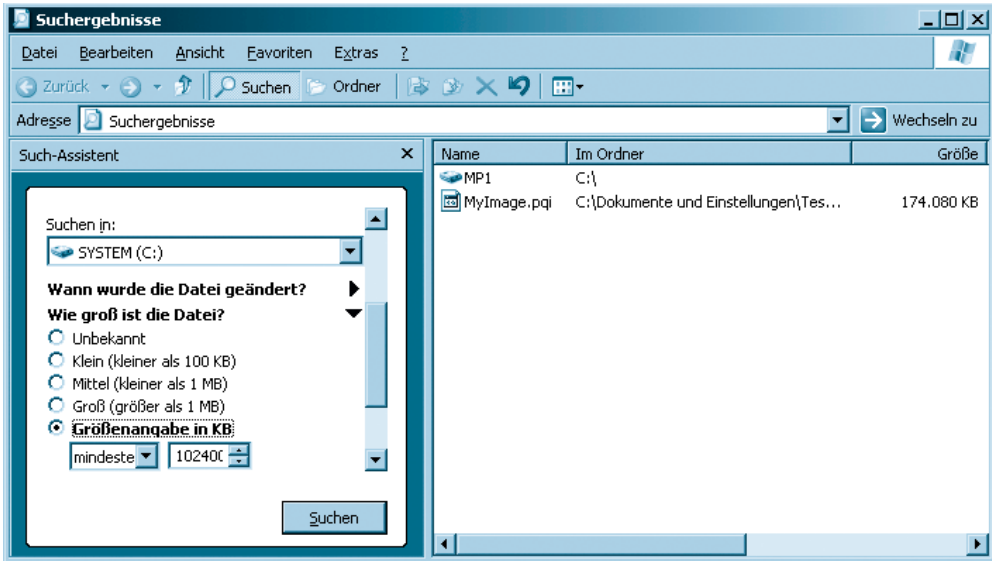


Abbildung 86: Nach Dateien mit einer bestimmten Größe suchen

Wenn Sie beispielsweise eine Aufstellung aller Dateien erzeugen möchten, die eine Größe von 100 MByte oder mehr haben, so wenden Sie den WMIC-Befehl unter Angabe des Alias `DataFile` an, selektieren die Dateien mit der `Where`-Klausel nach ihrer Größe (Angabe erfolgt in Bytes) und fragen z.B. die Werte der Eigenschaften `Name` (Dateiname) und `FileSize` (Dateigröße) ab.

```
@Echo off
WMIC DataFile Where "FileSize > '104857600'" Get Name, FileSize
```

Listing 172: 2_02_009.bat

Die Lösung mit VBScript

Auch mit VBScript gestaltet sich die Suche nach Dateien ab einer bestimmten Größe sehr einfach, da Sie – wie auch in der Eingabeaufforderung – nur eine Verbindung zur WMI-Klasse `CIM_DataFile` herstellen, die Dateien mit der `Where`-Klausel selektieren und die gewünschten Eigenschaften abfragen müssen, um eine Aufstellung der Dateien zu erhalten.

Das nachfolgende Listing zeigt, wie Sie eine Aufstellung aller Dateien erzeugen können, die eine Größe von 100 MByte oder mehr haben. Hierzu wird eine `ExecQuery`-Abfrage gegen die WMI-Klasse `CIM_DataFile` durchgeführt und eine Selektion aller Dateien vorgenommen, deren Größe 104.857.600 Bytes (100 Byte * 1024 * 1024) oder mehr beträgt. Die `ExecQuery`-Abfrage liefert das Ergebnis (die Dateien, die dem Suchkriterium entsprechend) daraufhin in Form einer Kollektion zurück. Mit Hilfe einer

For Each-Schleife können dann die Werte der Eigenschaften Name (Dateiname) und FileSize (Dateigröße) für jedes Element abgefragt und mit der WScript.Echo-Anweisung am Bildschirm ausgegeben werden.

Option Explicit

```
Dim oWMI
Dim strComputer
Dim colFile
Dim objItem
Dim intSize
```

```
strComputer = "."
intSize = 104857600
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colFile = oWMI.ExecQuery ("Select * From CIM_DataFile _
    Where FileSize > '" & intSize & "'")
```

```
For Each objItem in colFile
    Wscript.Echo "Filename: " & objItem.Name
    Wscript.Echo "Size: " & objItem.FileSize
Next
```

```
Set oWMI = Nothing
```

Listing 173: 2_02_009.vbs

36 Wann wurde die Datei oder der Ordner erstellt?

Wenn Sie den genauen Erstellungszeitpunkt einer Datei oder eines Ordners herausfinden müssen, weil Sie vielleicht zwei Datei- oder Ordnerversionen miteinander vergleichen wollen, so brauchen Sie nur die Eigenschaften der Datei oder des Ordners zu öffnen. Denn auf der Registerkarte ALLGEMEIN sind das Datum und auch die genaue Uhrzeit zu finden, wann das Objekt auf dem Dateisystem angelegt wurde.

Um beispielsweise herauszufinden, zu welchem Zeitpunkt die Systemdatei *Services.exe* erstellt wurde, rufen Sie den WINDOWS EXPLORER auf, markieren die Datei und rufen im Kontextmenü die EIGENSCHAFTEN auf. Auf der Registerkarte ALLGEMEIN finden Sie die Eigenschaft ERSTELLT und rechts daneben den genauen Erstellungszeitpunkt.

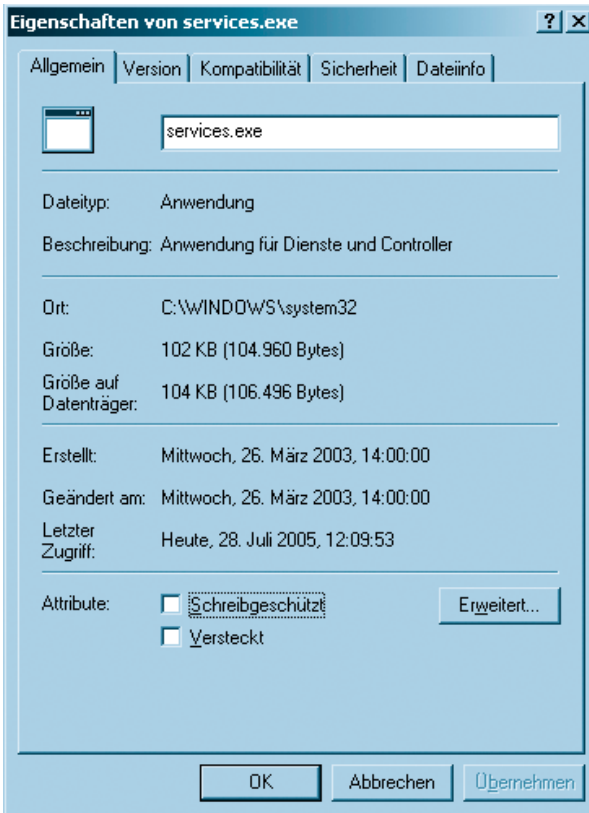


Abbildung 87: Den Erstellungszeitpunkt einer Datei anzeigen lassen

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung lässt sich der Erstellungszeitpunkt einer Datei oder eines Ordners am schnellsten mit dem `Dir`-Befehl ermitteln, den Sie in Verbindung mit dem Parameter `/t:c` gegen die betreffende Datei oder gegen den Ordner anwenden. Denn der Parameter `/t:c` legt fest, dass die Aufstellung, die am Ende der Datei-/Ordner-Auflistung angezeigt wird, nicht unter Angabe des letzten Änderungszeitpunktes (`/t:w`) erfolgt, sondern mit dem Erstellungszeitpunkt. Wenn Sie also herausfinden wollen, wann die Datei *Protokoll.txt*, die im Ordner *C:\LogFiles* abgelegt ist, erstellt wurde, so geben Sie folgende Befehlszeile ein:

```
@Echo off
Dir C:\LogFiles\Protokoll.txt /t:c
```

Listing 174: 2_02_010a.bat

Eine weitere Möglichkeit, den Erstellungszeitpunkt einer Datei oder eines Ordners zu ermitteln, bietet die WMIC-Abfrage. Um beispielsweise zu ermitteln, wann die Datei

Protokoll.txt im Ordner *C:\Logfiles* erstellt wurde, führen Sie eine Abfrage gegen den Alias *DataFile* aus, selektieren dabei alle Dateien, deren Name *C:\\LogFiles\\Protokoll.txt* entspricht, und lassen sich mit der *Get*-Anweisung den Wert der Eigenschaft *CreateDate* anzeigen.

```
@Echo off
WMIC DataFile WHERE _
    "Name = 'C:\\LogFiles\\Protokoll.txt'" _
    Get CreateDate
```

Listing 175: 2_02_010b.bat

Die gleiche Abfrage können Sie auch verwenden, um den Erstellungszeitpunkt eines Ordners zu ermitteln, allerdings müssen Sie die Abfrage gegen den Alias *FSDir* richten und nicht gegen *DataFile*:

```
@Echo off
WMIC FSDir WHERE _
    "Name = 'C:\\LogFiles'" Get CreateDate
```

Listing 176: 2_02_010c.bat

Die Lösung mit VBScript

Wenn Sie den Erstellungszeitpunkt einer Datei oder eines Ordners mit VBScript ermitteln möchten, so können Sie hierfür das *FileSystem*-Objekt verwenden und die Eigenschaft *DateCreated* auslesen, oder aber Sie führen – ähnlich der Lösung in der Eingabeaufforderung – eine WMI-Abfrage gegen die Klasse *CIM_DataFile* (zur Abfrage einer Datei) oder *Win32_Directory* (zur Abfrage eines Ordners) durch, filtern das betreffende Objekt heraus und fragen den Wert der Eigenschaft *CreationDate* ab.

Um beispielsweise den Erstellungszeitpunkt der Datei *C:\LogFiles\Protokoll.txt* mit Hilfe des *FileSystem*-Objektes zu ermitteln, erzeugen Sie zunächst eine neue Instanz des *FileSystem*-Objektes und binden mit der *GetFile*-Methode die betreffende Datei an eine Objektvariable. Anschließend können Sie den Wert der Eigenschaft *DateCreated* auslesen, woraufhin der Erstellungszeitpunkt im Format *dd.mm.yyyy hh:mm* zurückgeliefert wird.

```
Option Explicit

Dim oFS
Dim oFile
Dim strFile
```

Listing 177: 2_02_010a.vbs

```

strFile = "C:\LogFiles\Protokoll.txt"

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFile = oFS.GetFile(strFile)
Wscript.Echo oFile.DateCreated

Set oFS = Nothing
Set oFile = Nothing

```

Listing 177: 2_02_010a.vbs (Forts.)

Das nachfolgende Code-Beispiel löst die gleiche Aufgabe mit Hilfe einer WMI-Abfrage, indem zunächst alle Dateien selektiert werden, deren Name gleich C:\\LogFiles\\Protokoll.txt ist. Das Ergebnis dieser Abfrage kann innerhalb einer For Each-Schleife weiterverarbeitet werden. In diesem Fall wird der Wert der Eigenschaft CreationDate ermittelt, der den Erstellungszeitpunkt der Datei im UTC-Format enthält.

```

Option Explicit

Dim oWMI
Dim colFile
Dim objItem
Dim strComputer
Dim strFile

strComputer = "."
strFile = "'C:\\LogFiles\\Protokoll.old'"

Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\\root\\cimv2")
Set colFile = oWMI.ExecQuery ("SELECT * FROM _
    CIM_Datafile WHERE Name = " & strFile)

For Each objItem in colFile
    WScript.Echo objItem.CreationDate
Next

Set oWMI = Nothing

```

Listing 178: 2_02_010b.vbs

Mit der gleichen Abfrage können Sie auch den Erstellungszeitpunkt eines Ordners ermitteln, wobei Sie die Abfrage aber gegen die Klasse Win32_Directory ausführen müssen:

```

Option Explicit

Dim oWMI
Dim colFolder
Dim objItem
Dim strComputer
Dim strFolder

strComputer = "."
strFolder = "C:\\LogFiles"

Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colFolder = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Directory WHERE Name = " & strFolder)

For Each objItem in colFolder
    WScript.Echo objItem.CreationDate
Next

Set oWMI = Nothing

```

Listing 179: 2_02_010c.vbs

37 Wann wurde die Datei oder der Ordner zuletzt geändert?

Ähnlich wie Sie den Erstellungszeitpunkt einer Datei oder eines Ordners ermitteln können, so können Sie auch den Zeitpunkt der letzten Änderung herausfinden. Wenn Sie z.B. wissen möchten, wann der Ordner *LogFiles* zuletzt umbenannt oder inhaltlich verändert wurde, so brauchen Sie nur die **DETAIL**-Ansicht im **WINDOWS EXPLORER** zu öffnen.

Rufen Sie hierfür den **WINDOWS EXPLORER** auf, und wählen Sie im Menü **ANSICHT** den Eintrag **DETAILS** aus. Daraufhin werden in der rechten Fensterhälfte die Eigenschaften *Name*, *Größe*, *Typ*, *Geändert am* und *Attribute* des jeweiligen Objektes angezeigt. Der Spalte **GEÄNDERT AM** können Sie nun den letzten Änderungszeitpunkt entnehmen (siehe Abbildung 88).

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung können Sie sich den letzten Änderungszeitpunkt einer Datei oder wahlweise auch eines Ordners mit dem **Dir**-Befehl anzeigen lassen, den Sie normalerweise zum Auflisten von Dateien und Ordnern verwenden. Der **Dir**-Befehl gibt am Ende der Auflistung eine kurze Zusammenfassung über die Auflistung aus, womit der Datei-/Ordnername, dessen Größe und auch der letzte Änderungszeitpunkt ausgegeben werden.

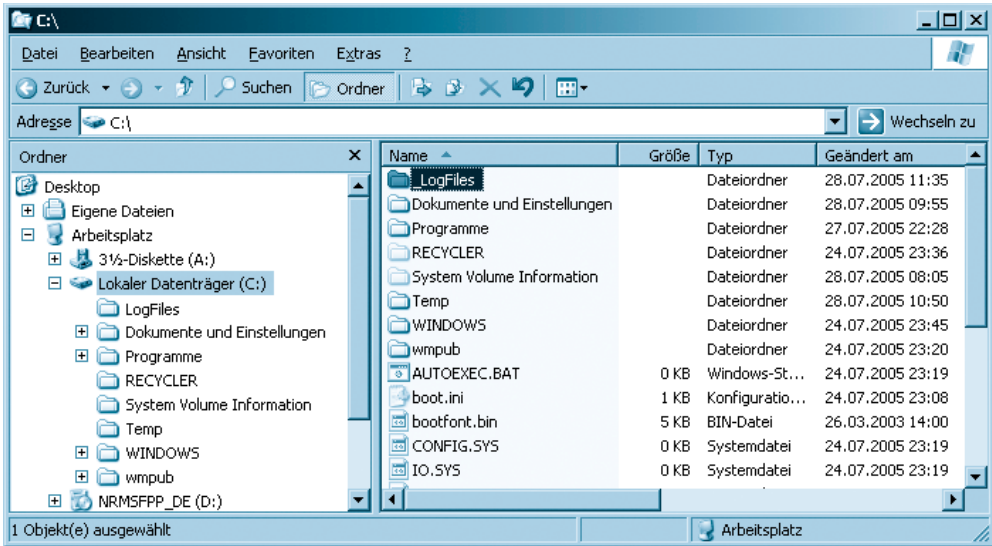


Abbildung 88: Die Detail-Ansicht des Windows Explorers zeigt den letzten Änderungszeitpunkt.

Wenn Sie also wissen möchten, wann die Datei *Services.exe* zuletzt geändert wurde, so listen Sie die Datei mit dem *Dir*-Befehl kurz auf und prüfen nur die Angaben in der Zusammenfassung:

```
@Echo off
Dir C:\Windows\System32\Services.exe
```

Listing 180: 2_02_011a.bat

Um den letzten Änderungszeitpunkt für einen Ordner zu ermitteln, verwenden Sie ebenfalls den *Dir*-Befehl, dem Sie lediglich den Namen des Ordners als Parameter übergeben. Am Anfang der Auflistung des Ordnerinhaltes werden die Einträge *.* und *..* angezeigt, die Aufschluss über den letzten Änderungszeitpunkt geben.

```
@Echo off
Dir C:\Windows
```

Listing 181: 2_02_011b.bat

Alternativ dazu können Sie natürlich auch eine *WMIC*-Abfrage durchführen. Wenn Sie beispielsweise wissen möchten, zu welchem Zeitpunkt die Datei *Services.exe* zuletzt geändert wurde, so führen Sie eine Abfrage gegen den Alias *DataFile* durch, selektieren alle Dateien, deren Name gleich *C:\Windows\System32\Services.exe* ist, und lesen mit Hilfe der *Get*-Anweisung den Wert der Eigenschaft *LastModified* aus.

```
@Echo off
WMIC DataFile WHERE _
    "Name = 'C:\\Windows\\System32\\Services.exe'" _
    Get LastModified
```

Listing 182: 2_02_011c.bat

Die gleiche Abfrage verwenden Sie auch, um den letzten Änderungszeitpunkt eines Ordners zu ermitteln, wobei Sie hierfür aber den Alias `FSDir` verwenden:

```
@Echo off
WMI FSDir WHERE "Name = 'C:\\Windows\\System32'" _
    Get LastModified
```

Listing 183: 2_02_011d.bat

Die Lösung mit VBScript

Der letzte Änderungszeitpunkt einer Datei oder eines Ordners lässt sich mit VBScript auf zwei verschiedenen Wegen ermitteln. Denn Sie können zum einen das `FileSystem-Objekt` verwenden und mit Hilfe der `GetFile-Methode` den Wert der Eigenschaft `DateLastModified` abfragen. Sie können zum anderen aber auch eine `WMI-Abfrage` durchführen und so den Wert der Eigenschaft `LastModified` ermitteln, womit Sie den letzten Änderungszeitpunkt im UTC-Format zurückerhalten.

Um den letzten Änderungszeitpunkt der Datei `C:\\LogFiles\\Protokoll.txt` mit Hilfe des `FileSystem-Objektes` zu ermitteln, binden Sie die betreffende Datei per `GetFile-Methode` an eine Objektvariable. Anschließend fragen Sie den Wert der `DateLastModified-Eigenschaft` ab und geben das Ergebnis z.B. mit der `WScript.Echo-Anweisung` am Bildschirm aus:

```
Option Explicit

Dim oFS
Dim oFile
Dim strFile

strFile = "C:\\LogFiles\\Protokoll.txt"

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFile = oFS.GetFile(strFile)

Wscript.Echo "Geändert am: " & oFile.DateLastModified
```

Listing 184: 2_02_011a.vbs

```
Set oFS = Nothing
Set oFile = Nothing
```

Listing 184: 2_02_011a.vbs (Forts.)

Das gleiche Skriptbeispiel können Sie natürlich auch verwenden, wenn Sie den letzten Änderungszeitpunkt eines Ordners ermitteln wollen. Allerdings müssen Sie die Bindung des Ordners dann mit Hilfe der `GetFolder`-Methode durchführen.

```
Set oFolder = oFS.GetFolder(strFolder)
```

Die nachfolgenden Code-Zeilen zeigen, wie sich der letzte Änderungszeitpunkt der Datei `C:\LogFiles\Protokoll.txt` mit Hilfe einer WMI-Abfrage ermitteln lässt. Zunächst folgt die `ExecQuery`-Abfrage gegen die WMI-Klasse `CIM_DataFile`, um die betreffende Datei (`strFile`) zu selektieren. Innerhalb einer `For Each`-Schleife wird dann der Wert der Eigenschaft `LastModified` ausgelesen und per `WScript.Echo`-Anweisung am Bildschirm ausgegeben.

```
Option Explicit

Dim oWMI
Dim colFile
Dim objItem
Dim strComputer
Dim strFile

strComputer = "."
strFile = "'C:\\LogFiles\\Protokoll.txt'"

Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\\root\\cimv2")
Set colFile = oWMI.ExecQuery ("SELECT * FROM _
    CIM_Datafile WHERE Name = " & strFile)

For Each objItem in colFile
    WScript.Echo objItem.LastModified
Next

Set oWMI = Nothing
```

Listing 185: 2_02_011b.vbs

Gleiches lässt sich natürlich auch zum Auslesen des letzten Änderungszeitpunktes für einen Ordner einsetzen, allerdings müssen Sie statt der `CIM_DataFile`-Klasse die Klasse `Win32_Directory` abfragen.

38 Datums- und Zeitstempel von Dateien und Ordnern ändern

Mit dem Anlegen einer neuen Datei oder eines neuen Ordners generiert das Betriebssystem automatisch einen Datums- und Zeitstempel, der genaue Auskunft darüber gibt, wann das Datei- oder Ordner-Objekt auf dem Dateisystem erstellt wurde. Gleiches geschieht auch, wenn Sie auf Dateien und Ordner zugreifen oder diese vom Inhalt her ändern. Diese Datums- und Zeitstempel sind in den Eigenschaften der Dateien und Ordner hinterlegt und können durch Aufruf der Datei- bzw. Ordner-eigenschaften jederzeit eingesehen werden.

Wenn Sie z.B. die Datums- und Zeitstempel der Datei *C:\LogFiles\Protokoll.txt* abfragen möchten, so starten Sie hierzu den WINDOWS EXPLORER und rufen aus dem Kontextmenü der Datei die EIGENSCHAFTEN auf. Auf der Registerkarte ALLGEMEIN finden Sie die Datums- und Zeitstempel für die Eigenschaften *Erstellt*, *Geändert am* und *Letzter Zugriff* und erfahren so, wann die Datei erzeugt wurde, wann zuletzt daraufhin zugegriffen und wann sie zuletzt geändert wurde.

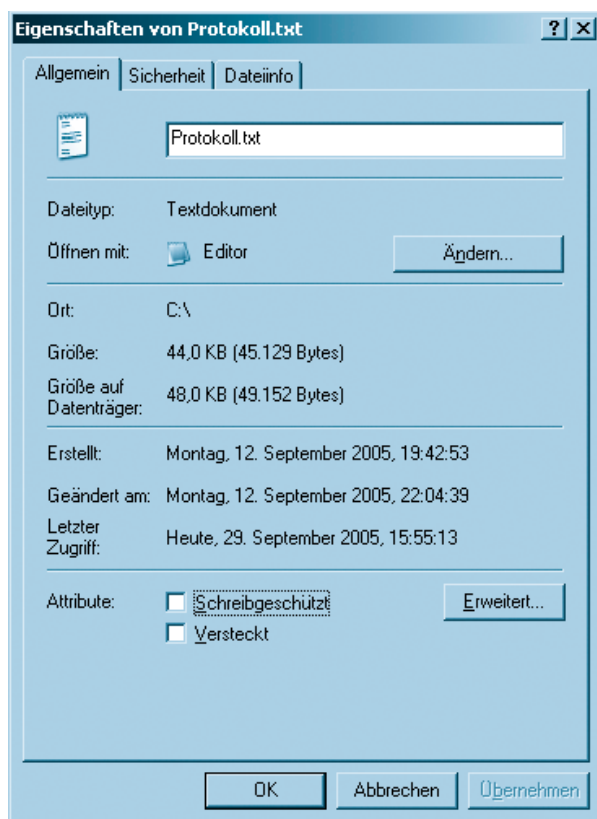


Abbildung 89: Die Datums- und Zeitstempel der Datei »Protokoll.txt«

In einigen Test- und Troubleshooting-Fällen kann es nun erforderlich sein, diese Datums- und Zeitstempel zu modifizieren, um beispielsweise das Verhalten nach Ablauf einer bestimmten Frist zu simulieren.

Leider stellt der Windows Server 2003 hierfür keinerlei Funktionen (und auch keine Tools) bereit, womit Sie die Datums- und Zeitwerte von Dateien und Ordner so ohne weiteres ändern können. Es wäre zwar möglich, die Systemzeit des Computersystems zu verändern und das Datei- oder Ordner-Objekt zu öffnen oder zu bearbeiten, womit ein neuer Stempel generiert wird, allerdings kann dies äußerst unangenehme Nebenwirkungen mit sich bringen. Denn wenn Sie beispielsweise das Systemdatum um einige Wochen oder Monate vorverlegen, so könnten Sie damit Kennwörter zum Ablauf bringen, was gesperrte Benutzerkonten zur Folge haben könnte.

Weitaus einfacher und zudem noch sicherer lassen sich die Datums- und Zeitstempel von Dateien und Ordnern mit dem GNU-Utility *Touch.exe* (siehe nächster Abschnitt) bearbeiten, da Sie mit Hilfe dieses Tools die Eigenschaften *Geändert am* und *Letzter Zugriff* von der Eingabeaufforderung aus modifizieren können, ohne dass Sie dafür die Systemzeit des Computersystems verändern müssen.

Die Lösung per Eingabeaufforderung

In den *GNU-Utilities für Win32* ist ein Tool mit der Bezeichnung *Touch.exe* enthalten, womit Sie auf recht einfache Art und Weise die Werte der Datei- und Ordner-eigenschaften *Geändert am* und *Letzter Zugriff* ändern können.

Wenn Sie z.B. den Datumsstempel für *Geändert am* auf den *01.01.* diesen Jahres und den Zeitstempel auf *00:00 Uhr* setzen möchten, so wenden Sie den *Touch*-Befehl unter Angabe des Parameters *-t* und des Wertes *01010000* (Format *mmddhhmm* oder wahlweise auch *mmddhhm myyss*) an, legen mit *-m* fest, dass die Eigenschaft *Geändert am* (Modified) geändert werden soll, und übergeben den Namen der Datei oder des Ordners, für die Sie den Datums- und Zeitstempel ändern möchten.

```
@Echo off
Touch -t 01010000 -m TestFile.txt
```

Listing 186: 2_02_012a.bat

Um dagegen die Eigenschaft *Letzter Zugriff* zu ändern, setzen Sie statt des Parameters *-m* (Modified) den Parameter *-a* (Accessed) ein.

```
@Echo off
Touch -t 01010000 -a TestFile.txt
```

Listing 187: 2_02_012b.bat

Wenn Sie den Datumsstempel für *Geändert am* und *Letzter Zugriff* nicht auf den 01.01. dieses Jahres setzen möchten, sondern z.B. auf den 01.01.2004, so übergeben Sie das neue Datum und auch die neue Uhrzeit im Format *mmddhhmm*, in diesem Beispiel also 0101000004.

```
@Echo off
Touch -t 0101000004 -m TestFile.txt
Touch -t 0101000004 -a TestFile.txt
```

Listing 188: 2_02_012c.bat

Die Lösung mit VBScript

Mit VBScript lassen sich die Datei- und Ordneigenschaften *Geändert am* und *Letzter Zugriff* zwar auf unterschiedlichsten Wegen auslesen, jedoch nicht bearbeiten. Wenn Sie die Werte der beiden Eigenschaften also aus einem VBScript heraus modifizieren möchten, so nutzen Sie die gleiche Möglichkeit wie auch in der Eingabeaufforderung, sprich: Sie setzen das GNU-Utility *Touch.exe* ein, das Sie mit der *Run*-Methode des *Shell*-Objektes auch per VBScript aufrufen können.

Das nachfolgende Listing startet das Tool *Touch.exe* und legt die Werte *01.01. diesen Jahres* und *00:00 Uhr* für die Eigenschaften *Geändert am* und *Letzter Zugriff* der Datei *TestFile.txt* fest.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Touch -t 01010000 -m TestFile.txt"
oShell.Run "%comspec% /c Touch -t 01010000 -a TestFile.txt"

Set oShell = Nothing
```

Listing 189: 2_02_012.vbs

39 Datei- und Ordnerattribute ändern

Einzelne Dateien und Ordner können mit speziellen Attributen versehen werden, worüber sich bestimmte Datei- bzw. Ordneigenschaften festlegen lassen. So lässt sich eine Datei beispielsweise durch Setzen des Attributes *Read-Only* mit einem Schreibschutz versehen und vor künftigen Änderungen schützen (bei einem Ordner hat dieses Attribut jedoch keine Auswirkung). Das Attribut *Hidden* dagegen bewirkt, dass das Objekt versteckt wird und – sofern in den Ordner-Optionen des WINDOWS EXPLORERS die Option VERSTECKTE DATEIEN UND ORDNER AUSBLENDEN aktiviert ist – im WINDOWS EXPLORER auch nicht mehr angezeigt wird.

Die nachfolgende Tabelle zeigt, welche Datei- und Ordner-Attribute unterstützt werden und bei Bedarf auch verändert werden können.

Attribut	Beschreibung
R (Read-Only)	Schreibgeschützte Datei, die zwar zum Lesen geöffnet, aber inhaltlich nicht verändert werden kann
H (Hidden)	Versteckte Datei
S (System)	Systemdatei
A (Archive)	Zu archivierende Datei, die mit dem nächsten Backup gesichert wird, falls die Datei ausgewählt wird

Tabelle 33: Die Datei- und Ordner-Attribute

Welche Datei bzw. welcher Ordner mit welchen Attributen versehen ist, können Sie sich jederzeit im WINDOWS EXPLORER anzeigen lassen, wenn Sie in die Ansicht DETAILS wechseln. Denn in der Spalte ATTRIBUTE werden alle Attribute der jeweiligen Datei (oder des Ordners) angezeigt, die gerade aktiviert sind.

Der WINDOWS EXPLORER zeigt dabei noch zwei weitere Attribute an, die in der vorherigen Tabelle nicht erwähnt wurden: C für *Compress* und E für *Encrypted*. Bei diesen beiden Attributen handelt es sich um NTFS-spezifische Attribute, die für ein komprimiertes (siehe Dateien und Ordner komprimieren und dekomprimieren auf Seite 292) oder verschlüsseltes (siehe Dateien und Ordner ver- und entschlüsseln auf Seite 298) Objekt gelten und auf anderen Dateisystemen nicht unterstützt werden.

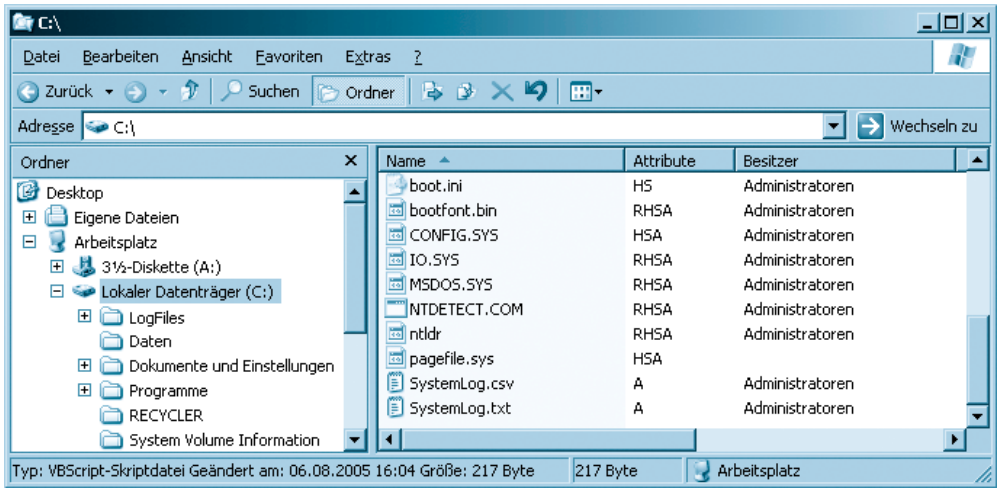


Abbildung 90: Die Anzeige der Attribute im Windows Explorer

Während Sie sich unter grafischer Oberfläche alle Attribute der Dateien und Ordner anzeigen lassen können, so können Sie per GUI nur die Attribute *Archive* und *Read-Only* verändern. Der Zugriff auf die Attribute *Hidden* und *System* bleibt unter grafi-

scher Oberfläche leider aus Sicherheitsgründen verwehrt, dies geht nur mit dem `Attrib`-Befehl in der Eingabeaufforderung (siehe nachfolgender Abschnitt).

Wenn Sie nun die Attribute *Archive* und *Read-Only* für ein bestimmtes Objekt ändern möchten, so starten Sie den WINDOWS EXPLORER, markieren die Datei oder den Ordner und rufen im Kontextmenü die EIGENSCHAFTEN auf. Auf der Registerkarte ALLGEMEIN können Sie das Attribut *Read-Only* setzen, indem Sie die Option SCHREIBGESCHÜTZT anklicken. Um das Attribut *Archive* zu setzen, wechseln Sie in die erweiterten Einstellungen und aktivieren die Option DATEI KANN ARCHIVIERT WERDEN, woraufhin das Objekt beim nächsten Backup berücksichtigt wird. Und wenn Sie die Attribute wieder entfernen möchten, so brauchen Sie die soeben genannten Optionen nur zu deaktivieren.

Die Lösung per Eingabeaufforderung

Die Attribute einer Datei oder eines Ordners lassen sich natürlich auch in der Eingabeaufforderung auslesen und ändern. Hierfür stellt der Windows Server 2003 den `Attrib`-Befehl bereit, der ohne Angabe weiterer Parameter die Attribute aller Dateien im aktuellen Ordner ausgibt.

Um ein Attribut zu setzen, wenden Sie den `Attrib`-Befehl gegen die betreffende Datei oder den Ordner an und stellen dem entsprechenden Attribut ein Pluszeichen voran. Möchten Sie beispielsweise der Datei `C:\LogFiles\Protokoll.txt` alle Attribute zuweisen, so verwenden Sie folgende Befehlszeile:

```
@Echo off
Attrib C:\LogFiles\Protokoll.txt +r +a +s +h
```

Listing 190: 2_02_013a.bat

Die soeben gesetzten Attribute lassen sich mit dem `Attrib`-Befehl auch wieder entfernen, indem Sie dem jeweiligen Attribut ein Minuszeichen voranstellen.

```
@Echo off
Attrib C:\LogFiles\Protokoll.txt -r -a -s -h
```

Listing 191: 2_02_013b.bat

Die Lösung mit VBScript

Mit Hilfe der `Attributes`-Eigenschaft des `FileSystem`-Objektes lassen sich die einzelnen Datei- und Ordner-Attribute auch mit VBScript problemlos auslesen und verändern. Hierbei ist jedem Attribut ein bestimmter Wert zugeordnet, den Sie der Eigenschaft nur zuweisen müssen, um das entsprechende Attribut zu setzen.

Die nachfolgende Tabelle gibt Auskunft über die möglichen Attribute, die sowohl für Dateien wie auch für Ordner gelten. Dabei sei allerdings angemerkt, dass einige dieser Attribute vom Typ *Read-Only* sind und daher nur ausgelesen, nicht aber verändert werden können.

Attribut	Wert
Normal	0
Read-Only	1
Hidden	2
System	4
Volume (Read-Only)	8
Directory (Read-Only)	16
Archive	32
Link (Read-Only)	64
Compressed (Read-Only)	2048

Tabelle 34: Die Attribute und ihre Werte

Wenn Sie beispielsweise der Datei *C:\LogFiles\Protokoll.txt* die Attribute *Read-Only*, *Hidden*, *System* und *Archive* zuweisen möchten, so erstellen Sie mit der `CreateObject`-Methode eine neue Instanz des `FileSystem`-Objektes und binden die zuvor genannte Datei an eine Objektvariable. Anschließend prüfen Sie mit der `IF`-Anweisung die Werte der entsprechenden Attribute. Falls die Werte noch nicht gesetzt sein sollten, so können Sie die Werte mit dem logischen Operator `XOR` setzen.

```
Option Explicit

Dim oFS
Dim oFile

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFile = oFS.GetFile("C:\LogFiles\Protokoll.txt")

If Not oFile.Attributes AND 1 Then
    oFile.Attributes = oFile.Attributes XOR 1
End If

If Not oFile.Attributes AND 2 Then
    oFile.Attributes = oFile.Attributes XOR 2
End If

If Not oFile.Attributes AND 4 Then
```

Listing 192: 2_02_013a.vbs

```

        oFile.Attributes = oFile.Attributes XOR 4
    End If

    If Not oFile.Attributes AND 32 Then
        oFile.Attributes = oFile.Attributes XOR 32
    End If

    Set oFS = Nothing

```

Listing 192: 2_02_013a.vbs (Forts.)

Um die soeben gesetzten Attribute wieder zu entfernen, prüfen Sie zunächst, ob der Wert bereits gesetzt ist. Wenn dies der Fall ist, so können Sie den Wert mit dem logischen Operator XOR entfernen.

```

Option Explicit

Dim oFS
Dim oFile

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFile = oFS.GetFile("C:\LogFiles\Protokoll.txt")

If oFile.Attributes AND 1 Then
    oFile.Attributes = oFile.Attributes XOR 1
End If

If oFile.Attributes AND 2 Then
    oFile.Attributes = oFile.Attributes XOR 2
End If

If oFile.Attributes AND 4 Then
    oFile.Attributes = oFile.Attributes XOR 4
End If

If oFile.Attributes AND 32 Then
    oFile.Attributes = oFile.Attributes XOR 32
End If

Set oFS = Nothing

```

Listing 193: 2_02_013b.vbs

40 Erweiterte Datei- und Ordneigenschaften auslesen

In den vorangegangenen Rezepten haben Sie erfahren, wie Sie sich die Attribute und auch die Erstellungs- und Zugriffsdaten von Dateien und Ordner anzeigen lassen können. Viele Datei-Objekte (manchmal auch Ordner-Objekte) enthalten darüber hinaus aber noch weitere Eigenschaften (z.B. versions-, produkt- oder medienspezifische Informationen), die meist in der Anwendung, mit der die Objekte erstellt wurden, gepflegt werden bzw. geändert werden können, die Sie sich aber auch innerhalb des WINDOWS EXPLORERS anzeigen lassen können. Denn der WINDOWS EXPLORER gibt in der Ansicht DETAIL weitaus mehr Informationen über Dateien und Ordner preis als zunächst angenommen.

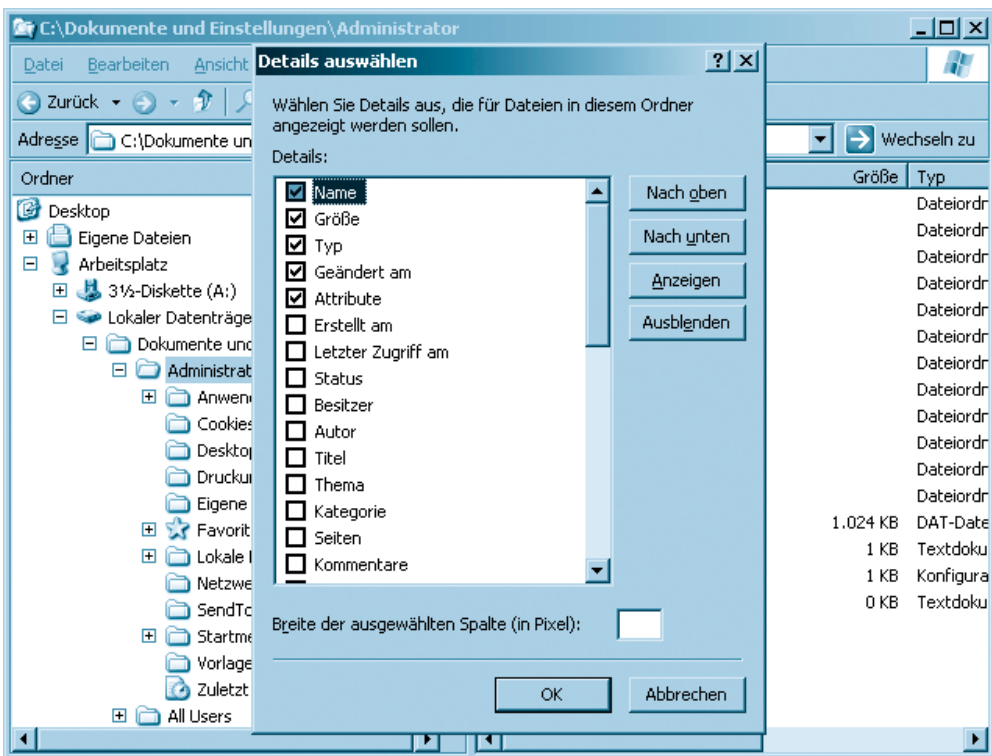


Abbildung 91: Die erweiterten Eigenschaften auswählen und einblenden

Um erweiterte Datei- und Ordneigenschaften im WINDOWS EXPLORER anzeigen zu lassen, müssen Sie lediglich in die Ansicht DETAIL wechseln und die entsprechenden Spalten einblenden. Hierzu wählen Sie im Menü ANSICHT den Eintrag DETAILS AUSWÄHLEN... aus und aktivieren die gewünschten Spalten. Mit einem Klick auf OK werden die geänderten Einstellungen übernommen, und die erweiterten Eigenschaften werden – sofern sie gepflegt wurden – in der rechten Fensterhälfte des WINDOWS EXPLORERS angezeigt.

Die Lösung per Eingabeaufforderung

Um die Eigenschaften einer Datei oder eines Ordners per Eingabeaufforderung auszu-
lesen, können Sie den WMIC-Befehl verwenden, dem Sie lediglich den entsprechenden
Alias und auch die betreffende Datei (oder den Ordner) als Parameter übergeben. Der
WMIC-Befehl stellt daraufhin eine Verbindung zu dem Objekt her, liest dessen Eigen-
schaften aus (siehe *Teil III*) und gibt das Ergebnis am Bildschirm aus.

Wenn Sie alle Eigenschaften der Datei *C:\LogFiles\Protokoll.txt* auflisten und in eine
Datei namens *File.txt* ausgeben möchten, so führen Sie eine WMIC-Abfrage gegen den
Alias *DataFile* durch, selektieren mit der Where-Klausel die betreffende Datei unter
Verwendung der Eigenschaft *Name* und schreiben das Ergebnis per Ausgabeumleitung
in die Zieldatei:

```
@Echo off
WMIC DataFile Where "Name = 'C:\\LogFiles\\Protokoll.txt'" > File.txt
```

Listing 194: 2_02_014a.bat

Möchten Sie dagegen die Eigenschaften eines Ordners ausgeben, so richten Sie die
Abfrage stattdessen gegen den Alias *FSDir* und übergeben den Namen des Ordners:

```
@Echo off
WMIC FSDir Where "Name = 'C:\\LogFiles'" > Dir.txt
```

Listing 195: 2_02_014b.bat

Die Lösung mit VBScript

Mit einer WMI-Abfrage lassen sich die Eigenschaften einer Datei oder eines Ordners
auch per VBScript auslesen. Um beispielsweise sämtliche Eigenschaften der Datei
C:\LogFiles\Protokoll.txt am Bildschirm auszugeben, führen Sie eine *ExecQuery*-
Abfrage gegen die WMI-Klasse *CIM_DataFile* durch und selektieren die betreffende
Datei unter Angabe der Eigenschaft *Name*.

Da das Ergebnis in Form einer Kollektion zurückgeliefert wird, setzen Sie eine *For Each*-
Schleife ein, um auf das Datei-Objekt zugreifen zu können, und fragen dann die jewei-
ligen Werte der Eigenschaften ab, die Sie mit Hilfe der *WScript.Echo*-Anweisung am
Bildschirm ausgeben.

```
Option Explicit
```

```
Dim oWMI
Dim colFile
Dim objItem
```

Listing 196: 2_02_014.vbs


```

Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=impersonate}!\" _
    & strComputer & "\root\cimv2")
Set colFile = oWMI.ExecQuery ("SELECT * FROM CIM_Datafile WHERE _
    Name = 'c:\\LogFiles\\Protokoll.txt'")

For Each objItem in colFile
    Wscript.Echo "Access mask: " & objItem.AccessMask
    Wscript.Echo "Archive: " & objItem.Archive
    Wscript.Echo "Caption: " & objItem.Caption
    Wscript.Echo "Compressed: " & objItem.Compressed
    Wscript.Echo "Compression method: " & objItem.CompressionMethod
    Wscript.Echo "CreationClassName: " & objItem.CreationClassName
    Wscript.Echo "Creation date: " & objItem.CreationDate
    Wscript.Echo "Computer system name: " & objItem.CSName
    Wscript.Echo "Description: " & objItem.Description
    Wscript.Echo "Drive: " & objItem.Drive
    Wscript.Echo "8.3 file name: " & objItem.EightDotThreeFileName
    Wscript.Echo "Encrypted: " & objItem.Encrypted
    Wscript.Echo "Encryption method: " & objItem.EncryptionMethod
    Wscript.Echo "Extension: " & objItem.Extension
    Wscript.Echo "File name: " & objItem.FileName
    Wscript.Echo "File size: " & objItem.FileSize
    Wscript.Echo "File type: " & objItem.FileType
    Wscript.Echo "InuseCount: " & objItem.InUseCount
    Wscript.Echo "FSCreationClassName: " & objItem.FSCreationClassName
    Wscript.Echo "Hidden: " & objItem.Hidden
    Wscript.Echo "InstallDate: " & objItem.InstallDate
    Wscript.Echo "Last accessed: " & objItem.LastAccessed
    Wscript.Echo "Last modified: " & objItem.LastModified
    Wscript.Echo "Manufacturer: " & objItem.Manufacturer
    Wscript.Echo "Name: " & objItem.Name
    Wscript.Echo "Path: " & objItem.Path
    Wscript.Echo "Readable: " & objItem.Readable
    Wscript.Echo "Status: " & objItem.Status
    Wscript.Echo "System: " & objItem.System
    Wscript.Echo "Version: " & objItem.Version
    Wscript.Echo "Writeable: " & objItem.Writeable
Next

Set oWMI = Nothing

```

Listing 196: 2_02_014.vbs (Forts.)

Mit dem gleichen Code-Beispiel können Sie auch die Eigenschaften eines Ordners auslesen, allerdings müssen Sie die Abfrage gegen die WMI-Klasse Win32_Directory richten

und dabei den Namen des Ordners übergeben. Die restlichen Code-Zeilen können Sie unverändert lassen, da die Win32_Directory-Klasse die gleichen Eigenschaften verwendet wie die CIM_DataFile-Klasse.

```
Set colFolder = oWMI.ExecQuery ("SELECT * FROM _
Win32_Directory WHERE Name = 'c:\\LogFiles'")
```

41 Dateien und Ordner kopieren und verschieben

Dank der Zwischenablage von Windows Server 2003 lassen sich einzelne Dateien wie auch komplette Ordner jederzeit kopieren und in einem beliebigen Pfad wieder einfügen. Einzelne Objekte können auch ausgeschnitten und in einem anderen Pfad wieder eingefügt werden, wodurch sich Dateien und Ordner problemlos verschieben lassen. Einzige Voraussetzung hierfür: Sie benötigen zumindest ein Leserecht auf das Objekt, das Sie kopieren wollen, und ein entsprechendes Schreibrecht im Zielordner, um das Objekt wieder einfügen zu können.

Einmal angenommen, Sie möchten aus Datensicherungsgründen den kompletten Inhalt des Ordners *C:\LogFiles* nach *C:\Backup\Logs* kopieren. Hierzu rufen Sie den WINDOWS EXPLORER auf und wechseln in den Ordner *C:\LogFiles*. Anschließend markieren Sie in der rechten Fensterhälfte den kompletten Ordnerinhalt (z.B. mit **[Strg] + [A]**) und wählen entweder im Kontextmenü KOPIEREN aus oder drücken die Tastenkombination **[Strg] + [C]**. Dann wechseln Sie in den Ordner *C:\Backup\Logs*, klicken in der rechten Fensterhälfte auf einen freien Bereich und wählen im Kontextmenü EINFÜGEN aus oder betätigen alternativ die Tastenkombination **[Strg] + [V]**.

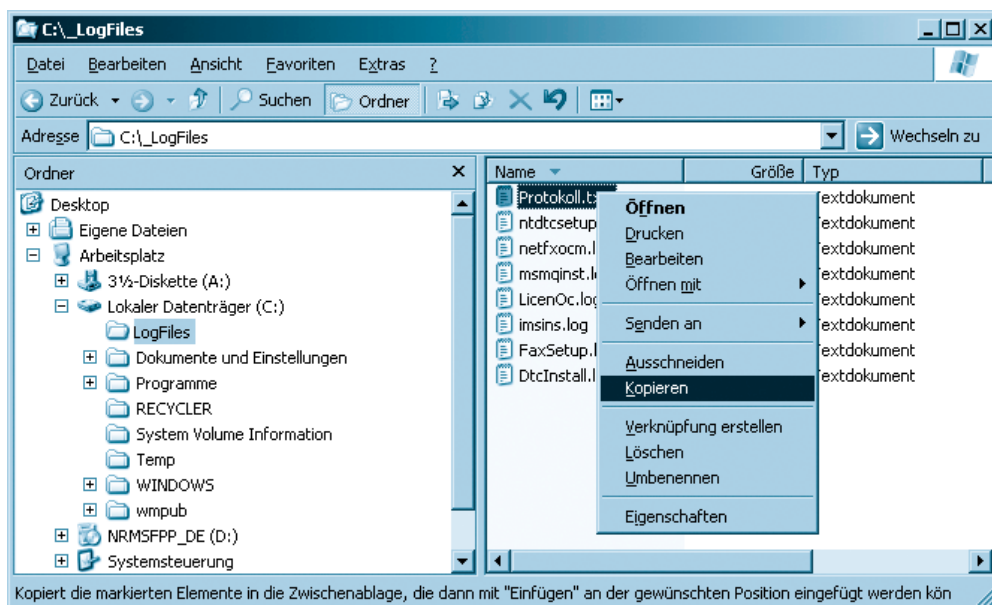


Abbildung 92: Eine Datei kopieren

Wenn Sie den Inhalt des Ordners *C:\LogFiles* aber nicht vervielfältigen, sondern verschieben möchten, so markieren Sie zunächst den Inhalt von *C:\LogFiles* und wählen im Kontextmenü nicht KOPIEREN, sondern AUSSCHNEIDEN aus. Damit werden die ausgewählten Objekte zwar auch in die Zwischenablage übertragen, von wo aus sie jederzeit wieder in einem beliebigen Pfad eingefügt werden können, allerdings werden die Objekte in ihrem Quellpfad gelöscht, womit der Ordner *C:\LogFiles* nach dem Ausschneidevorgang geleert wird.

Die Lösung per Eingabeaufforderung

Der Windows Server 2003 bringt von Haus aus entsprechende Befehle mit, womit Sie auch von der Eingabeaufforderung aus einzelne Dateien wie auch gesamte Ordner kopieren und verschieben können. Wenn Sie z.B. die Dateien des Ordners *C:\LogFiles* nach *C:\Backup\Logs* kopieren wollen, dann können Sie hierfür den *Copy*-Befehl wie auch den *Xcopy*-Befehl einsetzen, wobei Letzteres aber weitaus mehr Möglichkeiten bietet und daher auch in den nachfolgenden Beispielen als favorisierter Kopierbefehl verwendet wird.

Um nun die Dateien von *C:\LogFiles* nach *C:\Backup\Logs* zu kopieren, wenden Sie den *Xcopy*-Befehl unter Angabe der zu kopierenden Dateien wie auch des Zielpfades an. Für den Fall, dass der Zielpfad noch nicht existiert, so erfolgt eine Rückfrage, ob das angegebene Ziel eine Datei oder ein Verzeichnis ist. Um dies zu vermeiden und von vornherein die Option *Verzeichnis* zu bestimmen, fügen Sie der Befehlszeile den Parameter */i* an.

Und falls die zu kopierenden Dateien bereits im Zielpfad existieren, so müssen Sie das Überschreiben der jeweiligen Dateien von Hand bestätigen. Dies lässt sich aber mit dem zusätzlichen Parameter */y* unterdrücken, da der Parameter */y* das Überschreiben der Zieldateien erzwingt. Und sollte der Quellpfad einzelne Dateien mit den Attributen *System* oder *Versteckt* enthalten, so müssen Sie zusätzlich die Parameter */h* verwenden, da die betreffenden Dateien anderenfalls unberücksichtigt bleiben.

```
@Echo off
xcopy C:\LogFiles\*. * C:\Backup\Logs /i /y /h
```

Listing 197: 2_02_015a.bat

Um einen kompletten Ordner samt Inhalt zu kopieren, können Sie ebenfalls den *Xcopy*-Befehl einsetzen, indem Sie den Quellordner wie auch den Zielordner übergeben. Und damit auch die Inhalte der untergeordneten Ordner berücksichtigt werden, fügen Sie den zusätzlichen Parameter */s* an, um den Kopiervorgang rekursiv durchzuführen.

```
@Echo off
xcopy C:\LogFiles C:\Temp\LogFiles\ /s
```

Listing 198: 2_02_015b.bat

Wenn Sie die Dateien oder Ordner nicht kopieren, sondern gänzlich verschieben möchten, so verwenden Sie stattdessen den `Move`-Befehl. Um beispielsweise die Datei `C:\LogFiles\Protokoll.txt` nach `C:\Temp` zu verschieben, wenden Sie den `Move`-Befehl unter Angabe der zu verschiebenden Datei wie auch des Zielpfades an.

```
@Echo off
Move C:\LogFiles\Protokoll.txt C:\Temp
```

Listing 199: 2_02_015c.bat

Und um einen Ordner zu verschieben, setzen Sie den `Move`-Befehl unter Angabe des zu verschiebenden Ordners wie auch des Zielpfades ein.

```
@Echo off
Move C:\LogFiles C:\Temp
```

Listing 200: 2_02_015d.bat

Die Lösung mit VBScript

Die einfachste Möglichkeit, eine Datei oder auch einen Ordner per VBScript zu kopieren oder zu verschieben, bietet das `FileSystem`-Objekt. Denn mit den Methoden `CopyFile`, `CopyFolder`, `MoveFile` und `MoveFolder` lassen sich die Objekte mit nur wenigen Code-Zeilen vervielfältigen und verschieben. Sie müssen lediglich mit Hilfe der `CreateObject`-Methode eine neue Instanz des `FileSystem`-Objektes erzeugen, die jeweilige Methode gegen die Instanz des `FileSystem`-Objektes anwenden und den alten wie auch neuen Pfad als Argument übergeben.

```
Option Explicit

Dim oFS
Dim strFiles
Dim strNewPath

strFiles = "C:\LogFiles\*.*)"
strNewPath = "C:\Backup\Logs"

Set oFS = CreateObject("Scripting.FileSystemObject")
oFS.CopyFile strFiles, strNewPath

Set oFS = Nothing
```

Listing 201: 2_02_015a.vbs

Auf die gleiche Art lässt sich natürlich auch ein Ordner kopieren, wobei Sie dann natürlich statt der `CopyFile`-Methode die `CopyFolder`-Methode einsetzen müssen:

```
oFS.CopyFolder strFolder, strNewPath
```

Mit Hilfe der `MoveFile`- und `MoveFolder`-Methode lassen sich sowohl einzelne Dateien wie auch komplette Ordner in einen anderen Pfad verschieben. Wenn Sie z.B. die Datei `C:\LogFiles\Protokoll.txt` nach `C:\Temp` verschieben wollen, so wenden Sie die `MoveFile`-Methode an, der Sie die Datei wie auch den neuen Pfad als Argument übergeben.

Option Explicit

```
Dim oFS
Dim strFile
Dim strNewPath
```

```
strFile = "C:\LogFiles\Protokoll.txt"
strNewPath = "C:\Temp"
```

```
Set oFS = CreateObject("Scripting.FileSystemObject")
oFS.MoveFile strFile, strNewPath
```

```
Set oFS = Nothing
```

Listing 202: 2_02_015b.vbs

Um einen Ordner zu verschieben, verwenden Sie stattdessen die `MoveFolder`-Methode und übergeben den Ordnernamen wie auch den neuen Pfad:

```
oFS.MoveFolder strFolder, strNewPath
```

42 Dateien und Ordner umbenennen

Falls Sie eine Datei oder einen Ordner versehentlich unter falschem Namen angelegt haben, so können Sie das betreffende Objekt jederzeit umbenennen. Starten Sie einfach den `WINDOWS EXPLORER`, markieren Sie die Datei oder den Ordner, und drücken Sie entweder die Taste `[F2]`, oder wählen Sie im Kontextmenü des Objektes den Eintrag `UMBENENNEN` aus.

Daraufhin wird der Bearbeitungsmodus für den Datei-/Ordnernamen gestartet, und Sie können die aktuelle Bezeichnung durch eine neue Bezeichnung überschreiben. Drücken Sie anschließend die `[Enter]`-Taste, um die Änderung zu übernehmen.

Die `UMBENENNEN`-Funktion steht Ihnen nicht nur im `WINDOWS EXPLORER` zur Verfügung. Sie finden diese Funktion auch in den Kontextmenüs der `STARTMENÜ`-Einträge, der `DESKTOP`-Symbole und vielen weiteren mehr.

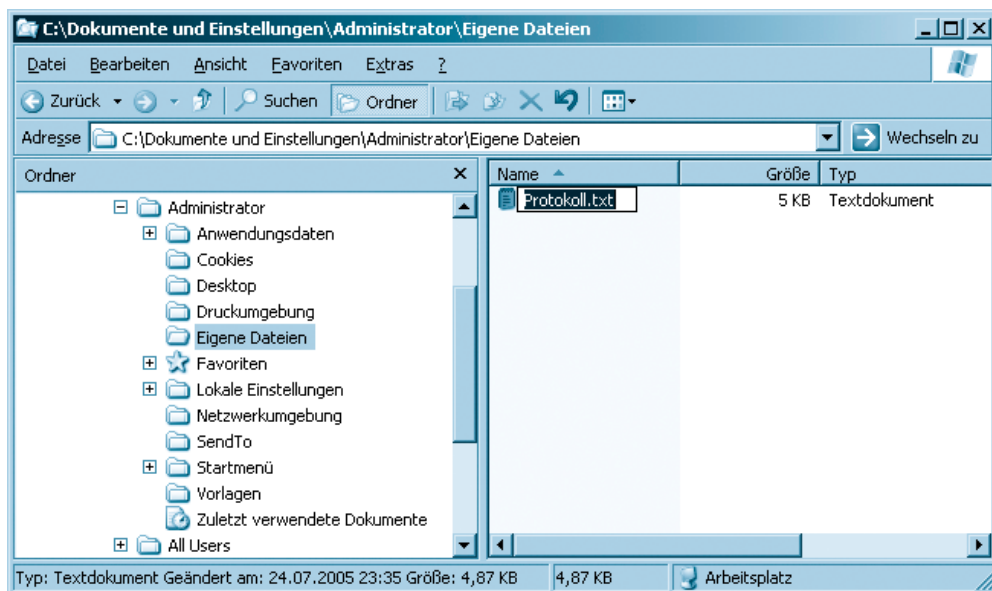


Abbildung 93: Eine Datei umbenennen

Die Lösung per Eingabeaufforderung

Um eine Datei oder wahlweise auch einen Ordner von der Eingabeaufforderung aus umzubenennen, verwenden Sie den `Rename`- bzw. `Ren`-Befehl, dem Sie lediglich den alten und auch den neuen Namen des umzubenennenden Objektes als Parameter übergeben. Der `Rename`- bzw. `Ren`-Befehl benennt dabei immer nur Dateien im gleichen Ordner um.

Wenn Sie z.B. die Datei *Protokoll.txt*, die sich im Ordner *C:\Dokumente und Einstellungen\Administrator\Eigene Dateien* befindet, nach *Protokoll.log* umbenennen wollen, dann wenden Sie den `Ren`-Befehl an und übergeben den aktuellen Dateinamen (inklusive der Pfadangabe) wie auch den neuen Dateinamen:

```
@Echo off
Ren C:\Dokumente und Einstellungen\Administrator\
  Eigene Dateien\Protokoll.txt Protokoll.log
```

Listing 203: 2_02_016a.bat

Mit der gleichen Befehlszeile können Sie natürlich auch einen kompletten Ordner umbenennen. Geben Sie lediglich den aktuellen Pfad wie auch den künftigen Ordnernamen, an:

```
@Echo off
Ren C:\_LogFiles LogFiles
```

Listing 204: 2_02_016b.bat

Die Lösung mit VBScript

Mit VBScript lässt sich das Umbenennen einer Datei oder wahlweise auch eines Ordners auf zweierlei Wegen realisieren. Denn Sie können zum einen die `MoveFile`- bzw. `MoveFolder`-Methode des `FileSystem`-Objektes verwenden, zum anderen aber auch die `Rename`-Methode der WMI-Klasse `CIM_DataFile` (für Dateien) bzw. `Win32_Directory` (für Ordner).

Ersteres lässt sich mit nur wenigen Code-Zeilen umsetzen, da Sie lediglich eine neue Instanz des `FileSystem`-Objektes erzeugen und die `MoveFile`- bzw. `MoveFolder`-Methode gegen die Objektvariable (`oFS`) anwenden müssen, der Sie den alten (`strOldFile`) und auch neuen Pfad (`strNewFile`) übergeben. Das nachfolgende Code-Beispiel benennt z.B. die Datei `C:\LogFiles\Protokoll.txt` nach `C:\LogFiles\Protokoll.log` um:

```
Option Explicit

Dim oFS
Dim strOldFile
Dim strNewFile

strOldFile = "C:\LogFiles\Protokoll.txt"
strNewFile = "C:\LogFiles\Protokoll.log"

Set oFS = CreateObject("Scripting.FileSystemObject")
oFS.MoveFile strOldFile , strNewFile

Set oFS = Nothing
```

Listing 205: 2_02_016a.vbs

Einen Ordner benennen Sie dagegen mit der `MoveFolder`-Methode um, der Sie ebenfalls den alten und auch neuen Pfad als Argument übergeben:

```
oFS.MoveFolder strOldFolder, strNewFolder
```

Eine weitere Möglichkeit, eine Datei oder auch einen Ordner umzubenennen, bieten die WMI-Klassen `CIM_DataFile` und `Win32_Directory`, die im Gegensatz zum `FileSystem`-Objekt eine entsprechende `Rename`-Methode bereitstellen.

Wenn Sie z.B. die Datei `C:\LogFiles\Protokoll.txt` nach `C:\LogFiles\Protokoll.log` umbenennen wollen, so stellen Sie zunächst eine Verbindung zur WMI-Klasse `CIM_DataFile` her, setzen den `Impersonation-Level` auf `Impersonate` und führen anschließend eine `ExecQuery`-Abfrage gegen die zuvor genannte WMI-Klasse durch, um die Datei mit dem

angegebenen Pfad (strOldFile) zu selektieren. Das Ergebnis wird daraufhin in Form einer Kollektion zurückgeliefert. Innerhalb einer For Each-Schleife wenden Sie dann die Rename-Methode gegen das Element an und übergeben den neuen Pfad (strNewFile) als Argument.

Option Explicit

```
Dim oWMI
Dim colFile
Dim objItem
Dim strComputer
Dim strOldFile
Dim strNewFile
```

```
strComputer = "."
strOldFile = "c:\\Dokumente und Einstellungen\\_
Administrator\\Eigene Dateien\\Protokoll.txt"
strNewFile = "C:\\Dokumente und Einstellungen\\_
Administrator\\Eigene Dateien\\Protokoll.log"
```

```
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
impersonate}!\" & strComputer & "\root\cimv2")
Set colFile = oWMI.ExecQuery ("SELECT * FROM CIM_Datafile _
WHERE Name = " & strOldFile)
```

```
For Each objItem in colFile
    Err = objItem.Rename (strNewFile)
Next
```

```
Set oWMI = Nothing
```

Listing 206: 2_02_016b.vbs

Das gleiche Code-Beispiel können Sie auch zum Umbenennen eines Ordners verwenden, allerdings stellen Sie die Verbindung dann nicht zur WMI-Klasse CIM_DataFile her, sondern zur WMI-Klasse Win32_Directory.

43 Dateien und Ordner komprimieren und dekomprimieren

Für den Fall, dass der freie Speicherplatz auf dem lokalen Datenträger so langsam zur Neige geht, die Aufrüstung der Plattenkapazität aber noch in weiter Ferne liegt, so können Sie mit Hilfe der Komprimierungsfunktion von Windows Server 2003 einzelne Dateien und auch komplette Ordner komprimieren und damit zusätzlichen Speicherplatz schaffen. Allerdings sei an dieser Stelle auch gesagt, dass die Kompri-

mierungsfunktion nicht für zusätzlichen Speicherplatz sorgt, wenn die Dateien und Ordner bereits mit einem externen Komprimierungstool (z.B. WinZip) gepackt wurden.

Die Nutzung der Windows Server 2003-eigenen Komprimierungsfunktion setzt jedoch voraus, dass der betreffende Datenträger mit NTFS formatiert wurde, da die Komprimierungsfunktion nur auf einem NTFS-Laufwerk zur Verfügung steht und nicht auf FAT- oder FAT32-Datenträgern genutzt werden kann.

Um beispielsweise eine einzelne Datei zu komprimieren und deren Speicherplatz auf dem Datenträger etwas zu limitieren, starten Sie den WINDOWS EXPLORER und rufen aus dem Kontextmenü der Datei die EIGENSCHAFTEN auf. Anschließend klicken Sie auf der Registerkarte ALLGEMEIN auf ERWEITERT, um in die erweiterten Einstellungen der Datei zu gelangen. Dort aktivieren Sie die Option INHALT KOMPRIMIEREN, UM SPEICHERPLATZ ZU SPAREN und speichern die geänderte Einstellung mit einem Klick auf ÜBERNEHMEN.

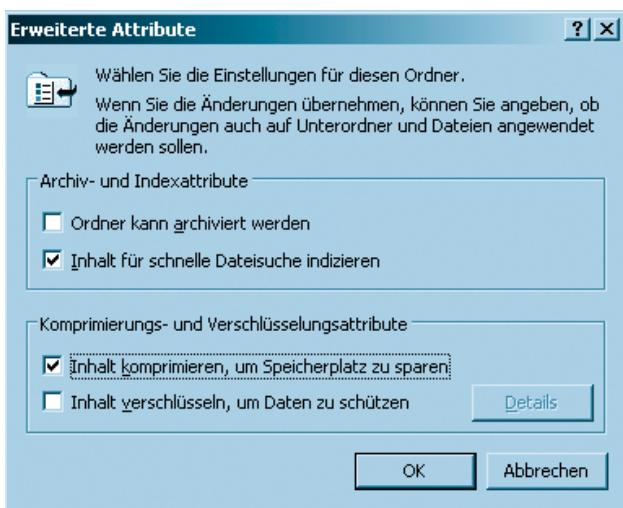


Abbildung 94: Einen Ordner komprimieren

Die Datei wird daraufhin komprimiert und – sofern Sie in den ORDNEROPTIONEN die Option VERSCHLÜSSELTE ODER KOMPRIMIERTE NTFS-DATEIEN IN ANDERER FARBE ANZEIGEN aktiviert haben – im WINDOWS EXPLORER auch in einer anderen Farbe dargestellt, woran Sie auf Anhieb eine komprimierte Datei erkennen können.

Auf die gleiche Art und Weise können Sie natürlich auch einen Ordner komprimieren, wobei Sie in diesem Fall aber angeben müssen, ob nur der Ordner selbst komprimiert werden soll oder auch dessen gesamter Inhalt. Letzteres hat zur Folge, dass Dateien und Ordner, die in einen komprimierten Ordner kopiert oder verschoben werden, das Komprimierungs-Flag des übergeordneten Objektes erben und damit automatisch gepackt werden.

Und für den Fall, dass Sie die Datei bzw. den Ordner wieder dekomprimieren möchten, so rufen Sie erneut die EIGENSCHAFTEN des Objektes auf und deaktivieren die Option INHALT KOMPRIMIEREN, UM SPEICHERPLATZ ZU SPAREN.

Achtung

Wenn Sie einzelne Dateien oder wahlweise auch komplette Ordner komprimiert haben und diese auf einen Datenträger kopieren oder verschieben, der nicht mit NTFS, sondern mit FAT16 oder FAT32 formatiert wurde, so wird das betreffende Objekt automatisch dekomprimiert. Sie sollten also vor dem Kopieren und Verschieben von Objekten zum einen das Dateisystem, zum anderen aber auch den freien Speicherplatz auf dem Zieldatenträger prüfen, damit der Vorgang nicht aufgrund mangelnden Speicherplatzes abbricht.

Die Lösung per Eingabeaufforderung

Mit dem `Compact`-Befehl lassen sich einzelne Dateien und komplette Ordner problemlos von der Eingabeaufforderung aus packen und wieder entpacken. Vorausgesetzt natürlich, dass das betreffende Objekt nicht bereits verschlüsselt ist, da die Komprimierungs- und Verschlüsselungsfunktion auch in der Eingabeaufforderung nicht gemeinsam verwendet werden kann.

Um beispielsweise eine einzelne Datei zu komprimieren, wenden Sie den `Compact`-Befehl an und den Parameter `/c` wie auch den Namen der betreffenden Datei:

```
@Echo off
Compact /c C:\LogFiles\Protokoll.txt
```

Listing 207: 2_02_017a.bat

Mit der gleichen Befehlszeile lässt sich auch ein kompletter Ordner komprimieren, wobei in diesem Fall aber nur der Ordner selbst und nicht die darin enthaltenen Objekte komprimiert werden würde. Um auch den Inhalt eines Ordners zu komprimieren, fügen Sie der Befehlszeile den Parameter `/s:` hinzu und übergeben den Namen des entsprechenden Ordners. Und um einen vorzeitigen Abbruch des `Compact`-Befehls zu vermeiden (falls ein Objekt einmal nicht komprimiert werden kann), übergeben Sie zusätzlich den Parameter `/i`.

```
@Echo off
Compact /c /s:C:\LogFiles /i
```

Listing 208: 2_02_017b.bat

Das Dekomprimieren einer Datei erfolgt mit dem gleichen Befehl, allerdings mit einem anderen Parameter. Wenden Sie den `Compact`-Befehl unter Angabe des Parameters `/u` an, und übergeben Sie den Namen der zu entpackenden Datei.

```
@Echo off
Compact /u C:\LogFiles\Protokoll.txt
```

Listing 209: 2_02_017c.bat

Gleiches gilt auch für die Dekomprimierung eines Ordners. Wenn Sie z.B. den Ordner `C:\LogFiles` komprimiert haben und diesen nun wieder entpacken möchten, so wenden Sie den `Compact`-Befehl mit dem Parameter `/u` an und übergeben mit dem Parameter `/s`: den Namen des Ordners. Idealerweise fügen Sie auch den Parameter `/i` an, da Letzteres den Abbruch der Befehlsausführung verhindert, falls ein Objekt – aus welchen Gründen auch immer – einmal nicht dekomprimiert werden kann.

```
@Echo off
Compact /u /s:C:\LogFiles /i
```

Listing 210: 2_02_017d.bat

Die Lösung mit VBScript

Wenn Sie Dateien und Ordner per VBScript komprimieren und dekomprimieren wollen, so können Sie diese Aufgabe ohne großen Aufwand mit WMI lösen. Denn die WMI-Klassen `CIM_DataFile` und `Win32_Directory` stellen jeweils eine `Compress-` und `Uncompress-Methode` bereit, die Sie nur gegen das betreffende Datei- oder Ordner-Objekt anwenden müssen.

Um beispielsweise die Datei `C:\LogFiles\Protokoll.txt` mit Hilfe von WMI zu komprimieren, führen Sie eine `ExecQuery`-Abfrage gegen die WMI-Klasse `CIM_DataFile` durch und selektieren die betreffende Datei unter Angabe des Dateinamens, woraufhin Sie das Ergebnis in Form einer Kollektion zurückerhalten. Mit einer `For Each`-Schleife können Sie dann auf das betreffende Objekt zugreifen und die `Compress-Methode` gegen das Datei-Objekt anwenden.

```
Option Explicit

Dim oWMI
Dim colFiles
Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colFiles = oWMI.ExecQuery ("SELECT * FROM _
```

Listing 211: 2_02_017a.vbs

```

        CIM_Datafile WHERE Name = 'C:\\LogFiles\\Protokoll.txt')

For Each objItem in colFiles
    Err = objItem.Compress
Next

Set oWMI = Nothing

```

Listing 211: 2_02_017a.vbs (Forts.)

Ein kompletter Ordner lässt sich auf die gleiche Art und Weise komprimieren. Nur mit dem Unterschied, dass Sie in diesem Fall die Win32_Directory-Klasse verwenden und nicht die CIM_DataFile-Klasse.

```

Option Explicit

Dim oWMI
Dim colFolder
Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colFolder = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Directory WHERE Name = 'C:\\LogFiles'")

For Each objItem in colFolder
    Err = objItem.Compress
Next

Set oWMI = Nothing

```

Listing 212: 2_02_017b.vbs

Und wenn Sie die soeben komprimierte Datei *C:\LogFiles\Protokoll.txt* wieder dekomprimieren möchten, so können Sie die gleichen Code-Zeilen wie beim Komprimieren der Datei verwenden, nur dass Sie diesmal statt der `Compress`-Methode die `Uncompress`-Methode gegen das Datei-Objekt anwenden.

```

Option Explicit

Dim oWMI
Dim colFiles

```

Listing 213: 2_02_017c.vbs

```

Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colFiles = oWMI.ExecQuery ("SELECT * FROM _
    CIM_Datafile WHERE Name = 'C:\\LogFiles\\Protokoll.txt'")

For Each objItem in colFiles
    Err = objItem.Uncompress
Next

Set oWMI = Nothing

```

Listing 213: 2_02_017c.vbs (Forts.)

Das Gleiche gilt für das Dekomprimieren eines Ordners. Um den Ordner *C:\LogFiles* wieder zu entpacken, nutzen Sie das gleiche Skript wie für die Komprimierung, nur dass Sie auch in diesem Fall die *Uncompress*-Methode gegen das Ordner-Objekt einsetzen.

```

Option Explicit

Dim oWMI
Dim colFolder
Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colFolder = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Directory WHERE Name = 'C:\\LogFiles'")

For Each objItem in colFolder
    Err = objItem.Uncompress
Next

Set oWMI = Nothing

```

Listing 214: 2_02_017d.vbs

44 Dateien und Ordner ver- und entschlüsseln

Auf ähnliche Art und Weise, wie Sie Dateien und Ordner komprimieren können, so können Sie die Objekte auch verschlüsseln und damit vor fremden Zugriffen schützen. Damit lassen sich Daten auch dann vor unautorisierten Zugriffen schützen, wenn die Festplatte in ein anderes Computersystem eingebaut wird. Denn die NTFS-Berechtigungen alleine würden in solch einem Fall nicht wirken.

Der Windows Server 2003 stellt hierfür eine eigene Verschlüsselungsfunktion bereit, die Sie allerdings nur auf einem mit NTFS formatierten Datenträger nutzen können und auch nur dann, wenn das betreffende Objekt nicht bereits komprimiert wurde. Denn die Komprimierungs- und Verschlüsselungsfunktion können nicht gemeinsam verwendet werden.

Wenn Sie die Datei *C:\LogFiles\SystemKonfig.log* verschlüsseln und die Datei damit vor fremden Zugriffen schützen möchten, so starten Sie den WINDOWS EXPLORER und rufen aus dem Kontextmenü der Datei die EIGENSCHAFTEN auf. Auf der Registerkarte ALLGEMEIN klicken Sie anschließend auf ERWEITERT, um in die erweiterten Einstellungen des Objektes zu gelangen. Dort aktivieren Sie die Option INHALT VERSCHLÜSSELN, UM DATEN ZU SCHÜTZEN und übernehmen die geänderte Einstellung mit einem Klick auf die Schaltfläche ÜBERNEHMEN. Die Datei wird daraufhin in codierter Form zurückgeschrieben und bleibt so lange verschlüsselt, bis sie entweder explizit entschlüsselt oder aber auf einen Datenträger verschoben wird, der nicht mit NTFS formatiert wurde.

Und genau so, wie Sie eine Datei verschlüsseln, können Sie auch einen kompletten Ordner verschlüsseln. Sie müssen nur festlegen, ob Sie lediglich den Ordner selbst oder aber auch dessen Inhalt verschlüsseln wollen. Wenn Sie den kompletten Ordner verschlüsseln, so werden auch alle neuen Objekte des Ordners automatisch verschlüsselt.

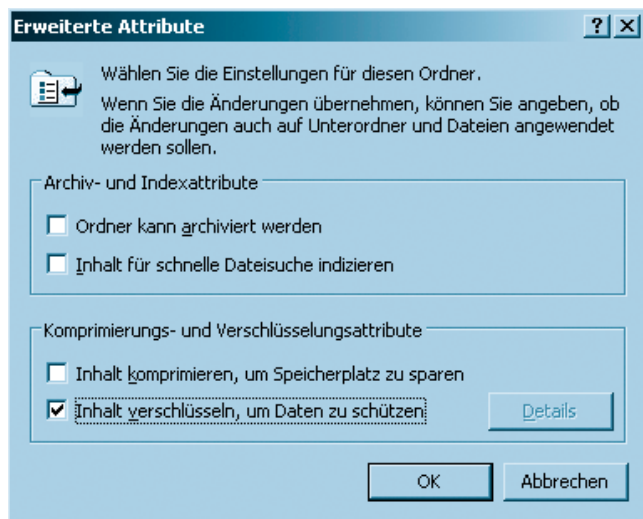


Abbildung 95: Einen Ordner verschlüsseln

Um eine verschlüsselte Datei oder einen verschlüsselten Ordner wieder zu dekodieren, rufen Sie erneut die EIGENSCHAFTEN des Objektes auf und deaktivieren in den erweiterten Einstellungen (Registerkarte ALLGEMEIN) die Option INHALT VERSCHLÜSSELN, UM DATEN ZU SCHÜTZEN.

Die Lösung per Eingabeaufforderung

Das Verschlüsseln von Dateien und Ordnern lässt sich natürlich auch in der Eingabeaufforderung bewerkstelligen, hierfür stellt der Windows Server 2003 den Cipher-Befehl bereit. Zum Verschlüsseln einer Datei erwartet der Cipher-Befehl zunächst die Parameter `/e` und `/a` wie auch die zu verschlüsselnde Datei. Der Parameter `/e` legt dabei fest, dass die Datei verschlüsselt (*encrypted*) werden soll, und mit `/a` bestimmen Sie, dass die Datei auch dann verschlüsselt werden soll, wenn der übergeordnete Ordner nicht verschlüsselt ist.

```
@Echo off
Cipher /e /a C:\LogFiles\Protokoll.txt
```

Listing 215: 2_02_018a.bat

Um einen kompletten Ordner zu verschlüsseln, wenden Sie ebenfalls den Cipher-Befehl unter Angabe des Parameter `/e` an und übergeben mit `/s:` den betreffenden Ordner. Mit dem zusätzlichen Parameter `/f` können Sie das Verschlüsseln aller Objekte in dem angegebenen Ordner erzwingen (selbst wenn diese bereits verschlüsselt sind), und durch Angabe des Parameters `/i` legen Sie fest, dass der Befehl auch dann weiter ausgeführt wird, falls es bei einem der enthaltenen Objekten zu einem Fehler während der Verschlüsselung kommt. Ohne Angabe des Parameters `/i` würde die Befehlsausführung abgebrochen werden, was oftmals nicht erwünscht ist.

```
@Echo off
Cipher /e /s:C:\LogFiles /f /i
```

Listing 216: 2_02_018b.bat

Der Cipher-Befehl erlaubt natürlich nicht nur das Verschlüsseln von Dateien und Ordnern, sondern auch deren spätere Entschlüsselung. Wenn Sie beispielsweise die Datei `C:\LogFiles\Protokoll.txt` mit dem Cipher `/e`-Befehl verschlüsselt haben, so können Sie die Datei mit dem Cipher-Befehl unter Angabe des Parameters `/d` (*decrypted*) wieder entschlüsseln, wobei Sie bei einer einzelnen Datei den Parameter `/a` anfügen müssen. Das Entschlüsseln einzelner Dateien funktioniert allerdings nur, wenn der übergeordnete Ordner nicht verschlüsselt ist, da die Datei anderenfalls das Verschlüsselungs-Flag des Ordners erben würde.

```
@Echo off
Cipher /d /a C:\LogFiles\Protokoll.txt
```

Listing 217: 2_02_018c.bat

Einen Ordner entschlüsseln Sie auf die gleiche Art und Weise, wofür Sie dem Cipher-Befehl zunächst den Parameter /d und dann mit /s: den Namen des Ordners übergeben. Um das Entschlüsseln zu erzwingen, fügen Sie der Befehlszeile den Parameter /f an, und mit /i bestimmen Sie, dass die Entschlüsselung auch im Falle eines Fehlers weiter ausgeführt werden soll.

```
@Echo off
Cipher /d /s:C:\LogFiles /f /i
```

Listing 218: 2_02_018d.bat

Die Lösung mit VBScript

Mit VBScript lässt sich zwar auf recht einfache Art und Weise ermitteln, ob eine Datei (oder ein Ordner) verschlüsselt ist, für die Verschlüsselung selbst stellt das Betriebssystem aber leider keine entsprechende Methode bereit. Will heißen, dass Sie Dateien und Ordner nicht per VBScript verschlüsseln können, sofern Sie nicht auf alternative Möglichkeiten zurückgreifen.

Die Run-Methode des Shell-Objektes erlaubt jedoch die Ausführung von Shell-Befehlen, womit Sie den Compact-Befehl auch aus einem VBScript heraus aufrufen und so dennoch eine Windows Server 2003-kompatible Verschlüsselung durchführen können. Hierzu erzeugen Sie mit der CreateObject-Methode lediglich eine neue Instanz des Shell-Objektes und wenden dann die Run-Methode an, der Sie die Compact-Befehlszeile als Argument übergeben.

Das nachfolgende Code-Beispiel verschlüsselt die Datei *C:\LogFiles\Protokoll.txt* unter Verwendung des Compact-Befehls, der mit Hilfe der Run-Methode des Shell-Objektes aufgerufen wird:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Cipher /e /a _
C:\LogFiles\Protokoll.txt"

Set oShell = Nothing
```

Listing 219: 2_02_018a.vbs

Die Entschlüsselung eines Objektes erfolgt auf dem gleichen Weg, nur dass Sie dem Compact-Befehl den Parameter /d für die Entschlüsselung übergeben und nicht den Parameter /c, womit Sie Dateien und Ordner verschlüsseln können.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Cipher /d /a _
C:\LogFiles\Protokoll.txt"

Set oShell = Nothing
```

Listing 220: 2_02_018b.vbs

45 Dateien splitten und wieder zusammenfügen

Hin und wieder kommt es vor, dass Dateien kopiert werden müssen, deren Größe die freie Speicherkapazität des Zieldatenträgers überschreitet. Dies ist beispielsweise des Öfteren der Fall, wenn Sie ein Abbild der Partition oder auch des gesamten Datenträgers erstellen und die Image-Datei auf ein MicroDrive kopieren oder auch auf CD brennen möchten. Oder wenn Sie eine große Anlage per E-Mail versenden möchten und dabei an die Größenbegrenzungen des Mailsystems stoßen. In solch einem Fall hilft meist auch das Komprimieren der Datei nicht weiter. Es bleibt also nur, die Datei in mehrere Fragmente aufzuteilen und dann Stück für Stück zu kopieren. Sprich: die Datei zu splitten und später wieder zusammenzufügen.

Leider bringt der Windows Server 2003 kein entsprechendes Tool mit, womit Sie eine Datei in mehrere Teile aufteilen und bei Bedarf wieder zusammenfügen können. Sie müssen in diesem Fall auf den Einsatz eines externen Tools zurückgreifen. Der Free- und Shareware-Markt hält jedoch einige nützliche Tools bereit, die Sie hierfür einsetzen können. Auf der Internetseite www.maklo.de finden Sie beispielsweise das Freeware-Tool mit der Bezeichnung *FileSplit*, womit Sie übergroße Dateien in unterschiedliche Größen aufteilen und mit Hilfe einer automatisch generierten Batch-Datei später auch wieder problemlos zusammenfügen können.

Wenn Sie eine Image-Datei mit einer Größe von 800 MByte aufteilen möchten, so laden Sie sich zunächst das zuvor genannte Tool von der Internetseite und installieren es auf dem betreffenden Computersystem. Anschließend starten Sie *FileSplit* aus dem STARTMENÜ heraus und legen zunächst die Größe der Dateifragmente fest. Anschließend klicken Sie auf START, woraufhin das Dialogfenster DATEI ZUM SPLITTEN AUSWÄHLEN geöffnet wird. Wählen Sie die Datei, die Sie splitten möchten, aus, und klicken Sie anschließend auf ÖFFNEN.

Die Datei wird daraufhin in mehrere Fragmente aufgeteilt und auf dem Datenträger abgelegt. Während dem Split-Vorgang wird zeitgleich auch eine BatchDatei erstellt, mit Hilfe derer Sie die Dateifragmente später wieder zusammenfügen können. Dieses

Verfahren hat den Vorteil, dass Sie die *FileSplit*-Applikation nicht auch auf dem Zielsystem installieren müssen, um die Fragmente wieder zusammenzufügen.

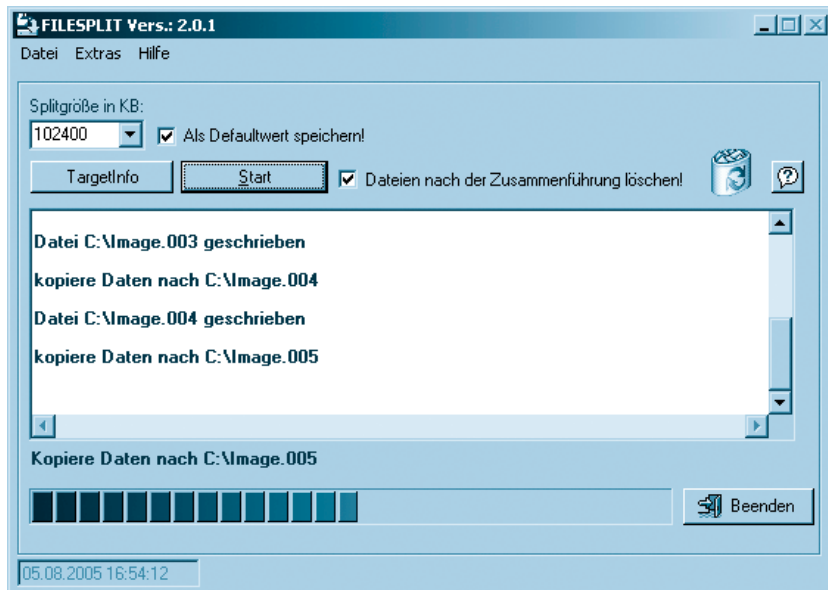


Abbildung 96: Eine Datei splitten

Wenn Sie die Dateifragmente auf das Zielsystem übertragen haben, dann führen Sie lediglich die Batch-Datei aus, woraufhin die einzelnen Dateifragmente per Copy-Befehl zu einer vollständigen Datei zusammengefügt werden.

Die Lösung per Eingabeaufforderung

Mit Hilfe der *GNU-Utilities für Win32* lässt sich eine Datei auch in der Eingabeaufforderung problemlos splitten. Denn in den GNU-Utilities ist das Tool *Split.exe* enthalten, womit Sie einzelne Dateien in beliebige Größen (oder im Falle einer Textdatei auch in beliebige Zeilen) aufteilen und später mit dem Copy- oder Xcopy-Befehl auch wieder zusammenfügen können.

Wenn Sie z.B. eine Datei namens *Image.pqi* mit einer Größe von 800 MByte in drei Fragmente aufteilen möchten, so wenden Sie den *Split*-Befehl an und übergeben mit dem Parameter *-b* zunächst die gewünschte Größe der Dateifragmente, wobei die Angabe der Größe standardmäßig in Bytes erfolgt. Mit Hilfe eines zusätzlichen Flags (siehe nachfolgende Tabelle) können Sie die Größe der Dateifragmente aber auch in KBytes oder MBytes angeben.

Flag	Beschreibung
b	Bytes
k	KBytes
m	MBytes

Tabelle 35: Die Größenbezeichnungen für den Split-Befehl

Des Weiteren geben Sie den Namen der Datei an, die aufgeteilt werden soll, und wahlweise auch das Präfix für die Namen der Dateifragmente. Der `Split`-Befehl verwendet standardmäßig das Präfix `x`, womit die Dateifragmente `xaa`, `xab`, `xac`, `xad` ... fortlaufend benannt werden. Sie können aussagekräftigere Dateinamen verwenden, wenn Sie den Namen der aufzuteilenden Datei als Präfix verwenden (z.B. `Imageaa`, `Imageab`, `Imageac` ...).

Das nachfolgende Beispiel teilt die Datei `C:\Temp\Image.pqi` in 300.000 KByte große Fragmente auf und verwendet die Bezeichnung `Image` als Präfix für die Dateinamen:

```
@Echo off
Split -b 300000k C:\Temp\Image.pqi Image
```

Listing 221: 2_02_019a.bat

Eine Textdatei lässt sich dabei nicht nur nach der Größe aufteilen, sondern auch nach der Anzahl der Zeilen. Wenn Sie z.B. eine Textdatei mit 100 Zeilen in zehn Dateien à zehn Zeilen aufteilen möchten, so ersetzen Sie den Parameter `-b` durch `-l` und übergeben nicht die Größe der Dateifragmente, sondern die Anzahl der gewünschten Zeilen.

```
@Echo off
Split -l 10 C:\LogFiles\Protokoll.txt Protokoll
```

Listing 222: 2_02_019b.bat

Wenn Sie die Datei-Fragmente zu einem späteren Zeitpunkt wieder zusammenfügen möchten, so verwenden Sie hierfür den Windows-eigenen `Copy`- oder `Xcopy`-Befehl und kopieren die einzelnen Fragmente in eine gemeinsame Zieldatei, wobei Sie die exakte Reihenfolge einhalten müssen. Um die Dateifragmente `Imageaa`, `Imageab` und `Imageac` zu einer Datei namens `Image.pqi` zusammenzufügen, wenden Sie folgenden `Copy`-Befehl unter Angabe des Parameters `/b` an, da in diesem Fall Binärdateien kopiert werden sollen.

```
@Echo off
Copy /b Imageaa + Imageab + Imageac Image.pqi
```

Listing 223: 2_02_019c.bat

Die Lösung mit VBScript

Mit der Run-Methode des Shell-Objektes lässt sich das Tool *Split.exe*, das Bestandteil der *GNU-Utilities für Win32* ist, auch aus einem VBScript heraus aufrufen. Um beispielsweise die Binärdatei *Image.pqi* in 300.000 KByte große Fragmente aufzuteilen, referenzieren Sie zunächst das Shell-Objekt und wenden dann die Run-Methode unter Angabe des zu verwendenden Befehlszeileninterpreters (%Comspec%) und des auszuführenden Split-Befehls an, wobei Sie Letzterem die Größe der Fragmente (-b 300000k) und auch die aufzuteilende Datei (C:\Temp\Image.pqi) als Parameter übergeben.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Split -b 300000k _
C:\Temp\Image.pqi Image"

Set oShell = Nothing
```

Listing 224: 2_02_019a.vbs

Auf die gleiche Art und Weise lässt sich auch eine Textdatei in eine bestimmte Anzahl an Zeilen aufteilen. Wenn Sie z.B. die Datei *C:\LogFiles\Protokoll.txt* in mehrere Fragmente à zehn Zeilen splitten möchten, so nutzen Sie wieder die Run-Methode des Shell-Objektes und führen das Tool *Split.exe* aus, dem Sie die Einheit (-l für Zeilen), die Anzahl (10) und auch die aufzuteilende Datei (*C:\LogFiles\Protokoll.txt*) als Parameter übergeben.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Split -l 10 _
C:\LogFiles\Protokoll.txt"

Set oShell = Nothing
```

Listing 225: 2_02_019b.vbs

Das Zusammenfügen der einzelnen Dateifragmente nehmen Sie – um bei der gleichen Methode zu bleiben – mit dem Copy-Befehl der Eingabeaufforderung vor, dem Sie die einzelnen Dateifragmente wie auch die Zieldatei als Parameter übergeben.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Copy /b Imageaa + Imageab + _
Imageac Image.pqi"

Set oShell = Nothing
```

Listing 226: 2_02_019c.vbs

46 Testdateien mit einer bestimmten Größe anlegen

Für den Fall, dass Sie zu Test- oder Troubleshooting-Zwecken eine Testdatei mit einer bestimmten Größe benötigen, so bleibt leider nur der Einsatz eines externen Tools oder eines Skriptes, da der Windows Server 2003 hierfür kein entsprechendes Bordmittel bereithält.

Sie können z.B. das Tool *Creatfil.exe* (siehe *Lösung per Eingabeaufforderung*) aus dem *Windows Server 2003 Resource Kit* verwenden, um eine solche Testdatei zu erzeugen. Oder Sie erstellen per VBScript eine Textdatei (siehe *Lösung mit VBScript*) und füllen die Datei mit einer entsprechenden Anzahl an Zeichen, bis die gewünschte Größe erreicht ist. Die nachfolgenden Abschnitte zeigen, auf welcher einfachen Art und Weise sich Testdateien mit einer bestimmten Größe erstellen lassen.

Die Lösung per Eingabeaufforderung

Die einfachste und zugleich eleganteste Lösung, eine Testdatei mit einer bestimmten Größe anzulegen, bietet das Tool *Creatfil.exe*, das Bestandteil der *Windows Server 2003 Resource Kit-Tools* ist. Denn der *Creatfil*-Befehl erwartet lediglich den Namen der zu erstellenden Datei wie auch die gewünschte Größe in Bytes als Parameter und erzeugt daraufhin ein entsprechendes Datei-Objekt.

Um beispielsweise im Ordner *C:\Temp* eine Datei mit der Bezeichnung *TestFile* und einer Größe von 100.000 Bytes anzulegen, verwenden Sie folgende Befehlszeile:

```
@Echo off
CreatFil C:\Temp\TestFile 100000
```

Listing 227: 2_02_020.bat

Die Lösung mit VBScript

Mit der *Write*-Methode des *FileSystem*-Objektes lassen sich ebenfalls Testdateien mit einer bestimmten Größe anlegen, da Sie nur eine neue Textdatei anlegen und diese mit einer bestimmten Anzahl an Zeichen füllen müssen. Wenn Sie beispielsweise eine

Datei mit einer Größe von 100.000 Bytes benötigen, so erzeugen Sie mit der `CreateObject`-Methode erst eine neue Instanz des `FileSystem`-Objektes, erstellen mit `OpenTextFile` die betreffende Datei und füllen dann das Datei-Objekt mit 100.000 Zeichen, indem Sie mit der `Write`-Methode ein Zeichen in die Ausgabedatei schreiben und die Methode so oft anwenden, wie Sie Zeichen, sprich: Bytes, benötigen.

Option Explicit

```
Dim oFS
Dim oFile
Dim strFile
Dim strText
Dim Counter
Dim maxCount
```

```
Const Write = 2
strFile = "C:\Temp\TestFile.txt"
MaxCount = 100000
```

```
Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFile = oFS.OpenTextFile(strFile, Write, True)
```

```
For Counter = 1 to maxCount
    oFile.Write "0"
Next
```

```
oFile.Close
Set oFile = Nothing
Set oFS = Nothing
```

Listing 228: 2_02_020.vbs

47 Geöffnete Dateien überschreiben

Wenn Sie eine einzelne Datei durch eine andere Dateiversion ersetzen müssen, so muss die Zieldatei, die überschrieben werden soll, geschlossen sein. Denn anderenfalls verweigert das Betriebssystem den Zugriff auf die zu überschreibende Datei, und der Vorgang bricht ab. Nun kann es aber vorkommen, dass Sie eine wichtige Systemdatei überschreiben und durch eine andere Version ersetzen müssen, die bereits während des Bootvorgangs des Betriebssystems geladen und erst beim Herunterfahren des Computersystems wieder freigegeben wird.

Unter grafischer Oberfläche lässt sich eine solche Datei nur dann überschreiben, wenn die Datei von einem Prozess verwendet wird, der sich temporär beenden lässt, womit die Datei dann wieder freigegeben wird. Handelt es sich aber um eine Datei, die zur Ausführung des Betriebssystems zwingend erforderlich ist, so können Sie noch versuchen, die entsprechende Datei mit Hilfe der Recovery Console von Win-

dows Server 2003 zu überschreiben, da Sie mit der Recovery Console auf das Dateisystem zugreifen können, ohne dass Windows Server 2003 im normalen oder zumindest im abgesicherten Modus gestartet sein muss.

Weitaus komfortabler lässt sich eine zur Laufzeit des Betriebssystems geöffnete Datei aber mit dem `Inuse`-Befehl (siehe nachfolgenden Abschnitt) überschreiben, der zum festen Befehlsumfang von Windows Server 2003 gehört. Denn der `Inuse`-Befehl tauscht die angegebene Quelldatei mit der Zieldatei zu Beginn des nächsten Bootvorgangs automatisch aus und erlaubt damit das Ersetzen einer Datei, noch bevor diese vom Betriebssystem verwendet wird.

Die Lösung per Eingabeaufforderung

Um eine Datei zu überschreiben, die während der Ausführung des Betriebssystems oder einer bestimmten Applikation in Benutzung ist, wenden Sie in der Eingabeaufforderung den `Inuse`-Befehl an und übergeben die Quell- und Zieldatei als Parameter. Mit dem zusätzlichen Parameter `/Y` können Sie das Überschreiben der Datei auch erzwingen und vermeiden damit die lästige Rückfrage, ob die Datei auch wirklich überschrieben werden soll.

Die nachfolgende `Inuse`-Befehlszeile zeigt ein Beispiel dafür, wie Sie mit dem nächsten Computerneustart die Datei `C:\Windows\System32\Svchost.exe` durch die Datei `C:\Windows\%NTService PackUninstall%\Svchost.exe` ersetzen und mit dem Parameter `/Y` das Überschreiben der Quelldatei erzwingen können.

```
@Echo off
Inuse C:\Windows\%NTServicePackUninstall%\Svchost.exe _
      C:\Windows\System32\Svchost.exe /Y
```

Listing 229: 2_02_021.bat

Die Lösung mit VBScript

VBScript bietet leider keine Möglichkeit an, womit Sie geöffnete Dateien so ohne weiteres überschreiben können. Mit der `Run`-Methode des `Shell`-Objektes können Sie aber den `Inuse`-Befehl der Eingabeaufforderung auch aus einem VBScript heraus ausführen und das Überschreiben einer geöffneten Datei damit auch per VBScript starten.

Das nachfolgende Code-Beispiel führt mit Hilfe der `Run`-Methode den `Inuse`-Befehl aus und erzwingt das Ersetzen der Datei `C:\Windows\System32\Svchost.exe` durch die Datei `C:\Windows\%NTServicePackUninstall%\Svchost.exe`:

```
Option Explicit
Dim oShell
```

Listing 230: 2_02_021.vbs

```
Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Inuse C:\Windows\$_NTServicePackUninstall$_
  Svchost.exe C:\Windows\System32\Svchost.exe /Y"
```

```
Set oShell = Nothing
```

Listing 230: 2_02_021.vbs (Forts.)

48 Neue Dateitypen registrieren

In der Regel sind alle Dateiendungen (z.B. *.txt, *.log, *.vbs ...), die Sie auf einem Computersystem vorfinden, mit einer der installierten Applikationen verknüpft. Wenn Sie z.B. im WINDOWS EXPLORER eine Textdatei (*.txt) doppelklicken, so wird automatisch die verknüpfte Anwendung (z.B. *Notepad.exe*) gestartet und die Datei darin geöffnet.

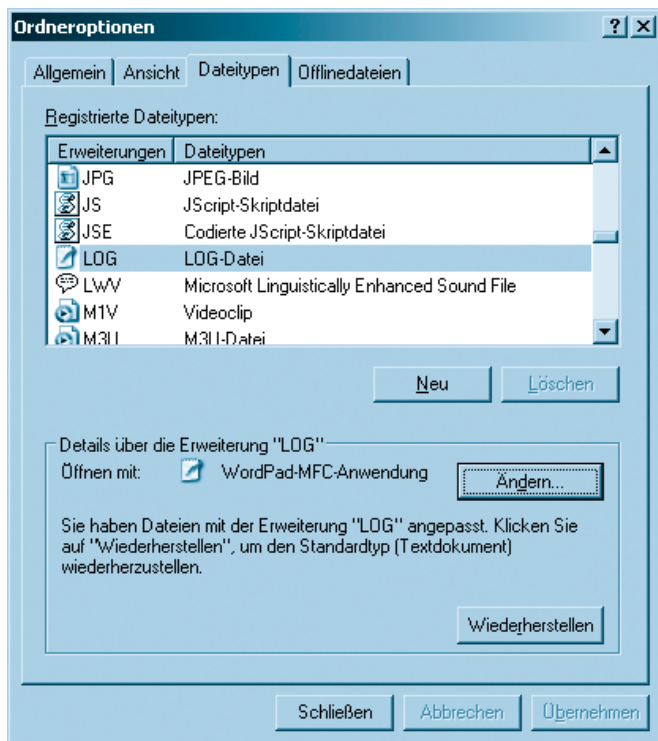


Abbildung 97: Einen neuen Dateityp registrieren

Für den Fall, dass eine bestimmte Dateierweiterung noch nicht (oder vielleicht fehlerhaft) mit einer Applikation verknüpft wurde, so können Sie dies recht schnell ändern. Starten Sie hierzu den WINDOWS EXPLORER, und wählen Sie im Menü EXTRAS den Eintrag ORDNEROPTIONEN aus. Wechseln Sie anschließend auf die Registerkarte DATEITY-

PEN, und wählen Sie die betreffende Dateierweiterung aus der Liste der Dateitypen aus, die Sie ändern möchten. Sollte die Dateierweiterung noch nicht in der Liste enthalten sein, so können Sie das Suffix auch neu erstellen, indem Sie auf die Schaltfläche NEU klicken und dann die Dateierweiterung eingeben (siehe Abbildung 97).

Wenn Sie die betreffende Dateierweiterung ausgewählt haben, klicken Sie auf die Schaltfläche ÄNDERN, um die entsprechende Applikation auszuwählen. Sie gelangen daraufhin in das Dialogfenster ÖFFNEN MIT, worin einige Standardapplikationen angezeigt werden. Wenn die Applikation, mit der Sie die Dateierweiterung verknüpfen möchten, in der Liste enthalten ist, so markieren Sie den Eintrag und klicken anschließend auf OK. Anderenfalls führen Sie einen Klick auf die Schaltfläche DURCHSUCHEN aus und wählen die gewünschte Applikation aus. Zu guter Letzt schließen Sie das Fenster mit einem Klick auf OK.

Die Lösung per Eingabeaufforderung

Mit dem `Assoc`-Befehl, der standardmäßig zum Befehlsumfang von Windows Server 2003 gehört, lässt sich ein Dateityp auch von der Eingabeaufforderung aus mit einer installierten Applikation verknüpfen. Allerdings findet diese Verknüpfung nicht auf direktem Wege (z.B. `*.txt -> Notepad.exe`) statt, sondern auf indirektem Wege. Das heißt, dass Sie das zu verknüpfende Dateisuffix (`*.txt`) erst einem vordefinierten Dateityp (`TxtFile`) zuweisen müssen. Der Dateityp entscheidet dann darüber, mit welcher Applikation das entsprechende Dateisuffix geöffnet wird. Welche Dateitypen zur Verfügung stehen und welche Verknüpfungen bereits vorgenommen wurden, können Sie sich durch die Eingabe von `Assoc` (ohne Angabe weiterer Parameter) jederzeit anzeigen lassen.

Mit dem `Assoc`-Befehl können Sie nun von der Eingabeaufforderung aus eine Dateierweiterung einem Dateityp zuweisen, wobei der `Assoc`-Befehl aber keine Möglichkeit bereitstellt, auch die dazugehörige Applikation auszuwählen. Dies geht nur über die grafische Oberfläche oder durch direktes Editieren der Registrierungseinstellungen. Doch dazu gleich mehr.

Einmal angenommen, Sie haben eine oder auch mehrere Dateien, deren Dateierweiterung `*.TeX` lautet. Nun möchten Sie diese Dateien einem Editor (z.B. `Notepad.exe` oder `Wordpad.exe`) zuweisen und die Datei künftig per Doppelklick öffnen können, ohne jedes Mal die Applikation, mit der die Datei geöffnet werden soll, auswählen zu müssen. In diesem Fall können Sie nun den `Assoc`-Befehl einsetzen und die Dateierweiterung `*.TeX` dem Dateityp `TxtFile` zuweisen, womit künftig alle `*.TeX`-Dateien als `TxtFile`-Dateien behandelt und daher mit dem Standardeditor `Notepad.exe` geöffnet werden. Letzteres ergibt sich dadurch, dass in den Registrierungseinstellungen des Dateityps (also `TxtFile`) die Befehlszeile zur Applikation hinterlegt ist.

```
@Echo off
Assoc *.TeX=TxtFile
```

Listing 231: `2_02_022a.bat`

Um eine solche Verknüpfung wieder zu lösen, übergeben Sie dem `Assoc`-Befehl lediglich die Dateierweiterung und verzichten auf die Angabe des Dateityps.

```
@Echo off
Assoc .TeX=
```

Listing 232: 2_02_022b.bat

Alternativ dazu können Sie die Änderung natürlich auch direkt in die Registrierung schreiben, wofür Sie in der Eingabeaufforderung den `Reg Add`-Befehl verwenden können. Diese Alternative hat den Vorteil, dass Sie dabei auch neue Dateitypen registrieren und andere Applikationen zuweisen können. Wenn Sie z.B. die Dateierweiterung `*.TeX` registrieren und mit der Applikation *Wordpad.exe* verknüpfen möchten, so erzeugen Sie zunächst einen neuen Registrierungseintrag für die Dateierweiterung und hinterlegen darin einen Link zum dazugehörigen Dateityp (z.B. *MyFile*). Anschließend erstellen Sie einen weiteren Registrierungseintrag für den Dateityp und nehmen darin die Befehlszeile auf, worüber die verknüpfte Anwendung gestartet wird.

Sicherlich könnten Sie die Befehlszeile auch direkt in der Registrierungseinstellung der Dateierweiterung hinterlegen. Der Konformität mit den Standardregistrierungseinstellungen wegen sollten Sie dieses Verfahren aber beibehalten, um fehlerhafte Verknüpfungen mit anderen Dateierweiterungen und -typen zu vermeiden.

Die nachfolgende `Reg Add`-Befehlszeile erzeugt in der Registrierung, genauer gesagt unter `HKEY_CLASSES_ROOT`, einen neuen Schlüssel mit der Bezeichnung `.TeX` und hinterlegt in der Zeichenfolge (*Default*) den Dateityp mit der Bezeichnung *MyFile*. Damit werden nun alle Dateien mit der Endung `*.TeX` dem Dateityp *MyFile* zugeordnet.

```
@Echo off
Reg Add HKCR\TEX /ve /t REG_SZ /d MyFile /f
```

Listing 233: 2_02_022c.bat

Nun müssen Sie mit dem `Reg Add`-Befehl den Dateityp (z.B. *MyFile*) erzeugen und die entsprechende Applikation zuweisen. Nachfolgend finden Sie ein Beispiel dafür, wie Sie die Registrierungseinstellung für den Dateityp *MyFile* erzeugen und so konfigurieren können, dass ab sofort die Applikation *Wordpad.exe* gestartet wird, wenn eine Datei mit der Endung `*.TeX` per Doppelklick gestartet wird.

```
@Echo off
Reg Add HKCR\MyFile\Shell\Open\Command /ve/t REG_EXPAND_SZ _
/d "C:\Programme\Windows NT\Wordpad.exe" /f
```

Listing 234: 2_02_022d.bat

Die Lösung mit VBScript

Um per VBScript die Zuordnung zwischen einer Dateieindung und einem Dateityp vorzunehmen, nutzen Sie die gleiche Möglichkeit wie auch in der Eingabeaufforderung. Sprich: Sie schreiben die Einstellung direkt in die Registrierung.

Wenn Sie beispielsweise die Dateieindung *.TeX dem Dateityp *MyFile* zuweisen möchten, so stellen Sie im ersten Schritt eine Verbindung zur WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`) her. Anschließend legen Sie mit der `CreateKey`-Methode unterhalb von `HKEY_CLASSES_ROOT` einen neuen Schlüssel mit der Bezeichnung `.TeX` an, wofür Sie der `CreateKey`-Methode den hexadezimalen Wert des Hauptzweiges (in diesem Fall `&H80000000`) und auch den Schlüsselnamen (`.TeX`) übergeben. Danach erzeugen Sie in dem soeben erstellten Schlüssel mit der `SetStringValue`-Methode eine neue Standardzeichenfolge und weisen ihr den Wert *MyFile* zu. Damit ist die Dateieindung *.TeX mit dem Dateityp *MyFile* verknüpft.

Option Explicit

```
Dim oReg
Dim strComputer
Dim strKey
Dim strValueName
Dim strValue
```

```
Const HKCR = &H80000000
strComputer = "."
strKey = ".TeX"
strValueName = ""
strValue = "MyFile"
```

```
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\default:StdRegProv")
oReg.CreateKey HKCR,strKey
oReg.SetStringValue HKCR, strKey, strValueName, strValue
```

```
Set oReg = Nothing
```

Listing 235: 2_02_022a.vbs

Um den Dateityp *MyFile* mit der Applikation *Wordpad.exe* zu verknüpfen, nutzen Sie ein ähnliches Listing und erzeugen mit der `CreateKey`-Methode zunächst den Schlüssel *MyFile* und weisen dann mit der `SetStringValue`-Methode die Befehlszeile zur Applikation (in diesem Fall `C:\Programme\Windows NT\Zubehör\Wordpad.exe`) zu.

```

Option Explicit

Dim oReg
Dim strComputer
Dim strKey
Dim strValueName
Dim strValue

Const HKCR = &H80000000
strComputer = "."
strKey = "MyFile"
strValueName = ""
strValue = "C:\Programme\Windows NT\Zubehör\Wordpad.exe"

Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\default:StdRegProv")
oReg.CreateKey HKCR,strKey
oReg.SetStringValue HKCR, strKey, strValueName, strValue

Set oReg = Nothing

```

Listing 236: 2_02_022b.vbs

49 Kopf- und Fußzeilen aus einer Textdatei entfernen

Viele Textdateien (insbesondere Export-, Protokoll- und sonstige Datendateien) enthalten am Anfang eine oder auch mehrere Kopfzeilen mit Dateibezeichnungen, Datums- und Uhrzeitinformationen oder einfach nur mit Spaltenüberschriften und oftmals am Ende der Datei auch Fußzeilen mit Zusammenfassungen etc. Wenn Sie ein solches Dateilayout automatisiert bearbeiten (z.B. auf- oder absteigend sortieren oder mit Zeilennummern versehen) möchten, so müssen Sie zunächst die Kopf- und Fußzeilen herauslöschen, da sie sonst als gewöhnliche Datenzeilen interpretiert und in die Verarbeitung mit einbezogen werden.

Um eine Datei unter grafischer Oberfläche von jeglichen Kopf- und Fußzeilen zu befreien, nehmen Sie einen der Windows-eigenen Editoren (NOTEPAD oder WORDPAD) zu Hilfe und löschen die betreffenden Zeilen von Hand heraus, indem Sie die jeweiligen Zeilen markieren und entweder im Kontextmenü den Eintrag LÖSCHEN wählen oder die Taste Entf drücken (siehe Abbildung 98).

Die Lösung per Eingabeaufforderung

Wenn Sie die Kopf- und Fußzeilen einer Textdatei von der Eingabeaufforderung aus entfernen möchten, so können Sie im einfachsten Fall den Editor der Eingabeaufforderung (*Edit.exe*) verwenden und – wie auch unter grafischer Oberfläche – die betreffenden Zeilen von Hand herauslöschen.

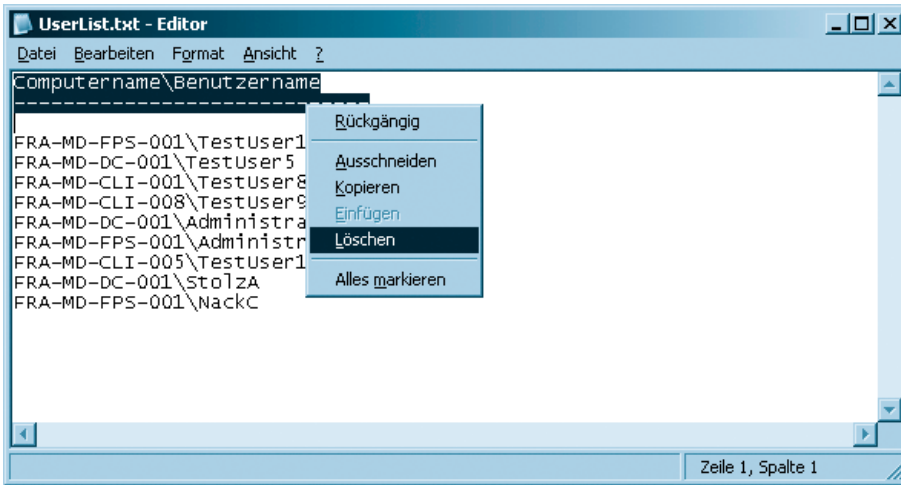


Abbildung 98: Die Kopfzeilen einer Textdatei entfernen

Sie können aber auch die Tools *Tail.exe* und *Head.exe* verwenden, die Bestandteil der *GNU-Utilities für Win32* sind, wobei das Tool *Tail.exe* aber auch in den *Windows Server 2003 Resource Kit-Tools* zu finden ist.

Mit *Tail.exe* haben Sie die Möglichkeit, die Ausgabe ab einer bestimmten Zeile oder auch die letzten *n* Zeilen einer Textdatei auszulesen und per Ausgabeumleitung in eine gesonderte Datei zu schreiben, womit sich auf recht einfache Art und Weise die Kopfzeilen herauslöschten lassen. Denn haben Sie beispielsweise eine Textdatei mit einer unbestimmten Anzahl an Zeilen, wovon die ersten vier Zeilen jedoch Kopfzeilen sind, so können Sie mit *Tail +5* alle Zeilen über der Zeilennummer 5 und damit nur die eigentlichen Datenzeilen auslesen, da in diesem Fall die ersten vier Zeilen ignoriert und nicht verarbeitet werden.

Die nachfolgende *Tail*-Befehlszeile liest z.B. den Inhalt der Datei *C:\Userlist.txt* ab der fünften Zeile aus und schreibt das Ergebnis per Ausgabeumleitung in die Datei *C:\Userlist_New.txt*.

```
@Echo off
Tail +5 C:\Userlist.txt > C:\Userlist_New.txt
```

Listing 237: 2_02_023a.bat

Mit dem Tool *Head.exe* können Sie dagegen die ersten *n* Zeilen einer Textdatei auslesen. Wenn Sie eine Textdatei mit 100 Zeilen haben, wovon die letzten vier Zeilen Fußzeilen sind, so können Sie mit *Head -n 96* die ersten 96 Zeilen einlesen und per Ausgabeumleitung in eine gesonderte Textdatei schreiben. Die Ausgabedatei enthält damit nur die eigentlichen Datenzeilen, da die letzten vier Zeilen, sprich: die Fußzeilen, ignoriert werden.

```
@Echo off
Head -n 96 C:\Userlist.txt > C:\Userlist_New.txt
```

Listing 238: 2_02_023b.bat

Der Einsatz dieser beiden Tools setzt natürlich voraus, dass die Anzahl der Zeilen und auch die Anzahl der Kopf-/Fußzeilen bekannt sind. Die Anzahl der Zeilen, die in einer Textdatei enthalten sind, lassen sich in der Eingabeaufforderung mit dem GNU-Utility `Wc.exe` (siehe Die Zeilenanzahl einer Textdatei ermitteln auf Seite 349) ermitteln.

Die Lösung mit VBScript

Mit der `ReadLine`-, `SkipLine`- und `WriteLine`-Methode des `FileSystem`-Objektes lassen sich die Kopfzeilen einer Textdatei auch per VBScript auf recht einfache Art und Weise herauslöschen. Hierzu öffnen Sie die betreffende Textdatei im Lesemodus und überspringen mit der `SkipLine`-Methode die ersten `n` Zeilen. Anschließend lesen Sie den restlichen Inhalt der Textdatei mit der `ReadLine`-Methode zeilenweise ein und schreiben den Zeileninhalt mit der `WriteLine`-Methode in eine Ausgabedatei.

Das nachfolgende Listing zeigt ein solches Beispiel: Zunächst wird mit der `OpenTextFile`-Methode die Quelldatei `C:\UserList.txt` (`strInputFile`) im Lesemodus und die Zieldatei `C:\UserList_New.txt` (`strOutputFile`) im Schreibmodus geöffnet. Um die ersten vier Zeilen (die Kopfzeilen) der Quelldatei zu überspringen und den Positionszeiger in die Spalte 1 der Zeile 5 zu setzen, wird die `SkipLine`-Methode in einer `For Next`-Schleife angewendet. Nach dem Setzen des Positionszeigers wird der Inhalt der Quelldatei mit der `ReadLine`-Methode zeilenweise eingelesen und per `WriteLine`-Methode in die Zieldatei geschrieben. Dies erfolgt innerhalb einer `Do Loop`-Schleife, die so lange wiederholt wird, bis das Ende der Quelldatei erreicht ist.

Option Explicit

```
Dim oFS
Dim oInputFile
Dim oOutputFile
Dim strInputFile
Dim strOutputFile
Dim strText
Dim intCounter

Const Read = 1
Const Write = 2
strInputFile = "C:\UserList.txt"
strOutputFile = "C:\UserList_New.txt"
```

Listing 239: 2_02_023a.vbs

```

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oInputFile = oFS.OpenTextFile(strInputFile, Read)
Set oOutputFile = oFS.OpenTextFile(strOutputFile, Write, True)

For intCounter = 1 to 4
    oInputFile.SkipLine
Next

Do While Not (oInputFile.AtEndOfStream)
    strText = oInputFile.ReadLine
    oOutputFile.WriteLine strText
Loop

oInputFile.Close
oOutputFile.Close
Set oInputFile = Nothing
Set oOutputFile = Nothing
Set oFS = Nothing

```

Listing 239: 2_02_023a.vbs (Forts.)

Wenn Sie dagegen die Fußzeilen und damit die letzten *n* Zeilen aus einer Textdatei löschen möchten, so erzeugen Sie zunächst ein dynamisches Array, lesen den Inhalt der Quelldatei mit der `ReadLine`-Methode zeilenweise ein und übertragen die Zeileninhalte jeweils in das Array, das Sie mit der `Redim Preserve`-Anweisung sukzessive vergrößern. Anschließend setzen Sie die `WriteLine`-Methode ein, um die Array-Elemente in eine gesonderte Ausgabedatei zu schreiben, wobei Sie als Zähler für die `For Next`-Schleife die Anzahl der Zeilen (= Größe des Arrays) abzüglich der Fußzeilen verwenden.

```

Option Explicit

Dim oFS
Dim oInputFile
Dim oOutputFile
Dim arrLines()
Dim intCounter

Const Read = 1
Const Write = 2
intCounter = 0

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oInputFile = oFS.OpenTextFile("C:\UserList.txt", Read)

```

Listing 240: 2_02_023b.vbs

```

Set oOutputFile = oFS.OpenTextFile("C:\UserList_New.txt", Write, True)

Do While Not (oInputFile.AtEndOfStream)
    Redim Preserve arrLines(intCounter)
    arrLines(intCounter) = oInputFile.ReadLine
    intCounter = intCounter + 1
Loop

For intCounter = 0 to Ubound(arrLines) - 4
    oOutputFile.WriteLine arrLines(intCounter)
Next

oInputFile.Close
oOutputFile.Close
Set oInputFile = Nothing
Set oOutputFile = Nothing
Set oFS = Nothing

```

Listing 240: 2_02_023b.vbs (Forts.)

50 Inhalte von Textdateien miteinander vergleichen

Wenn zwei oder auch mehrere Dateien auf den ersten Blick gleich sind und den gleichen Dateinamen haben, die gleichen Datums- und Zeitstempel tragen und auch von der Dateigröße her gleich sind, so muss das nicht heißen, dass es sich auch wirklich um die gleichen Dateien handelt. Denn oftmals sind es nur wenige Zeichen im Inneren der Dateien, die den Unterschied ausmachen.

Um herauszufinden, ob zwei Dateien gleich sind oder ob sie sich im Inhalt her unterscheiden, können Sie die Inhalte der Dateien miteinander vergleichen und sich die möglichen Differenzen genau anzeigen lassen. Der Windows Server 2003 hält allerdings per Default kein entsprechendes Werkzeug bereit, womit Sie einen solchen Vergleich unter grafischer Oberfläche durchführen können (dies geht ohne Einsatz weiterer Tools nur von der Eingabeaufforderung aus oder per Skript). Wenn Sie jedoch die Support-Tools, die im Ordner *\Support\Tools* auf der Installations-CD von Windows Server 2003 zu finden sind, nachinstallieren, so können Sie das Tool *Windiff.exe* einsetzen und den Vergleich auch per GUI vornehmen.

Wenn Sie die Support-Tools eingerichtet oder zumindest die Datei *Windiff.exe* aus der Cab-Datei heraus extrahiert haben, dann starten Sie das Tool *Windiff.exe* und wählen im Menü FILE den Eintrag COMPARE FILES... aus, woraufhin Sie in das Dialogfenster SELECT FIRST FILE gelangen und die erste zu vergleichende Datei auswählen müssen. Markieren Sie also die betreffende Datei, und klicken Sie anschließend auf ÖFFNEN, um mit der Auswahl der zweiten zu vergleichenden Datei fortzufahren.

Sie gelangen daraufhin in eine Ansicht, in der die beiden Dateien gegenübergestellt werden und die zuletzt bearbeitete Datei mit IS MORE RECENT kenntlich gemacht wird. Markieren Sie nun die beiden Dateien, bzw. klicken Sie den Eintrag im Dialogfenster WINDIFF, an und führen Sie anschließend einen Klick auf die Schaltfläche EXPAND aus, woraufhin Sie eine etwas detailliertere Ansicht vorfinden. Denn in dieser Ansicht werden nun alle Zeilen farblich dargestellt, die Differenzen enthalten und damit nicht gleich sind. Der ersten Spalte können Sie dabei die Farbe der zuerst geöffneten und auch die Farbe der zweiten Datei entnehmen. Die zweite Spalte gibt Auskunft darüber, in welcher Zeile die Differenz jeweils gefunden wurde, und den eigentlichen Inhalt der Zeile entnehmen Sie der rechten Hauptspalte.

Die nachfolgende Abbildung zeigt, wie die Differenzen, die sich aus dem Vergleich zweier Dateien ergeben haben, dargestellt werden. Der Vergleich ergab, dass der Inhalt der Dateien bis auf drei Zeilen übereinstimmt, wobei sich die Unterschiede den jeweils farblich gekennzeichneten Zeilen genau entnehmen lassen.

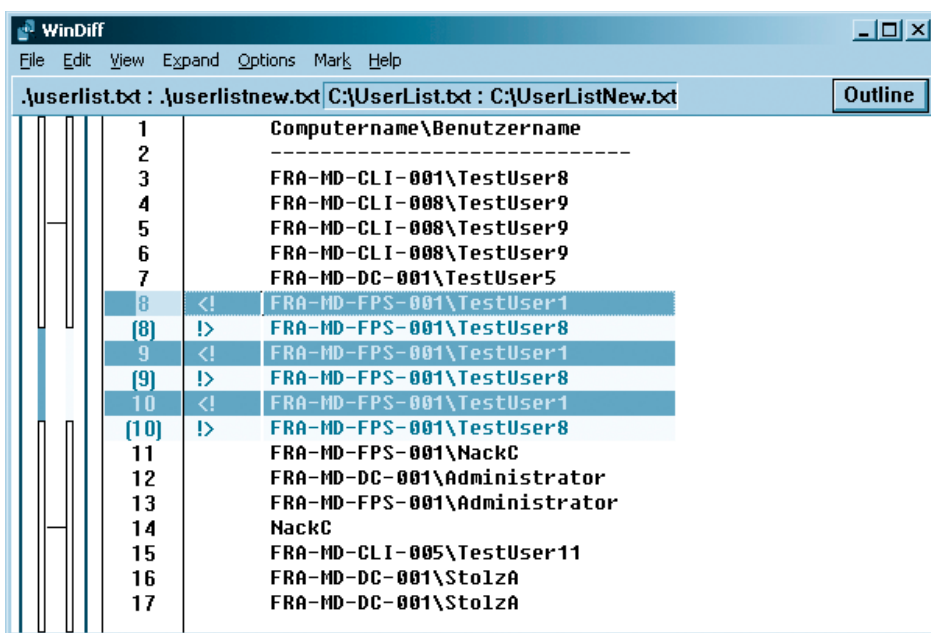


Abbildung 99: Die Inhalte zweier Textdateien miteinander vergleichen

Für den Fall, dass Sie vorhandene Differenzen beseitigen möchten, so können Sie zum einen natürlich die ursprüngliche Datei erneut kopieren, zum anderen aber auch direkt aus dem Dialogfenster WINDIFF heraus editieren, indem Sie im Kontextmenü den Eintrag EDIT LEFT FILE bzw. EDIT RIGHT FILE auswählen. Die betreffende Datei wird daraufhin in einem Editor geöffnet, und Sie können – leider durch manuelles Suchen der betreffenden Zeile – den Inhalt korrigieren.

Tipp

Mit dem Tool *Windiff.exe*, das Bestandteil der *Support-Tools* von Windows Server 2003 ist, können Sie nicht nur einzelne Dateien miteinander vergleichen, sondern auch die Inhalte von Ordnern. So können Sie beispielsweise durch einen Vergleich zweier Ordner auf recht einfache Art und Weise herausfinden, ob zwei Ordner die gleichen Dateien beinhalten und ob sich die einzelnen Dateien von ihrem Inhalt her unterscheiden. Ein solcher Vergleich ist z.B. nach der Restaurierung von Dateien, die in einem alternativen Pfad wiederhergestellt wurden, sehr nützlich.

Die Lösung per Eingabeaufforderung

Dateien lassen sich natürlich auch von der Eingabeaufforderung aus vergleichen. Hierfür stellt der Windows Server 2003 den *Comp*- und *FC*-Befehl bereit, womit Sie die Inhalte zweier Dateien auf mögliche Unterschiede hin überprüfen können.

Der *Comp*-Befehl erwartet dabei zumindest die Angabe der ersten und auch der zweiten zu vergleichenden Datei als Parameter und prüft daraufhin erst einmal die Größe der Dateien. Wenn die zu vergleichenden Dateien unterschiedliche Dateigrößen aufweisen, so bricht der Vergleich ab, und der *Comp*-Befehl gibt die Meldung *Dateien sind unterschiedlich groß* aus. Sind die Dateien jedoch gleich groß, so werden die Inhalte miteinander verglichen. Werden während des Vergleichs mehr als zehn Unterschiede gefunden, so bricht die *Comp*-Befehlsausführung ebenfalls ab und die Meldung *10 Unterschiede - Vergleich beendet* erscheint. Anderenfalls – wenn weniger als zehn Unterschiede festgestellt werden – gibt der *Comp*-Befehl nach dem Vergleich die Unterschiede im hexadezimalen Format aus.

Um die Unterschiede im ASCII-Format auszugeben, fügen Sie der *Comp*-Befehlszeile den zusätzlichen Parameter */a* an, und wenn Sie dabei auch die Zeilennummern ausgeben lassen möchten, so verwenden Sie zusätzlich den Parameter */l*.

```
@Echo off
Comp C:\UserList.txt C:\UserListNew.txt /a /l
```

Listing 241: 2_02_024a.bat

Eine weitere, wenngleich sehr komfortable Möglichkeit, zwei Dateien von der Eingabeaufforderung aus miteinander zu vergleichen, bietet das Tool *Diff.exe*, das Bestandteil der *GNU-Utilities für Win32* ist und ohne Einrichtung einer zusätzlichen Shell sofort eingesetzt werden kann. Im Gegensatz zum *Comp*-Befehl von Windows Server 2003 ermöglicht das Tool *Diff.exe* auch den Vergleich von Dateien, die mehr als zehn Unterschiede beinhalten, und bietet auch gegenüber dem *FC*-Befehl weitaus mehr Parameter, womit sich der durchzuführende Vergleich sehr exakt bestimmen und auch die Ausgabe sehr gut formatieren lässt.

So können Sie mit dem Parameter `-y` beispielsweise festlegen, dass die zu vergleichenden Dateien in zwei Spalten angezeigt und zeilenweise gegenüber gestellt werden, womit sich der Vergleich auch am Bildschirm (oder durch eine Ausgabeumleitung in eine Datei) sehr gut nachvollziehen lässt.

```
@Echo off
Diff -y C:\UserList.txt C:\UserListNew.txt > Result.txt
```

Listing 242: 2_02_024b.bat

Mit dem Parameter `-c` dagegen bewirken Sie, dass zunächst die Zeilen der ersten Datei unter Kennzeichnung der Unterschiede und dann die Zeilen der zweiten Datei (ebenfalls mit Markierungen hinsichtlich der Unterschiede) ausgegeben werden. Per Ausgabeumleitung lässt sich das Ergebnis natürlich auch in eine Datei schreiben, womit Sie das Ergebnis auch zu einem späteren Zeitpunkt noch auswerten können.

```
@Echo off
Diff -c C:\UserList.txt C:\UserListNew.txt > Result.txt
```

Listing 243: 2_02_024c.bat

Die Lösung mit VBScript

Auch mit VBScript lassen sich die Inhalte von Dateien recht einfach miteinander vergleichen. Hierzu öffnen Sie lediglich die zu vergleichenden Dateien im Lesemodus, lesen die Inhalte jeweils zeilenweise ein und vergleichen deren Inhalte. Wenn die Zeilen gleich sind, fahren Sie mit der nächsten Zeile fort, anderenfalls geben Sie die Zeilen, in denen Unterschiede festgestellt wurden, entweder am Bildschirm aus oder schreiben die Inhalte der Zeilen in eine gesonderte Ausgabedatei. Und mit Hilfe eines Zählers, den Sie mit jedem Einlesen einer Zeile um den Wert 1 erhöhen, können Sie zusätzlich auch die Zeilennummern, in denen die Unterschiede festgestellt wurden, ausgeben.

Das nachfolgende Code-Beispiel zeigt, wie Sie die Inhalte zweier Textdateien miteinander vergleichen können und wie Sie die Zeilen, in denen die Unterschiede festgestellt wurden, unter Angabe der Zeilennummer und auch der jeweiligen Zeileninhalte in eine Ausgabedatei mit der Bezeichnung *Result.txt* schreiben können.

Die zu vergleichenden Dateien werden mit der `OpenTextFile`-Methode des `FileSystem`-Objektes im Lesemodus (`Read = 1`) und die Ausgabedatei im Schreibmodus (`Write = 2`) geöffnet, wobei Letztere neu angelegt wird, sofern die Datei noch nicht existiert. Anschließend werden die Inhalte der ersten Datei (`strFile1`) und der zweiten Datei (`strFile2`) mit der `ReadLine`-Methode zeilenweise eingelesen und an die Variablen `strTmp1` und `strTmp2` übergeben. Die Werte dieser beiden Variablen werden inhaltlich miteinander verglichen. Sind die Inhalte gleich, so werden jeweils die nächsten beiden Zeilen eingelesen, miteinander verglichen, und auch der Zähler für die Zeilennummerierung wird um den Wert 1 erhöht. Anderenfalls – wenn Unterschiede festgestellt wur-

den – werden die Werte von `intCounter`, `strTmp1` und `strTmp2` mit der `WriteLine`-Methode in die Ausgabedatei geschrieben, und die Schleife wird verlassen. Diese Schritte werden so lange wiederholt, bis das Ende der beiden Dateien erreicht wurde.

Option Explicit

```
Dim oFS
Dim oFile1
Dim oFile2
Dim oOutputFile
Dim strFile1
Dim strFile2
Dim strOutputFile
Dim strTmp1
Dim strTmp2
Dim intCounter
```

```
strFile1 = "C:\UserList.txt"
strFile2 = "C:\UserListNew.txt"
strOutputFile = "C:\Result.txt"
intCounter = 1
```

```
Const Read = 1
Const Write = 2
```

```
Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFile1 = oFS.OpenTextFile(strFile1, Read)
Set oFile2 = oFS.OpenTextFile(strFile2, Read)
Set oOutputFile = oFS.OpenTextFile(strOutputFile, Write, True)
```

```
Do While Not (oFile1.AtEndOfStream AND oFile2.AtEndOfStream)
    strTmp1 = oFile1.ReadLine
    strTmp2 = oFile2.ReadLine
```

```
    If strTmp1 <> strTmp2 Then
        oOutputFile.WriteLine "Line: " & intCounter
        oOutputFile.WriteLine "File1: " & strTmp1
        oOutputFile.WriteLine "File2: " & strTmp2
        oOutputFile.WriteBlankLines(1)
        Exit Do
    End If
```

```
    intCounter = intCounter + 1
Loop
```

```
oFile1.Close  
oFile2.Close  
oOutputFile.Close  
Set oFS = Nothing  
Set oFile1 = Nothing  
Set oFile2 = Nothing
```

Listing 244: 2_02_024.vbs (Forts.)

51 Textdateien nach bestimmten Zeichenfolgen durchsuchen

Nicht selten müssen Textdateien nach bestimmten Zeichenfolgen durchsucht werden. Dies ist unter anderem dann der Fall, wenn Sie spezielle Einträge in einer Log-Datei suchen müssen, den Inhalt einer Datendatei unter die Lupe nehmen wollen oder die Einstellungen in einer Konfigurationsdatei prüfen möchten. Und gerade bei größeren Dateien kann sich die Suche nach bestimmten Zeichenfolgen recht mühsam erweisen, da der Windows Server 2003 hierfür nur wenige Hilfsmittel zur Verfügung stellt.

Wenn Sie z.B. herausfinden möchten, ob in der Log-Datei *SystemProtokoll.txt* die Zeichenfolge *Administrator* enthalten ist, so können Sie natürlich einen der Windows-eigenen Editoren (NOTEPAD oder WORDPAD) zu Hilfe nehmen, die Zeichenfolge als Suchbegriff festlegen und den gesamten Text nach möglichen Übereinstimmungen durchsuchen. Die Windows-eigenen Editoren führen dabei ausschließlich eine Volltextsuche durch, womit der gesamte Text zeichenweise verarbeitet und mit dem Suchbegriff verglichen wird. Sobald eine Übereinstimmung gefunden wurde, wird die betreffende Position am Bildschirm angezeigt, und Sie können – wenn Sie nach weiteren Übereinstimmungen suchen möchten – mit der Taste **[F3]** so lange nach weiteren Übereinstimmungen fahnden, bis das Ende des Textes erreicht ist.

Um lediglich das Vorhandensein einer bestimmten Zeichenfolge zu überprüfen oder um kleinere Textdateien nach bestimmten Zeichenketten zu durchsuchen, ist die Volltextsuche in den meisten Fällen mit Sicherheit ausreichend. Wenn Sie jedoch größere Textdateien verarbeiten müssen und beispielsweise nach dem *n*-ten Auftreten einer bestimmten Zeichenfolge oder gar an bestimmten Positionen innerhalb des Textes nach einem Suchbegriff suchen müssen, so sollten Sie einen alternativen Editor einsetzen, der die Verwendung regulärer Ausdrücke (engl. = *Regular Expressions*) unterstützt. Denn mit Hilfe von regulären Ausdrücken können Sie erweiterte Suchmuster festlegen und die Suche nach bestimmten Zeichenfolgen dadurch um ein Vielfaches verfeinern und auch optimieren.

Als alternative Editoren können Sie z.B. den VI-Clone *WinVi32* (www.winvi.de) oder auch *Lemmy* (www.softwareonline.org) verwenden, da beide Editoren die Verwendung regulärer Ausdrücke unterstützen und auch sonst weitaus mehr Funktionen (z.B. Aufruf externer Befehle, Zeichensatzkonvertierung, Hexadezimalmodus etc.) bieten als die Windows-eigenen Editoren NOTEPAD und WORDPAD. Am Beispiel von

WinVi32 soll nachfolgend einmal die Verwendung einfacher regulärer Ausdrücke unter grafischer Oberfläche demonstriert werden:

Einmal angenommen, Sie haben eine Textdatei mit der Bezeichnung *UserList.txt*, in der unterschiedliche Benutzernamen im Format *Computername\Benutzername* aufgelistet sind. Die Textdatei enthält dabei auch Zeilen, in der die beiden Benutzer *TestUser1* und *NackC* ohne Angabe des Computernamens auftauchen. Diese Zeilen möchten Sie nun herauslöschen, so dass zum Schluss nur Einträge im Format *Computername\Benutzername* enthalten sind.

Unter Verwendung einer Volltextsuche würden Sie nun nach den Begriffen *TestUser1* und *NackC* suchen und so lange auf WEITERSUCHEN klicken, bis eine Position angezeigt wird, in der die Suchbegriffe am Anfang einer Zeile stehen. Mit Hilfe von regulären Ausdrücken können Sie diese Positionen aber direkt ansteuern und die Suche damit erheblich effizienter gestalten:

Starten Sie hierfür den Editor *WinVi32.exe*, und öffnen Sie die betreffende Textdatei. Wählen Sie im Menü SUCHEN den Eintrag SUCHEN/ERSETZEN aus, oder drücken Sie alternativ dazu die Tastenkombination **[Alt] + [F3]**, um die Such-Funktion aufzurufen. Aktivieren Sie die Option ALS REGULÄRER AUSDRUCK, geben Sie im Feld SUCHMUSTER die Zeichenfolge *^TestUser1* (das Zeichen *^* steht hierbei für »Suche am Zeilenanfang«) ein, und klicken Sie anschließend auf WEITERSUCHEN, um die erste Position anzusteuern, in der *TestUser1* am Anfang einer Zeile steht. Damit werden nun alle Zeilen ignoriert, die nicht mit *TestUser1* beginnen, was die Suche erheblich beschleunigt.

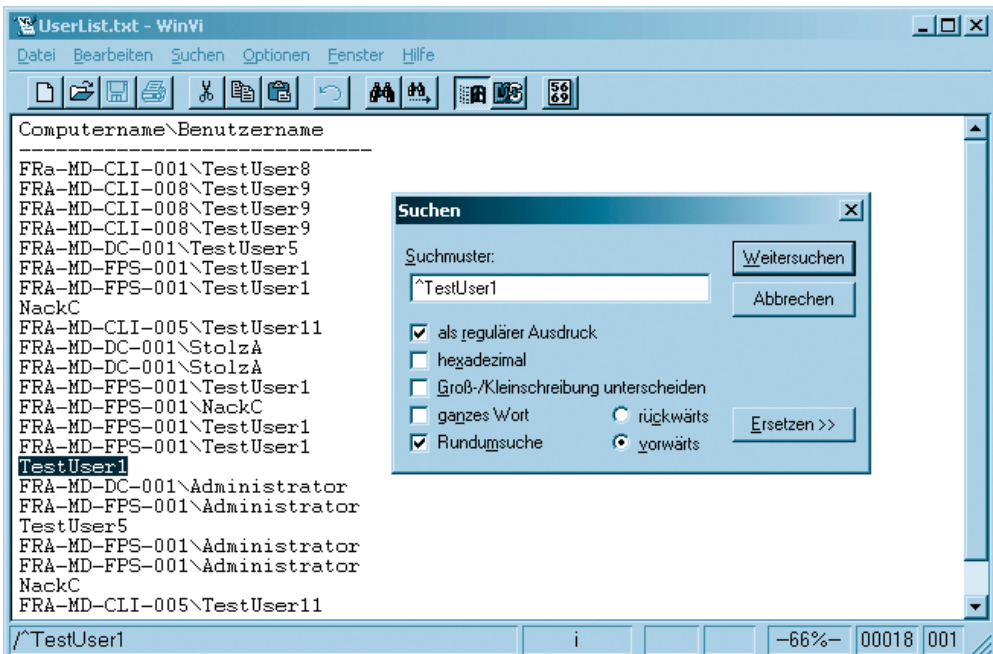


Abbildung 100: Nach einem regulären Ausdruck suchen

Wenn Sie stattdessen nur die Positionen ansteuern möchten, in der *TestUser1* am Ende der Zeile steht, so geben Sie im Feld SUCHMUSTER die Zeichenfolge *TestUser1\$* ein (das Zeichen \$ steht für »Suche am Zeilenende«) und klicken auf WEITERSUCHEN, woraufhin alle Zeilen ignoriert werden, die nicht mit *TestUser1* enden.

Und um die Suche noch weiter zu optimieren, können Sie in nur einem Suchvorgang auch nach beiden Zeichenfolgen (*TestUser1* und *NackC*) suchen. Hierzu geben Sie im Feld SUCHMUSTER die Zeichenfolge *\(TestUser1\|NackC\)* ein, woraufhin alle Positionen angesteuert werden, die entweder *TestUser1* oder *NackC* enthalten.

Die nachfolgende Tabelle zeigt eine kurze Übersicht über die wichtigsten regulären Ausdrücke unter Angabe eines kurzen Beispiels und auch einer kurzen Beschreibung.

Regulärer Ausdruck	Beispiel	Beschreibung
<code>^</code>	<code>^TestUser1</code>	Sucht am Anfang einer Zeile nach dem Ausdruck <i>TestUser1</i> und steuert alle Positionen an, die mit dem angegebenen Ausdruck beginnen.
<code>\$</code>	<code>TestUser1\$</code>	Sucht am Ende einer Zeile nach dem Ausdruck <i>TestUser1</i> und steuert alle Positionen an, die mit dem angegebenen Ausdruck enden.
<code>.</code>	<code>TestUser.1</code>	Sucht nach Zeichenfolgen, die mit <i>TestUser</i> beginnen, dann ein beliebiges Zeichen enthalten und mit <i>1</i> enden.
<code>\</code>	<code>\\$</code>	Hebt die Sonderbedeutung eines Steuerzeichens auf, womit innerhalb des Textes auch nach einem Steuerzeichen (z.B. \$) gesucht werden kann.
<code>\<</code>	<code>\<TestUser1</code>	Sucht am Anfang eines Wortes nach dem Ausdruck <i>TestUser1</i> und steuert alle Positionen an, die ein Wort enthalten, das mit <i>TestUser1</i> beginnt.
<code>\></code>	<code>TestUser1\></code>	Sucht am Ende eines Wortes nach dem Ausdruck <i>TestUser1</i> und steuert alle Positionen an, die ein Wort enthalten, das mit <i>TestUser1</i> endet.
<code>[List]</code>	<code>[DC]</code>	Sucht nach einem der angegebenen Zeichen und steuert alle Positionen an, die entweder das Zeichen <i>D</i> oder das Zeichen <i>C</i> enthalten.
<code>[^List]</code>	<code>[^DC]</code>	Sucht nicht nach einem der angegebenen Zeichen und steuert alle Positionen an, die weder das Zeichen <i>D</i> noch das Zeichen <i>C</i> enthalten.

Tabelle 36: Übersicht der wichtigsten regulären Ausdrücke

Regulärer Ausdruck	Beispiel	Beschreibung
[x-y]	[0-9]	Sucht nach einem Zeichen in dem angegebenen Bereich und steuert alle Positionen an, die 0, 1, 2, 3, 4, 5, 6, 7, 8 oder 9 enthalten (mögliche Bereiche sind beispielsweise 0 bis 9, a bis z und A bis Z).
*	Te*	Sucht nach dem vorherigen Zeichen kein oder auch mehrere Male. Das Beispiel <i>Te*</i> steuert alle Positionen mit <i>T</i> , <i>Te</i> , <i>Tee</i> usw. an.
\(Ausdruck\ Ausdruck\)	\(TestUser1\ NackC\)	Sucht nach dem Ausdruck <i>TestUser1</i> und alternativ nach dem Ausdruck <i>NackC</i> und steuert damit alle Positionen an, die entweder <i>TestUser1</i> oder <i>NackC</i> enthalten.

Tabelle 36: Übersicht der wichtigsten regulären Ausdrücke (Forts.)

Die Lösung per Eingabeaufforderung

Wenn Sie die Suche nach einer bestimmten Zeichenfolge von der Eingabeaufforderung aus durchführen möchten, so können Sie hierfür den `Findstr`-Befehl verwenden, der zum festen Befehlsumfang von Windows Server 2003 gehört.

Um beispielsweise den Inhalt der Datei `C:\UserList.txt` nach der Zeichenfolge *TestUser1* zu durchsuchen und alle Zeilen, die eine exakte Übereinstimmung mit dem Suchbegriff enthalten, in die Datei `Result.txt` auszugeben, wenden Sie den `Findstr`-Befehl unter Angabe des Parameters `/C`: an und übergeben den Suchbegriff wie auch die zu durchsuchende Datei. Des Weiteren fügen Sie der Befehlszeile eine Ausgabeumleitung an, womit das Ergebnis nicht am Bildschirm ausgegeben, sondern in die angegebene Datei geschrieben wird.

```
@Echo off
Findstr /C:TestUser1 C:\UserList.txt > Result.txt
```

Listing 245: 2_02_025a.bat

Die zuvor genannte `Findstr`-Befehlszeile führt dabei eine Volltextsuche durch und gibt damit alle Zeilen aus, in denen die Zeichenfolge *TestUser1* enthalten ist. Und zwar unabhängig davon, ob der Suchbegriff *TestUser1* als eigenständiges Wort in der Zeile vorkommt oder Bestandteil einer anderen Zeichenfolge ist. Auch die Position, an der eine Übereinstimmung gefunden wurde, bleibt hierbei unberücksichtigt.

Der `Findstr`-Befehl erlaubt jedoch die Verwendung von regulären Ausdrücken, womit Sie die Suche erheblich verfeinern können (vgl. auch vorheriger Aberschnitt). Denn wenn Sie z.B. nur die Zeilen herausfiltern möchten, die am Anfang einer Zeile *TestUser1* enthalten, so fügen Sie dem `Findstr`-Befehl den Parameter `/R` an, womit

die Verwendung von regulären Ausdrücken eingeleitet wird, und legen dann mit `^` die Spaltenposition (Anfang der Zeile) fest, womit Sie die Suche weitaus genauer spezifizieren können.

```
@Echo off
Findstr /R "^TestUser1" C:\UserList.txt > Result.txt
```

Listing 246: 2_02_025b.bat

Die nachfolgende Tabelle zeigt, welche regulären Ausdrücke mit *Findstr.exe* verwendet werden können, und zeigt jeweils ein kurzes Beispiel zu den einzelnen Ausdrücken.

Regulärer Ausdruck	Beispiel	Beschreibung
.	19...2005	Der Punkt dient als Platzhalter für ein beliebiges Zeichen. Beispiel: Der Suchbegriff <i>19...2005</i> gibt alle Zeilen aus, die eine zehnstellige Zeichenfolge enthalten, dabei mit <i>19</i> beginnen und mit <i>2005</i> enden.
*	19*	Der Stern dient als Platzhalter für beliebige Zeichen. Beispiel: Der Suchbegriff <i>19*</i> gibt alle Zeilen aus, die mit <i>1</i> beginnen und danach beliebig viele oder auch keine <i>9</i> enthalten.
^	^W32Time	Das <code>^</code> -Zeichen bestimmt, dass nur am Zeilenanfang nach dem angegebenen Begriff gesucht werden soll. Beispiel: Gibt alle Zeilen aus, die mit <i>W32Time</i> beginnen.
\$	W32Time\$	Das <code>\$</code> -Zeichen bestimmt dagegen, dass nur am Zeilenende nach dem Begriff gesucht werden soll. Beispiel: Gibt alle Zeilen aus, die mit <i>W32Time</i> enden.
[Klasse]	[TN]	Mit diesem Ausdruck wird jeweils nach einem der angegebenen Zeichen gesucht, und es werden alle Zeilen ausgegeben, die zumindest eines der angegebenen Zeichen enthalten. Beispiel: Gibt alle Zeilen aus, die entweder <i>T</i> oder <i>N</i> enthalten.
[^Klasse]	[^TN]	Mit diesem Ausdruck werden nur die Zeilen ausgegeben, die keines der angegebenen Zeichen enthalten. Beispiel: Gibt alle Zeilen aus, die nicht <i>T</i> oder <i>N</i> enthalten.
[x-y]	[5-9]	Gibt alle Zeilen aus, die ein oder auch mehrere Zeichen aus dem angegebenen Bereich (<i>0-9</i> , <i>a-z</i> oder <i>A-Z</i>) enthalten. Beispiel: Gibt alle Zeilen aus, die eine Ziffer zwischen <i>5</i> und <i>9</i> enthalten.
\x	\\$	Hebt die Sonderbedeutung eines Steuerzeichens auf, womit innerhalb des Textes auch nach einem Steuerzeichen (z.B. <code>\$</code>) gesucht werden kann.

Tabelle 37: Die regulären Ausdrücke

Regulärer Ausdruck	Beispiel	Beschreibung
<xyz	<W32Time	Legt fest, dass nach dem Suchbegriff am Anfang eines Wortes gesucht werden soll. Beispiel: Gibt alle Zeilen aus, die ein Wort enthalten, das mit <i>W32Time</i> beginnt.
xyz >	W32Time >	Legt fest, dass nach dem Suchbegriff am Ende eines Wortes gesucht werden soll. Beispiel: Gibt alle Zeilen aus, die ein Wort enthalten, das mit <i>W32Time</i> endet.

Tabelle 37: Die regulären Ausdrücke (Forts.)

Möchten Sie stattdessen nur die Zeilen ausgeben, die *TestUser1* am Ende einer Zeile enthalten, so wenden Sie den `Findstr`-Befehl mit dem Parameter `/R` an, bestimmen mit dem regulären Ausdruck `$` die Zeilenposition (Ende der Zeile) und übergeben dann die Datei, deren Inhalt durchsucht werden soll. Und mit einer Ausgabeumleitung können Sie das Ergebnis auch hier wieder in eine Datei ausgeben lassen.

```
@Echo off
Findstr /R "TestUser1$" C:\UserList.txt > Result.txt
```

Listing 247: 2_02_025c.bat

Tipp

In den *GNU-Utilities für Win32* ist noch eine ganze Reihe weiterer Tools enthalten, womit Sie Dateiinhalte nach regulären Ausdrücken durchsuchen können. Eine Einzeldarstellung all dieser Tools ist aus Platzgründen leider nicht möglich, dennoch sollen diese Tools zumindest namentlich erwähnt werden, da sie mit ihren unzähligen Parametern die Angabe sehr komplexer Suchmuster ermöglichen und daher in vielen Fällen sehr hilfreich sein können. So können Sie beispielsweise die Tools *agrep.exe*, *egrep.exe*, *fgrep.exe* und *ggrep.exe* verwenden, um die Inhalte gleich mehrerer Textdateien (z.B. die Dateien eines Ordners) nach komplexen Suchmustern zu durchforsten und durch Angabe zusätzlicher Parameter erweiterte Informationen (z.B. Name der Datei, Anzahl der Übereinstimmungen etc.) auszugeben. Die Tools *gawk.exe* und *sed.exe* ermöglichen sogar die Ausführung selbst erstellter Programme zur Textmanipulation, die in Form eines Skriptes zur Ausführung übergeben werden können.

Die Lösung mit VBScript

Der Inhalt einer Textdatei lässt sich natürlich auch per VBScript nach einem bestimmten Suchbegriff durchforsten, wobei Sie auch hier wieder zwischen einer Volltextsuche und der Suche unter Verwendung regulärer Ausdrücke wählen können. Um beispielsweise den Inhalt der Datei *C:\UserList.txt* in Form einer Volltextsuche nach der Zeichenfolge *TestUser1* zu durchsuchen, öffnen Sie zunächst die angegebene Datei mit der `OpenTextFile`-Methode des `FileSystem`-Objektes, wenden dann die

`ReadLine`- oder wahlweise auch die `ReadAll`-Methode an, um den Inhalt der Textdatei zeilenweise oder komplett einzulesen, und setzen anschließend die `InStr`-Funktion ein, um nach Übereinstimmungen mit der gesuchten Zeichenfolge zu suchen.

Die `InStr`-Funktion erwartet dabei die Angabe zweier Parameter, die Sie mit dem Aufruf der Funktion übergeben müssen. Folgende Parameter können verwendet werden:

- ▶ **Start:** Der optionale Parameter *Start* erlaubt die Angabe der Position, an der die Suche nach der gesuchten Zeichenfolge beginnen soll. So legt der Wert 10 beispielsweise fest, dass mit der Suche erst ab dem zehnten Zeichen des Textes nach möglichen Übereinstimmungen mit dem Suchbegriff gesucht werden soll. Wird dieser Parameter nicht angegeben, so beginnt die Suche per Default ab der ersten Position.
- ▶ **String1:** Der Parameter *String1* enthält die Zeichenfolge bzw. den Text, der nach dem Suchbegriff durchsucht werden soll, und muss daher auf jeden Fall angegeben werden.
- ▶ **String2:** Mit dem Parameter *String2* übergeben Sie die Zeichenfolge, nach der gesucht werden soll. Auch dieser Parameter muss auf jeden Fall angegeben werden.
- ▶ **Compare:** Der optionale Parameter *Compare* bestimmt, ob ein binärer Vergleich oder ein Textvergleich durchgeführt werden soll. Bei einem binären Vergleich (dies ist die Standardeinstellung) definieren Sie zu Beginn des Skriptes die Konstante *vbBinaryCompare* und weisen ihr den Wert 0 zu, bei einem Textvergleich dagegen können Sie die Konstante *vbTextCompare* verwenden, der Sie den Wert 1 zuweisen.

Das nachfolgende Listing zeigt ein Beispiel dafür, wie Sie mit Hilfe der `InStr`-Funktion den Inhalt der Datei `C:\UserList.txt` nach der Zeichenfolge `TestUser1` durchsuchen und die Zeilen, in denen eine Übereinstimmung gefunden wurde, in die Datei `C:\Result.txt` ausgeben können.

Option Explicit

```
Dim oFS
Dim oInputFile
Dim oOutputFile
Dim strInputFile
Dim strOutputFile
Dim strText
Dim strResult
Dim strString
```

```
strInputFile = "C:\UserList.txt"
```

Listing 248: 2_02_025a.vbs

```

strOutputFile = "C:\Result.txt"
strString = "TestUser1"
Const Read = 1
const Write = 2

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oInputFile = oFS.OpenTextFile(strInputFile, Read)
Set oOutputFile = oFS.OpenTextFile(strOutputFile, Write, True)

Do While Not (oInputFile.AtEndOfStream)
    strText = oInputFile.ReadLine
    strResult = InStr(strText, strString)
    If strResult <> 0 Then
        oOutputFile.WriteLine strText
    End If
Loop

oInputFile.Close
oOutputFile.Close
Set oFS = Nothing
Set oInputFile = Nothing
Set oOutputFile = Nothing

```

Listing 248: 2_02_025a.vbs (Forts.)

Gleiches lässt sich natürlich auch unter Verwendung regulärer Ausdrücke realisieren, womit Sie die Suche in vielen Fällen schneller und effizienter gestalten können. Denn wenn Sie nach einem regulären Ausdruck suchen, so können Sie beispielsweise auch die Position bestimmen, an der nach dem gesuchten Begriff gefahndet werden soll, und das Ergebnis damit weiter eingrenzen.

Das nachfolgende Code-Beispiel führt eine ähnliche Suche wie in dem vorangegangenen Listing durch, allerdings wird diesmal keine Volltextsuche verwendet, sondern nach einem regulären Ausdruck (^TestUser1) gesucht. Um reguläre Ausdrücke in einem VBScript verwenden zu können, erzeugen Sie mit der `CreateObject`-Methode erst eine neue Instanz des `RegExp`-Objektes, übergeben der Eigenschaft `Pattern` den regulären Ausdruck (siehe Tabelle 37: Die regulären Ausdrücke auf Seite 325 und *Teil III* dieses Buches), weisen der Eigenschaft `Global` den Wert `True` zu, damit sich die Suche auf die gesamte Zeichenfolge bezieht, und führen dann die Suche mit der `Execute`-Methode durch, der Sie den zu durchsuchenden Text als Argument übergeben. Die `Execute`-Methode liefert daraufhin eine Kollektion der Übereinstimmungen zurück, deren Elemente mit einer `For Each`-Schleife ausgelesen, am Bildschirm ausgegeben oder auch in eine Datei geschrieben werden können.

Option Explicit

```
Dim oFS
Dim oInputFile
Dim oOutputFile
Dim oRE
Dim strInputFile
Dim strOutputFile
Dim strText
Dim colMatches
Dim objItem
```

```
strInputFile = "C:\UserList.txt"
strOutputFile = "C:\Result.txt"
Const Read = 1
const Write = 2
```

```
Set oRE = CreateObject("VBScript.RegExp")
oRE.Pattern = "^TestUser"
oRE.Global = True
```

```
Set oFS = CreateObject("Scripting.FileSystemObject")
Set oInputFile = oFS.OpenTextFile(strInputFile, Read)
Set oOutputFile = oFS.OpenTextFile(strOutputFile, Write, True)
```

```
Do While Not (oInputFile.AtEndOfStream)
    strText = oInputFile.ReadLine
    Set colMatches = oRE.Execute(strText)
    If colMatches.Count > 0 Then
        oOutputFile.WriteLine strText
    End If
Loop
```

```
oInputFile.Close
oOutputFile.Close
Set oFS = Nothing
Set oInputFile = Nothing
Set oOutputFile = Nothing
Set oRE = Nothing
```

Listing 249: 2_02_025b.vbs

52 Zeichenfolgen suchen und ersetzen

Die meisten Editoren stellen eine entsprechende Funktion bereit, womit Sie den Inhalt einer Datei nicht nur nach bestimmten Zeichenfolgen durchsuchen, sondern diese Zeichenfolgen auch gleichzeitig durch andere Zeichenfolgen ersetzen können. Diese Suchen-und-Ersetzen-Funktion ermöglicht es Ihnen beispielsweise, den Inhalt einer Konfigurations- oder Datendatei auf recht einfache Art und Weise zu überarbeiten und ohne großen Aufwand an neue Gegebenheiten anzupassen. Haben Sie z.B. eine Datei, in der diverse Benutzer im Format *Computer- oder Domänenname\Benutzername* hinterlegt sind, so können Sie mit der Suchen-und-Ersetzen-Funktion nach dem Computer- bzw. Domännennamen suchen und diese Zeichenfolgen durch andere Computer- und Domännennamen ersetzen, womit Sie diese Datei dann auch auf anderen Computersystemen oder in anderen Domänen verwenden können.

Die Windows Server 2003-eigenen Editoren namens NOTEPAD und WORDPAD verfügen zwar auch über eine Suchen-und-Ersetzen-Funktion, allerdings wenden diese beiden Editoren ausschließlich eine Volltextsuche (vgl. Textdateien nach bestimmten Zeichenfolgen durchsuchen auf Seite 321) an, um den Inhalt der Datei nach dem gewünschten Suchbegriff zu durchforsten. Wenn der Suchbegriff aber an unterschiedlichen Positionen auftaucht, so können Sie mit einem Editor, der die Verwendung regulärer Ausdrücke unterstützt, das Suchen und Ersetzen bestimmter Zeichenfolgen wesentlich effizienter gestalten, da Sie hierbei nicht nur den Suchbegriff als solches definieren können, sondern auch die Position, an der nach einer Übereinstimmung gesucht werden soll.

Einmal angenommen, Sie haben eine Datei, in der zahlreiche Benutzernamen (Format *Computername\Benutzername*) und zusätzlich auch noch die Computernamen aufgelistet sind. Der Inhalt der Datei sieht in etwa so aus:

```
...
FRA-MD-CLI-001\TestUser8 FRA-MD-CLI-001
FRA-MD-CLI-008\TestUser9 FRA-MD-CLI-008
FRA-MD-CLI-008\TestUser9 FRA-MD-CLI-008
FRA-MD-CLI-008\TestUser9 FRA-MD-CLI-008
FRA-MD-DC-001\TestUser5 FRA-MD-DC-001
FRA-MD-FPS-001\TestUser1 FRA-MD-FPS-001
FRA-MD-FPS-001\TestUser1 FRA-MD-FPS-001
...
```

Nun möchten Sie in allen Zeilen die Zeichenfolge *FRA-MD* durch *FRA-DC* ersetzen, wobei aber nur die Computernamen berücksichtigt werden sollen, die jeweils vor dem Benutzernamen stehen. Die zusätzlichen Computernamen, die am Ende einer jeden Zeile enthalten sind, sollen nicht geändert werden.

Sie können nun einen der Windows Server 2003-eigenen Editoren verwenden und nach der Zeichenfolge *FRA-MD* suchen, die Sie anschließend durch *FRA-DC* ersetzen. Da diese Editoren aber nur eine Volltextsuche unterstützen, werden während des Suchvorgangs beide Computernamen angezeigt, da beide Computernamen die Zeichenfolge *FRA-MD* enthalten.

Mit Hilfe eines Editors, der die Verwendung regulärer Ausdrücke unterstützt, lässt sich diese Aufgabe jedoch einfacher (und auch schneller) durchführen. Denn mit dem Ausdruck `^FRA-MD` können Sie explizit am Zeilenanfang nach der Zeichenfolge `FRA-MD` suchen und diese Zeichenfolge dann durch `FRA-DC` ersetzen.

Wenn Sie z.B. WinVi32 (www.winvi.de) einsetzen, so öffnen Sie zunächst die Textdatei und wählen dann im Menü SUCHEN den Eintrag SUCHEN/ERSETZEN aus. Diese Funktion können Sie auch mit der Tastenkombination **[Alt] + [F3]** aufrufen. Anschließend geben Sie im Feld SUCHMUSTER den Ausdruck `^FRA-MD` ein, womit Sie am Anfang einer Zeile (^) nach der Zeichenfolge `FRA-MD` suchen können. Aktivieren Sie dann die Option ALS REGULÄRER AUSDRUCK, und klicken Sie auf WEITERSUCHEN, um zur ersten Übereinstimmung zu gelangen. Sobald die Zeichenfolge gefunden wurde, wird die Position angezeigt. Klicken Sie auf ERSETZEN, und geben Sie die gewünschte Zeichenfolge für den markierten Bereich an. In diesem Beispiel ersetzen Sie `FRA-MD` durch `FRA-DC`. Mit einem Klick auf ALLES ERSETZEN werden alle Übereinstimmungen auf einmal ersetzt, die einzelnen Übereinstimmungen werden damit nicht mehr angezeigt, sondern sofort geändert.

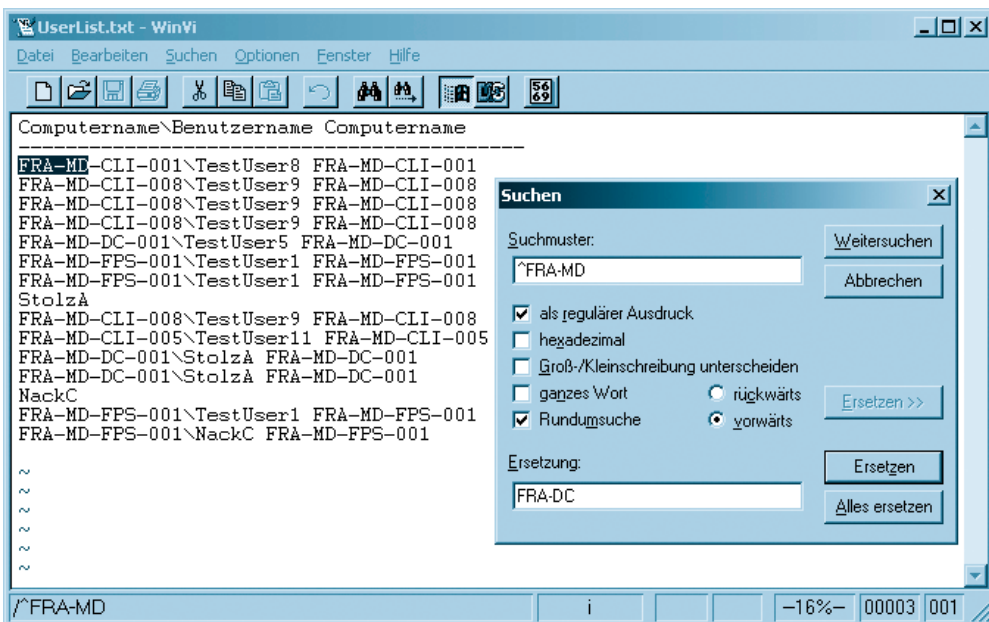


Abbildung 101: Zeichenfolgen suchen und ersetzen

Die Lösung per Eingabeaufforderung

Mit Hilfe der *GNU-Utilities für Win32* lässt sich die gleiche Aufgabe auch von der Eingabeaufforderung aus lösen und damit sogar automatisieren. Sie können z.B. das GNU-Utility *Sed.exe* einsetzen, womit Sie den Inhalt einer Textdatei nach bestimmten Zeichenfolgen durchsuchen, diese durch andere Zeichenfolgen ersetzen und das Ergebnis

entweder am Bildschirm ausgeben oder per Ausgabeumleitung auch in eine gesonderte Ausgabedatei schreiben können.

Um beispielsweise alle Zeilen der Datei *C:\UserList.txt* einzulesen, jeweils am Zeilenanfang nach *FRA-MD* zu suchen und Übereinstimmungen durch *FRA-DC* zu ersetzen, verwenden Sie die nachfolgende *Sed*-Befehlszeile, die das Ergebnis per Ausgabeumleitung in die Datei *C:\Result.txt* schreibt.

```
@Echo off
Sed -e s/^FRA-MD/FRA-DC/ C:\UserList.txt > C:\Result.txt
```

Listing 250: 2_02_026.bat

Die Lösung mit VBScript

Selbst mit VBScript stellt das Suchen und Ersetzen bestimmter Zeichenfolgen kein Problem dar. Wenn Sie beispielsweise den Inhalt der Datei *C:\UserList.txt* nach der Zeichenfolge *FRA-MD* durchsuchen, dabei nur Übereinstimmungen am Zeilenanfang berücksichtigen und die Zeichenfolge durch *FRA-DC* ersetzen möchten, so öffnen Sie zunächst die betreffende Datei mit der *OpenTextFile*-Methode des *FileSystem*-Objektes und lesen deren Inhalt mit der *ReadLine*-Methode zeilenweise ein. Unter Verwendung des regulären Ausdrucks *^FRA-MD* prüfen Sie dann, ob die Zeile mit der Zeichenfolge *FRA-MD* beginnt. Wird eine Übereinstimmung gefunden, so wenden Sie die *Replace*-Methode des *RegExp*-Objektes an, um die bisherige Zeichenfolge (*FRA-MD*) mit der neuen Zeichenfolge (*FRA-DC*) zu überschreiben, wofür Sie der *Replace*-Methode den Text, der die zu ersetzende Zeichenfolge enthält, wie auch die neue Zeichenfolge als Argument übergeben.

Das nachfolgende Listing zeigt ein Beispiel dafür, wie Sie den Inhalt der Datei *C:\UserList.txt* zeilenweise einlesen, den Zeilenanfang per regulärem Ausdruck auf die Zeichenfolge *FRA-MD* hin überprüfen, Übereinstimmungen mit der *Replace*-Methode durch *FRA-DC* ersetzen und das Ergebnis in die Datei *Result.txt* ausgeben.

```
Option Explicit

Dim oFS
Dim oInputFile
Dim oOutputFile
Dim oRE
Dim strInputFile
Dim strOutputFile
Dim strInputText
Dim strOutputText
Dim colMatches
Dim objItem
```

Listing 251: 2_02_026.vbs


```

strInputFile = "C:\UserList.txt"
strOutputFile = "C:\Result.txt"
Const Read = 1
const Write = 2

Set oRE = CreateObject("VBScript.RegExp")
oRE.Pattern = "^FRA-MD"
oRE.Global = True

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oInputFile = oFS.OpenTextFile(strInputFile, Read)
Set oOutputFile = oFS.OpenTextFile(strOutputFile, Write, True)

Do While Not (oInputFile.AtEndOfStream)
    strInputText = oInputFile.ReadLine
    strInputText = oRE.Replace (strInputText, "FRA-DC")
    strOutputText = strOutputText & strInputText & VBCRLF
End If
Loop

oOutputFile.WriteLine strOutputText

oInputFile.Close
oOutputFile.Close
Set oFS = Nothing
Set oInputFile = Nothing
Set oOutputFile = Nothing
Set oRE = Nothing

```

Listing 251: 2_02_026.vbs (Forts.)

53 Leerzeilen entfernen

Wenn Sie des Öfteren Daten ex- und wieder importieren oder einfach nur Dateiinhalte per Automatismus nachbearbeiten müssen, so können Sie sicher bestätigen, dass überflüssige Leerzeilen zu einer ganzen Reihe unerwünschter Nebeneffekte führen können. Denn wenn Sie beispielsweise den Inhalt einer Datendatei in eine Datenbank importieren möchten, die zu importierende Datei aber eine oder auch mehrere Leerzeilen enthält, so werden diese Leerzeilen meist als gewöhnliche Datenzeilen interpretiert und damit auch importiert, womit in der Datenbank dann leere Datensätze entstehen.

Um nun den Inhalt einer Datei von überflüssigen Leerzeilen zu befreien und die Datei für weitere Verarbeitungsvorgänge entsprechend aufzubereiten, können Sie die Datei zum einen sortieren und die leeren Zeilen in einem Rutsch herauslöschen oder aber (wenn eine Sortierung nicht in Frage kommt) einen der Windows Server 2003-eige-

nen Editoren (NOTEPAD oder WORDPAD) einsetzen, den Dateiinhalt von Hand nach leeren Zeilen durchsuchen und die jeweiligen Zeilen manuell herauslöschen. Letzteres geht mit dem VI-Clone *WinVi32* jedoch etwas einfacher, da *WinVi32* das Erstellen eines Makros unter Verwendung regulärer Ausdrücke unterstützt, womit Sie das Suchen und Löschen leerer Zeilen zumindest ein Stück weit vereinfachen können.

Öffnen Sie hierzu die aufzubereitende Datei mit *WinVi32* (www.winvi.de), und wählen Sie im Menü OPTIONEN den Eintrag EINSTELLUNGEN aus. Holen Sie die Registerkarte TASTATUR-MAKROS in den Vordergrund, aktivieren Sie dort die Option MAP, und geben Sie im Feld NAME eine kurze Beschreibung für das neue Makro ein. Wechseln Sie anschließend in das Feld EINGABE-TASTENFOLGE und hinterlegen Sie die Taste, mit der Sie das Makro aufrufen möchten (geben Sie z.B. DBL für *Delete Blank Lines*) ein. Geben Sie dann im Feld AUSZUFÜHRENDE TASTENFOLGE den Ausdruck `:/^s*$/d` ein, und aktivieren Sie daraufhin die Optionen EINFÜGE-/ERSETZUNGSMODUS, BEFEHLSMODUS und BEFEHLSZEILEN-MODUS (EX-BEFEHL). Aktivieren Sie außerdem die Option DAUERHAFT EINSTELLUNG, und klicken Sie anschließend auf SETZEN, um dieses Makro jederzeit aufrufen zu können.

Schließen Sie das Dialogfenster EINSTELLUNGEN wieder, und geben Sie nun die zuvor festgelegte Eingabetastenfolge (z.B. DBL) ein, um das Makro auszuführen. Das Makro sucht nun nach der nächsten leeren Zeile und löscht diese Zeile. Rufen Sie das Makro so lange auf, bis das Ende der Datei erreicht ist.

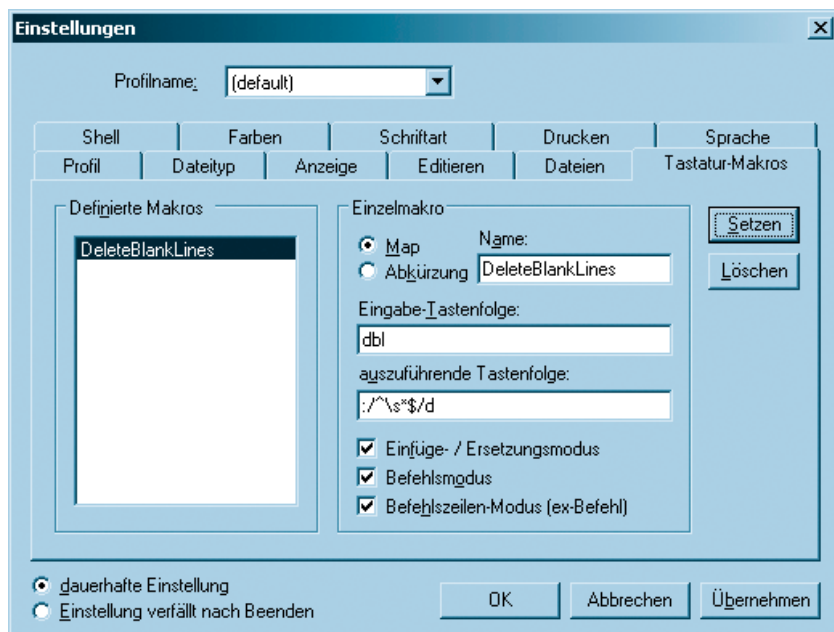


Abbildung 102: Ein Makro zum Löschen einer leeren Zeile definieren

Die Lösung per Eingabeaufforderung

Wenn Sie die Leerzeilen einer Datei von der Eingabeaufforderung aus löschen möchten, so können Sie hierfür den `Findstr`-Befehl einsetzen, womit Sie die Aufgabe vollständig automatisieren und gerade bei größeren Dateien enorm viel Aufwand und Zeit einsparen können.

Um beispielsweise alle Leerzeilen, die in der Datei `C:\UserList.txt` enthalten sind, zu entfernen und die Datei ohne jegliche Leerzeilen als `C:\Result.txt` zu speichern, verwenden Sie die folgende `Findstr`-Befehlszeile:

```
@Echo off
Findstr /r /c: "." C:\UserList.txt > C:\Result.txt
```

Listing 252: 2_02_027.bat

Die Lösung mit VBScript

Die Leerzeilen einer Datei lassen sich auch mit VBScript sehr einfach und noch dazu vollautomatisch herauslöschen. Hierzu öffnen Sie die betreffende Eingabedatei mit der `OpenTextFile`-Methode des `FileSystem`-Objektes im Lesemodus (`Read = 1`) und die Ausgabedatei, die das bereinigte Ergebnis enthalten wird, im Schreibmodus (`Write = 2`).

Anschließend lesen Sie den Inhalt der Eingabedatei mit der `ReadLine`-Methode zeilenweise ein und prüfen den Inhalt der Zeichenfolge. Enthält die Zeichenfolge mindestens ein Zeichen, so schreiben Sie die Zeichenfolge mit der `WriteLine`-Methode in die Ausgabedatei, anderenfalls setzen Sie die Verarbeitung der nächsten Zeile fort.

Das Einlesen der Zeilen, das Überprüfen der Zeichenfolge und auch das Füllen der Ausgabedatei definieren Sie dabei in einer `Do Loop`-Schleife, deren Inhalt so oft wiederholt wird, bis das Ende der Eingabedatei erreicht ist.

```
Option Explicit

Dim oFS
Dim oInputFile
Dim oOutputFile
Dim strInputFile
Dim strOutputFile
Dim strText

Const Read = 1
Const Write = 2
strInputFile = "C:\UserList.txt"
strOutputFile = "C:\Result.txt"
```

Listing 253: 2_02_027.vbs

```

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oInputFile = oFS.OpenTextFile(strInputFile, Read)
Set oOutputFile = oFS.OpenTextFile(strOutputFile, Write, True)

Do While Not oInputFile.AtEndOfStream
    strText = oInputFile.ReadLine
    If Not strText = "" Then
        oOutputFile.WriteLine strText
    End If
Loop

oInputFile.Close
oOutputFile.Close
Set oFS = Nothing
Set oInputFile = Nothing
Set oOutputFile = Nothing

```

Listing 253: 2_02_027.vbs (Forts.)

54 Nur die letzten x Zeilen einer Textdatei ausgeben

Gerade beim automatisierten Aufbereiten von Log-Dateien besteht oftmals der Bedarf, nur eine bestimmte Anzahl von Zeilen und vorzugsweise nur die *n* Zeilen am Ende der Log-Datei auszulesen, um beispielsweise nur die neu hinzugekommenen Einträge in die Verarbeitung mit einzubeziehen. Was sich mit VBScript nun auf recht einfache Art und Weise realisieren lässt, stellt unter der grafischen Oberfläche und auch in der Eingabeaufforderung leider ein ziemliches Problem dar, da der Windows Server 2003 hierfür keine eigenen Bordmittel bereitstellt.

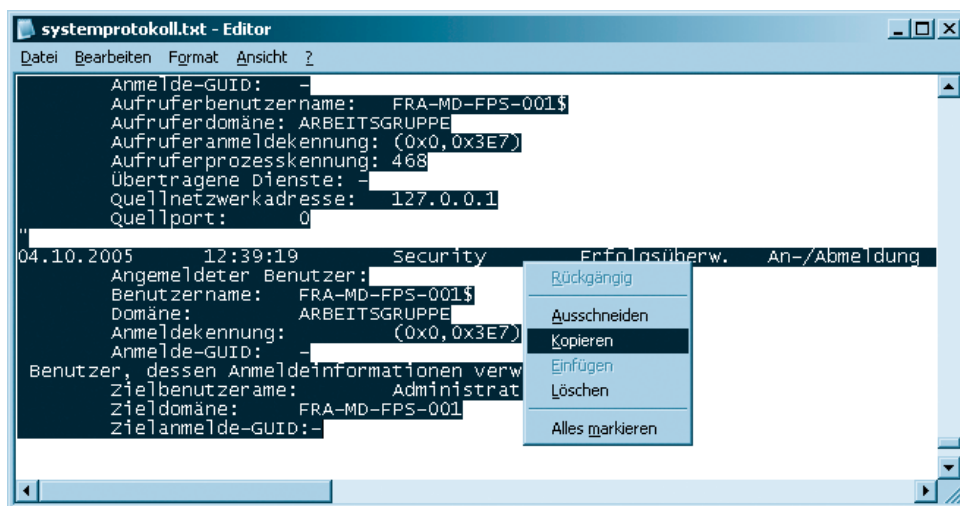


Abbildung 103: Die letzten Zeilen per Copy & Paste in eine neue Datei übertragen

Wenn Sie eine derartige Aufgabe aber dennoch unter grafischer Oberfläche ausführen müssen, so bleibt nur das manuelle Herauskopieren der betreffenden Zeilen. Hierzu öffnen Sie die entsprechende Datei mit einem Editor (z.B. NOTEPAD), scrollen bis zum Ende der Datei, markieren die gewünschten Zeilen und übertragen sie per *Copy & Paste* in eine neue Datei (siehe Abbildung 103).

Die Lösung per Eingabeaufforderung

Wenn Sie die letzten *n* Zeilen einer Log-Datei von der Eingabeaufforderung aus in eine gesonderte Datei übertragen möchten, so können Sie hierfür das Tool *Tail.exe* verwenden, das zum einen Bestandteil der *Windows Server 2003 Resource Kit-Tools* (www.microsoft.com) ist, zum anderen aber auch den *GNU-Utilities für Win32* angehört. Letzteres bietet gegenüber der Microsoft-Version einige Optionen mehr, weshalb im nachfolgenden Beispiel das GNU-Utility *Tail.exe* zum Einsatz kommt.

Möchten Sie beispielsweise die letzten 20 Zeilen der Datei *C:\Systemprotokoll.txt* in die Datei *C:\Result.txt* kopieren, so führen Sie das *Tail-Tool* unter Angabe der gewünschten Zeilenanzahl (*-n 20*) und dem Namen der Eingabedatei (*C:\Systemprotokoll.txt*) an und übertragen das Ergebnis per Ausgabeumleitung in die Datei *C:\Result.txt*.

```
@Echo off
Tail -n 20 C:\Systemprotokoll.txt > C:\Result.txt
```

Listing 254: 2_02_028a.bat

Um stattdessen die letzten 6.000 Zeichen auszugeben, wenden Sie statt des Parameters *-n* den Parameter *-c* an und übergeben die Anzahl der Zeichen, die – beginnend vom Ende der Datei – ausgegeben werden sollen.

```
@Echo off
Tail -c 6000 C:\Systemprotokoll.txt > C:\Result.txt
```

Listing 255: 2_02_028b.bat

Tipp

Wenn Sie herausfinden möchten, wie viele Zeilen eine Textdatei enthält, so können Sie hierfür das GNU-Utility *Wc.exe* einsetzen, womit Sie zum einen die Anzahl der Zeilen, zum anderen aber auch die Anzahl der Wörter und auch die der Bytes abfragen können. Weitere Informationen zu *Wc.exe* finden Sie in den Rezepten Die Anzahl der Wörter einer Textdatei ermitteln (Seite 347) und Die Zeilenanzahl einer Textdatei ermitteln (Seite 348).

Die Lösung mit VBScript

Um per VBScript die letzten *n* Zeilen aus einer Textdatei zu lesen und in eine gesonderte Ausgabedatei zu kopieren, erzeugen Sie zunächst ein neues `ADOR.Recordset`-Objekt, lesen den Inhalt der Textdatei zeilenweise ein und übertragen jede eingelesene Zeile in ein Datenfeld des `Recordset`-Objektes. Anschließend ermitteln Sie die Anzahl der Datensätze des `Recordset`-Objektes, subtrahieren davon die gewünschte Anzahl an Zeilen, die es herauszukopieren gilt, und positionieren den Datensatzzeiger auf den ermittelten Startwert. Und zu guter Letzt lesen Sie die Inhalte der verbleibenden `Recordset`-Datenfelder aus und schreiben die einzelnen Zeilen, gefolgt von einem Zeilenumbruch (`VBCRLF`), in die entsprechende Ausgabedatei.

```
Option Explicit
```

```
Dim oFS
Dim oInputFile
Dim oOutputFile
Dim oDL
Dim strInputFile
Dim strOutputFile
Dim strInputText
Dim strOutputText
Dim intLines
Dim lngStartRecord
```

```
strInputFile = "C:\Systemprotokoll.txt"
strOutputFile = "C:\Result.txt"
```

```
Const adVarChar = 200
Const MaxCharacters = 255
Const Read = 1
const Write = 2
intLines = 20
```

```
Set oFS = CreateObject("Scripting.FileSystemObject")
Set oInputFile = oFS.OpenTextFile(strInputFile, Read)
Set oOutputFile = oFS.OpenTextFile(strOutputFile, Write, True)
```

```
Set oDL = CreateObject ("ADOR.Recordset")
oDL.Fields.Append "Line", adVarChar, MaxCharacters
oDL.Open
```

```
Do While Not oInputFile.AtEndOfStream
    strInputText = oInputFile.ReadLine
    oDL.AddNew
    oDL("Line") = strInputText
```

```

    oDL.Update
Loop

LngStartRecord= oDL.Recordcount - CLng(intLines)
oDL.MoveFirst
oDL.Move LngStartRecord

Do While Not oDL.EoF
    strOutputText = strOutputText & oDL.Fields.Item("Line") & VBCRLF
    oDL.MoveNext
Loop

oOutputFile.WriteLine strOutputText

oInputFile.Close
oOutputFile.Close
Set oFS = Nothing
Set oInputFile = Nothing
Set oOutputFile = Nothing
Set oDL = Nothing

```

Listing 256: 2_02_028.vbs (Forts.)

55 Doppelte Einträge finden und löschen

Wenn Sie das Öffnen von Daten exportieren und die Exportdatei von doppelten Datenzeilen befreien müssen, so kann sich diese Aufgabe als recht aufwändig und mühsam erweisen. Denn ohne Einsatz externer Tools (z.B. einer leistungsfähigen Textverarbeitung) bleiben unter grafischer Oberfläche nur der Einsatz der Windows-eigenen Editoren NOTEPAD oder WORDPAD und das manuelle Herausfiltern der doppelten Einträge, da der Windows Server 2003 hierfür keinerlei Hilfsmittel bereitstellt.

Um doppelte Einträge schneller finden zu können, sollte der Inhalt der Datendatei zunächst sortiert werden. Allerdings stellt sich hierbei das Problem, dass die Windows-eigenen Editoren keine Sortierfunktion enthalten, weshalb Sie die betreffende Datei erst mit dem Sort-Befehl der Eingabeaufforderung (siehe Textdateiinhalte sortieren ab Seite 350) aufbereiten sollten. Nach der Sortierung öffnen Sie die Textdatei mit NOTEPAD und WORDPAD, markieren die doppelten Zeilen und löschen diese mit der Entf-Taste heraus oder wählen im Kontextmenü den Eintrag LÖSCHEN aus (siehe Abbildung 104).

Die Lösung per Eingabeaufforderung

Wenn Sie die doppelten Einträge einer Textdatei von der Eingabeaufforderung aus herausfiltern möchten, so können Sie natürlich den gleichen mühsamen Weg gehen wie unter grafischer Oberfläche. Will heißen, dass Sie die Datei zunächst mit dem Sort-Befehl aufbereiten und anschließend mit dem Editor der Eingabeaufforderung (*Edit.exe*) von Hand bereinigen können.

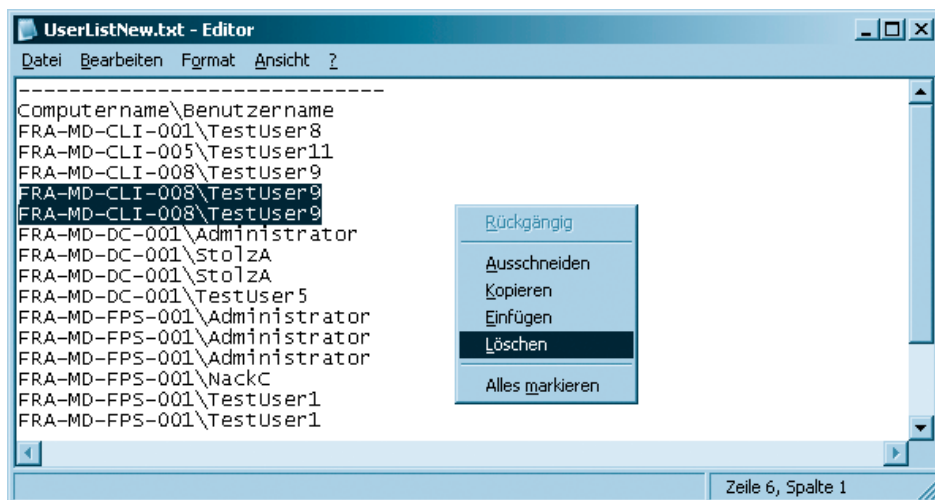


Abbildung 104: Doppelte Datenzeilen löschen

Weitaus komfortabler lassen sich die doppelten Datenzeilen aber mit den *GNU-Utilities für Win32* entfernen, da in den GNU-Utilities die Tools *Sort.exe* und *Uniq.exe* enthalten sind, womit Sie das Entfernen doppelter Einträge vollständig automatisieren können. So können Sie mit dem GNU-Utility *Sort.exe* beispielsweise den Inhalt einer Textdatei sortieren und mit den Parametern `-u -m` die Ausgabe der sortierten Datenzeilen dahingehend filtern, dass jeweils nur eindeutige Zeilen ausgegeben werden. Dadurch lassen sich alle Datenzeilen, die mehrfach vorkommen, von vornherein ausklammern. Gleiches lässt sich auch mit dem GNU-Utility *Uniq.exe* in Verbindung mit dem Parameter `-u` bewerkstelligen, allerdings mit dem Unterschied, dass *Uniq.exe* keine Sortierfunktion beinhaltet und damit nur bereits sortierte Dateien verarbeitet.

Die nachfolgende (GNU-)Sort-Befehlszeile führt eine Sortierung der Datei `C:\UserList.txt` durch und schreibt jeweils nur eindeutige Datenzeilen in die Ausgabedatei `C:\UserListNew.txt`, womit in der Ausgabedatei keine doppelten oder mehrfach vorkommenden Datenzeilen mehr enthalten sind.

```
@Echo off
Sort C:\UserList.txt -u -m > C:\UserListNew.txt
```

Listing 257: 2_02_029a.bat

Wenn eine Datei bereits in sortierter Form vorliegt, so können Sie auch den *Uniq*-Befehl verwenden, um doppelte Einträge herauszufiltern.

```
@Echo off
Uniq C:\UserList.txt -u > C:\UserListNew.txt
```

Listing 258: 2_02_029b.bat

Die Lösung mit VBScript

Selbst mit VBScript ist das Löschen von doppelten Einträgen kein Problem. Denn ähnlich der vorangegangenen Lösung unter grafischer Oberfläche und auch in der Eingabeaufforderung lesen Sie zunächst den Inhalt der Textdatei ein, sortieren die eingelesenen Zeilen in aufsteigender Reihenfolge, löschen dann die doppelten bzw. mehrfach vorkommenden Zeilen heraus und geben zum Schluss nur die eindeutigen Zeilen am Bildschirm oder auch in eine zusätzliche Ausgabedatei aus.

Zur Lösung dieser Aufgabe verwenden Sie idealerweise ein *disconnected Recordset*, womit Sie zur Laufzeit des Skriptes eine Datenbank erzeugen können, die sich ausschließlich im Arbeitsspeicher befindet und unmittelbar nach der Skriptausführung wieder gelöscht wird. Denn das *disconnected Recordset* bietet gegenüber dem Array hier den entscheidenden Vorteil, dass Sie die Daten mit Hilfe der *Sort*-Methode sortieren können und keinen eigenen Sortieralgorithmus implementieren müssen (siehe Textdateiinhalte sortieren auf Seite 352).

Das nachfolgende Listing zeigt, wie Sie den Inhalt einer Textdatei zeilenweise einlesen, in das *disconnected Recordset* übertragen, anschließend sortieren und von doppelten Einträgen befreien. Hierzu wird im ersten Schritt die zu sortierende Textdatei (*strInputFile*) im Lesemodus und die spätere Ausgabedatei (*strOutputFile*), die nur die eindeutigen Zeilen enthalten wird, im Schreibmodus geöffnet, wobei Letztere neu angelegt wird, sofern die Datei noch nicht existiert. Anschließend wird mit der *CreateObject*-Methode eine neue Instanz des ADO *Recordset*-Objektes erzeugt und mit der *Append*-Methode ein Datenfeld mit der Bezeichnung *UserName* angelegt, das vom Datentyp (siehe Tabelle 38: Die wichtigsten OLE DB-Datentypen und ihre Werte auf Seite 353) *Variant* (*adVarChar* = 200) ist und eine Feldlänge von 255 Zeichen (*MaxCharacters*=255) besitzt. Dann wird das *Recordset* mit der *Open*-Methode geöffnet, womit die Datenbank zur Aufnahme neuer Daten bereit ist.

Innerhalb einer *Do Loop*-Schleife werden nun die einzelnen Zeilen mit der *ReadLine*-Methode des *FileSystem*-Objektes eingelesen, mit der *AddNew*-Methode des *Recordset*-Objektes in das *Recordset* übertragen und per *Update*-Methode darin gespeichert. Die Schleife wird dabei so oft durchlaufen, bis alle Zeilen eingelesen und in das *Recordset* übertragen wurden.

Dann folgt der Aufruf der *Sort*-Methode, womit die Daten des *Recordsets* automatisch sortiert werden, und der Datensatzzeiger wird mit der *MoveFirst*-Methode auf den ersten Datensatz positioniert.

Nun folgt eine weitere *Do Loop*-Schleife, worin die doppelten bzw. mehrfach vorkommenden Zeilen gelöscht werden. Hierzu wird der erste Datensatz des *Recordsets* eingelesen und in der Variablen *strTmpText* zwischengespeichert. Dann wird zum nächsten Datensatz gewechselt und der Inhalt mit *strTmpText* verglichen. Bei einer Übereinstimmung wird der aktuelle *Record*-Datensatz mit der *Delete*-Methode gelöscht, womit der nächste Datensatz automatisch nachrückt und der nächste Vergleich stattfinden kann. Diese Schleife wird so lange wiederholt, bis der letzte Datensatz des *Recordsets* verarbeitet wurde.

Das Recordset enthält nun ausschließlich eindeutige Datensätze, die mit der Write-Line-Methode des FileSystem-Objektes in die Ausgabedatei (strOutputFile) geschrieben werden.

```
Option Explicit
```

```
Dim oFS
Dim oInputFile
Dim oOutputFile
Dim oDL
Dim strInputFile
Dim strOutputFile
Dim strInputText
Dim strOutputText
Dim strTmpText
```

```
strInputFile = "C:\UserList.txt"
strOutputFile = "C:\UserListNew.txt"
```

```
Const adVarChar = 200
Const MaxCharacters = 255
Const Read = 1
const Write = 2
```

```
Set oFS = CreateObject("Scripting.FileSystemObject")
Set oInputFile = oFS.OpenTextFile(strInputFile, Read)
Set oOutputFile = oFS.OpenTextFile(strOutputFile, Write, True)
```

```
Set oDL = CreateObject ("ADOR.Recordset")
oDL.Fields.Append "UserName", adVarChar, MaxCharacters
oDL.Open
```

```
Do While Not oInputFile.AtEndOfStream
    strInputText = oInputFile.ReadLine
    oDL.AddNew
    oDL("UserName") = strInputText
    oDL.Update
Loop
```

```
oDL.Sort = "UserName"
oDL.MoveFirst
```

```
Do While Not oDL.EOF
    On Error Resume Next
    strOutputText = strOutputText & oDL.Fields.Item("UserName") & VBCRLF
    strTmpText = oDL.Fields.Item("UserName")
```

```
oDL.MoveNext
If oDL.Fields.Item("UserName") = strTmpText Then
    oDL.Delete
End if
Loop

oOutputFile.WriteLine strOutputText

oInputFile.Close
oOutputFile.Close
Set oFS = Nothing
Set oInputFile = Nothing
Set oOutputFile = Nothing
Set oDL = Nothing
```

Listing 259: 2_02_029.vbs (Forts.)

56 Textdateien mit einer Zeilennummerierung versehen

Hin und wieder besteht der Bedarf, eine Textdatei mit einer Zeilennummerierung zu versehen, um beispielsweise eine Skriptdatei besser dokumentieren zu können oder eine Datendatei übersichtlicher zu gestalten. Nun bringen die Windows-eigenen Editoren (NOTEPAD und WORDPAD) leider keine entsprechende Funktion mit, womit Sie eine Textdatei zeilenweise durchnummerieren können. Sie müssen also auch in diesem Fall wieder auf alternative Möglichkeiten zurückgreifen, da der Windows Server 2003 die Nummerierung von Textdateiinhalten unter grafischer Oberfläche nicht unterstützt.

Wenn Sie die Zeilennummerierung nur im Bearbeitungsmodus benötigen, so können Sie hierfür die Editoren *WinVi32.exe* oder *Lemmy.exe* einsetzen, denn mit diesen Editoren lässt sich die Zeilennummerierung jederzeit einblenden. Unter *WinVi32.exe* rufen Sie lediglich die EINSTELLUNGEN im Menü OPTIONEN auf, wechseln auf die Registerkarte ANZEIGE und aktivieren die Option ZEILENNUMMERN, bei *Lemmy.exe* dagegen rufen Sie im Menü VIEW den Eintrag PROPERTIES auf, wechseln auf die Registerkarte VI SETTINGS und aktivieren dort die Option NUMBER. Wenn Sie die geänderte Einstellung übernehmen, wird das geöffnete Dokument mit einer entsprechenden Zeilennummerierung versehen.

Nachteil hierbei ist, dass die Zeilennummerierung nur angezeigt und nicht in der Datei gespeichert wird. Wenn Sie die Nummerierung der Zeilen in der Datei selbst speichern und damit übernehmen wollen, so können Sie hierfür das Tool *Nl.exe* (siehe nächster Abschnitt) einsetzen, das Bestandteil der *GNU-Utilities für Win32* ist.

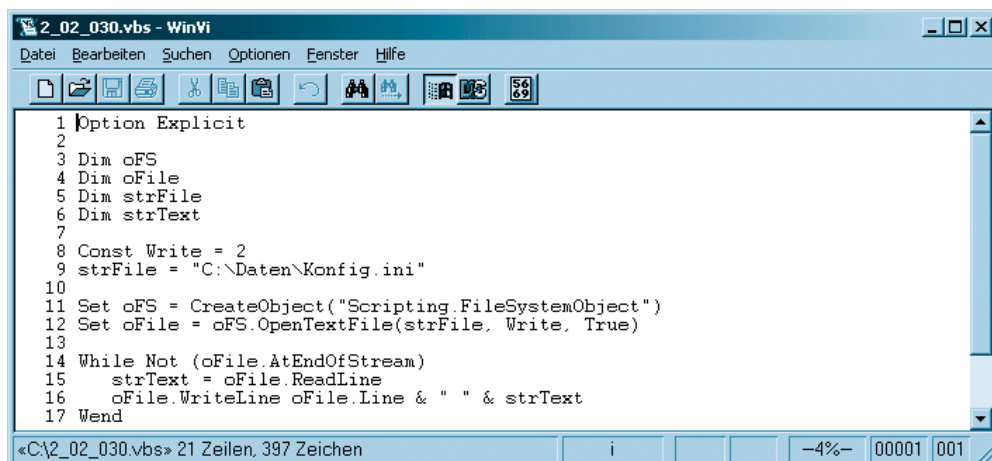


Abbildung 105: Die Zeilennummerierung mit WinVi32

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung können Sie zum einen den `Findstr`-Befehl einsetzen, zum anderen aber auch das GNU-Utility `Nl.exe`, um Textdateien mit einer Zeilennummerierung zu versehen. Wenn Sie z.B. alle Zeilen der Datei `C:\LogFiles\Protokoll.txt` nummerieren und das Ergebnis am Bildschirm ausgeben möchten, so wenden Sie im einfachsten Fall den `Findstr`-Befehl mit den Parametern `/N /C: " "` an und übergeben den Namen der Datei.

```
@Echo off
Findstr /N /C: " " C:\LogFiles\Protokoll.txt
```

Listing 260: 2_02_030a.bat

Alternativ dazu können Sie auch das GNU-Utility `Nl.exe` unter Angabe der Datei und des Parameters `-b a` einsetzen, um eine Zeilennummerierung durchzuführen. Möchten Sie die Zeilennummerierung z.B. dauerhaft übernehmen, so leiten Sie das Ergebnis mit Hilfe der Ausgabeumleitung in eine separate Datei um.

```
@Echo off
Nl C:\LogFiles\Protokoll.txt -b a > C:\LogFiles\ProtokollNew.txt
```

Listing 261: 2_02_030b.bat

Wenn Sie dem `Nl`-Befehl statt `-b a` den Parameter `-b t` übergeben, so werden nur die Zeilen nummeriert, die nicht leer sind. Alle leeren Zeilen bleiben damit von der Nummerierung unberücksichtigt.

```
@Echo off
Nl C:\LogFiles\Protokoll.txt -b t > C:\LogFiles\ProtokollNew.txt
```

Listing 262: 2_02_030c.bat

Die Lösung mit VBScript

Mit VBScript ist das Einfügen einer Zeilennummerierung kein Problem. Denn mit der `ReadLine`-Methode des `FileSystem`-Objektes lässt sich der Inhalt der Datei zeilenweise einlesen und per `WriteLine`-Methode auch wieder zeilenweise zurückschreiben. Mit Hilfe eines Counters, der bei jedem Lese- und Schreibvorgang um den Wert 1 erhöht wird, lässt sich dabei die Zeilennummer ermitteln, die Sie der Ausgabezeile als Zeilennummerierung voranstellen können.

Das nachfolgende Code-Beispiel zeigt eine solche Lösung. Nach dem Erzeugen einer neuen Instanz des `FileSystem`-Objektes wird die Eingabedatei `C:\LogFiles\Protokoll.txt` (`strInFile`) zum Lesen und die Ausgabedatei `C:\LogFiles\ProtokollNew.txt` (`strOutFile`) zum Schreiben geöffnet, wobei die Ausgabedatei neu angelegt wird, sofern sie noch nicht existiert.

Innerhalb einer `Do Loop`-Schleife wird der Inhalt der Eingabedatei zeilenweise eingelesen und in der Variablen `strText` zwischengespeichert. Mit der `WriteLine`-Methode werden dann der Wert des Zählers (`intCounter`) und auch der Inhalt der Variablen `strText` in die Ausgabedatei geschrieben. Anschließend wird der Zähler um den Wert 1 erhöht, und die Verarbeitung der nächsten Zeile beginnt. Die Schleife wird dabei so oft durchlaufen, bis das Ende der Eingabedatei (`oFileIn.AtEndOfStream`) erreicht ist.

Option Explicit

```
Dim oFS
Dim oFileIn
Dim oFileOut
Dim strInFile
Dim strOutFile
Dim strText
Dim intCounter

Const Write = 2
strInFile = "C:\LogFiles\Protokoll.txt"
strOutFile = "C:\LogFiles\ProtokollNew.txt"
intCounter = 1

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFileIn = oFS.OpenTextFile(strInFile)
Set oFileOut = oFS.OpenTextFile(strOutFile, Write, True)
```

Listing 263: 2_02_030a.vbs

```

Do While Not (oFileIn.AtEndOfStream)
    strText = oFileIn.ReadLine
    oFileOut.WriteLine intCounter & " " & strText
    intCounter = intCounter + 1
Loop

oFileIn.Close
oFileOut.Close
Set oFileIn = Nothing
Set oFileOut = Nothing
Set oFS = Nothing

```

Listing 263: 2_02_030a.vbs (Forts.)

Während das vorherige Code-Beispiel alle Zeilen durchnummeriert, fügt das nachfolgende Listing nur dann eine Zeilennummer ein, wenn die Zeile nicht leer ist. Dies wurde mit Hilfe einer IF-Anweisung umgesetzt, womit der Inhalt der Variablen `strText` geprüft wird. Wenn `strText` einen Wert enthält, so wird erst der Wert des Zählers und dann der Inhalt von `strText` zurückgeschrieben. Besitzt `strText` dagegen keinen Wert, so bleibt der Wert des Zählers unberücksichtigt.

```

Option Explicit

Dim oFS
Dim oFileIn
Dim oFileOut
Dim strInFile
Dim strOutFile
Dim strText
Dim intCounter

Const Write = 2
strInFile = "C:\LogFiles\Protokoll.txt"
strOutFile = "C:\LogFiles\ProtokollNew.txt"
intCounter = 1

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFileIn = oFS.OpenTextFile(strInFile)
Set oFileOut = oFS.OpenTextFile(strOutFile, Write, True)

Do While Not (oFileIn.AtEndOfStream)
    strText = oFileIn.ReadLine
    If strText <> "" Then
        oFileOut.WriteLine intCounter & " " & strText
    End If
    intCounter = intCounter + 1
Loop

```

Listing 264: 2_02_030b.vbs

```
        intCounter = intCounter + 1
    Else
        oFileOut.WriteLine strText
    End If
Loop

oFileIn.Close
oFileOut.Close
Set oFileIn = Nothing
Set oFileOut = Nothing
Set oFS = Nothing
```

Listing 264: 2_02_030b.vbs (Forts.)

57 Die Anzahl der Wörter einer Textdatei ermitteln

Für den Fall, dass Sie die Anzahl der Wörter einer Textdatei ermitteln müssen, weil Sie vielleicht die genaue Anzahl der Elemente einer Datendatei benötigen, so bleibt nur der Einsatz eines externen Tools, da der Windows Server 2003 hierfür kein entsprechendes Werkzeug bereitstellt.

Wenn der Server über eine Internetverbindung verfügt und wenn Sie nicht den Inhalt einer sensitiven Datei verarbeiten müssen, so können Sie beispielsweise einen Online-Wordcounter wie ECLaunchsite.com/ECLWordCounter.htm verwenden. Hierzu rufen Sie lediglich die zuvor genannte Internetseite auf, kopieren die zählenden Zeilen in das WordCounter-Fenster und klicken anschließend auf COUNT, woraufhin die im WordCounter-Fenster enthaltenen Wörter gezählt werden.

Alternativ dazu können Sie auch das GNU-Utility *Wc.exe* (siehe nächster Abschnitt) verwenden, um die Anzahl der Wörter zu ermitteln, die in einer Textdatei enthalten sind.

Die Lösung per Eingabeaufforderung

Mit Hilfe der *GNU-Utilities für Win32* lässt sich diese Aufgabe prima von der Eingabeaufforderung aus lösen, denn dieses Sammelurium enthält unter anderem das Tool *Wc.exe*, womit Sie die Anzahl der Zeilen, Wörter und Zeichen einer Datei ermitteln können.

Um beispielsweise die Anzahl der Wörter zu ermitteln, die in der Datei *C:\Daten\ID.Dat* enthalten sind, wenden Sie den *WC*-Befehl mit dem Parameter *-w* an und übergeben den Namen der Datei. Der *WC*-Befehl liefert daraufhin die Anzahl der Wörter zurück und gibt das Ergebnis am Bildschirm aus.

```
@Echo off
Wc -w C:\Daten\ID.Dat
```

Listing 265: 2_02_031.bat

Die Lösung mit VBScript

Mit Hilfe des GNU-Utilities `Wc.exe` und der `Run`-Methode des `Shell`-Objektes lässt sich die Anzahl der Wörter, die in einer Textdatei enthalten sind, auch per VBScript sehr einfach ermitteln. Denn Sie brauchen nur den `Wc`-Befehl unter Angabe des Parameters `-w` und der betreffenden Datei auszuführen, um die gewünschte Information zu erhalten.

Das nachfolgende Code-Beispiel zeigt, mit welcher wenigen Code-Zeilen sich die Aufgabe lösen lässt: Nach dem Erzeugen einer neuen Instanz des `Shell`-Objektes wird die `Run`-Methode angewandt, um den `Wc`-Befehl auszuführen.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c wc -w C:\Daten\ID.Dat"

Set oShell = Nothing
```

Listing 266: 2_02_031.vbs

58 Die Zeilenanzahl einer Textdatei ermitteln

Wenn Sie auf die Schnelle die Zeilenanzahl einer Textdatei ermitteln müssen, so brauchen Sie die betreffende Textdatei nur mit einem Editor (z.B. `NOTEPAD`, `WORDPAD`) zu öffnen und den Cursor an das Ende der Textdatei zu setzen. Denn die meisten Editoren zeigen in der Statuszeile die Position des Cursors unter Angabe der aktuellen Zeile und Spalte an, woran Sie die Zeilenanzahl erkennen können.

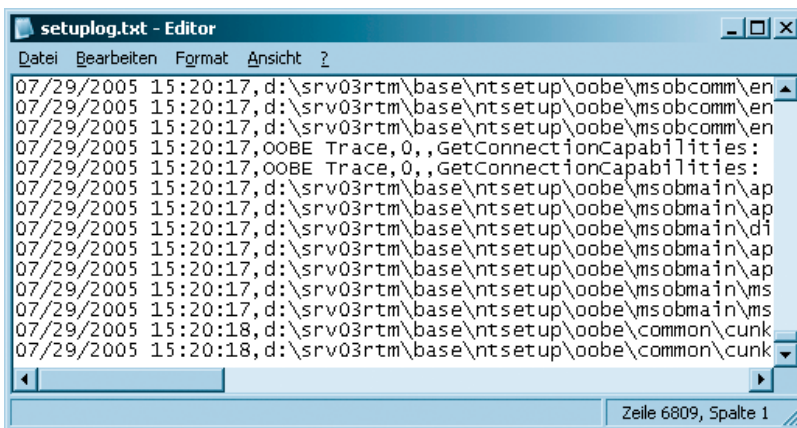


Abbildung 106: Die Zeilenanzahl einer Datei ermitteln

Um beispielsweise die Zeilenanzahl der Datei `C:\Windows\System32\Setuplog.txt` zu ermitteln, öffnen Sie die Datei mit dem Windows-eigenen Editor namens `NOTEPAD` und setzen den Cursor mit Hilfe der Tastenkombination `[Strg] + [Ende]` an das Ende der Datei. Daraufhin wird in der Statuszeile des Notepad-Fensters (falls diese nicht sichtbar ist, können Sie die Statuszeile im Menü `ANSICHT` einschalten) die aktuelle Cursor-Position angezeigt, und Sie können anhand des Wertes `ZEILE` erkennen, wie viele Zeilen die Textdatei enthält.

Falls Sie *Lemmy* oder *WinVi32* einsetzen, so müssen Sie den Cursor nicht an das Ende der Datei setzen. Denn diese Editoren geben die aktuelle Cursor-Position und auch die Anzahl der Zeilen getrennt voneinander aus, womit Sie die Zeilenanzahl jederzeit der Statuszeile entnehmen können.

Die Lösung per Eingabeaufforderung

Die Zeilenanzahl einer Textdatei lässt sich auch per Eingabeaufforderung ermitteln, wobei Sie hierfür aber ein alternatives Tool einsetzen müssen, da der Windows Server 2003 keinen entsprechenden Befehl bereithält. Verwenden Sie z.B. das Tool `Wc.exe`, das Bestandteil der *GNU-Utilities für Win32* ist, um die Anzahl der Zeilen zu ermitteln.

Wenn Sie z.B. herausfinden müssen, wie viele Zeilen die Datei `C:\LogFiles\Setuplog.txt` enthält, so setzen Sie den `Wc`-Befehl unter Angabe des Parameters `-l` und der betreffenden Datei ein. Der `Wc`-Befehl liefert daraufhin die Anzahl der Zeilen zurück und gibt das Ergebnis am Bildschirm aus.

```
@Echo off
Wc -l C:\LogFiles\Setuplog.txt
```

Listing 267: 2_02_032.bat

Die Lösung mit VBScript

Gleiches lässt sich auch per VBScript realisieren, indem Sie die Datei zunächst zeilenweise einlesen und am Ende der Datei die aktuelle Zeilennummer ermitteln. Um beispielsweise die Zeilenanzahl der Datei `C:\LogFiles\Setuplog.txt` herauszufinden, erzeugen Sie mit der `CreateObject`-Methode eine neue Instanz des `FileSystem`-Objektes und öffnen die betreffende Datei (`strFile`) zum Lesen (`Read = 1`).

Innerhalb einer `Do Loop`-Schleife lesen Sie dann den Inhalt der angegebenen Datei zeilenweise ein, wofür das `FileSystem`-Objekt die `ReadLine`-Methode bereitstellt. Diese Schleife wird so lange durchlaufen, bis die Eigenschaft `AtEndOfStream` den Wert `True` erhält, womit das Ende der Datei gekennzeichnet wird. Wenn das Ende der Datei erreicht ist, wird die Schleife verlassen und der Wert der Eigenschaft `Line` ermittelt, die Auskunft über die aktuelle Zeilennummer gibt. Das Ergebnis wird daraufhin mit der `WScript.Echo`-Anweisung am Bildschirm ausgegeben.

```

Option Explicit

Dim oFS
Dim oFile
Dim strFile
Dim strText
dim intlines

Const Read = 1
strFile = "C:\LogFiles\Setuplog.txt"

Set oFS = CreateObject("Scripting.FileSystemObject")
Set oFile = oFS.OpenTextFile(strFile, Read)

Do While Not oFile.AtEndOfStream
    strText = oFile.ReadLine
Loop

WScript.Echo "Anzahl der Zeilen: " & oFile.Line -1
oFile.Close

oFile.Close
Set oFile = Nothing
Set oFS = Nothing

```

Listing 268: 2_02_032.vbs

59 Textdateiinhalte sortieren

Die Windows-eigenen Editoren *Notepad* und *Wordpad* eignen sich zwar recht gut für die schnelle Bearbeitung von Textdateien, für die Sortierung von Dateiinhalten lassen sich diese Tools aber nicht einsetzen, da diese Textverarbeitungswerkzeuge nicht mit entsprechenden Sortier-Funktionen ausgestattet sind.

Wenn Sie des Öfteren Dateiinhalte sortieren müssen und dies vorzugsweise unter grafischer Oberfläche tun möchten, so empfiehlt sich der Einsatz eines alternativen Editors, der neben den Standardfunktionen einer Textverarbeitung auch über entsprechende Sortier-Funktionen verfügt oder zumindest den Aufruf eines externen Befehls erlaubt, wie dies beim VI-Clone *WinVi* und auch bei *Lemmy* der Fall ist. Denn Letzteres ermöglicht – trotz fehlender Sortier-Funktion – das Sortieren eines Textes, indem der Textinhalt an den `Sort`-Befehl der Eingabeaufforderung übergeben und dadurch kurzerhand sortiert wird, ohne dass hierfür die Eingabeaufforderung selbst aufgerufen werden muss. Zugegeben, dieser Workaround mag auf den ersten Blick etwas aufwändig klingen, ist es aber nicht.

Um den Inhalt einer Textdatei z.B. mit *Lemmy* zu sortieren, rufen Sie zunächst das Tool *Lemmy* auf und öffnen die betreffende Textdatei. Falls die Textdatei neben dem zu sortierenden Dateninhalt auch zusätzliche Kopf- und Fußzeilen mit Spaltenüberschriften, Zusammenfassungen etc. enthält, so notieren Sie sich zunächst die Start- und Endezeile des eigentlichen Dateninhaltes, um die Kopf- und Fußzeilen aus der Sortierung ausklammern zu können, oder entfernen die Kopf- und Fußzeilen komplett aus der Datei. Anschließend wählen Sie im Menü EDIT den Eintrag EX COMMAND... aus, woraufhin Sie in das Dialogfenster EX COMMAND gelangen und einen externen Befehl ausführen können.

Wenn Sie alle Zeilen sortieren möchten, so geben Sie den Befehl `%!SORT` ein, womit Sie den Textinhalt an den Sort-Befehl der Eingabeaufforderung zur Verarbeitung übergeben können, der daraufhin in sortierter Form im *Lemmy*-Fenster angezeigt wird. Wenn Sie stattdessen nur die Zeilen 3 bis 10 sortieren wollen, so verwenden Sie den Befehl `3,10!Sort` und klicken dann auf OK, woraufhin der Sort-Befehl nur auf die ausgewählten Zeilen ausgeführt wird. Um die Datei daraufhin in sortierter Form zu speichern, wählen Sie im Menü FILE den Eintrag SAVE und – wenn Sie *Lemmy* beenden möchten – auch CLOSE aus.

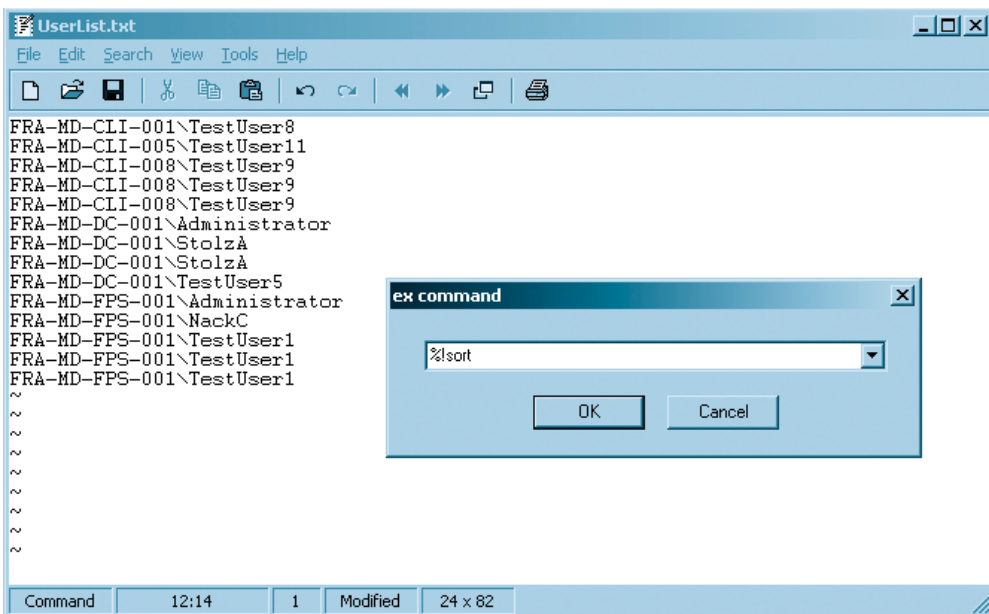


Abbildung 107: Den Inhalt einer Textdatei mit dem Ex-Kommando »%!sort« sortieren

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung ist das Sortieren von Textdateiinhalten auch ohne Einsatz eines externen Tools möglich, da der Windows Server 2003 hierfür den `Sort`-Befehl bereitstellt, womit Sie die Inhalte einzelner Dateien auf- und auch absteigend sortieren können.

Wenn Sie z.B. den Inhalt der Datei *C:\Userlist.txt* aufsteigend sortieren und das Ergebnis in die Ausgabedatei *C:\UserlistNew.txt* schreiben möchten, so verwenden Sie in der Eingabeaufforderung folgende Sort-Befehlszeile:

```
@Echo off
Sort C:\UserList.txt /O C:\UserListNew.txt
```

Listing 269: 2_02_033a.bat

Möchten Sie den Inhalt der Datei *C:\UserList.txt* dagegen in absteigender Reihenfolge sortieren, so fügen Sie dem Sort-Befehl den zusätzlichen Parameter */R* an.

```
@Echo off
Sort C:\UserList.txt /R /O C:\UserListNew.txt
```

Listing 270: 2_02_033b.bat

Die Lösung mit VBScript

Wenn Sie den Inhalt einer Textdatei dagegen mit VBScript sortieren möchten, so lesen Sie den Inhalt der Textdatei zunächst zeilenweise ein, nutzen entweder einen Sortieralgorithmus wie *Bubble Sort* oder *Quick Sort*, oder aber Sie verwenden ein *disconnected Recordset* und sortieren die Daten, sprich: die Zeilen, mit der Sort-Methode auf automatischem Wege.

Die nachfolgende Übersicht soll einen kleinen Überblick über die wichtigsten Sortier-Methoden mit VBScript geben und jeweils die Vor- und Nachteile sowie deren primäre Einsatzgebiete aufzeigen:

- ▶ **Bubble Sort:** Beim Bubble Sort handelt es sich um einen sehr einfach zu implementierenden, aber dafür recht langsamen Sortieralgorithmus, der primär für kleine Datenmengen in eindimensionalen Tabellen verwendet wird. Die Sortierung erfolgt dabei immer durch den Vergleich von zwei nebeneinander liegenden Elementen der Tabelle. Ist das erste Element größer als das zweite Element, so werden beide Elemente getauscht. Ist das erste Element dagegen kleiner, so werden die nächsten beiden Elemente miteinander verglichen. Durch dieses Verfahren gelangt das größte Element sukzessive an das Ende der Tabelle, wodurch das sortierte Ergebnis entsteht.
- ▶ **Quick Sort:** Der etwas schnellere, dafür aber etwas aufwändiger zu implementierende Quick Sort zieht lediglich unsortierte Elemente der Tabelle für den Vergleich heran und nicht – wie beim Bubble Sort – auch bereits sortierte Elemente. Hierzu wird ein Element aus der Tabelle als Vergleichselement gewählt. Der Vergleich mit den anderen Elementen erfolgt nun von der Mitte aus jeweils nach links und nach rechts, wobei das jeweils größere Element von links bzw. von rechts zur Mitte hin verschoben wird und die linke bzw. rechte Tabellengrenze mit jedem Durchlauf neu gesetzt wird, womit sich die Anzahl der zu vergleichenden Elemente sukzessive verringert.

- **Disconnected Recordset:** Eine weitere Möglichkeit, Daten per VBScript zu sortieren, bietet das disconnected Recordset, das eine im Speicher befindliche Datenbank darstellt, die erst zur Laufzeit des Skriptes erzeugt und nach dem Beenden des Skriptes auch direkt wieder gelöscht wird (vergleichbar mit einem Array). Der Vorteil des disconnected Recordsets besteht darin, dass Sie sich nicht um den eigentlichen Sortieralgorithmus kümmern müssen, da das disconnected Recordset die Sort-Methode bereitstellt, womit sich die Daten durch den Methodenaufruf automatisch sortieren lassen. Das disconnected Recordset ermöglicht zudem eine schnelle Sortierung auch größerer Datenmengen, lässt sich recht einfach implementieren und ist insbesondere dann von Vorteil, wenn Datensätze aus mehreren Datenfeldern bestehen. Denn gerade Letzteres kann den Implementierungsaufwand bei Verwendung des Bubble-Sort- oder Quick-Sort-Verfahrens immens erhöhen.

Da das disconnected Recordset eine sehr einfach zu implementierende, wenngleich recht komfortable Sortiermöglichkeit darstellt, soll im Folgenden einmal demonstriert werden, wie sich Daten mit dem disconnected Recordset sortieren lassen.

Hierzu öffnen Sie zunächst die zu sortierende Datei im Lesemodus und die spätere Zieldatei, die das sortierte Ergebnis enthalten wird, im Schreibmodus. Anschließend erzeugen Sie mit der `CreateObject`-Methode eine neue Instanz des ADO Recordset-Objektes, definieren mit der `Append`-Methode die erforderlichen Felder und übergeben dabei den Feldnamen, den Datentyp (siehe nachfolgende Tabelle) und auch die Feldlänge als Argument.

Datentyp	Wert	Beschreibung
adBoolean	11	Boole'scher Wert
adDBDate	133	Datumswert
adDBTime	134	Zeitwert
adGUID	72	GUID-Wert
adInteger	3	Ganzzahl mit einer Länge von 4 Byte
advarBinary	204	Binärwert
adVarChar	200	Zeichenfolge

Tabelle 38: Die wichtigsten OLE DB-Datentypen und ihre Werte

Im nächsten Schritt öffnen Sie mit der `Open`-Methode das Recordset, um neue Daten aufnehmen zu können. Nun lesen Sie den Inhalt der Quelldatei mit der `ReadLine`-Methode des `FileSystem`-Objektes zeilenweise ein, erzeugen mit der `AddNew`-Methode des ADO Recordset-Objektes jeweils einen neuen Datensatz, weisen dabei den Inhalt der eingelesenen Zeile als Feldinhalt zu und speichern den Datensatz mit der `Update`-Methode ab. Wenn alle Zeilen im Recordset hinterlegt sind, können Sie die `Sort`-Methode anwenden, um den Inhalt des Recordsets zu sortieren.

Und zu guter Letzt lesen Sie die – in sortierter Form vorliegenden – Feldinhalte des Recordsets aus und schreiben die Informationen mit der WriteLine-Methode des FileSystem-Objektes in die gewünschte Zieldatei.

```
Option Explicit
```

```
Dim oFS
Dim oInputFile
Dim oOutputFile
Dim oDL
Dim strInputFile
Dim strOutputFile
Dim strInputText
Dim strOutputText
```

```
strInputFile = "C:\UserList.txt"
strOutputFile = "C:\UserListNew.txt"
```

```
Const adVarChar = 200
Const MaxCharacters = 80
Const Read = 1
const Write = 2
```

```
Set oFS = CreateObject("Scripting.FileSystemObject")
Set oInputFile = oFS.OpenTextFile(strInputFile, Read)
Set oOutputFile = oFS.OpenTextFile(strOutputFile, Write, True)
```

```
Set oDL = CreateObject ("ADOR.Recordset")
oDL.Fields.Append "UserName", adVarChar, MaxCharacters
oDL.Open
```

```
Do While Not oInputFile.AtEndOfStream
    strInputText = oInputFile.ReadLine
    oDL.AddNew
    oDL("UserName") = strInputText
    oDL.Update
Loop
```

```
oDL.Sort = "UserName"
oDL.MoveFirst
```

```
Do While Not oDL.EOF
    strOutputText = strOutputText & oDL.Fields.Item("UserName") & VBCRLF
    oDL.MoveNext
Loop
```

```
oOutputFile.WriteLine strOutputText

oInputFile.Close
oOutputFile.Close
Set oFS = Nothing
Set oInputFile = Nothing
Set oOutputFile = Nothing
Set oDL = Nothing
```

Listing 271: 2_02_033.vbs (Forts.)

60 Dateien und Ordner sichern und wiederherstellen

Der Windows Server 2003 ist bereits von Haus aus mit einem Backup-Utility ausgestattet, das Sie zur Sicherung und auch zur Wiederherstellung der Dateien und Ordner verwenden können. Dieses Backup-Utility trägt den Namen NTBACKUP und setzt – entgegen einiger anderer Backup-/Recovery-Lösungen – keine aufwändige Konfigurationsarbeit voraus, womit sich NTBACKUP sehr gut zur schnellen Datensicherung einsetzen lässt.

Das Windows Server 2003-eigene Datensicherungstool unterstützt dabei zahlreiche Speichermedien, die Sie als Ziellaufwerk für die Datensicherung nutzen können. Bei der Wahl des geeigneten Speichermediums müssen zumindest die folgenden Kriterien berücksichtigt werden:

- ▶ Kapazität des Speichermediums
- ▶ Hardware-Kompatibilität mit Windows Windows Server 2003
- ▶ Geschwindigkeit (Datentransferrate, Hardware-Kompression ...)
- ▶ Möglichkeiten der Fehlerkorrektur

Grundsätzlich können Sie jedes Speichermedium für die Datensicherung mit NTBACKUP verwenden, das mit dem Wechselmediendienst verwaltet werden kann und das keine Kompatibilitätsprobleme mit Windows Server 2003 aufweist. Auch können Sie lokale Festplatten als Ziellaufwerk festlegen. Eines wird aber von NTBACKUP nicht unterstützt: die direkte Datensicherung auf CD-R/RW oder DVD-R/RW. Grund dafür ist, dass die Wechselmedienverwaltung keine CD- und DVD-Datenträger als Medien für die Datensicherung akzeptiert, wenngleich eine Verwaltung dieser Medien mit der Wechselmedienverwaltung aber möglich ist.

Die folgende Übersicht zeigt, welche Speichermedien für die Datensicherung genutzt werden können und über welche speziellen Leistungsmerkmale die jeweiligen Medien verfügen:

- ▶ Streamer: Das Bandlaufwerk ist das gängigste Speichermedium für die regelmäßige Datensicherung. In der Regel werden QIC-Kassetten (Quarter-Inch Cartridge), DAT-Bänder (Digital Audio Type), 8-mm-Kassetten und DLT-Bänder (Digital Linear Type) verwendet. Die Bänder haben eine Kapazität zwischen 2 und 400 GByte.

- ▶ CD-R/RW und DVD-R/RW: In Verbindung mit der Packet-Writing-Technologie eignet sich die CD und DVD sehr gut als Speichermedium. Je nach verwendetem Medientyp sind Kapazitäten von 650–4.500 MByte möglich, für eine inkrementelle, differenzielle oder tägliche Sicherung geringer Datenmengen also oft schon ausreichend. In Verbindung mit NTBackup kann die CD-R/RW und DVD-R/RW jedoch nur genutzt werden, wenn die Sicherungsdatei temporär auf der lokalen Festplatte zwischengespeichert wird. Eine direkte Sicherung auf diese Medien ist leider nicht möglich.
- ▶ Festplatten: Interne Festplatten oder Wechselplatten sind empfehlenswert, wenn des Öfteren Datenbestände gesichert werden müssen und ein schneller Zugriff auf das Medium erforderlich ist. Mit einem Fassungsvermögen von bis zu 200 GByte eignet sich die Harddisk als Backup-Lösung, um mehrere Datenbestände zentral zu sichern. Die Laufwerke sind für alle gängigen Anschlusstypen (IDE, SCSI, USB, Firewire) erhältlich und erfordern keine aufwändige Konfiguration.
- ▶ MO-Laufwerke: MO-Medien (magnetooptische Medien) eignen sich aufgrund der hohen Datensicherheit und einer Kapazität von bis zu 2,5 GByte als professionelle Backup-Medien. Die Medien werden bei Temperaturen von über 300 Grad magnetisiert, wodurch das Risiko minimiert werden kann, dass Daten aufgrund von Magnetfeldern zerstört werden. MO-Laufwerke gibt es für alle gängigen Anschlusstypen (IDE, SCSI, Firewire, Parallel ...). MO-Laufwerke gibt es sowohl für den internen Einbau wie auch als externes Gerät.
- ▶ ZIP-Drive: Das ZIP-Drive eignet sich für die Datensicherung mit geringem Sicherungsvolumen. Eine Kapazität von 100 bis 750 MB erlaubt demnach keine Gesamtsicherung der Systempartition auf einem Medium. Das ZIP-Drive hat geringe Anschaffungskosten und ist für den IDE-, SCSI-, USB- und parallelen Anschluss erhältlich. Vorteilhaft ist, dass das ZIP-Drive ohne großen Aufwand an unterschiedlichen Computern genutzt werden kann.

Neben der Wahl des geeigneten Backup-Mediums gilt es natürlich auch, eine entsprechende Backup-Strategie zu entwickeln, die zum einen eine zuverlässige Datensicherung gewährleistet, zum anderen aber auch eine schnelle Wiederherstellung der Daten garantiert und zudem noch den Backup-Aufwand und auch die Sicherungszeit berücksichtigt.

Bei einer gängigen Backup-Methode, die sich in der Praxis sehr bewährt hat und daher in vielen IT-Infrastrukturen eingesetzt wird, findet beispielsweise einmal wöchentlich (bei kleineren Datenmengen auch einmal monatlich) eine vollständige Datensicherung aller Dateien und Ordner unter Verwendung der Sicherungsart *Normal* statt. Diese vollständige Datensicherung wird meist am Wochenende durchgeführt. Unter der Woche finden dann nur *inkrementelle* oder *tägliche* Datensicherungen statt, womit jeweils nur die Dateien und Ordner gesichert werden, die seit der letzten vollständigen, inkrementellen oder täglichen Datensicherung erstellt oder geändert wurden. Damit wird der Backup-Aufwand unter der Woche auf ein Minimum reduziert, und es wird weitaus weniger Backup-Speicherplatz verwendet. Nachteil ist jedoch, dass zur vollständigen Wiederherstellung aller Daten mehrere Datensicherungen restauriert werden müssen,

da erst die vollständige Sicherung und dann die inkrementellen bzw. täglichen Sicherungen wiederhergestellt werden müssen.

Je nachdem, welche Daten Sie in welchem Rhythmus sichern möchten, gilt es also, zunächst die geeignete Sicherungsart (siehe nachfolgende Tabelle) auszuwählen und dann den eigentlichen Zeitplan festzulegen.

Sicherungsart	Beschreibung
Normal	Mit Auswahl der Sicherungsart <i>Normal</i> werden alle ausgewählten Dateien/Ordner auf das zuvor festgelegte Medium oder in den angegebenen Pfad gesichert. Diese Sicherungsart entfernt bei allen gesicherten Datei- und Ordner-Objekten das <i>Archiv</i> -Attribut, womit die ausgewählten Dateien und Ordner als gesichert markiert und bei der nächsten inkrementellen, differenziellen und täglichen Sicherung nur berücksichtigt werden, wenn die Datei- und Ordner-Objekte zwischenzeitlich geändert wurden. Diese Sicherungsart wird vorzugsweise für die vollständige Datensicherung eingesetzt und daher oftmals auch als <i>Full-Backup</i> bezeichnet.
Kopieren	Die Sicherungsart <i>Kopieren</i> sichert ebenfalls alle ausgewählten Dateien und Ordner auf das zuvor festgelegte Medium oder in den angegebenen Pfad, wobei mit Einsatz dieser Sicherungsart das <i>Archiv</i> -Attribut jedoch nicht zurückgesetzt wird, sondern weiterhin bestehen bleibt. Diese Sicherungsart wird meist für eine Datensicherung außerhalb des regulären Backup-Rhythmus (Ad-hoc-Backup für zwischendurch) gewählt, da das Archiv-Attribut nicht verändert und damit die Datei-/Ordner-Auswahl bei der nächsten inkrementellen, differenziellen oder täglichen Datensicherung nicht beeinflusst wird.
Inkrementell	Die Auswahl der Sicherungsart <i>Inkrementell</i> hat zur Folge, dass nur die Dateien und Ordner auf das zuvor festgelegte Medium oder in den angegebenen Pfad gesichert werden, die seit der letzten Datensicherung vom Typ <i>Normal</i> oder <i>Inkrementell</i> erstellt oder geändert wurden. Bei der inkrementellen Datensicherung wird das Archiv-Attribut eines jeden Datei- oder Ordner-Objektes entfernt, womit die Dateien und Ordner als »gesichert« markiert werden.
Differenziell	Die Sicherungsart <i>Differenz</i> bewirkt, dass alle Dateien und Ordner auf das festgelegte Medium oder in den angegebenen Pfad gesichert werden, die seit dem letzten Backup des Typs <i>Normal</i> oder <i>Inkrementell</i> erstellt oder geändert wurden, wobei unter Verwendung der <i>Differenz</i> -Sicherung die Archiv-Attribute nicht verändert werden. Die gesicherten Datei- und Ordner-Objekte werden somit nicht als »gesichert« markiert.
Täglich	Mit Auswahl der Sicherungsart <i>Täglich</i> werden alle ausgewählten Dateien und Ordner auf das festgelegte Medium oder in den angegebenen Pfad gesichert, die an dem gleichen Tag, an dem die Datensicherung durchgeführt wird, erstellt oder geändert wurden. Die tägliche Sicherung führt nicht dazu, dass das Archiv-Attribut entfernt wird, womit die gesicherten Datei- und Ordner-Objekte auch nicht als »gesichert« markiert werden.

Tabelle 39: Die Sicherungsarten von NTBackup

Um nun die Dateien und Ordner mit Hilfe von NTBACKUP zu sichern, starten Sie zunächst das Datensicherungstool, wählen dann die entsprechenden Dateien und Ordner aus bzw. selektieren die gewünschte Sicherungsart, legen das Backup-Medium oder den Datensicherungspfad fest und starten dann die Datensicherung, woraufhin alle selektierten Datei- und Ordner-Objekte gesichert werden.

Hierzu öffnen Sie das STARTMENÜ, erweitern ALLE PROGRAMME, klicken auf SYSTEMPROGRAMME und wählen dann den Eintrag SICHERUNG aus, um NTBACKUP zu starten. Alternativ dazu können Sie das Datensicherungstool auch direkt aufrufen, indem Sie erst auf START klicken, AUSFÜHREN auswählen, anschließend NTBACKUP eingeben und dann die -Taste drücken oder die Schaltfläche OK betätigen.

Mit dem ersten Aufruf von NTBACKUP wird automatisch der SICHERUNGS- ODER WIEDERHERSTELLUNGS-ASSISTENT gestartet. Klicken Sie auf ERWEITERTEN MODUS, um den Assistenten zu beenden und in die eigentliche Oberfläche von NTBACKUP zu gelangen. Wechseln Sie anschließend auf die Registerkarte SICHERN. Sie sehen nun eine WINDOWS EXPLORER-ähnliche Ansicht, worüber Sie die zu sichernden Dateien und Ordner markieren können. Um beispielsweise eine vollständige Datensicherung aller Dateien und Ordner des Laufwerks C durchzuführen, markieren Sie in der linken Fensterhälfte das Root des Laufwerks (*Lokaler Datenträger C:*), woraufhin alle untergeordneten Dateien und Objekte automatisch selektiert werden.

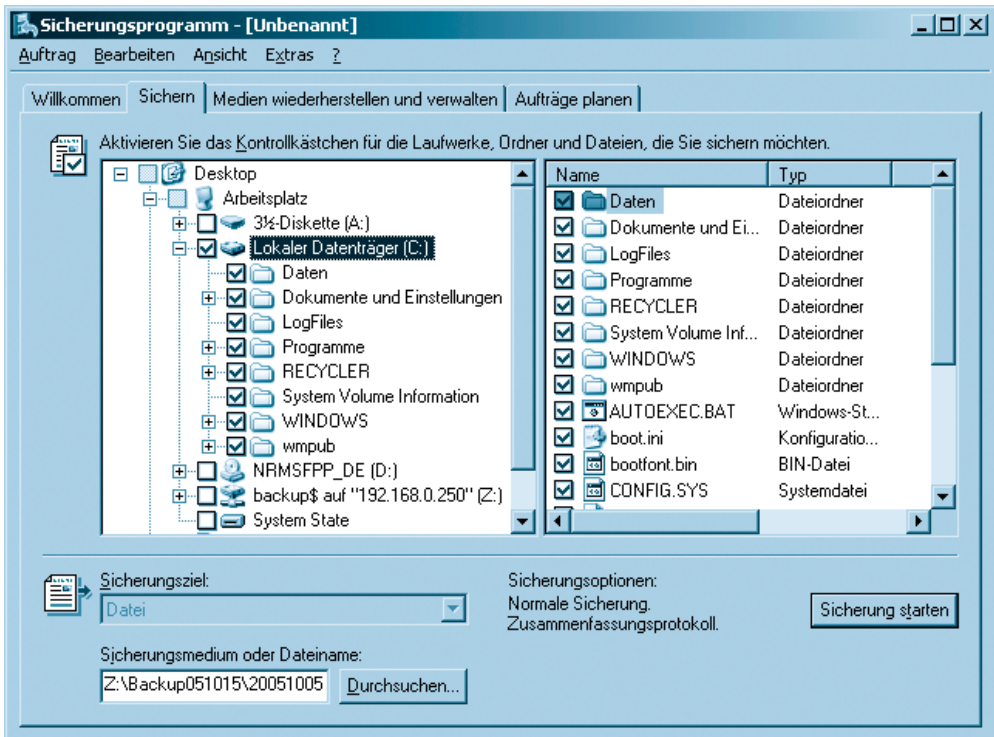


Abbildung 108: Die zu sichernden Dateien auswählen

Klicken Sie am unteren Bildschirmrand auf DURCHSUCHEN, um das Speichermedium oder den Zielpfad zu bestimmen, und legen Sie den Pfad wie auch den Dateinamen für die Sicherungsdatei fest. Klicken Sie anschließend auf SICHERUNG STARTEN, um in die erweiterten Einstellungen für diesen Datensicherungsauftrag zu gelangen. Im Dialogfenster INFORMATIONEN ZUM SICHERUNGS-AUFTRAG haben Sie nun die Möglichkeit, eine kurze Beschreibung zu der Datensicherung (z.B. *Satz am 08.10.2005 um 04:00 erstellt*) einzugeben und durch Auswahl der Optionen SICHERUNGSKOPIE DEM MEDIUM ANHÄNGEN und DATEN AUF DEM MEDIUM DURCH DIESE SICHERUNG ERSETZEN festzulegen, ob diese neue Datensicherung einer vorhandenen Sicherung angehängt werden soll oder ob – falls bereits Sicherungen vorhanden sind – bestehende Sicherungen ersetzt werden sollen. Letzteres führt dazu, dass alle Sicherungen, die sich auf einem Band befinden, vor dem Sicherungsvorgang gelöscht werden.

Wählen Sie die gewünschte Option aus, und klicken Sie anschließend auf ERWEITERT, um die Sicherungsart (siehe Tabelle 39: Die Sicherungsarten von NTBackup auf Seite 357) auszuwählen und auch die Überprüfung der Sicherung zu aktivieren. Schließen Sie die erweiterten Einstellungen wieder, und klicken Sie dann auf STARTEN, um die zuvor selektierten Dateien und Ordner auf das Speichermedium bzw. in den Zielpfad zu sichern.

Sobald die Datensicherung beendet ist, können Sie den Datensicherungsauftrag im Menü AUFTRAG als Auswahl speichern, womit Sie die Einstellungen künftig nicht noch einmal setzen müssen, da alle Einstellungen in einer Konfigurationsdatei hinterlegt und mit dem nächsten Öffnen der Auswahl automatisch übernommen werden. Eine solche Auswahldatei benötigen Sie auch, wenn Sie die Datensicherung von der Eingabeaufforderung (siehe Die Lösung per Eingabeaufforderung auf Seite 361) aus durchführen möchten, da Sie von der Eingabeaufforderung aus keine Datei- und Ordner-Selektion vornehmen können und die Auswahl daher vorab per GUI konfigurieren müssen.

Tipp

Einen Datensicherungsauftrag können Sie mit Hilfe des NTBACKUP-eigenen Zeitplaners auch automatisieren und damit an ausgewählten Tagen und zu festgelegter Uhrzeit automatisch ausführen lassen. Hierzu rufen Sie zunächst NTBACKUP auf, wechseln in den erweiterten Modus und holen die Registerkarte AUFTRÄGE PLANEN in den Vordergrund. Dann klicken Sie auf AUFTRAG HINZUFÜGEN, woraufhin der SICHERUNGS-ASSISTENT gestartet wird. Legen Sie die Konfigurationsparameter für die Datensicherung fest, und klicken Sie auf die Schaltfläche ZEITPLAN FESTLEGEN, um das Ausführungsdatum und auch die Uhrzeit festzulegen. Legen Sie dann den Benutzernamen und auch das Kennwort des Benutzers fest, unter dessen Verwendung die Datensicherung ausgeführt werden soll, und beenden Sie den Assistenten. Die Datensicherung wird – sofern geeignete Backup-Medien zur Verfügung stehen – dann automatisch ausgeführt. Weitere Informationen zur Automatisierung von Verwaltungsaufgaben finden Sie auch in der Kategorie *Automatisierung*.

Sollte nun der Fall der Fälle eintreten und ein Datenverlust entstehen, so können Sie die zuvor gesicherten Dateien und Ordner mit Hilfe von NTBACKUP auch wieder restaurieren. Das Wiederherstellen einer Datensicherung setzt dabei nicht voraus, dass die Daten von dem Computersystem aus, auf dem die Sicherung erstellt wurde, auch wieder restauriert werden. Zur Wiederherstellung der Daten benötigen Sie lediglich das Tool NTBACKUP, das Medium, auf dem die Datensicherung abgelegt ist, und auch den Sicherungskatalog, der die Sicherung enthält.

Zur Restaurierung einer Datensicherung starten Sie NTBACKUP aus dem Startmenü heraus, wechseln in den erweiterten Modus und holen dann die Registerkarte MEDIEN WIEDERHERSTELLEN UND VERWALTEN nach vorne. Wählen Sie in der linken Fensterhälfte die Datensicherung aus, deren Inhalt Sie restaurieren möchten, und selektieren Sie daraufhin in der rechten Fensterhälfte die wiederherzustellenden Dateien und Ordner.

Wählen Sie im Feld DATEIEN WIEDERHERSTELLEN IN durch Auswahl der entsprechenden Option aus, ob die Daten in ihrem ursprünglichen Bereich wiederhergestellt werden sollen, womit die Originaldaten, die sich in dem Ordner befinden, überschrieben werden, oder ob die gesicherten Daten in einen alternativen Bereich oder in einen einzelnen Ordner restauriert werden sollen, und legen Sie gegebenenfalls den Zielordner fest. Klicken Sie anschließend auf WIEDERHERSTELLUNG STARTEN, um mit der Restaurierung der Daten zu beginnen.

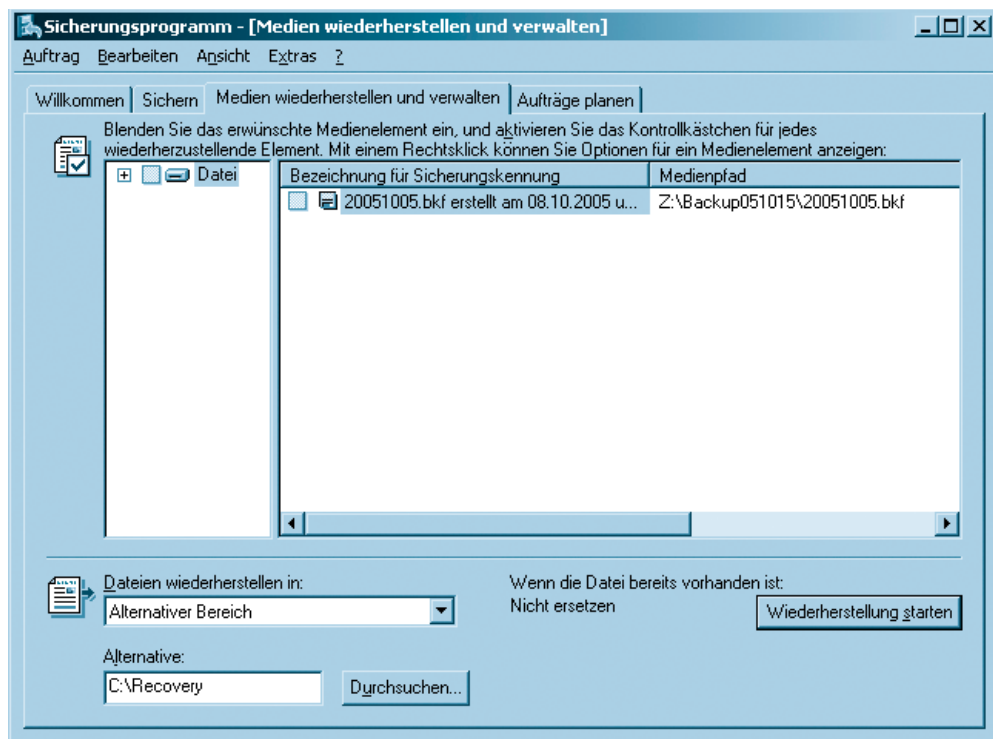


Abbildung 109: Eine Datensicherung in einen alternativen Bereich restaurieren

Die Lösung per Eingabeaufforderung

Das Backup-Utility namens NTBACKUP lässt sich mit Hilfe des zusätzlichen Parameters `Backup` auch von der Eingabeaufforderung aus verwenden, wobei Sie in der Eingabeaufforderung aber nur die Funktion *Sicherung* verwenden können. Denn die Funktion *Wiederherstellen* steht leider nur unter grafischer Oberfläche zur Verfügung und kann daher nicht zur Restaurierung von Daten von der Eingabeaufforderung aus genutzt werden.

Wenn Sie NTBACKUP in der Eingabeaufforderung einsetzen möchten, so müssen Sie zunächst unter grafischer Oberfläche einen Datensicherungsauftrag anlegen, die zu sichernden Daten auswählen und den Auftrag als Auswahl speichern, da Sie in der Eingabeaufforderung keine Datei- und Ordner-Auswahl treffen können. Oder aber Sie erzeugen mit einem der Windows-eigenen Editoren eine Datei im Unicode-Format und nehmen jeweils pro Zeile ein Laufwerk oder den Pfad eines Ordners oder einer Datei auf und speichern die Datei als Datensicherungsauftrag mit der Endung **.bks* ab. Erzeugen Sie also zunächst einen entsprechenden Auftrag, und speichern Sie die Konfiguration als **.bks*-Datei.

Anschließend können Sie den zuvor angelegten Datensicherungsauftrag von der Eingabeaufforderung aus (z.B. aus einer Batch-Datei heraus) starten, indem Sie den Befehl `NTBackup Backup` eingeben, den Namen der Auswahldatei als Parameter übergeben (dieser muss ein *@*-Zeichen vorangestellt sein!) und mit dem Parameter `/F` das Speichermedium bzw. den Zielpfad festlegen.

Mit dem Parameter `/J` können Sie zusätzlich einen Auftragsnamen (z.B. *FullBackup*) bestimmen, mit `/D` eine Beschreibung zur Sicherung übergeben, mit `/V:Yes` die Überprüfung der Datensicherung veranlassen, mit `/L:S` den Protokolltyp (`F` = vollständiges Protokoll, `S` = Zusammenfassung, `N` = Kein Protokoll) festlegen und mit dem Parameter `/M` die Sicherungsart (`Normal`, `Copy`, `Differencial`, `Incremental`, `Daily`) festlegen.

Die nachfolgende `NTBackup`-Befehlszeile führt beispielsweise den Sicherungsauftrag *FullBackup.bks* aus, legt den Auftragsnamen *FullBackup* fest, sichert die Daten in die Datei *Z:\Backup051015.bkf*, verwendet dabei die Beschreibung *Fullbackup_05.10.2005_22:00*, führt eine Überprüfung der Sicherung durch, gibt am Ende der Sicherung eine kurze Zusammenfassung aus und verwendet dabei die Sicherungsart *Normal*.

```
@Echo off
NTBackup Backup @FullBackup.bks /J FullBackup _
/F Z:\Backup051015.bkf /D FullBackup_05.10.2005_22:00 _
/V:Yes /L:S /M Normal
```

Listing 272: 2_02_034a.bat

Wenn Sie die Sicherung nicht in einem bestimmten Pfad erstellen möchten, sondern beispielsweise auf einem Band, so legen Sie mit dem Parameter /T den Namen des Bandes fest und bestimmen mit /A, dass die Sicherung dem Band angehängt werden soll und dass bereits vorhandene Sicherungen nicht ersetzt werden dürfen.

```
@Echo off
NTBackup Backup @FullBackup.bks /J FullBackup _
/T Backup /D FullBackup_05.10.2005_22:00 _
/A /V:Yes /L:S /M Normal
```

Listing 273: 2_02_034b.bat

Die Lösung mit VBScript

Das Backup-Utility namens NTBACKUP stellt leider keine entsprechende API bereit, womit Sie die Sicherung und auch die Wiederherstellung von Datei- und Ordner-Objekten per VBScript steuern können.

Ein bereits definierter Datensicherungsauftrag lässt sich aber dennoch aus einem VBScript heraus starten, da Sie mit der Run-Methode des Shell-Objektes jeden Eingabeaufforderungsbefehl – und damit auch *NTBackup.exe* – aufrufen und ausführen können, womit Sie hinsichtlich des Backups zumindest die gleichen Möglichkeiten haben wie auch in der Eingabeaufforderung (siehe vorheriger Abschnitt).

Wenn Sie also unter grafischer Oberfläche bereits einen Datensicherungsauftrag erstellt haben, so können Sie diesen Task nun mit Hilfe der Run-Methode ausführen. Hierzu erzeugen Sie mit der CreateObject-Methode zunächst eine neue Instanz des Shell-Objektes, binden diese an eine Objektvariable und wenden dann die Run-Methode unter Angabe der entsprechenden NTBackup-Befehlszeile gegen die Objektvariable an.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c NTBackup Backup @FullBackup.bks /J FullBackup _
/F Z:\192.168.0.250\Backup$\Backup051015.bkf _
/D FullBackup_05.10.2005_22:00 /V:Yes /L:S /M Normal"

Set oShell = Nothing
```

Listing 274: 2_02_034.vbs

Datenträgerverwaltung

Der Windows Server 2003 verfügt bereits von Haus aus über entsprechende Bordmittel, mit denen Sie einzelne Datenträger verwalten und natürlich auch warten können. So lassen sich unter grafischer Oberfläche beispielsweise problemlos neue Festplatten einbinden und initialisieren, primäre und erweiterte Partitionen anlegen, logische Laufwerke einrichten, dynamische Volumes erstellen und sogar softwarebasierende RAID-Systeme konfigurieren, ohne dass hierfür auf externe Tools zurückgegriffen werden muss. Auch für das Defragmentieren und Einrichten von Datenträgerkontingenten hält das Betriebssystem entsprechende Möglichkeiten bereit.

Die datenträgerbezogenen Aufgaben lassen sich aber nicht nur unter grafischer Oberfläche ausführen. Denn im Befehlsumfang von Windows Server 2003 ist eine ganze Reihe nützlicher Tools enthalten, womit Sie die Konvertierung der Datenträger, die Partitionierung, die Einrichtung dynamischer Volumes und viele weitere Verwaltungs- und Wartungsaufgaben auch von der Eingabeaufforderung aus erledigen und sogar automatisieren können. Die nachfolgenden Rezepte zeigen daher nicht nur, wie Sie datenträgerbezogene Aufgaben unter grafischer Oberfläche lösen, sondern wie Sie diese Aufgaben auch von der Eingabeaufforderung aus oder durch Einsatz eines Skriptes ausführen und damit automatisieren können.

61 Laufwerkinformationen auslesen

Der Windows Server 2003 stellt unter grafischer Oberfläche mehrere Möglichkeiten bereit, worüber Sie sich die physikalischen Laufwerke, die auf dem Server eingerichtet sind, anzeigen lassen können. Wenn Sie z.B. eine einfache Aufstellung der installierten Disketten-, Festplatten- und CD-/DVD-Laufwerke unter Angabe ihres Laufwerkbuchstabens benötigen, so können Sie sich die Laufwerke im WINDOWS EXPLORER unterhalb von ARBEITSPLATZ anzeigen lassen, wobei Sie den jeweiligen Laufwerkstyp (Festplatte, CD/DVD ...) an dem vorangestellten Symbol erkennen können.

Benötigen Sie dagegen erweiterte Hardwareinformationen wie z.B. Hersteller, Modell, Treiber etc., so rufen Sie das Systemsteuerungsobjekt SYSTEM auf, wechseln auf die Registerkarte HARDWARE und klicken auf GERÄTE-MANAGER, um in die Übersicht der installierten Geräte zu wechseln. Je nach Hardwareausstattung des Computersystems finden Sie dort die Abschnitte DISKETTENLAUFWERKE, DVD/CD-LAUFWERKE und LAUFWERKE und unterhalb davon jeweils die Geräteobjekte vor. In den Eigenschaften der Geräte-Objekte finden Sie die Informationen zum Hersteller, zum Modell, zu den verwendeten Treibern wie auch zur Geräteerkennung.

Wenn Sie dagegen eine Übersicht der installierten Laufwerke unter Angabe erweiterter Laufwerkinformationen (z.B. Laufwerksbuchstabe, Laufwerkstyp, Speicherkapazität, Volume-Name und -Seriennummer ...) benötigen, so nehmen Sie die SYSTEMINFORMATIONEN zu Hilfe: Öffnen Sie hierzu das STARTMENÜ, erweitern Sie ALLE PROGRAMME, dann ZUBEHÖR, anschließend SYSTEMPROGRAMME, klicken Sie auf SYSTEMINFORMATIONEN, oder

wählen Sie im STARTMENÜ den Eintrag AUSFÜHREN aus, und geben Sie `Minfo32` ein, womit Sie die SYSTEMINFORMATIONEN auch direkt aufrufen können.

Im Dialogfenster SYSTEMINFORMATIONEN erweitern Sie in der linken Fensterhälfte den Abschnitt KOMPONENTEN, darunter SPEICHERGERÄTE, und markieren dann den Eintrag LAUFWERKE, woraufhin die installierten Laufwerke unter Angabe laufwerksspezifischer Informationen recht übersichtlich angezeigt und bei Bedarf auch exportiert werden können.

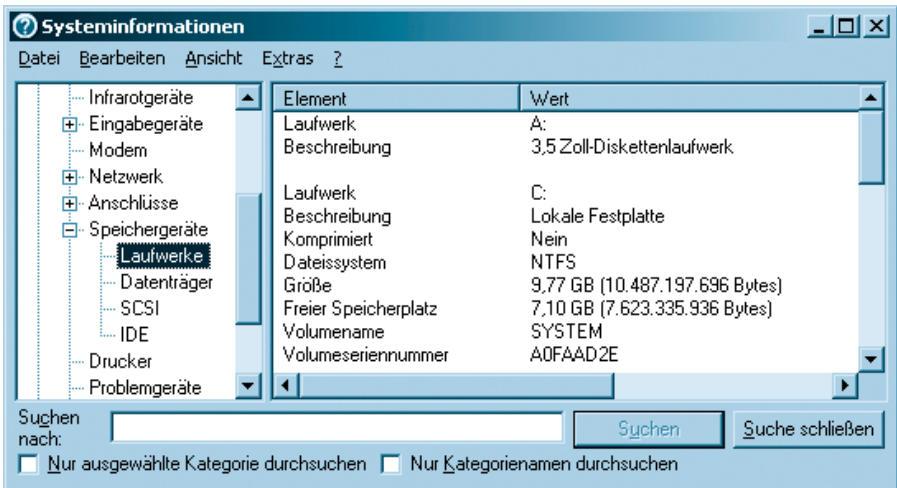


Abbildung 110: Die Übersicht der Laufwerke in den Systeminformationen

Die Lösung per Eingabeaufforderung

Wenn Sie die Laufwerkinformationen von der Eingabeaufforderung aus abfragen möchten (beispielsweise zu Inventarisierungszwecken aus einer Batch-Datei heraus), so können Sie hierfür den WMIC-Befehl mit dem Alias `LogicalDisk` verwenden und damit eine Abfrage gegen die WMI-Klasse `Win32_LogicalDisk` durchführen, die eine ganze Reihe wertvoller Informationen zu den einzelnen Laufwerken besitzt.

Die nachfolgende Tabelle enthält einen Teil der Eigenschaften, die von `Win32_LogicalDisk` bereitgestellt und mit dem WMIC-Befehl auch von der Eingabeaufforderung aus ausgelesen werden können. Eine vollständige Übersicht aller Eigenschaften finden Sie im *Teil III* dieses Buches.

Eigenschaft	Beschreibung
Caption	Bezeichnung des Laufwerks
Description	Beschreibung des Laufwerks
FileSystem	Verwendetes Dateisystem

Tabelle 40: Die wichtigsten Eigenschaften der Win32_LogicalDisk-Klasse

Eigenschaft	Beschreibung
FreeSpace	Freie Speicherkapazität
Size	Gesamtkapazität
VolumeName	Datenträgerbezeichnung
VolumeSerialNumber	Datenträger-Seriennummer

Tabelle 40: Die wichtigsten Eigenschaften der Win32_LogicalDisk-Klasse (Forts.)

Um beispielsweise eine Aufstellung aller installierten Laufwerke unter Angabe der Bezeichnung, der Beschreibung, des verwendeten Dateisystems, der freien und gesamten Speicherkapazität zu generieren und am Bildschirm auszugeben, wenden Sie den WMIC-Befehl mit dem Alias `LogicalDisk` an und übergeben mit `Get` die Eigenschaften, deren Werte Sie auslesen möchten (ohne Angabe von `Get` werden alle Eigenschaften ausgegeben).

```
@Echo off
WMIC LogicalDisk Get Caption, Description, FileSystem, FreeSpace, Size
```

Listing 275: 2_03_001.bat

Die Lösung mit VBScript

Wenn Sie die Laufwerkinformationen dagegen per VBScript abfragen möchten, so stellen Sie ebenfalls eine Verbindung zur WMI-Klasse `Win32_LogicalDisk` her und fragen lediglich die Werte der entsprechenden Eigenschaften (siehe *Teil III*) ab.

Um beispielsweise alle Laufwerke unter Angabe der Bezeichnung, der Beschreibung, des Dateisystems, der Kapazitäten und der Volume-Informationen (Name und Seriennummer) am Bildschirm auszugeben, stellen Sie mit der `GetObject`-Methode zunächst eine Verbindung zum WMI-Namespace `root\cimv2` des betreffenden Computersystems her und setzen zur Personifizierung den `Impersonation-Level` auf `Impersonate`. Anschließend führen Sie eine `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_LogicalDisk` durch, woraufhin die Laufwerke in Form einer Kollektion zurückgeliefert werden.

Innerhalb einer `For Each`-Schleife können Sie nun für jedes Element der Kollektion, sprich: für jedes Laufwerk, den Wert der gewünschten Eigenschaft (z.B. `Caption`, `Description`, `FileSystem`, `FreeSpace`, `Size` ...) abfragen und mit der `WScript.Echo`-Anweisung am Bildschirm ausgeben.

```
Option Explicit

Dim oWMI
Dim colDisk
```

Listing 276: 2_03_001.vbs

```

Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colDisk = oWMI.ExecQuery ("SELECT * FROM Win32_LogicalDisk")

For Each objItem in colDisk
    WScript.Echo "Caption: " & objItem.Caption
    WScript.Echo "Description: " & objItem.Description
    WScript.Echo "FileSystem: " & objItem.FileSystem
    WScript.Echo "FreeSpace: " & objItem.FreeSpace
    WScript.Echo "Size: " & objItem.Size
    WScript.Echo "Volume-Name: " & objItem.VolumeName
    WScript.Echo "Volume-Serialnumber: " & objItem.VolumeSerialNumber & VBCRLF
Next

Set oWMI = Nothing

```

Listing 276: 2_03_001.vbs (Forts.)

62 Aktuelle Startkonfiguration ermitteln

Mit der Installation von Windows Server 2003 wird im Root der Systempartition eine Konfigurationsdatei mit der Bezeichnung *Boot.ini* angelegt, worin die aktuelle Startkonfiguration hinterlegt ist. Diese Datei enthält genaue Angaben darüber, von welchem Datenträger und von welcher Partition gestartet werden soll, in welchem Ordner die Windows-Dateien zu finden sind und welche Startoptionen verwendet werden sollen.

Um nun die aktuelle Startkonfiguration zu ermitteln, öffnen Sie die Datei *Boot.ini* und prüfen deren Einträge. Da diese Datei standardmäßig die Attribute *Schreibgeschützt* und *Versteckt* besitzt, müssen Sie zunächst – falls noch nicht geschehen – die Anzeige der Systemdateien aktivieren, da diese Einstellung standardmäßig deaktiviert ist. Hierzu rufen Sie den WINDOWS EXPLORER auf und wählen im Menü EXTRAS den Eintrag ORDNER-OPTIONEN aus. Auf der Registerkarte ANSICHT aktivieren Sie die Option GESCHÜTZTE SYSTEMDATEIEN AUSBLENDEN (EMPFOHLEN) und übernehmen die Einstellung mit einem Klick auf OK.

Anschließend markieren Sie im WINDOWS EXPLORER die Systempartition, woraufhin in der rechten Fensterhälfte die *Boot.ini*-Datei angezeigt wird. Öffnen Sie nun die Datei durch einen Doppelklick, womit Sie eine ähnliche Konfiguration vorfinden, wie in der folgenden Abbildung zu sehen ist.

Die Sektion BOOT LOADER enthält für gewöhnlich den Eintrag *Timeout=xx*, womit die Anzeigedauer des Boot-Menüs zur Auswahl der Windows-Installation definiert wird. Des Weiteren enthält die BOOT LOADER-Sektion die Default-Konfiguration, die automatisch gestartet wird, wenn keine andere Windows-Installation ausgewählt wird.

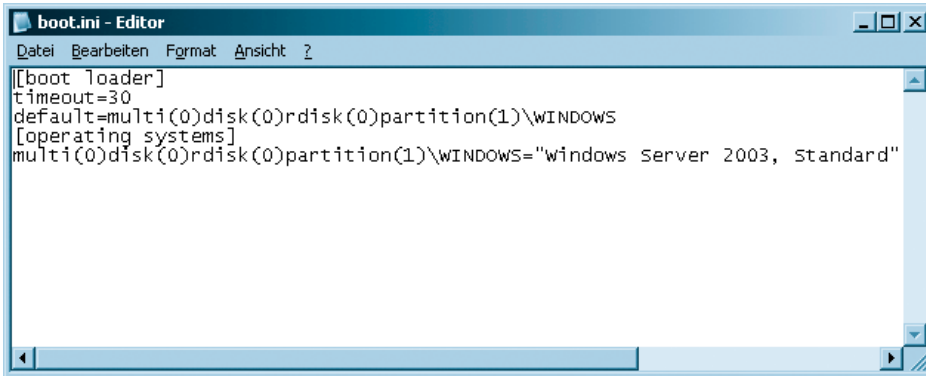


Abbildung 111: Die Startkonfiguration von Windows Server 2003

Die Sektion OPERATING SYSTEMS enthält eine Aufstellung aller Windows-Installationen (und anderer Betriebssysteme, die über den Bootlader von Windows gestartet werden können), die auf dem betreffenden Computersystem eingerichtet sind. Standardmäßig wird in dieser Sektion nur eine Konfiguration angezeigt, bei Parallelinstallationen können aber auch mehrere Einträge vorhanden sein. Enthält die Sektion nur einen Eintrag, wird der oben angegebene Wert für Timeout nicht beachtet. Das Element *Rdisk* gibt dabei die Nummer des Datenträgers und das Element *Partition* die fortlaufende Nummer der Partition an, auf der die jeweilige Installation zu finden ist. In welchem Ordner sich die jeweilige Windows-Installation befindet, ist ebenfalls angegeben, diese Informationen finden Sie direkt hinter der Partitionsnummer (*\Windows*). Detaillierte Erklärungen der Nummerierung können unter <http://support.microsoft.com/default.aspx?scid=kb;en-us;102873> eingesehen werden.

Jede Windows-Installation kann mit bestimmten Optionen gestartet werden, womit sich beispielsweise eine Protokollierung des Boot-Vorgangs, das Starten des Debug-Modus oder der Aufruf des abgesicherten Modus aktivieren lässt. Wenn Startoptionen verwendet werden, so sind diese Optionen am Ende der Konfigurationszeile in Form von Parametern zu finden (eine Beschreibung der einzelnen Startoptionen finden Sie in Tabelle 43: Die Betriebssystem-Startoptionen ab Seite 371).

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung lässt sich die aktuelle Startkonfiguration mit dem `Bootcfg`-Befehl ermitteln, der ohne Angabe weiterer Parameter alle Einträge der Datei *Boot.ini* am Bildschirm ausgibt.

```
@Echo off
Bootcfg
```

Listing 277: 2_03_002a.bat

Eine weitere Möglichkeit, die aktuelle Startkonfiguration von der Eingabeaufforderung aus zu ermitteln, bietet der WMIC-Befehl in Verbindung mit dem Alias `BootConfig`. Der `WMIC BootConfig`-Befehl erlaubt eine Abfrage der WMI-Klasse `Win32_BootConfiguration`, worin Informationen zu den einzelnen Windows-Installationen enthalten sind.

Die nachfolgende WMIC-Befehlszeile gibt den Windows-Installationsordner, die Datenträger- und Partitionsinformation der Windows-Installation wie auch den Laufwerksbuchstaben und den Temp-Pfad aus und schreibt das Ergebnis der Abfrage in die Datei `BootConfig.txt`.

```
@Echo off
WMIC BootConfig > BootConfig.txt
```

Listing 278: `2_03_002b.bat`

Die Lösung mit VBScript

Mit Hilfe von WMI ist das Auslesen der aktuellen Startkonfiguration auch per VBScript kein Problem, da Sie nur eine Verbindung zur WMI-Klasse `Win32_BootConfiguration` herstellen und die Werte der entsprechenden Eigenschaften auslesen müssen. Welche Eigenschaften bereitgestellt werden, zeigt die nachfolgende Tabelle.

Eigenschaft	Beschreibung
BootDirectory	Pfad zu den Windows-Systemdateien
Caption	Bezeichnung der Windows-Installation
ConfigurationPath	Pfad zu den Windows-Systemdateien (entspricht BootDirectory)
Description	Beschreibung der Windows-Installation
LastDrive	Letzter verwendeter Laufwerksbuchstabe
Name	Name der Windows-Installation
ScratchDirectory	Pfad des Temp-Verzeichnisses, das während des Boot-Vorgangs verwendet wird
SettingID	Einstellungs-ID
TempDirectory	Pfad des Temp-Verzeichnisses

Tabelle 41: Die Eigenschaften der `Win32_BootConfiguration`-Klasse

Das nachfolgende Code-Beispiel führt eine solche `Win32_BootConfiguration`-Abfrage durch und gibt die Werte der zuvor genannten Eigenschaften am Bildschirm aus. Dazu wird mit der `GetObject`-Methode eine Verbindung zum WMI-Namespace `root\cimv2` des lokalen Computersystems hergestellt und der Personifizierung wegen der `Impersonation-Level` auf `Impersonate` gesetzt. Dann folgt die `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_BootConfiguration`, woraufhin die Konfigurationen als Kollektion zurückgeliefert werden. Mit einer `For Each`-Schleife werden dann die Werte der Eigenschaften ausgelesen und per `WScript.Echo`-Anweisung am Bildschirm ausgegeben.

```
Option Explicit

Dim oWMI
Dim colConfig
Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colConfig = oWMI.ExecQuery ("SELECT * FROM Win32_BootConfiguration")

For Each objItem in colConfig
    WScript.Echo "Boot-Directory: " & objItem.BootDirectory
    WScript.Echo "Caption: " & objItem.Caption
    WScript.Echo "Configuration-Path: " & objItem.ConfigurationPath
    WScript.Echo "Description: " & objItem.Description
    WScript.Echo "LastDrive: " & objItem.LastDrive
    WScript.Echo "Name: " & objItem.Name
    WScript.Echo "Scratch-Directory: " & objItem.ScratchDirectory
    WScript.Echo "Setting-ID: " & objItem.SettingID
    WScript.Echo "Temp-Directory: " & objItem.TempDirectory
Next

Set oWMI = Nothing
```

Listing 279: 2_03_002.vbs

63 Die Startkonfiguration ändern

Die Startkonfiguration lässt sich natürlich jederzeit ändern, wobei Sie dies aber mit Bedacht tun sollten, da eine fehlerhafte Startkonfiguration dazu führen kann, dass das Betriebssystem nicht mehr startet. Sicherheitshalber sollten Sie nie die einzige vorhandene Einstellung ändern, sondern immer nur eine Kopie der aktuellen Einstellung, um im Notfall auf die ursprüngliche Konfiguration zurückgreifen zu können.

Wenn Sie beispielsweise auf einer zweiten Partition oder auf einer zusätzlichen Festplatte eine weitere Windows Server 2003-Installation vorgenommen haben und das Computersystem künftig mit der neuen Windows-Installation starten möchten, so können Sie die neue Windows-Installation in der Startkonfiguration als neuen Standard definieren. Damit wird die neue Windows-Installation ab sofort per Default gestartet und muss nicht mehr im Boot-Menü von Hand ausgewählt werden.

Um das Standardbetriebssystem zu ändern, rufen Sie in der SYSTEMSTEUERUNG das Objekt SYSTEM auf und wechseln auf die Registerkarte ERWEITERT. In der Sektion STARTEN UND WIEDERHERSTELLEN (im unteren Drittel des Fensters) klicken Sie auf EINSTELLUNGEN, woraufhin Sie in das Dialogfenster STARTEN UND WIEDERHERSTELLEN gelangen. Im Auswahlfeld STANDARDBETRIEBSSYSTEM können Sie nun die zweite Windows-Installation auswählen und damit als neuen Standard festlegen.

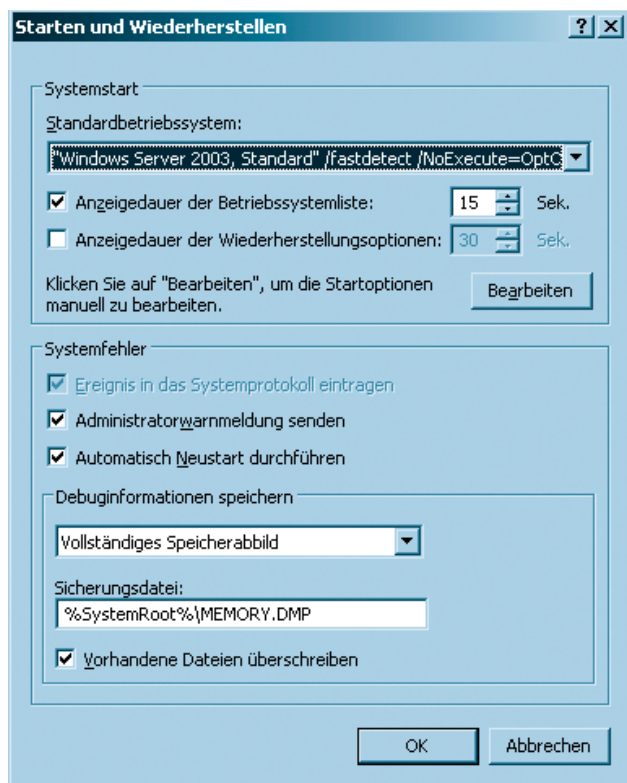


Abbildung 112: Die Startkonfiguration bearbeiten

Falls die neue Windows-Installation wider Erwarten nicht in der Betriebssystemliste erscheinen sollte, so können Sie die Startkonfiguration auch von Hand ändern. Klicken Sie hierzu auf BEARBEITEN, woraufhin die Datei *Boot.ini* geöffnet wird. Unterhalb der Sektion OPERATING SYSTEMS sehen Sie nun alle vom Betriebssystem erkannten Windows-Installationen unter Angabe des Datenträgers, der Partition und auch des Ordners. Der Verweis zur jeweiligen Windows-Installation wird nach der ARC-Konvention (Advanced RISC Computing) angegeben. Die einzelnen Elemente der Pfadangabe lassen sich folgendermaßen interpretieren:

Element	Beschreibung
Multi(x) oder SCSI(x)	Ist für den Boot-Vorgang ein BIOS-Zugriff (INT13-Aufruf) möglich, so wird <i>Multi</i> verwendet, beim Laden von Starttreibern und ohne BIOS-Zugriff dagegen wird <i>SCSI</i> verwendet. Die Nummer gibt dabei die Ordnungszahl für den IDE- oder SCSI-Adapter an und ist in den meisten Fällen 0.
Disk(x)	Ist das erste Element <i>Multi(x)</i> , so wird dieses Element nicht benötigt und enthält 0. Ist das erste Element dagegen <i>SCSI(x)</i> , so erhält <i>Disk(x)</i> die SCSI-Kennung des Datenträgers.
Rdisk(x)	Ist das erste Element <i>Multi(x)</i> , so erhält <i>Rdisk</i> die Nummer des Datenträgers (0 = erster Datenträger, 1 = zweiter Datenträger ...). Ist das erste Element dagegen <i>SCSI(x)</i> , so wird <i>Rdisk</i> nicht benötigt und enthält den Wert 0.
Partition(x)	Nummer der Partition (1 = erste Partition, 2 = zweite Partition ...)
Ordner	Name des Ordners, worin sich die Windows-Systemdateien befinden (z.B. \Windows)
Bezeichnung	Bezeichnung der Installation, wie sie beispielsweise im Boot-Menü angezeigt wird
Startoptionen	Die Startoptionen, womit das Betriebssystem gestartet wird (siehe Tabelle 43: Die Betriebssystem-Startoptionen auf Seite 371).

Tabelle 42: Die Elemente des ARC-Pfades für x86-basierende Computersysteme

Sollten Sie – trotz einer Parallelinstallation von Windows Server 2003 – unterhalb von OPERATING SYSTEMS nur einen Eintrag vorfinden, so kopieren Sie diesen Eintrag, fügen ihn in der Folgezeile ein und ändern – je nach Konfiguration – die Datenträger- und Partitionsnummer wie auch den Windows-Ordner.

Jede Windows-Installation kann dabei mit unterschiedlichen Startoptionen aufgerufen werden, indem der Pfadangabe die entsprechenden Parameter (siehe nachfolgende Tabelle) angefügt werden. So lässt sich beispielsweise mit Hilfe der Start-Option */BaseVideo* festlegen, dass die betreffende Windows-Installation grundsätzlich nur im Standard-VGA-Modus mit einer Auflösung von 640 * 480 Pixel bei 16 Farben gestartet wird. Folgende Optionen sind möglich:

Parameter	Beschreibung
/3GB	Ermöglicht die Nutzung von 3 GB (statt standardmäßigen 2 GB) virtuellen Speichers für Programme und 1 GB für den Kernel. Diese Option wird von Anwendungen benötigt, die mehr als 2 GB virtuellen Speicher zur Programmausführung erfordern.
/BaseVideo	Startet die Windows-Installation mit der Standard-VGA-Auflösung (640 * 480 bei 16 Farben) und dem Standard-VGA-Treiber von Microsoft.

Tabelle 43: Die Betriebssystem-Startoptionen

Parameter	Beschreibung
/BaudRate=xxx	Legt die Übertragungsrate für den Debug-Kommunikationsport fest und erwartet die Angabe der zu verwendenden Übertragungsrate. Standardmäßig wird eine Übertragungsrate von 9.600 kbps verwendet.
/BootLog	Aktiviert die Protokollierung des Boot-Vorgangs (%Systemroot%\Ntbtlog.txt).
/BurnMemory=xxx	Beschränkt den zur Verfügung stehenden Arbeitsspeicher auf den angegebenen Arbeitsspeicher (Angabe in MByte).
/Channel	Ermöglicht die Übertragung von Debug-Informationen über den angegebenen Firewire-Channel (0–62). Diese Option ist nur in Verbindung mit /Debug und /DebugPort 1394 zu verwenden.
/CrashDebug	Aktiviert den Kernel-Debugger während des Systemstarts und ermöglicht die automatische Nutzung des Debug-Ports bei einem Absturz.
/Debug	Aktiviert den Kernel-Debugger während des Systemstarts.
/DebugPort=xxx	Legt den zu verwendenden COM-Port (z.B. COM1, COM2 ...) für das Debuggen von einem Remote-System fest und aktiviert automatisch die Option /Debug.
/NoSerialMice /NoSerialMice:COMx /NoSerialMice:COMx,y	Deaktiviert die automatische Mauserkennung an allen oder an dem angegebenen COM-Port während des Bootvorgangs.
/HAL=xxx	Legt die während des Boot-Vorgangs zu verwendende HAL-Datei fest (z.B. HAL.dll). Die Datei muss im Windows-Ordner \System32 liegen.
/Kernel=xxx	Legt die während des Boot-Vorgangs zu verwendende Kernel-Datei fest. Die Datei muss im Windows-Ordner \System32 liegen.
/MaxMem=xxx	Bestimmt den für Windows zur Verfügung stehenden Arbeitsspeicher.
/NoDebug	Deaktiviert das Debugging.
/NoGUIBoot	Deaktiviert das Windows-Logo während des Boot-Vorgangs.
/NumProc=xxx	Legt die Anzahl der zu verwendenden Prozessoren fest.
/PAE	Ermöglicht den normalen Windows-Start bei PAE-Unterstützung.
/PCILock	Verhindert die dynamische Ressourcen-Zuweisung durch das Betriebssystem.
/SafeBoot:Parameter	Startet Windows im abgesicherten Modus mit der angegebenen Option (Minimal, Network oder Minimal (Alternative)). Diesen Schalter sollten Sie jedoch nie selbst setzen, da er vom System gesetzt wird, wenn der Startvorgang mit der Taste F8 unterbrochen wird.
/SOS	Veranlasst die Anzeige der geladenen Treiber während des Boot-Vorgangs.

Tabelle 43: Die Betriebssystem-Startoptionen (Forts.)

Wenn Sie die Einstellungen überprüft bzw. Änderungen vorgenommen haben, so schließen Sie die Datei *Boot.ini* wieder. Sie gelangen daraufhin automatisch in das Dialogfenster STARTEN UND WIEDERHERSTELLEN zurück.

Für den Fall, dass während des Systemstarts ein Boot-Menü eingeblendet wird, worüber Sie die verfügbaren Windows-Installationen auswählen können, so lässt sich das Boot-Menü im Dialogfenster STARTEN UND WIEDERHERSTELLEN deaktivieren, in dem Sie die Option ANZEIGEDAUER DER BETRIEBSSYSTEMLISTE deaktivieren. Sie können natürlich auch die Anzeigedauer des Bootmenüs verkürzen oder verlängern, indem Sie einen neuen Wert (0–999) eingeben.

Wenn Sie alle Änderungen durchgeführt haben, so klicken Sie auf OK, um die neuen Einstellungen zu übernehmen.

Die Lösung per Eingabeaufforderung

Die Startkonfiguration lässt sich auch von der Eingabeaufforderung aus ändern. Hierfür stellt der Windows Server 2003 den `Bootcfg`-Befehl bereit, womit Sie die Anzeigedauer des Boot-Menüs, das Standardbetriebssystem und auch die Startoptionen modifizieren können.

Um beispielsweise die Anzeigedauer des Bootmenüs von 30 auf 15 Sekunden zu reduzieren, wenden Sie den `Bootcfg`-Befehl mit dem Parameter `/Timeout` an und übergeben die Anzahl der Sekunden (0–999), für die das Boot-Menü ab dem nächsten Systemstart angezeigt werden soll.

`Bootcfg /Timeout 15`

Haben Sie auf einem Computersystem mehrere Windows-Installationen vorgenommen, so können Sie mit dem Parameter `/Default /ID x` das Standardbetriebssystem ändern, wobei Sie `x` durch die Starteintragskennung ersetzen. Die Starteintragskennung können Sie sich mit dem Befehl `Bootcfg` (ohne Angabe weiterer Parameter) anzeigen lassen, wobei die Werte aber nicht statisch sind, sondern mit der Änderung des Standardbetriebssystems neu vergeben werden (dies gilt es, gerade im Batch-Betrieb zu berücksichtigen!)

`Bootcfg /Default /ID 2`

Und last but not least, können Sie mit dem `Bootcfg`-Befehl auch Änderungen an den Startoptionen (siehe Tabelle 43: Die Betriebssystem-Startoptionen auf Seite 371) vornehmen. Wenn Sie z.B. das Betriebssystem ab sofort mit der Startoption `/Bootlog` starten und damit eine Protokollierung des Boot-Vorgangs veranlassen möchten, so wenden Sie den `Bootcfg`-Befehl mit dem Parameter `/Raw` an, übergeben die Startoption, legen mit `/A` fest, dass die Option angefügt werden soll, und bestimmen mit `/ID` die Starteintragskennung der entsprechenden Windows-Installation.

`Bootcfg /Raw "/BootLog" /A /ID 2`

Die nachfolgende Batch-Datei enthält die erforderlichen Befehlszeilen, womit die Anzeigedauer des Bootmenüs auf 15 Sekunden gesetzt, die zweite Windows-Installa-

tion als Standardbetriebssystem definiert und das neue Standardbetriebssystem mit der Startoption */Bootlog* gestartet wird.

```
@Echo off
Bootcfg /Timeout 15
Bootcfg /Default /ID 2
Bootcfg /Raw "/BootLog" /A /ID 2
```

Listing 280: 2_03_003.bat

Die Lösung mit VBScript

Wenn Sie die Startkonfiguration von Windows Server 2003 per VBScript bearbeiten möchten, so nutzen Sie die *Run*-Methode des *Shell*-Objektes und führen den *Bootcfg*-Befehl der Eingabeaufforderung aus, dem Sie die entsprechenden Parameter (siehe *Lösung per Eingabeaufforderung*) übergeben.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Bootcfg /Timeout 15"
oShell.Run "%comspec% /c Bootcfg /Default /ID 2"
oShell.Run "%comspec% /c Bootcfg /Raw /BootLog /A /ID 2"

Set oShell = Nothing
```

Listing 281: 2_03_003.vbs

64 Den Datenträgerstatus ermitteln

Wenn Sie einen neuen Datenträger (zum Beispiel eine neue Festplatte) eingebaut haben und dessen Status (*Online/Offline*) ermitteln möchten oder vielleicht den Datenträgertyp (*Basis/Dynamisch*) und das Partitionsformat (*MBR/GPT*) überprüfen wollen, so wechseln Sie in die DATENTRÄGERVERWALTUNG von Windows Server 2003 und erzeugen eine Datenträgerliste, womit all diese Informationen in einer übersichtlichen Ansicht dargestellt werden.

Rufen Sie aus dem Kontextmenü des ARBEITSPLATZ-Symbols VERWALTEN auf, um die COMPUTERVERWALTUNG zu öffnen. Anschließend erweitern Sie den Abschnitt DATENSPEICHER und markieren unterhalb davon DATENTRÄGERVERWALTUNG. Im Menü ANSICHT wählen Sie erst ANSICHT oben und dann DATENTRÄGERLISTE aus. Sie sehen nun in der rechten Fensterhälfte (oberhalb der grafischen Datenträgerübersicht) eine Aufstellung aller Datenträger unter Angabe des Typs, der Kapazität, des Status, des Gerätetyps und auch des Partitionsstils.

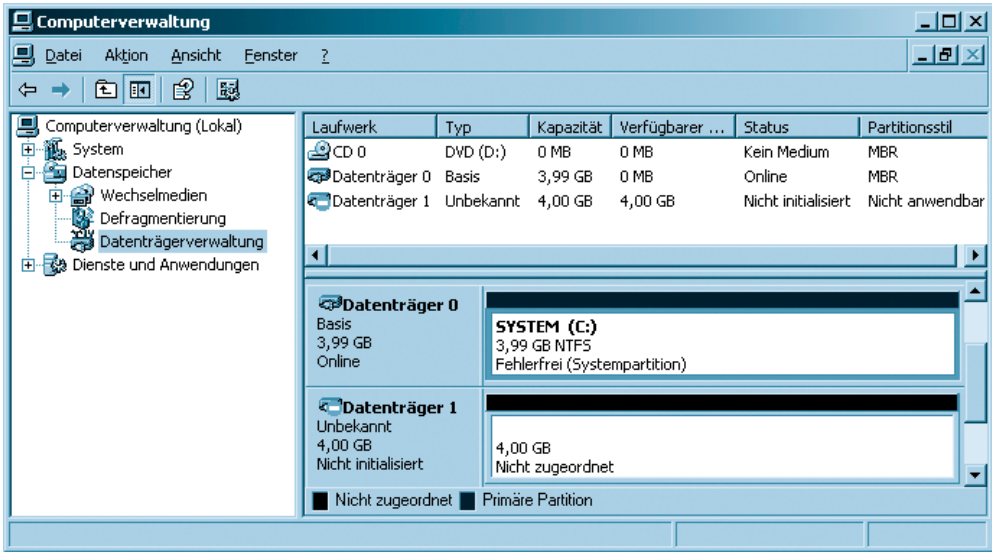


Abbildung 113: Die Übersicht der Datenträger und Partitionen

Die Lösung per Eingabeaufforderung

Wenn Sie den Datenträgerstatus von der Eingabeaufforderung aus ermitteln möchten, so verwenden Sie hierfür den `Diskpart`-Befehl und erzeugen mit `List Disk` eine Aufstellung der installierten Datenträger. Öffnen Sie hierzu das Eingabeaufforderungsfenster, und geben Sie zunächst `Diskpart` ein, um in die `Diskpart`-eigene Eingabeaufforderung zu gelangen. Anschließend geben Sie den Befehl `List Disk` ein, woraufhin die installierten Datenträger unter Angabe der Datenträgernummer, des Status, der Größe, der freien Kapazität und der Kennung für dynamische Datenträger und des GPT-Partitionsformats ausgegeben werden.

Diese Aufstellung können Sie natürlich auch erzeugen, ohne die `Diskpart`-eigene Eingabeaufforderung aufrufen zu müssen. Erstellen Sie einfach eine Textdatei, und nehmen Sie den Befehl `List Disk` darin auf. Speichern Sie die Datei ab (z.B. *Datenträgerliste.txt*), und führen Sie anschließend folgende Befehlszeile aus, um den `Diskpart`-Befehl `List Disk` nicht von der Tastatur, sondern per Eingabeumleitung aus der soeben erstellten Textdatei zu lesen:

```
@Echo off
Diskpart /s Datenträgerliste.txt
```

Listing 282: 2_03_004.bat

Die Lösung mit VBScript

Um den Status eines Datenträgers per VBScript zu ermitteln, können Sie zum einen die `IsReady`-Eigenschaft des `Drive`-Objektes (`FileSystemObject`) ermitteln, Sie können aber auch eine WMI-Abfrage gegen die Klasse `Win32_DiskDrive` durchführen und den Wert der Eigenschaft `Status` abfragen.

Das nachfolgende Listing nutzt beispielsweise eine WMI-Abfrage, um den Status eines Datenträgers (`Status`) zu ermitteln, und gibt dabei auch die Laufwerkskennung (`Cap- tion`), die Datenträgernummer (`Index`), die Anzahl der Partitionen (`Partitions`), eine Statusinformation (`StatusInfo`) und auch die Datenträgergröße (`Size`) aus. Dazu werden die Datenträger des Typs *Fixed hard disk media* mit einer `ExecQuery`-Abfrage gegen die `Win32_DiskDrive`-Klasse selektiert. Innerhalb einer `For Each`-Schleife werden dann die Werte der Eigenschaften ausgelesen und mit `WScript.Echo` am Bildschirm ausgegeben.

Option Explicit

```
Dim oWMI
Dim colDisk
Dim objItem
Dim strComputer
```

```
strComputer = "."
```

```
Set oWMI = GetObject("winmgmts:{impersonationLevel=_
    Impersonate}!\\" & strComputer & "\root\cimv2")
Set colDisk = oWMI.ExecQuery("SELECT * FROM Win32_DiskDrive _
    WHERE MediaType='Fixed hard disk media'")
```

```
For Each objItem in colDisk
    WScript.Echo "Disk: #" & objItem.Index
    WScript.Echo "Partitions: " & objItem.Partitions
    WScript.Echo "Status: " & objItem.Status
    WScript.Echo "Status-Info: " & objItem.StatusInfo
    WScript.Echo "Size: " & objItem.Size & "Bytes" & VBCRLF
Next
```

```
Set oWMI = Nothing
```

Listing 283: 2_03_004.vbs

65 Die Auslastung eines Datenträgers ermitteln

Wenn Sie die Auslastung eines Datenträgers ermitteln und damit herausfinden möchten, über wie viel freien Speicher der Datenträger noch verfügt, so können Sie sich diese Information im `WINDOWS EXPLORER` anzeigen lassen, indem Sie im Fenster `ARBEITSPLATZ` aus dem Kontextmenü des jeweiligen Datenträgers die `EIGENSCHAFTEN`

aufrufen. Denn auf der Registerkarte ALLGEMEIN wird sowohl die Gesamtkapazität des Datenträgers angezeigt wie auch dessen freier Speicher. Und mit Hilfe des Diagramms lässt sich die Auslastung des Datenträgers sehr gut erfassen. Vorsicht ist jedoch bei komprimierten Datenträgern geboten, da die Anzeige der freien Kapazität kleiner sein kann als der tatsächlich noch zur Verfügung stehende Platz.

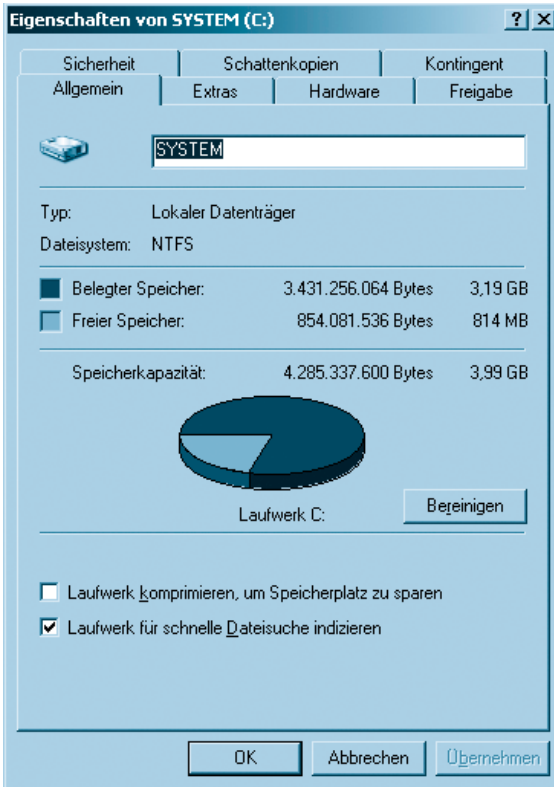


Abbildung 114: Die Auslastung des Datenträgers anzeigen lassen

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung stehen Ihnen unterschiedliche Möglichkeiten zur Verfügung, womit Sie die Werte der Datenträger-Auslastung ermitteln können. Wenn Sie beispielsweise nur den freien Speicher des aktuellen Laufwerks abfragen möchten, so können Sie den `Dir`-Befehl verwenden, der am Ende der Datei- und Ordner-Auflistung nicht nur die Anzahl der Dateien und Ordner ausgibt, sondern auch die freie Speicherkapazität (in Bytes) des aktuellen Laufwerks.

Wenn Sie stattdessen nur die Gesamtkapazität des Datenträgers und dessen freien Speicher ausgeben möchten, so können Sie den `WMIC`-Befehl mit dem Alias `LogicalDisk` anwenden und die Werte der Eigenschaften `Size` und `FreeSpace` der WMI-Klasse `Win32_LogicalDisk` abfragen. Um beispielsweise eine Auflistung aller Datenträger

unter Angabe des Laufwerkbuchstabens (Caption), des verwendeten Dateisystems (FileSystem), der Gesamtkapazität (Size) und auch des freien Speichers (FreeSpace) zu erzeugen, verwenden Sie folgende WMIC-Befehlszeile:

```
@Echo off
WMIC LogicalDisk Get Caption, FileSystem, Size, FreeSpace
```

Listing 284: 2_03_005.bat

Die Lösung mit VBScript

Mit VBScript lassen sich die Gesamtkapazität und der verfügbare Speicher eines Datenträgers zum einen mit der TotalSize- und FreeSpace-Eigenschaft des Drive-Objektes (FileSystemObject) ermitteln, zum anderen aber auch mit der Size- und FreeSpace-Eigenschaft der WMI-Klasse Win32_LogicalDisk. Der Einsatz einer WMI-Abfrage hat hierbei den Vorteil, dass Sie mit der ExecQuery-Abfrage alle Datenträgerobjekte selektieren können und die einzelnen Datenträger nicht explizit angeben müssen, wie es beim Drive-Objekt der Fall ist.

Um beispielsweise eine Aufstellung aller Datenträger unter Angabe des Laufwerkbuchstabens (Caption), des Dateisystems (Filesystem), der Gesamtkapazität in Bytes (Size) und des freien Speichers (FreeSpace) zu erzeugen, können Sie das nachfolgende Listing einsetzen. Das Skript führt eine ExecQuery-Abfrage gegen die WMI-Klasse Win32_LogicalDisk durch, woraufhin eine Kollektion aller Datenträger zurückgeliefert wird. Mit Hilfe einer For Each-Schleife werden dann pro Datenträger die Werte der Eigenschaften Caption, Filesystem, Size und FreeSpace ermittelt und mit WScript.Echo am Bildschirm ausgegeben.

```
Option Explicit

Dim oWMI
Dim colDisk
Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:{impersonationLevel=_
    Impersonate}!\\" & strComputer & "\root\cimv2")
Set colDisk = oWMI.ExecQuery("SELECT * FROM Win32_LogicalDisk")

For Each objItem in colDisk
    WScript.Echo "Caption: " & objItem.Caption
    WScript.Echo "Filesystem: " & objItem.Filesystem
    WScript.Echo "Size : " & objItem.Size & " Bytes"
```

Listing 285: 2_03_005.vbs

```
wScript.Echo "FreeSpace : " & objItem.FreeSpace & " Bytes" & VBCRLF
Next
```

```
Set oWMI = Nothing
```

Listing 285: 2_03_005.vbs (Forts.)

66 Partitionen und Volumes eines Datenträgers anzeigen lassen

Wenn Sie auf die Schnelle herausfinden müssen, in welche Partitionen ein Basisdatenträger aufgeteilt ist bzw. welche Volumes ein dynamischer Datenträger aufweist, so brauchen Sie unter grafischer Oberfläche nur die DATENTRÄGERVERWALTUNG zu öffnen. Denn in der DATENTRÄGERVERWALTUNG werden alle eingebauten Datenträger unter Angabe der Partitionen bzw. Volumes angezeigt.

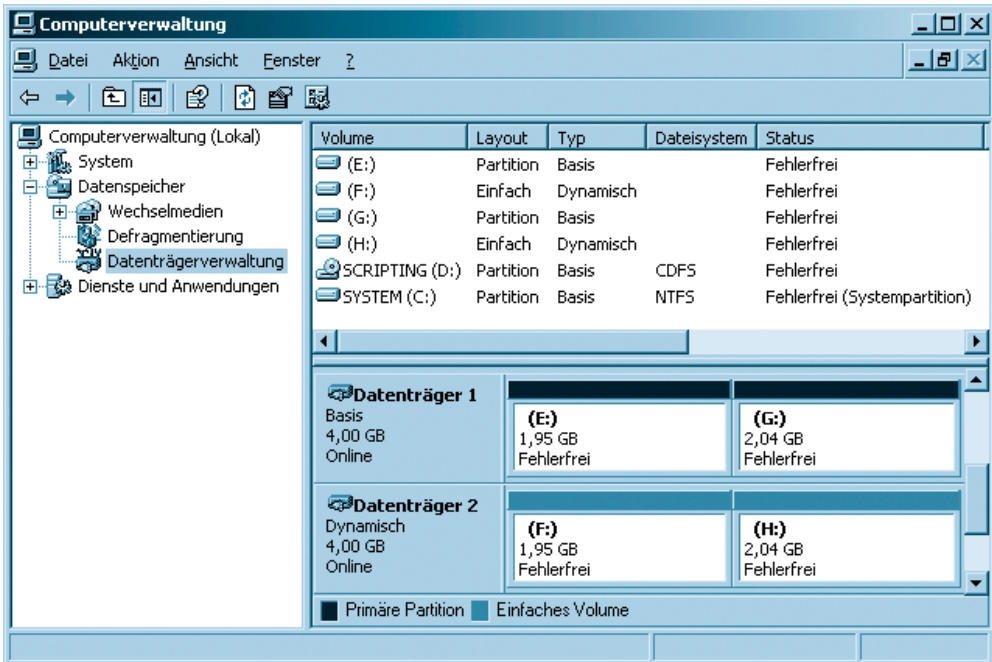


Abbildung 115: Die Übersicht der Partitionen und Volumes der Datenträger

Um beispielsweise die Anzahl und auch die Größen der Partitionen eines Basisdatenträgers bzw. der Volumes eines dynamischen Datenträgers zu erfahren, rufen Sie zunächst aus dem Kontextmenü des ARBEITSPLATZ-Symbols VERWALTEN auf, um in die COMPUTERVERWALTUNG zu gelangen. Dort erweitern Sie den Abschnitt DATENSPEICHER und markieren unterhalb davon DATENTRÄGERVERWALTUNG. Im unteren Bereich der rechten Fensterhälfte werden nun alle Datenträger unter Angabe der jeweiligen Parti-

tionierung bzw. der Volumes angezeigt. Anhand der farbig dargestellten Datenträgerbereiche können Sie die Anzahl der Partitionen/Volumes ermitteln und anhand der Kurzinformation, die in den jeweiligen Datenträgerbereichen eingeblendet wird, die Partitions- bzw. Volumengröße.

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus die Aufteilung eines Datenträgers in Partitionen bzw. Volumes ermitteln möchten, so verwenden Sie hierfür den `Diskpart`-Befehl, der standardmäßig zum Befehlsumfang von Windows Server 2003 gehört.

Geben Sie `Diskpart` ein, um in die `Diskpart`-eigene Eingabeaufforderung zu gelangen, und markieren Sie mit `Select Disk=x` den entsprechenden Datenträger, wobei Sie `x` durch die Nummer des Datenträgers ersetzen. Wenn Sie z.B. die Aufteilung des ersten Datenträgers herausfinden möchten, so setzen Sie den Fokus mit `Select Disk=0` auf die erste Festplatte. Anschließend geben Sie den Befehl `List Partition` ein, womit Sie eine Auflistung aller Partitionen des zuvor ausgewählten Datenträgers erhalten. Wenn Sie die Volumes eines dynamischen Datenträgers abfragen möchten, so verwenden Sie in diesem Fall ebenfalls den Befehl `List Partition`, da der `List Volume`-Befehl nicht die Volumes eines dynamischen Datenträgers ausgibt, sondern eine Aufstellung der verfügbaren Laufwerke erzeugt.

Die Aufstellung der Partitionen können Sie natürlich auch aus einer Batch-Datei heraus erzeugen, ohne dass hierfür eine Interaktion mit der `Diskpart`-Eingabeaufforderung erforderlich ist. Hierzu rufen Sie den `Diskpart`-Befehl mit dem Parameter `/s` auf und übergeben ein so genanntes `Diskpart`-Skript, worin alle erforderlichen `Diskpart`-Befehle hinterlegt sind. Wenn Sie z.B. per Batch-Datei eine Aufstellung der Partitionen des ersten Datenträgers (*Disk #0*) erzeugen und das Ergebnis in die Datei *Partition.txt* ausgeben möchten, so nehmen Sie in der Batch-Datei die folgende `Diskpart`-Befehlszeile auf:

```
@Echo off
Diskpart /s ListPartition.txt > Partition.txt
```

Listing 286: 2_03_006a.bat

Anschließend erstellen Sie eine Textdatei mit der Bezeichnung *ListPartition.txt* und nehmen die `diskpart`-spezifischen Befehle zum Markieren des Datenträgers (`Select Disk=0`) und zur Auflistung der Partitionen (`List Partition`) auf und kopieren die Textdatei in den gleichen Pfad, in dem sich auch die Batch-Datei befindet. Dann führen Sie die Batch-Datei aus, woraufhin der Inhalt des `Diskpart`-Skriptes abgearbeitet und das Ergebnis in die Datei *Partition.txt* geschrieben wird.


```
Select Disk=0  
List Partition
```

Listing 287: ListPartition.txt

Eine weitere Möglichkeit, eine Aufstellung der Partitionen zu erzeugen, bietet der WMIC-Befehl mit dem Alias `Partition`, womit Sie von der Eingabeaufforderung aus eine WMI-Abfrage gegen die Klasse `Win32_DiskPartition` durchführen und Informationen zu den einzelnen Partitionen abfragen können. Wenn Sie z.B. alle Partitionen unter Angabe ihrer Bezeichnung (`Caption`), der Größe (`Size`) und des Typs (`Type`) in die Datei `Partition.txt` ausgeben möchten, so wenden Sie den WMIC-Befehl unter Angabe des Alias `Partition` an, fragen mit der `Get`-Anweisung explizit die Werte der entsprechenden Eigenschaften ab und leiten das Ergebnis mit Hilfe einer Ausgabumleitung in die angegebene Datei um.

```
@Echo off  
WMIC Partition Get Caption, Size, Type > Partition.txt
```

Listing 288: 2_03_006b.bat

Die Lösung mit VBScript

Die WMI-Klasse `Win32_DiskPartition`, die Informationen zu den einzelnen Partitionen und Volumes besitzt, stellt eine ganze Reihe von Eigenschaften bereit, deren Werte sich natürlich auch per VBScript abfragen lassen.

Die nachfolgende Tabelle zeigt einen kleinen Auszug der Eigenschaften, die sich zur Auflistung der Partitionen und Volumes sehr gut eignen. Eine vollständige Übersicht aller Eigenschaften finden Sie in *Teil III* dieses Buches.

Eigenschaft	Beschreibung
DeviceID	Gibt die Datenträger- und Partitionsnummer an.
Bootable	Enthält den Wert <code>True</code> , wenn von der Partition gebootet werden kann, anderenfalls ist der Wert <code>False</code> .
BootPartition	Enthält den Wert <code>True</code> , wenn die Partition die Boot-Partition ist, anderenfalls ist der Wert <code>False</code> .
PrimaryPartition	Enthält den Wert <code>True</code> , wenn die Partition vom Partitionstyp Primär ist, anderenfalls ist der Wert <code>False</code> .
Size	Enthält die Gesamtkapazität der Partition.
Systemname	Enthält den Computernamen.

Tabelle 44: Einige Eigenschaften der Win32_DiskPartition-Klasse

Wenn Sie z.B. eine Aufstellung aller Partitionen erzeugen möchten und die Werte der Eigenschaften, die in der obigen Tabelle enthalten sind, ausgeben möchten, so führen Sie eine ExecQuery-Abfrage gegen die Klasse Win32_DiskPartition durch, um eine Kollektion aller Partitionen zu erhalten. Innerhalb einer For Each-Schleife können Sie dann die Werte der Eigenschaften ermitteln und mit der WScript.Echo-Anweisung am Bildschirm ausgeben.

```
Option Explicit

Dim oWMI
Dim colPartition
Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "[impersonationLevel=_
    impersonate]!\\" & strComputer & "\root\cimv2")
Set colPartition = oWMI.ExecQuery ("SELECT * FROM Win32_DiskPartition")

For Each objItem in colPartition
    WScript.Echo "Device-ID: " & objItem.DeviceID
    WScript.Echo "Bootable: " & objItem.Bootable
    WScript.Echo "Bootpartition: " & objItem.Bootpartition
    WScript.Echo "Primary Partition: " & objItem.PrimaryPartition
    WScript.Echo "Size: " & objItem.Size
    WScript.Echo "Systemname: " & objItem.Systemname
Next

Set oWMI = Nothing
```

Listing 289: 2_03_006.vbs

67 Basisdatenträger in dynamische Datenträger konvertieren

Der Windows Server 2003 erlaubt das Konvertieren eines gewöhnlichen Datenträgers (Basisdatenträger genannt) in einen dynamischen Datenträger, womit Ihnen erweiterte Datenträgerfunktionen zur Verfügung stehen. Denn auf dynamischen Datenträgern lassen sich softwarebasierte RAID-Systeme einrichten, womit Sie beispielsweise ein Laufwerk (auf dynamischen Datenträgern *Volume* genannt) erstellen können, das sich über mehrere Datenträger hinweg erstreckt oder Datenträger der Ausfallsicherheit wegen auch spiegeln kann.

Die nachfolgende Tabelle enthält eine Übersicht der Datenträgerkonfigurationen, die Sie mit Hilfe eines dynamischen Datenträgers erstellen können, und gibt Auskunft über die Mindestanzahl an Datenträgern und der wesentlichen Funktionsmerkmale:

Volume-Typ	Anzahl der Datenträger	Beschreibung
Einfach	1	Ein einfaches Volume stellt ein Laufwerk dar, das auf nur einem dynamischen Datenträger eingerichtet ist und sich nicht über mehrere Datenträger hinweg erstreckt. Einfache Volumes können durch Einsatz weiterer dynamischer Datenträger bzw. Volumes zu übergreifenden und gespiegelten Volumes geändert werden.
Übergreifend	2	Als übergreifendes Volume wird ein Laufwerk bezeichnet, das sich über mehrere dynamische Datenträger hinweg erstreckt. So lässt sich ein großes Laufwerk erstellen, das sich aus dem Speicherplatz von zwei oder auch mehreren dynamischen Datenträgern zusammensetzt.
Stripeset	2	Ein Stripeset setzt sich aus mindestens zwei dynamischen Datenträgern zusammen. Die Datenblöcke werden abwechselnd auf beide Datenträger geschrieben und so über beide Datenträger gleichmäßig verteilt. Das Volume verfügt damit über die Kapazität beider Datenträger. Diese Konfiguration wird auch als RAID 0 bezeichnet. Ein Stripeset kann nicht erweitert oder gespiegelt werden, bietet jedoch eine Steigerung der Performance, weil die Zugriffe von beiden Datenträgern parallel verarbeitet werden können.
Gespiegelt	2	Ein gespiegeltes Volume (auch als RAID 1 bezeichnet) erfordert genau zwei dynamische Datenträger und bietet eine gewisse Ausfallsicherheit, da die Daten des einen Volumes 1:1 auf das zweite Volume dupliziert werden. Fällt das primäre Volume aus, so verwendet das Betriebssystem automatisch das sekundäre Volume. Ein gespiegeltes Volume kann nicht über mehrere Datenträger erweitert werden.
RAID 5	3	Ein RAID 5-Volume setzt sich aus mindestens drei dynamischen Datenträgern zusammen. Die Daten werden dabei abwechselnd auf die drei Datenträger geschrieben und so gleichmäßig über alle drei Datenträger verteilt. Das RAID 5-Volume speichert Paritätsinformationen auf den Datenträgern, womit sich ein Datenträger, der ausgefallen ist, aufgrund der Paritätsinformationen sehr schnell wiederherstellen lässt. Das RAID 5-Volume bietet somit eine sehr gute Ausfallsicherheit, lässt sich jedoch nicht erweitern oder spiegeln.

Tabelle 45: Die Volume-Typen und ihre wesentlichen Merkmale

Die Konvertierung eines Basisdatenträger in einen dynamischen Datenträger nehmen Sie unter grafischer Oberfläche mit Hilfe der DATENTRÄGERVERWALTUNG vor. Um diese zu öffnen, wählen Sie im Kontextmenü des ARBEITSPLATZ-Symbols VERWALTEN aus, wor-

aufhin die COMPUTERVERWALTUNG gestartet wird. In der COMPUTERVERWALTUNG erweitern Sie den Abschnitt DATENSPEICHER und markieren unterhalb davon den Eintrag DATENTRÄGERVERWALTUNG.

Achtung

Die Konvertierung eines Basisdatenträgers in einen dynamischen Datenträger kann nur durch eine erneute Konvertierung rückgängig gemacht werden. Diese erneute Konvertierung setzt voraus, dass alle Volumes, die auf dem dynamischen Datenträger eingerichtet wurden, zuvor gelöscht wurden. Andernfalls kann der dynamische Datenträger nicht in einen Basisdatenträger zurückverwandelt werden.

Datenträger-
verwaltung

Im unteren Bereich der rechten Fensterhälfte sehen Sie nun eine Auflistung aller Datenträger. Markieren Sie das graue Feld des Datenträgers, den Sie konvertieren möchten, und wählen Sie im Kontextmenü IN DYNAMISCHEN DATENTRÄGER KONVERTIEREN... aus. Sie gelangen daraufhin in das Dialogfenster IN DYNAMISCHEN DATENTRÄGER KONVERTIEREN und müssen die Auswahl des zuvor markierten Datenträgers noch einmal bestätigen. Klicken Sie auf OK, um die Konvertierung zu starten. Nach Abschluss der Konvertierung (die meist nur einen Augenblick dauert) wird der Datenträger mit DYNAMISCH ausgewiesen, und die Konvertierung ist abgeschlossen.

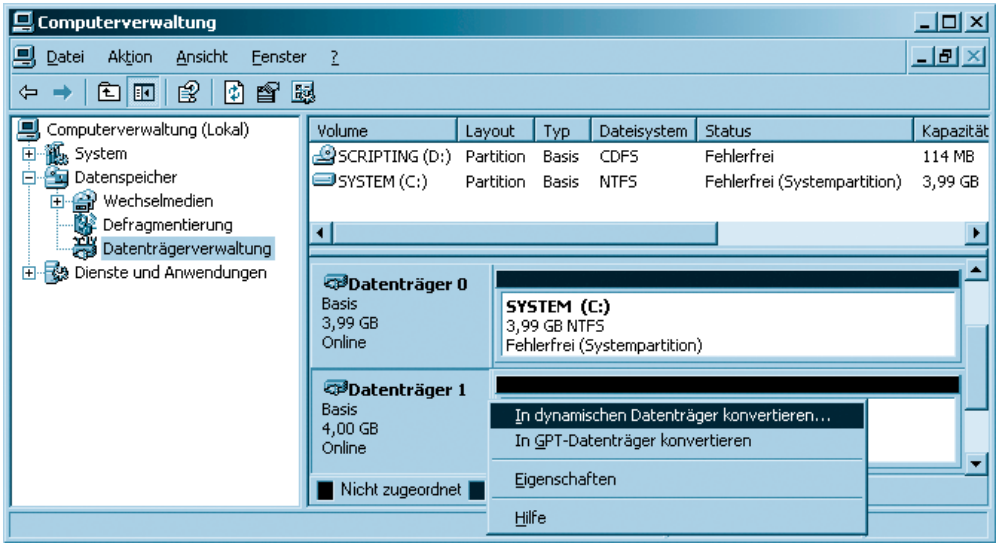


Abbildung 116: Einen Basisdatenträger in einen dynamischen Datenträger konvertieren

Die Lösung per Eingabeaufforderung

Mit dem Diskpart-Befehl, der standardmäßig im Befehlsumfang von Windows Server 2003 enthalten ist, lassen sich Basisdatenträger auf recht einfache Art und Weise auch von der Eingabeaufforderung aus in einen dynamischen Datenträger konvertieren:

In der Eingabeaufforderung geben Sie den Befehl `Diskpart` ein, woraufhin Sie in die `Diskpart`-eigene Eingabeaufforderung gelangen. Anschließend selektieren Sie mit `Select Disk=x` den Datenträger, den Sie zu einem dynamischen Datenträger konvertieren möchten, wobei Sie das `x` durch die Nummer des Datenträgers (0 = erster Datenträger, 1 = zweiter Datenträger usw.) ersetzen. Dann führen Sie den Befehl `Convert Dynamic` aus, um die eigentliche Konvertierung des Datenträgers durchzuführen. Nach Abschluss der Konvertierung (die meist nur wenige Sekunden dauert) verlassen Sie die `Diskpart`-eigene Eingabeaufforderung mit `Exit`.

Wenn Sie die Konvertierung aus einer Batch-Datei heraus aufrufen möchten, so können Sie die `Diskpart`-spezifischen Befehle, die Sie zur Konvertierung eingeben müssen, auch in einer Textdatei hinterlegen und den `Diskpart`-Befehl mit dem Parameter `/s` aufrufen, womit die Befehle dann aus der angegebenen Skriptdatei gelesen werden. Und damit Sie die Rückmeldungen, die der `Diskpart`-Befehl während der Befehlsausführung ausgibt, zu einem späteren Zeitpunkt noch einsehen können, sollten Sie das Ergebnis per Ausgabeumleitung in eine gesonderte Textdatei schreiben.

```
@Echo off
Diskpart /s ConvertToDynamic.txt > Result.txt
```

Listing 290: 2_03_007.bat

Die `Diskpart`-Befehlszeile führt den Inhalt der Datei `ConvertToDynamic.txt` aus, ohne dass hierfür eine Interaktion mit der `Diskpart`-eigenen Eingabeaufforderung notwendig ist. Denn die Datei `ConvertToDynamic.txt` enthält die erforderlichen Befehlszeilen, womit zunächst der Datenträger selektiert (`Select Disk=1`) und dann die Konvertierung durchgeführt (`Convert Dynamic`) wird:

```
Select Disk=1
Convert Dynamic
```

Listing 291: ConvertToDynamic.txt

Die Lösung mit VBScript

Da VBScript leider keine Möglichkeiten zur Datenträgerkonvertierung bietet, bleibt nur der Einsatz der `Run`-Methode des `Shell`-Objektes, womit Sie den `Diskpart`-Befehl der Eingabeaufforderung auch aus einem VBScript heraus aufrufen können.

Um die Konvertierung eines Basisdatenträgers in einen dynamischen Datenträger per VBScript zu starten, erzeugen Sie mit der `CreateObject`-Methode einen Verweis auf das `WScript.Shell`-Objekt und wenden dann die `Run`-Methode zum Ausführen der Befehlszeile `Diskpart /s ConvertToDynamic.txt > Result.txt` an. Diese Befehlszeile führt die `Diskpart`-spezifischen Befehle, die in der Datei `ConvertToDynamic.txt` (siehe Lösung per Eingabeaufforderung) hinterlegt sind, aus und leitet das Ergebnis der Befehlsausführung in die Datei `Result.txt` um.

```
Option Explicit
Dim oShell
```

```
Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Diskpart /s ConvertToDynamic.txt > Result.txt"
```

```
Set oShell = Nothing
```

Listing 292: 2_03_007.vbs

68 Dynamischen Datenträger in Basisdatenträger konvertieren

Ein dynamischer Datenträger kann natürlich auch wieder in einen Basisdatenträger zurückverwandelt werden. Dies setzt allerdings voraus, dass alle Daten, die auf den einzelnen Volumes des dynamischen Datenträgers abgelegt sind, auf ein anderes Medium verschoben oder zumindest gesichert worden sind, da die Konvertierung eines dynamischen Datenträgers in einen Basisdatenträger erst dann durchgeführt werden kann, wenn alle Volumes (die Partitionen und damit auch deren Inhalte) gelöscht wurden.

Die Konvertierung eines dynamischen Datenträgers in einen Basisdatenträger nehmen Sie unter grafischer Oberfläche ebenfalls mit Hilfe der DATENTRÄGERVERWALTUNG von Windows Server 2003 vor: Öffnen Sie die DATENTRÄGERVERWALTUNG, und markieren Sie im unteren Bereich der rechten Fensterhälfte das graue Feld des dynamischen Datenträgers, den Sie in einen Basisdatenträger konvertieren möchten. Wenn der dynamische Datenträger noch über eingerichtete Volumes verfügt und die Volumes Bestandteil einer übergreifenden, einer Stripeset-, einer gespiegelten oder gar RAID 5-Festplattenkonfiguration sind, so müssen Sie die jeweilige Festplattenkonfiguration aus dem Kontextmenü des Volumes zunächst aufheben und das Volume danach löschen.

Anschließend wählen Sie im Kontextmenü des Datenträgers den Eintrag **IN EINEN BASISDATENTRÄGER KONVERTIEREN** aus. Der Datenträger wird daraufhin in einen Basisdatenträger konvertiert und wird in der Datenträgerliste wieder als **BASIS** ausgewiesen (siehe Abbildung 117).

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung nehmen Sie die Konvertierung eines dynamischen Datenträgers in einen Basisdatenträger wieder mit dem **Diskpart**-Befehl vor. Öffnen Sie hierzu das Eingabeaufforderungsfenster, und geben Sie **Diskpart** ein, um in die Diskpart-eigene Eingabeaufforderung zu gelangen. Dann selektieren Sie mit **Select Disk=x** den dynamischen Datenträger, den Sie konvertieren möchten, und ersetzen das **x** durch die Nummer des Datenträgers (0, 1, 2 ...). Anschließend geben Sie **Convert Basic** ein, woraufhin der Datenträger konvertiert wird, und verlassen die Diskpart-Eingabeaufforderung mit **Exit**.

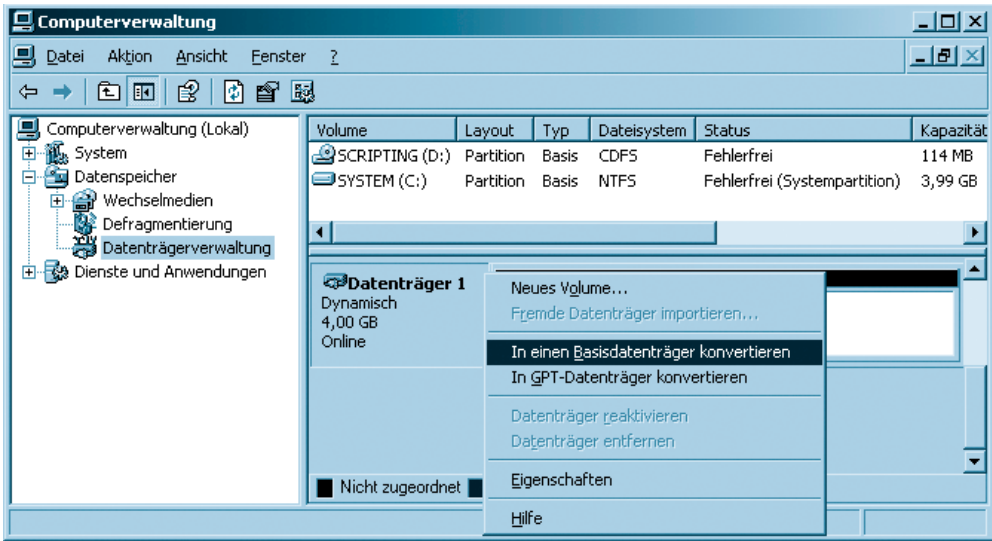


Abbildung 117: Einen dynamischen Datenträger in einen Basisdatenträger konvertieren

Auch die Konvertierung eines dynamischen Datenträgers in einen Basisdatenträger lässt sich durch Einsatz eines Diskpart-Skriptes wieder automatisieren. Führen Sie hierfür den Diskpart-Befehl unter Angabe des Parameters /s aus, übergeben Sie den Namen des Diskpart-Skriptes, und verwenden Sie (bei Bedarf) eine Ausgabeumleitung, womit die Rückmeldung der Befehlsausführung in eine gesonderte Textdatei geschrieben wird.

```
@Echo off
Diskpart /s ConvertToBasic.txt > Result.txt
```

Listing 293: 2_03_008.bat

Der Diskpart-Befehl führt daraufhin die Befehle aus, die in der angegebenen Datei (in diesem Fall *ConvertToBasic.txt*) hinterlegt sind:

```
Select Disk=1
Convert Basic
```

Listing 294: ConvertToBasic.txt

Die Lösung mit VBScript

Wenn Sie die Konvertierung eines dynamischen Datenträgers in einen Basisdatenträger aus einem VBScript heraus starten möchten, so bleibt nur der Einsatz der Run-Methode des Shell-Objektes, womit Sie den Diskpart-Befehl der Eingabeaufforderung auch aus einem VBScript heraus aufrufen können:

```
Option Explicit
Dim oShell
```

```
Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Diskpart /s ConvertToBasic.txt > Result.txt"
```

```
Set oShell = Nothing
```

Listing 295: 2_03_008.vbs

69 Eine primäre Partition anlegen

Neue Partitionen sind dank der komfortablen Datenträgerverwaltung von Windows Server 2003 schnell erstellt. Wenn Sie beispielsweise gerade einen zusätzlichen Datenträger eingebaut haben oder der aktuelle Datenträger noch über nicht zugeordnete Speicherbereiche verfügt, so können Sie auf dem freien Speicherbereich eine primäre Partition einrichten und später eine Betriebssysteminstallation durchführen oder die Partition z.B. als eigenständiges Datenlaufwerk nutzen.

Um unter grafischer Oberfläche eine primäre Partition anzulegen, wechseln Sie in die DATENTRÄGERVERWALTUNG. Hierzu rufen Sie aus dem Kontextmenü des ARBEITSPLATZ-Symbols VERWALTEN auf, woraufhin die COMPUTERVERWALTUNG gestartet wird. In der COMPUTERVERWALTUNG erweitern Sie den Abschnitt DATENSPEICHER und markieren unterhalb davon den Eintrag DATENTRÄGERVERWALTUNG.

Im unteren Bereich der rechten Fensterhälfte sehen Sie nun alle Datenträger unter Angabe der jeweiligen Partitionierung. Freie Speicherbereiche, die partitioniert werden können, sind dabei als NICHT ZUGEORDNET ausgewiesen. Markieren Sie nun den freien Speicherbereich, den Sie partitionieren möchten, und wählen Sie im Kontextmenü den Eintrag NEUE PARTITION... aus. Daraufhin wird der ASSISTENT ZUM ERSTELLEN NEUER PARTITIONEN gestartet. Schließen Sie den Willkommensbildschirm mit einem Klick auf WEITER, und wählen Sie im nächsten Schritt die Option PRIMÄRE PARTITION aus, um den Partitionierungstyp *Primär* festzulegen (siehe Abbildung 118).

Im nächsten Schritt legen Sie die Größe der neuen Partition fest. Die Mindestgröße beträgt dabei immer 8 MByte, die maximale Größe hängt vom verfügbaren freien Speicherbereich ab. Wenn Sie die Größe eingegeben haben, klicken Sie auf WEITER, um einen Laufwerkbuchstaben für die neue Partition festzulegen. Wählen Sie einen der freien Laufwerkbuchstaben aus, oder aktivieren Sie die Option KEINEN LAUFWERKBUCHSTABEN ODER -PFAD ZUWEISEN, wenn Sie dies später tun möchten oder keinen Laufwerkbuchstaben benötigen.

Anschließend haben Sie die Möglichkeit, die neue Partition mit *FAT*, *FAT32* oder idealerweise mit *NTFS* zu formatieren, wobei Sie auch die Größe der Zuordnungseinheiten auf dem Datenträger festlegen können. Danach werden die Einstellungen noch einmal angezeigt, und mit einem Klick auf FERTIG STELLEN wird die Partition angelegt (und je nach Auswahl auch sofort formatiert).

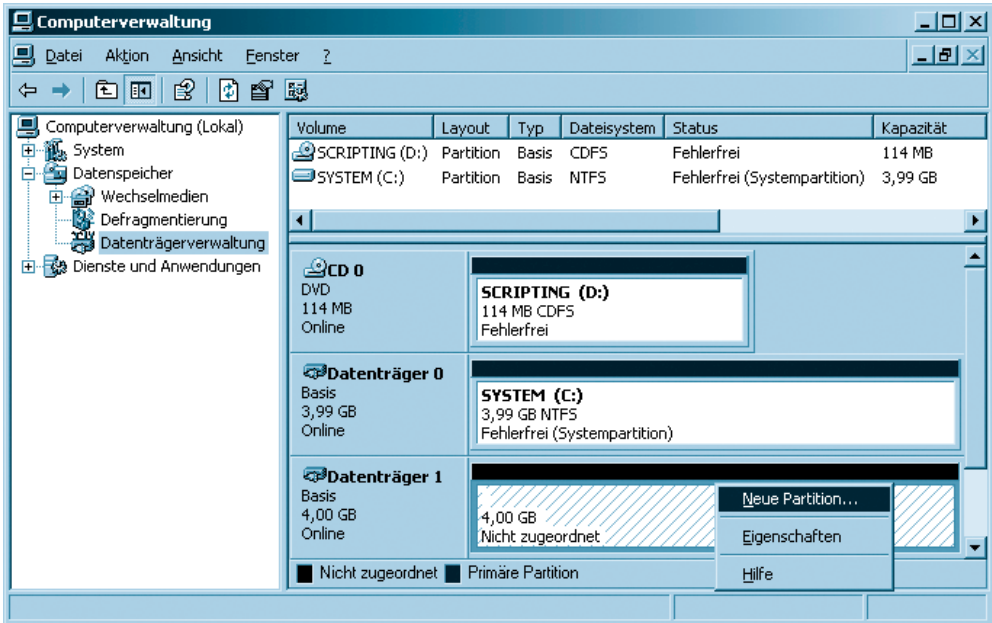


Abbildung 118: Eine neue Partition anlegen

Die Lösung per Eingabeaufforderung

Der Windows Server 2003 unterstützt das Anlegen einer primären Partition auch von der Eingabeaufforderung aus. Denn mit dem `Diskpart`-Befehl lassen sich auf recht einfache Art und Weise neue Partitionen erstellen und gegebenenfalls auch wieder löschen. Sie haben dabei die Möglichkeit, die `Diskpart`-Eingabeaufforderung zu nutzen und die Partitionierung durch direkte Eingabe der `Diskpart`-Befehle vorzunehmen oder die Partitionierung mit Hilfe eines Skriptes durchzuführen, das die entsprechenden `Diskpart`-Befehle enthält.

Wenn Sie auf dem lokalen Computersystem mit `Diskpart` eine zweite Partition vom Typ *Primär* anlegen möchten, weil der Datenträger noch über ungenutzten Speicherplatz verfügt, so rufen Sie in der Eingabeaufforderung zunächst `Diskpart` auf, um in die `Diskpart`-Eingabeaufforderung zu gelangen. Anschließend setzen Sie mit dem Befehl `Disk=x` den Fokus auf den Datenträger, dessen Partitionierung Sie einsehen oder bearbeiten möchten, wobei Sie `x` durch die Nummer des Datenträgers (0 = erster Datenträger, 1 = zweiter Datenträger usw.) ersetzen. Wenn nur eine Festplatte eingebaut ist, selektieren Sie die Festplatte also mit `Disk=0`.

Um eine Partition anzulegen, verwenden Sie den Befehl `Create Partition`, dem Sie zumindest den Typ der neuen Partition (in diesem Fall *Primary*) wie auch die Größe der Partition (Angabe erfolgt in MByte) als Parameter übergeben. Eine primäre Partition mit einer Größe von 5.000 MByte legen Sie mit dem Befehl `Create Partition Primary Size=5000` an, wobei die Partition standardmäßig am Anfang des freien Speicherbereichs angelegt wird. Nach dem Erstellen der Partition können Sie die `Diskpart`-Eingabeaufforderung mit dem Befehl `Quit` wieder verlassen.

Die gleiche Aufgabe lässt sich mit Hilfe eines `Diskpart`-Skriptes auch automatisieren. Wenn Sie z.B. aus einer Batch-Datei heraus eine Partitionierung vornehmen möchten, so rufen Sie aus der Batch-Datei heraus den `Diskpart`-Befehl mit dem Parameter `/s` auf und übergeben den Namen des Skriptes, worin die einzelnen `Diskpart`-Befehle hinterlegt sind.

Die nachfolgende Befehlszeile startet `Diskpart` und führt die Befehle aus, die in der Datei `CreatePrimaryPartition.txt` hinterlegt sind, ohne die `Diskpart`-Eingabeaufforderung zu starten. Das Ergebnis der `Diskpart`-Ausführung wird per Ausgabeumleitung in die Datei `Result.txt` geschrieben, damit der Status später eingesehen werden kann:

```
@Echo off
Diskpart /s CreatePrimaryPartition.txt > Result.txt
```

Listing 296: 2_03_009.bat

Das Skript `CreatePrimaryPartition.txt` enthält dabei die entsprechenden `Diskpart`-Befehle, womit zunächst der Datenträger selektiert und dann die Partition angelegt wird:

```
Select Disk=0
Create Partition Primary Size=5000
```

Listing 297: CreatePrimaryPartition.txt

Die Lösung mit VBScript

Wenn Sie die primäre Partition per VBScript anlegen möchten, so führen Sie mit Hilfe der `Run`-Methode des `Shell`-Objektes ebenfalls den `Diskpart`-Befehl unter Angabe der einzelnen `Diskpart`-Anweisungen aus, da VBScript leider keine entsprechenden Möglichkeiten bietet, womit Sie einen Datenträger partitionieren können.

Das nachfolgende Listing legt auf dem ersten Datenträger des lokalen Computersystems eine primäre Partition mit einer Größe von 5.000 MByte an. Hierzu wird mit der `Run`-Methode der `Diskpart`-Befehl mit dem Parameter `/s` aufgerufen und das `Diskpart`-Skript mit der Bezeichnung `CreatePrimaryPartition.txt` übergeben, worin die einzelnen `Diskpart`-Anweisungen hinterlegt sind. Damit die Partitionierung durchgeführt werden kann, muss das `Diskpart`-Skript im gleichen Ordner abgelegt sein wie auch das VBScript.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
```

Listing 298: 2_03_009.vbs

```
oShell.Run "%comspec% /c Diskpart /s CreatePrimaryPartition.txt > Result.txt"
```

Set oShell = Nothing

Listing 298: 2_03_009.vbs (Forts.)

70 Eine erweiterte Partition anlegen

Ein Basisdatenträger lässt sich in maximal vier primäre Partitionen aufteilen. Wenn Sie eine weitere Aufteilung des Basisdatenträgers benötigen, so können Sie statt der vierten primären Partition eine erweiterte Partition einrichten und die erweiterte Partition dann mit mehreren logischen Laufwerken versehen. Logische Laufwerke können mit FAT, FAT32 und auch mit NTFS formatiert sowie mit einem Laufwerksbuchstaben versehen werden, so dass sie wie eigenständige Datenträger erscheinen. Wie viele logische Laufwerke Sie einrichten können, hängt dabei von der Größe der erweiterten Partition und von den verfügbaren Laufwerksbuchstaben (A–Z) ab.

Um nun unter grafischer Oberfläche eine erweiterte Partition anzulegen und damit die Voraussetzung für die Einrichtung logischer Laufwerke zu schaffen, nutzen Sie wieder die DATENTRÄGERVERWALTUNG von Windows Server 2003: Rufen Sie aus dem Kontextmenü des ARBEITSPLATZ-Symbols VERWALTEN auf, um die COMPUTERVERWALTUNG zu öffnen. Erweitern Sie in der COMPUTERVERWALTUNG den Abschnitt DATENSPEICHER, und markieren Sie unterhalb davon DATENTRÄGERVERWALTUNG.

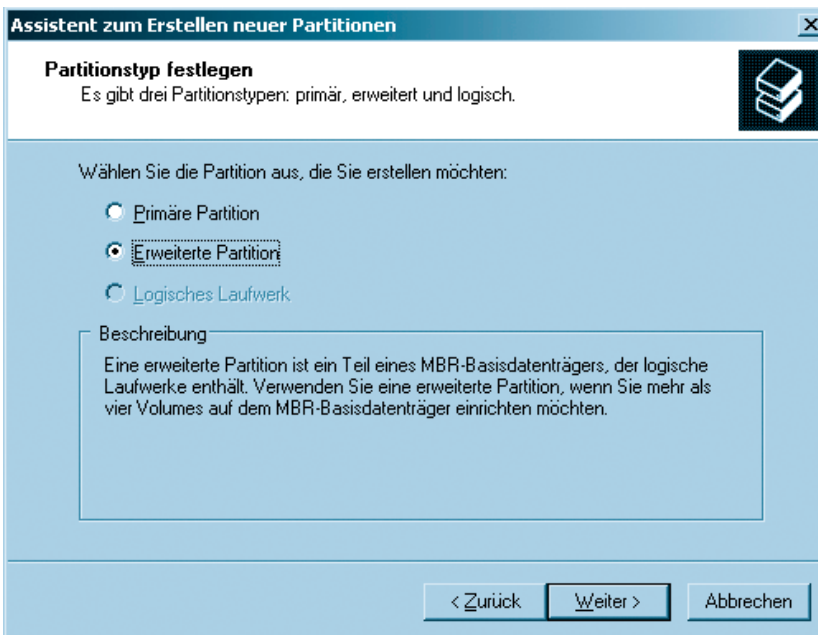


Abbildung 119: Eine erweiterte Partition anlegen

In der rechten Fensterhälfte finden Sie nun eine Übersicht der vorhandenen Datenträger vor. Markieren Sie im unteren Teil der rechten Fensterhälfte den freien Speicherbereich (dieser ist mit NICHT ZUGEORDNET ausgewiesen), auf dem Sie eine erweiterte Partition anlegen möchten, und wählen Sie im Kontextmenü des schraffierten Feldes NEUE PARTITION... aus. Daraufhin wird der ASSISTENT ZUM ERSTELLEN NEUER PARTITIONEN gestartet. Schließen Sie die erste Dialogseite mit einem Klick auf WEITER, und wählen Sie im nächsten Schritt den Partitionstyp ERWEITERTE PARTITION aus. Legen Sie daraufhin die Größe der erweiterten Partition fest, und beenden Sie den Assistenten mit einem Klick auf FERTIG STELLEN.

Die Lösung per Eingabeaufforderung

Wenn Sie die erweiterte Partition dagegen in der Eingabeaufforderung erstellen möchten, so verwenden Sie den `Diskpart`-Befehl von Windows Server 2003: Rufen Sie `Diskpart` auf, um in die `Diskpart`-eigene Eingabeaufforderung zu gelangen, und wählen Sie den Datenträger aus, auf dem Sie die erweiterte Partition anlegen möchten. Die Auswahl des Datenträgers nehmen Sie mit dem Befehl `Select Disk=x` vor, wobei Sie `x` durch die Nummer des Datenträgers (0 = erster Datenträger, 1 = zweiter Datenträger ...) ersetzen. Den ersten Datenträger wählen Sie demnach mit `Select Disk=0` aus.

Zum Anlegen einer neuen Partition verwenden Sie den `Create`-Befehl, dem Sie den Typ der Partition (in diesem Fall `Extended`) wie auch die Partitionsgröße (die Angabe erfolgt in MByte) als Parameter übergeben. Wenn Sie die Größe der Partition nicht explizit festlegen, so wird automatisch die Gesamtkapazität verwendet. Um beispielsweise eine erweiterte Partition mit einer Größe von 5.000 MByte anzulegen, geben Sie `Create Partition Extended Size=5000` ein. Die erweiterte Partition wird daraufhin angelegt, und Sie können die `Diskpart`-Eingabeaufforderung mit `Exit` wieder verlassen.

Der `Diskpart`-Befehl erlaubt natürlich auch in diesem Fall die Verwendung eines Skriptes, in der alle `Diskpart`-spezifischen Befehle hinterlegt sind. Wenn Sie die Partition beispielsweise aus einer Batch-Datei heraus anlegen und eine Interaktion mit der `Diskpart`-Eingabeaufforderung vermeiden möchten, so führen Sie den `Diskpart`-Befehl mit dem Parameter `/s` aus und übergeben die Textdatei, in der alle Subbefehle von `Diskpart` hinterlegt sind.

Die nachfolgende `Diskpart`-Befehlszeile liest z.B. alle `Diskpart`-spezifischen Befehle aus der Datei `CreateExtendedPartition.txt` und schreibt das Ergebnis der Befehlsausführungen in die Datei `Result.txt`:

```
@Echo off
Diskpart /s CreateExtendedPartition.txt > Result.txt
```

Listing 299: 2_03_010.bat

Das Skript *CreateExtendedPartition.txt* enthält alle erforderlichen Befehlszeilen, womit zunächst der Datenträger 0 selektiert und anschließend die erweiterte Partition mit einer Größe von 5.000 MByte angelegt wird:

```
Select Disk=0  
Create Partition Extended Size=5000
```

Listing 300: CreateExtendedPartition.txt

Die Lösung mit VBScript

VBScript stellt leider keine entsprechenden Möglichkeiten bereit, womit Sie eine Partition anlegen können. Mit der *Run*-Methode des *Shell*-Objektes können Sie aber alle Eingabeaufforderungsbefehle auch aus einem VBScript heraus ausführen und die Partition damit auf die gleiche Art und Weise erstellen wie auch in der Eingabeaufforderung (siehe vorheriger Abschnitt).

Das nachfolgende Listing erzeugt mit der *CreateObject*-Methode zunächst eine neue Instanz des *Shell*-Objektes und wendet dann die *Run*-Methode unter Angabe des zu verwendenden Befehlszeileninterpreters (dieser ist in der Systemvariablen *%Comspec%* hinterlegt) und der *Diskpart*-Befehlszeile an, woraufhin der *Diskpart*-Befehl ausgeführt und der Inhalt des *Diskpart*-Skriptes *CreateExtendedPartition.txt* verarbeitet wird.

```
Option Explicit  
Dim oShell  
  
Set oShell = CreateObject ("WScript.Shell")  
oShell.Run "%comspec% /c Diskpart /s CreateExtendedPartition.txt >  
Result.txt"  
  
Set oShell = Nothing
```

Listing 301: 2_03_010.vbs

71 Logische Laufwerke einrichten

Wenn Sie – wie im vorherigen Rezept beschrieben – eine erweiterte Partition angelegt haben, so können Sie nun logische Laufwerke einrichten und die Laufwerke zur Datenaufnahme vorbereiten. Öffnen Sie hierfür wieder die DATENTRÄGERVERWALTUNG, indem Sie aus dem Kontextmenü des ARBEITSPLATZ-Symbols VERWALTEN wählen, woraufhin Sie zunächst in die COMPUTERVERWALTUNG gelangen. Dort erweitern Sie den Abschnitt DATENSPEICHER und klicken unterhalb davon auf DATENTRÄGERVERWALTUNG.

Im unteren Bereich der rechten Fensterhälfte sehen Sie nun alle Datenträger. Markieren Sie den freien Speicherplatz des Basisdatenträgers, auf dem Sie die erweiterte

Partition angelegt haben, und wählen Sie aus dem Kontextmenü des schraffierten Feldes NEUES LOGISCHES LAUFWERK... aus. Daraufhin wird der ASSISTENT ZUM ERSTELLEN NEUER PARTITIONEN gestartet. Klicken Sie auf WEITER, um den Willkommensbildschirm zu schließen, und übernehmen Sie im nächsten Schritt die einzig verfügbare Option LOGISCHES LAUFWERK mit einem Klick auf WEITER (siehe Abbildung 120).

Legen Sie nun die gewünschte Größe für das logische Laufwerk fest, und weisen Sie dem Laufwerk anschließend einen freien Laufwerksbuchstaben zu. Im nächsten Schritt haben Sie die Möglichkeit, das logische Laufwerk zu formatieren. Übernehmen Sie hierfür die Standardoption DIESE PARTITION MIT FOLGENDEN EINSTELLUNGEN FORMATIEREN, und wählen Sie das gewünschte Dateisystem (*FAT*, *FAT32* oder idealerweise *NTFS*) aus. Die ausgewählten Optionen werden daraufhin noch einmal angezeigt. Klicken Sie auf FERTIG STELLEN, um das logische Laufwerk mit den zuvor ausgewählten Optionen zu erstellen und – sofern ausgewählt – auch zu formatieren.

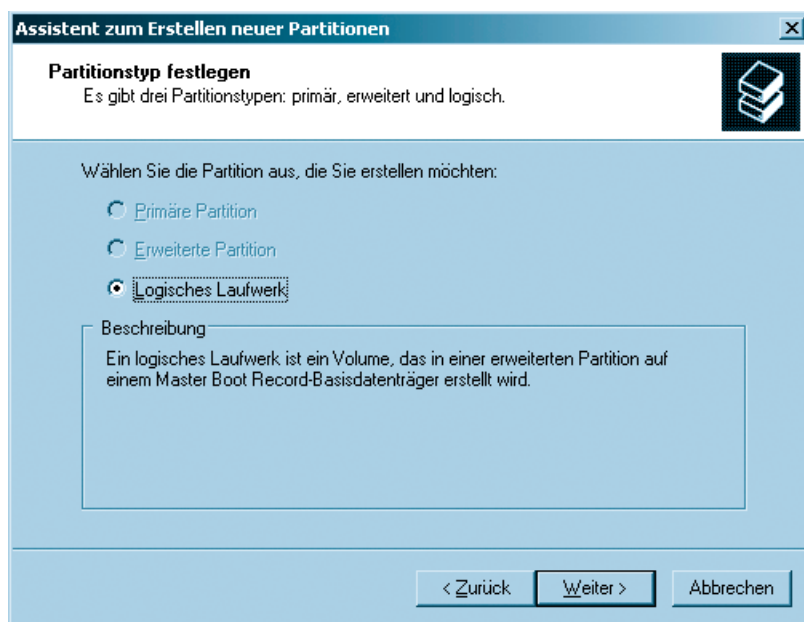


Abbildung 120: Ein logisches Laufwerk einrichten

Die Lösung per Eingabeaufforderung

Mit dem Diskpart-Befehl lassen sich logische Laufwerke auch von der Eingabeaufforderung aus erstellen. Geben Sie zunächst Diskpart ein, um in die Diskpart-eigene Eingabeaufforderung zu gelangen, und selektieren Sie mit Select Disk=x den Datenträger, auf dem die erweiterte Partition existiert (den Platzhalter x ersetzen Sie dabei durch die Nummer des Datenträgers). Anschließend wenden Sie den Create Partition-Befehl an, dem Sie den Typ (Logical) und auch die Größe des logischen Laufwerks (Size=xxxx), als Parameter übergeben.

Um beispielsweise ein logisches Laufwerk mit einer Größe von 5.000 MByte anzulegen, geben Sie im Anschluss an die Datenträgerauswahl den Befehl `Create Partition Logical Size=5000` ein.

Wenn Sie das logische Laufwerk aus einer Batch-Datei heraus anlegen und damit die Interaktion mit der `Diskpart`-eigenen Eingabeaufforderung vermeiden möchten, so rufen Sie den `Diskpart`-Befehl mit dem Parameter `/s` auf und übergeben die Textdatei, in der die `Diskpart`-spezifischen Befehle hinterlegt sind:

```
@Echo off
Diskpart /s CreateLogicalDisc.txt > Result.txt
```

Listing 302: 2_03_011.bat

Das Skript *CreateLogicalDisc.txt* enthält alle erforderlichen Befehle, die zum Erstellen eines logischen Laufwerks erforderlich sind. Die erste Befehlszeile selektiert den Datenträger 0, die zweite Befehlszeile führt den `Create Partition`-Befehl unter Angabe des Typs (Logical) und der Größe (Size=5000) aus:

```
Select Disk=0
Create Partition Logical Size=5000
```

Listing 303: CreateLogicalDisc.txt

Die Lösung mit VBScript

Da VBScript keine Möglichkeiten zum Anlegen von Partitionen und demzufolge auch nicht zum Erstellen logischer Laufwerke bietet, bleibt nur das Anwenden der `Run`-Methode des `Shell`-Objektes, womit Sie den `Diskpart`-Befehl der Eingabeaufforderung auch aus einem VBScript heraus aufrufen können.

Das nachfolgende Listing führt z.B. den `Diskpart`-Befehl unter Angabe des Parameters `/s` aus, womit dem `Diskpart`-Befehl das Skript *CreateLogicalDisc.txt* (siehe vorheriger Abschnitt) übergeben und die `Diskpart`-spezifischen Befehle ohne jegliche Interaktion ausgeführt werden:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Diskpart /s CreateLogicalDisc.txt > Result.txt"

Set oShell = Nothing
```

Listing 304: 2_03_011.vbs

72 Partitionen aktivieren

Das Computersystem wird stets von einer aktiven, primären Partition gestartet. Wenn Sie also eine weitere primäre Partition eingerichtet haben und vielleicht zu Test- oder Übungszwecken eine Parallelinstallation von Windows Server 2003 vornehmen möchten, so müssen Sie die primäre Partition, die Sie nachträglich eingerichtet haben, zunächst als *Aktiv* markieren. Denn anderenfalls werden die Startdateien der neuen Windows-Installation auf der momentanen Partition, die als *Aktiv* gekennzeichnet ist, eingerichtet.

Beim Aktivieren einer primären Partition gilt zu berücksichtigen, dass immer nur eine der primären Partitionen (es können max. vier sein) einer Festplatte als *Aktiv* markiert sein kann. Sie können zwar weitere Festplatten einbauen und auf jeder Festplatte eine primäre Partition als *Aktiv* kennzeichnen, allerdings wird das Betriebssystem immer versuchen, von der aktiven Partition der ersten Festplatte zu booten.

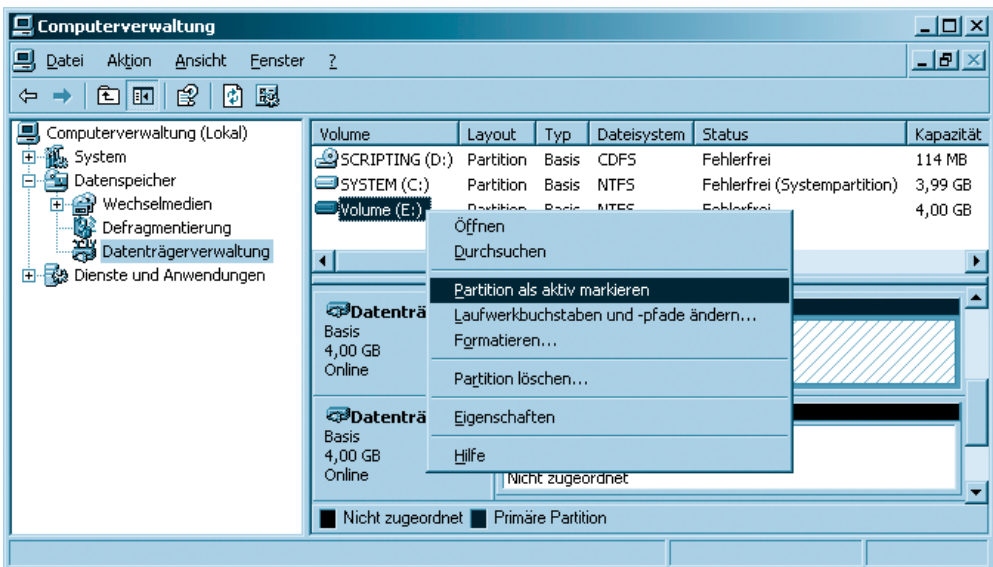


Abbildung 121: Eine primäre Partition aktivieren

Um eine primäre Partition unter grafischer Oberfläche als *Aktiv* zu markieren, rufen Sie zunächst die DATENTRÄGERVERWALTUNG auf. Wählen Sie hierzu aus dem Kontextmenü des ARBEITSPLATZ-Symbols VERWALTEN aus, womit Sie in die COMPUTERVERWALTUNG gelangen, und markieren Sie in der COMPUTERVERWALTUNG – unterhalb von DATENSPEICHER – den Eintrag DATENTRÄGERVERWALTUNG.

Im unteren Bereich der rechten Fensterhälfte sehen Sie nun alle Datenträger und die jeweiligen Partitionen (und ggf. die Volumes und logischen Laufwerke). Markieren Sie die primäre Partition, die Sie aktivieren möchten, und wählen Sie im Kontextmenü den Eintrag PARTITION ALS AKTIV MARKIEREN. Starten Sie das Computersystem neu, um von dieser Partition zu booten.

Die Lösung per Eingabeaufforderung

Wenn Sie eine primäre Partition von der Eingabeaufforderung aus als *Aktiv* markieren möchten, so verwenden Sie wieder den `Diskpart`-Befehl. Geben Sie zunächst `Diskpart` ein, um in die `Diskpart`-eigene Eingabeaufforderung zu gelangen, und selektieren Sie mit `Select Disk=x` die Festplatte, deren primäre Partition Sie auf *Aktiv* setzen möchten (wobei Sie das `x` durch die Festplattennummer ersetzen). Anschließend setzen Sie den Fokus mit `Select Partition=x` (das `x` ersetzen Sie durch die Partitionsnummer) den Fokus auf die zu aktivierende Partition und geben `Active` ein, worauf die Partition als *Aktiv* markiert wird. Danach können Sie die `Diskpart`-Eingabeaufforderung mit `Exit` wieder verlassen.

Um das Aktivieren einer primären Partition ohne Interaktion mit der `Diskpart`-eigenen Eingabeaufforderung vorzunehmen, rufen Sie den `Diskpart`-Befehl mit dem Parameter `/s` auf und übergeben das `Diskpart`-Skript, das alle erforderlichen Befehle zum Aktivieren einer Partition enthält. Die nachfolgende `Diskpart`-Befehlszeile verwendet beispielsweise das Skript *Activate.txt* und schreibt das Ergebnis der Befehlsausführung per Ausgabeumleitung in die Datei *Result.txt*:

```
@Echo off
Diskpart /s Activate.txt > Result.txt
```

Listing 305: 2_03_012.bat

Nachfolgend finden Sie den Inhalt des `Diskpart`-Skriptes mit der Bezeichnung *Activate.txt*, das alle erforderlichen Befehle zum Selektieren der Festplatte und der Partition, wie auch für das Aktivieren der Partition, enthält:

```
Select Disk=0
Select Partition=2
Active
```

Listing 306: Activate.txt

Die Lösung mit VBScript

Um das Aktivieren einer primären Partition aus einem VBScript heraus vorzunehmen, verwenden Sie die `Run`-Methode des `Shell`-Objektes und führen – wie auch in der Eingabeaufforderung – den `Diskpart`-Befehl unter Angabe des `Diskpart`-Skriptes aus.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
```

Listing 307: 2_03_012.vbs

```
oShell.Run "%comspec% /c Diskpart /s Activate.txt > Result.txt"
```

```
Set oShell = Nothing
```

Listing 307: 2_03_012.vbs (Forts.)

73 Volumes auf dynamischen Datenträgern erstellen

Die Einteilung eines dynamischen Datenträgers erfolgt nicht durch Anlegen von Partitionen und logischen Laufwerken, sondern durch Erstellen so genannter Volumes, die im Gegensatz zu Partitionen und logischen Laufwerken in einer Datenträgerdatenbank (und nicht mit Hilfe der Partitionstabelle) verwaltet werden.

Beim Anlegen eines Volumes auf einem dynamischen Datenträger gilt es, dabei zwischen den folgenden Volume-Typen zu unterscheiden:

- ▶ **Einfach:** Ein einfaches Volume wird zunächst auf einem einzelnen dynamischen Datenträger eingerichtet und ist vergleichbar mit einem logischen Laufwerk. Mit der Einrichtung weiterer dynamischer Datenträger können einfache Volumes auch über mehrere Datenträger erweitert oder auf einen anderen dynamischen Datenträger gespiegelt werden.
- ▶ **Übergreifend:** Ein übergreifendes Volume wird zunächst auf einem einzelnen dynamischen Datenträger eingerichtet und kann dann über weitere dynamische Datenträger erweitert werden. Mit diesem Volume-Typ können Sie also ein Laufwerk einrichten, das sich über mehrere Datenträger hinweg erstreckt.
- ▶ **Stripeset:** Der Volume-Typ Stripeset dient der Erhöhung der Zugriffsgeschwindigkeit und erfordert den Einsatz von mindestens zwei dynamischen Datenträgern. Die Daten werden dabei in Datenblöcke aufgeteilt und diese auf beide Datenträger verteilt, wodurch sich eine Performance-Steigerung erzielen lässt. Diese Konfiguration ist jedoch nicht fehlertolerant und führt immer zu einem Fehler, wenn einer der beiden Datenträger ausfällt.
- ▶ **Gespiegelt:** Ein gespiegeltes Volume erfordert den Einsatz von mindestens zwei dynamischen Datenträgern, wobei der gesamte Speicherbereich des ersten Datenträgers in einer 1:1-Kopie auf dem zweiten Datenträger abgebildet (gespiegelt) wird. Dieser Volume-Typ bietet eine gewisse Ausfallsicherheit, da beim Ausfall eines Datenträgers automatisch der zweite Datenträger benutzt wird, ohne dass es hierbei zu einer Unterbrechung während der Betriebssystem-Ausführung kommt.
- ▶ **RAID5:** Der Volume-Typ RAID5 erlaubt das Einrichten einer fehlertoleranten Festplattenkonfiguration, wofür mindestens drei dynamische Datenträger benötigt werden. Mit RAID5 werden Daten und Paritätsinformationen abwechselnd auf drei Datenträger geschrieben. Fällt einer der Datenträger aus, so kann dieser aufgrund der Daten und Paritätsinformationen der anderen beiden Datenträger sehr schnell wiederhergestellt werden, ohne dass es hierbei zu einer Unterbrechung des Serverbetriebes kommt.

Um nun ein solches Volume einzurichten, nehmen Sie unter grafischer Oberfläche wieder die Datenträgerverwaltung zu Hilfe. Öffnen Sie das Kontextmenü des Arbeitsplatz-Symbols, und wählen Sie **Verwalten** aus, um in die Computerverwaltung zu gelangen. Dort erweitern Sie den Abschnitt **Datenspeicher** und markieren unterhalb davon **Datenträgerverwaltung**.

Im unteren Teil der rechten Fensterhälfte sehen Sie alle verfügbaren Datenträger. Markieren Sie den freien Speicherbereich (dieser ist mit NICHT ZUGEORDNET gekennzeichnet) des dynamischen Datenträgers, auf dem Sie ein neues Volume erstellen möchten, und wählen Sie im Kontextmenü den Eintrag **NEUES VOLUME...** aus. Daraufhin wird der **ASSISTENT ZUM ERSTELLEN NEUER VOLUMES** gestartet. Schließen Sie den Willkommensbildschirm mit einem Klick auf **WEITER**, und wählen Sie im nächsten Schritt den gewünschten Volume-Typ (*Einfach*, *Übergreifend*, *Stripeset*, *Gespiegelt* oder *RAID5*) aus.

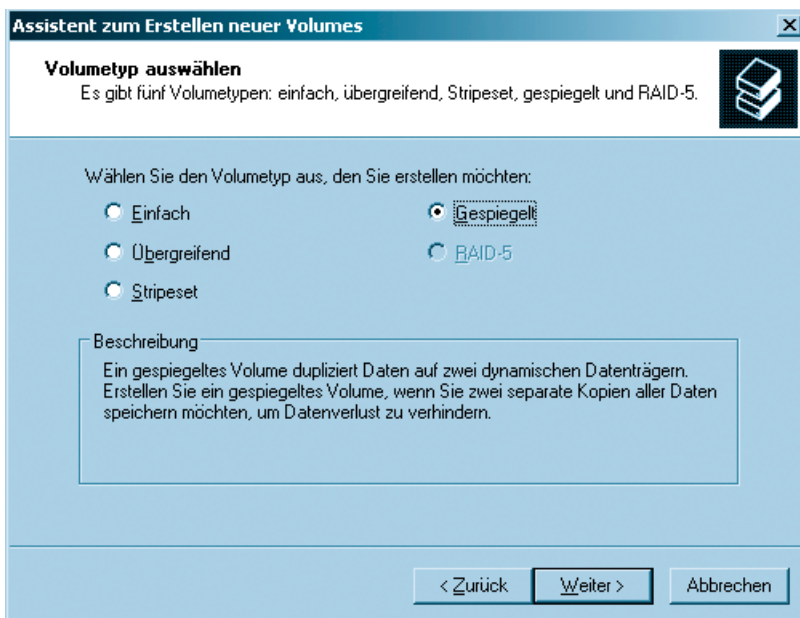


Abbildung 122: Ein gespiegeltes Volume einrichten

Mit einem Klick auf **WEITER** gelangen Sie zur Dialogseite **DATENTRÄGER AUSWÄHLEN**, worüber Sie nun die dynamischen Datenträger selektieren müssen. Standardmäßig ist bereits der Datenträger ausgewählt, den Sie zu Beginn in der **DATENTRÄGERVERWALTUNG** markiert haben. Wenn Sie in der vorherigen Dialogseite den Volume-Typ *Übergreifend*, *Stripeset* oder *RAID5* ausgewählt haben, so müssen Sie nun weitere dynamische Datenträger selektieren, da diese Konfigurationen nicht auf einem einzelnen Datenträger erstellt werden können. Wählen Sie also die entsprechenden Datenträger aus, und klicken Sie auf **WEITER**.

Im nächsten Schritt können Sie einen Laufwerkbuchstaben festlegen oder das Laufwerk auch in einem leeren Ordner eines bereits vorhandenen Laufwerks bereitstellen und dem Laufwerk damit einen Laufwerkpfad zuweisen. Markieren Sie die gewünschte Option, und klicken Sie auf WEITER, woraufhin Sie zur Dialogseite VOLUME FORMATIEREN gelangen und die Formatierungsoptionen (*Dateisystem*, *Größe der Zuordnungseinheit* und *Volume-Bezeichnung*) festlegen können. Falls Sie das Laufwerk zu einem späteren Zeitpunkt formatieren möchten, so aktivieren Sie die Option DIESES VOLUME NICHT FORMATIEREN und klicken anschließend auf WEITER. Im letzten Schritt werden noch einmal die zuvor ausgewählten Optionen angezeigt. Klicken Sie auf FERTIG STELLEN, um das Volume anzulegen, oder wählen Sie ZURÜCK, wenn Sie bestimmte Optionen ändern möchten.

Die Lösung per Eingabeaufforderung

Mit dem Diskpart-Befehl lassen sich neue Volumes auch von der Eingabeaufforderung aus ohne großen Aufwand erstellen. Wenn Sie z.B. auf einer zweiten Festplatte (die bereits zu einem dynamischen Datenträger konvertiert wurde) ein einfaches Volume einrichten möchten, so wechseln Sie durch Eingabe von Diskpart zunächst in die Diskpart-eigene Eingabeaufforderung.

Anschließend setzen Sie den Fokus auf den Datenträger, auf dem Sie das Volume einrichten möchten. Hierzu geben Sie `Select Disk=x` ein, wobei Sie das `x` durch die Nummer des Datenträgers (0 = erster Datenträger, 1 = zweiter Datenträger usw.) ersetzen. Eine zweite Festplatte selektieren Sie demnach mit `Select Disk=1`. Wenn Sie den Datenträger selektiert haben, dann geben Sie den Befehl `Create Volume Simple` ein, um ein einfaches Volume einzurichten, dessen Größe der Gesamtkapazität des Datenträgers entspricht. Um nur einen Teil der Datenträgerkapazität für das Volume zu verwenden, fügen Sie dem `Create Volume Simple`-Befehl den Parameter `Size=x` an und ersetzen `x` durch die gewünschte Größe (in MByte). Nachdem das Volume erstellt wurde, verlassen Sie die Diskpart-eigene Eingabeaufforderung durch Eingabe von `Exit`.

Die gleiche Aufgabe lässt sich natürlich auch ohne Einsatz der Diskpart-eigenen Eingabeaufforderung durchführen. Hierzu erstellen Sie einfach eine Textdatei und nehmen alle Diskpart-spezifischen Befehle zur Einrichtung des Volumes auf. Dann führen Sie den Diskpart-Befehl mit dem Parameter `/s` aus und übergeben die Textdatei, in der die einzelnen Befehle hinterlegt sind.

Die nachfolgende Diskpart-Befehlszeile führt beispielsweise die Befehle aus der Textdatei `CreateVolume.txt` aus und schreibt das Ergebnis der Diskpart-Ausführung per Ausgabeumleitung in die Datei `Result.txt`:

```
@Echo off
Diskpart /s CreateVolume.txt > Result.txt
```

Listing 308: 2_03_013.bat

Das Diskpart-Skript *CreateVolume.txt* enthält die folgenden Befehlszeilen, womit zunächst der Datenträger selektiert und dann das Volume erstellt wird:

```
Select Disk=1  
Create Volume Simple
```

Listing 309: CreateVolume.txt

Wenn Sie statt eines einfachen Volumes ein Volume vom Typ *Übergreifend* erstellen möchten, so selektieren Sie zunächst den ersten Datenträger des künftigen Datenträgerverbundes und erstellen mit *Create Volume Simple* ein einfaches Volume. Dann erweitern Sie das Laufwerk durch Eingabe von *Extend Disk=x*, wobei Sie das *x* durch die Nummer des nächsten Verbunddatenträgers ersetzen. Um beispielsweise auf dem Datenträger 2 ein einfaches Volume zu erstellen und das Laufwerk durch Einbeziehen des Datenträgers 3 zu erweitern, können Sie das folgende Diskpart-Skript verwenden:

```
Select Disk=1  
Extend Disk=2
```

Listing 310: CreateExtendedVolume.txt

Ein Stripeset-Volume ist ebenfalls schnell erstellt. Wenn Sie z.B. auf den Datenträgern 2 und 3 ein Stripeset-Volume einrichten möchten, so können Sie in diesem Fall auf die Selektion des Datenträgers verzichten, da Sie nur den Befehl *Create Volume Stripe Disk=1,2* eingeben müssen.

```
Create Volume Stripe Disk=1,2
```

Listing 311: CreateStripesetVolume.txt

Mit dem Diskpart-Befehl lassen sich natürlich auch gespiegelte Volumes einrichten. Wenn Sie z.B. den Datenträger 2 als primäres Volume einrichten und den Datenträgerinhalt auf Datenträger 3 spiegeln möchten, so selektieren Sie mit *Select Disk=1* den zweiten Datenträger, erstellen mit *Create Volume Simple* ein einfaches Volume und leiten mit *Add Disk=2* die Spiegelung ein.

```
Select Disk=1  
Add Disk=2
```

Listing 312: CreateMirror.txt

Und zu guter Letzt können Sie mit Diskpart natürlich auch eine RAID5-Konfiguration erstellen, wofür Sie allerdings mindestens drei dynamische Datenträger benöti-

gen. Wenn Sie z.B. eine RAID5-Konfiguration erstellen und die Datenträger 2, 3 und 4 dafür verwenden möchten, so geben Sie lediglich den Befehl `Create Volume RAID Disk=1,2,3` ein.

```
Create Volume RAID Disk=2,3,4
```

Listing 313: CreateRAID.txt

Die Lösung mit VBScript

VBScript stellt leider keine entsprechenden Möglichkeiten bereit, womit Sie auf einem dynamischen Datenträger ein Volume einrichten können. Dennoch lässt sich das Erstellen eines Volumes auch aus einem VBScript heraus starten, da Sie mit der `Run`-Methode des `Shell`-Objektes den `Diskpart`-Befehl der Eingabeaufforderung auch per VBScript aufrufen können.

Das nachfolgende Listing führt den `Diskpart`-Befehl unter Angabe eines `Diskpart`-Skriptes (*CreateVolume.txt*) aus, das die erforderlichen Befehlszeilen zum Anlegen eines einfachen Volumes auf einem dynamischen Datenträger (siehe *Lösung per Eingabeaufforderung*) enthält.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Diskpart /s CreateVolume.txt > Result.txt"

Set oShell = Nothing
```

Listing 314: 2_03_013.vbs

74 Partitionen, Volumes und logische Laufwerke löschen

Für den Fall, dass Sie zu Test- oder Übungszwecken Partitionen, Volumes oder logische Laufwerke eingerichtet haben und diese künftig nicht mehr benötigen, so können Sie die Aufteilung des Datenträgers aufheben, indem Sie die überflüssigen Partitionen, Volumes und logischen Laufwerke löschen. Bei Bedarf können Sie die Hauptpartition oder das Haupt-Volume danach auch erweitern und den frei gewordenen Speicherplatz mitbenutzen, ohne eine neue Partition oder ein neues Volume einrichten zu müssen (siehe *Lösung per Eingabeaufforderung*).

Achtung

Bevor Sie eine Datenträgeraufteilung aufheben, müssen Sie zunächst alle Daten, die sich auf zu löschenden Speicherbereichen befinden, sichern. Denn mit dem Löschen einer Partition, eines Volumes oder eines logischen Laufwerks werden alle Daten unwiderruflich gelöscht und können nicht mehr wiederhergestellt werden (zumindest nicht so ohne weiteres).

Um eine Partition, ein Volume oder ein logisches Laufwerk zu löschen, wechseln Sie zunächst in die DATENTRÄGERVERWALTUNG. Hierzu wählen Sie aus dem Kontextmenü des ARBEITSPLATZ-Symbols VERWALTEN aus, woraufhin die COMPUTERVERWALTUNG gestartet wird. In der COMPUTERVERWALTUNG erweitern Sie DATENSPEICHER und markieren unterhalb davon DATENTRÄGERVERWALTUNG.

Im unteren Bereich der rechten Fensterhälfte sehen Sie nun alle Datenträger und deren Aufteilung in Partitionen, Volumes und logische Laufwerke. Markieren Sie das schraffierte Feld, das den zu löschenden Speicherbereich darstellt, und wählen Sie im Kontextmenü den Eintrag PARTITION LÖSCHEN... bzw. LOGISCHES LAUFWERK LÖSCHEN... aus. Sofern sich keine geöffneten Dateien auf der Partition befinden, wird der Speicherbereich daraufhin wieder freigegeben und mit NICHT ZUGEORDNET gekennzeichnet. Anderenfalls – wenn Dateien in Benutzung sind – erhalten Sie einen Hinweis, dass die Partition gerade in Verwendung ist. Sie müssen das Löschen der Partition und damit auch das etwas »unsanfte« Schließen der Dateien von Hand bestätigen.

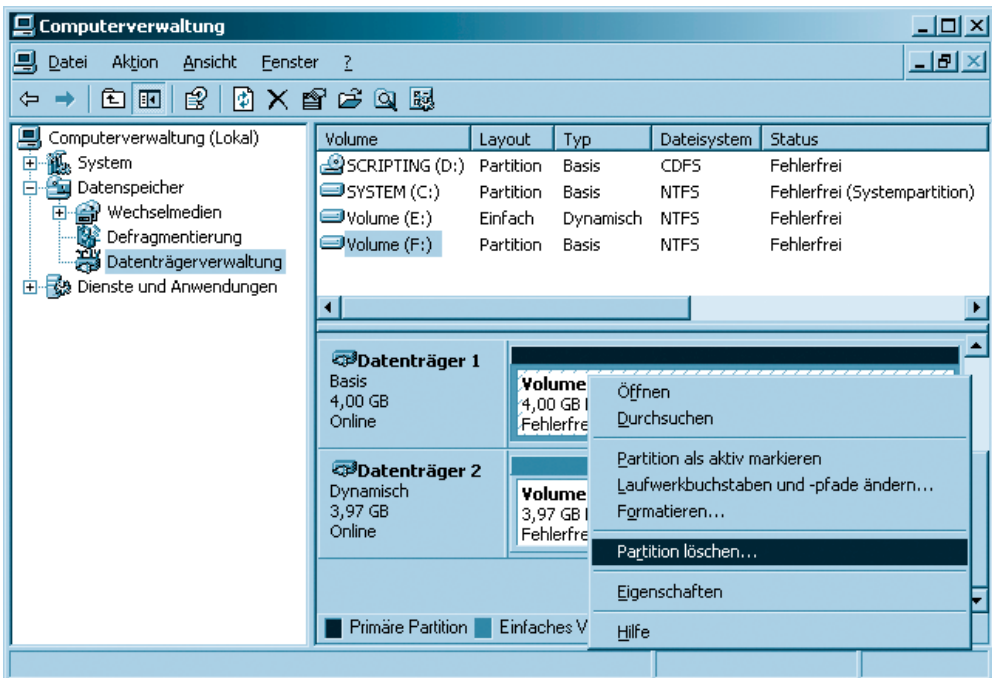


Abbildung 123: Eine primäre Partition wieder löschen

Die Lösung per Eingabeaufforderung

Um Partitionen, Volumes und logische Laufwerke von der Eingabeaufforderung aus zu löschen, können Sie den `Diskpart`-Befehl verwenden, der die befehszeilenorientierte Datenträgerverwaltung von Windows Server 2003 darstellt.

Wenn Sie z.B. auf einer zweiten Festplatte zwei primäre Partitionen eingerichtet haben und die zweite Partition nun löschen möchten, so geben Sie in der Eingabeaufforderung zunächst `Diskpart` ein, um in die `Diskpart`-eigene Eingabeaufforderung zu gelangen. Anschließend setzen Sie den Fokus auf das Objekt, das Sie löschen möchten (in diesem Fall eine Partition). Selektieren Sie mit `Select Disk=x` die Festplatte, auf der sich die zu löschende Partition befindet, wobei Sie `x` durch die Nummer der Festplatte ersetzen (0 = erste Festplatte, 1 = zweite Festplatte ...), und setzen Sie den Fokus dann mit `Select Partition=2` auf die zweite Partition (die erste Partition hat den Wert 1, die zweite Partition den Wert 2 usw.). Dann geben Sie `Delete Partition` ein, um das soeben ausgewählte Objekt zu löschen, und beenden die `Diskpart`-Sitzung durch Eingabe von `Exit`.

Gleiches lässt sich natürlich auch aus einer Batch-Datei heraus und ohne Interaktion mit der `Diskpart`-Eingabeaufforderung erledigen. Wenn Sie die zweite Partition (Partition 2) auf der zweiten Festplatte (Disk 1) löschen möchten, so wenden Sie den `Diskpart`-Befehl unter Angabe des Parameters `/s` an, übergeben das `Diskpart`-Skript und leiten das Ergebnis der `Diskpart`-Befehlsausführung bei Bedarf in eine Ausgabe-datei um.

```
@Echo off
Diskpart /s DeletePartition.txt > Result.txt
```

Listing 315: 2_03_014.bat

Das `Diskpart`-Skript muss dabei alle Befehle enthalten, die Sie auch in der `Diskpart`-Eingabeaufforderung eingeben müssen. Das nachfolgende Skript zeigt ein solches Beispiel: Mit der ersten Befehlszeile wird die Festplatte selektiert, mit der zweiten Befehlszeile die Partition, und mit der dritten Befehlszeile wird die Partition gelöscht.

```
Select Disk=1
Select Partition=2
Delete Partition
```

Listing 316: DeletePartition.txt

Volumes und logische Laufwerke werden ebenfalls mit dem `Diskpart`-spezifischen Befehl `Delete Partition` gelöscht, wobei Sie das zu löschende Volume oder Laufwerk zuvor mit `Select Partition=x` (das `x` ersetzen Sie dabei durch die Nummer der Partition) markieren müssen.

Die Lösung mit VBScript

Da VBScript leider keine Möglichkeiten bietet, womit Sie Partitionen, Volumes und logische Laufwerke löschen können, bleibt nur der Einsatz der `Run`-Methode des `Shell`-Objektes, womit Sie den `Diskpart`-Befehl der Eingabeaufforderung auch aus einem VBScript heraus aufrufen können.

Das nachfolgende Listing führt den `Diskpart`-Befehl mit dem Parameter `/s` aus und wendet die `Diskpart`-spezifischen Befehle an, die in der Datei *DeletePartition.txt* (siehe *Lösung per Eingabeaufforderung*) hinterlegt sind. Das Ergebnis der `Diskpart`-Befehlsausführung wird per Ausgabeumleitung in die Datei *Result.txt* geschrieben:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Diskpart /s DeletePartition.txt > Result.txt"

Set oShell = Nothing
```

Listing 317: 2_03_014.vbs

75 Partitionen, Volumes und logische Laufwerke formatieren

Wenn Sie die Formatierung beim Einrichten einer Partition, eines Volumes oder eines logischen Laufwerks übersprungen haben und dies nun nachholen möchten oder wenn Sie ein Laufwerk aus sonstigen Gründen neu formatieren möchten, so können Sie die Formatierung direkt im WINDOWS EXPLORER vornehmen, indem Sie das Kontextmenü des Laufwerks öffnen und **FORMATIEREN...** auswählen.

Der besseren Übersicht wegen sollten Sie die Formatierung aber mit Hilfe der DATENTRÄGERVERWALTUNG durchführen, da die DATENTRÄGERVERWALTUNG alle Datenträger und dazugehörigen Partitionen, Volumes und logische Laufwerke sehr übersichtlich darstellt und ein versehentliches Formatieren eines falschen Laufwerks dadurch leichter vermieden werden kann.

Um nun ein Laufwerk zu formatieren, starten Sie zunächst die DATENTRÄGERVERWALTUNG. Hierzu öffnen Sie das Kontextmenü des ARBEITSPLATZ-Symbols und wählen **VERWALTEN** aus, woraufhin die COMPUTERVERWALTUNG gestartet wird. In der COMPUTERVERWALTUNG erweitern Sie den Abschnitt **DATENSPEICHER** und markieren – unterhalb von **DATENSPEICHER** – den Eintrag **DATENTRÄGERVERWALTUNG**. Im unteren Bereich der rechten Fensterhälfte werden nun alle Datenträger und die jeweils eingerichteten Partitionen, Volumes und logischen Laufwerke angezeigt. Markieren Sie das Laufwerk, das Sie formatieren möchten, und rufen Sie aus dem Kontextmenü **FORMATIEREN...** auf.

Sie gelangen daraufhin in ein Dialogfenster, worüber Sie die Formatierungsoptionen festlegen können. Im Feld VOLUME-BEZEICHNUNG können Sie eine Bezeichnung für das Laufwerk (z.B. *System*, *Daten*, *Backup* ...) eingeben, womit das Laufwerk später leichter zu identifizieren ist, da die Datenträgerbezeichnungen in der Regel in den Laufwerksübersichten angezeigt werden. Im Feld ZU VERWENDENDENES DATEISYSTEM können Sie – sofern es sich nicht um ein Volume auf einem dynamischen Datenträger handelt – zwischen *FAT*, *FAT32* und *NTFS* wählen (die wesentlichen Unterschiede dieser Dateisysteme sind in der nachfolgenden Tabelle aufgeführt). Ein Volume, das auf einem dynamischen Datenträger eingerichtet wurde, kann dagegen nur mit *NTFS* formatiert werden, andere Dateisysteme werden nicht unterstützt.

Dateisystem	Beschreibung
FAT	Das FAT-Dateisystem kommt auf kleinen Datenträgern (bis max. 4 GByte) zum Einsatz und kann von allen MS DOS-, Windows 3.x-, Windows 9.x-, Windows NT-, Windows 2000-, Windows XP- und Windows 2003-Systemen gelesen werden. FAT enthält keine Berechtigungs-, Komprimierungs- und Verschlüsselungsfunktionen, wie dies bei NTFS der Fall ist.
FAT32	Das FAT32-Dateisystem kommt auf Datenträgern mit bis zu 32 GByte Speicherkapazität zum Einsatz und kann von allen Windows 95 Release 2-, Windows 98-, Windows 2000-, Windows XP- und Windows 2003-Systemen gelesen werden. FAT32 enthält ebenfalls keine Berechtigungs-, Verschlüsselungs- und Komprimierungsfunktionen. Für Windows NT 4 existieren Zusatzprogramme, die einen Zugriff erlauben.
NTFS	Das Dateisystem NTFS kann ab einer Datenträgergröße von 10 MByte eingesetzt werden und eignet sich insbesondere für große Datenträger im GByte- und TByte-Bereich. Das NTFS-Dateisystem kann von Systemen mit Windows NT 4.0 mit SP 4 und höheren Betriebssystemversionen gelesen werden und enthält im Gegensatz zu FAT und FAT32 Berechtigungs-, Verschlüsselungs- und Komprimierungsfunktionen, womit die Daten besser vor fremden Zugriffen geschützt werden können. Das NTFS-Dateisystem findet vorwiegend in Windows NT-, Windows 2000-, Windows XP- und Windows 2003-Umgebungen Einsatz und ist beispielsweise Voraussetzung für die Einrichtung von Active Directory.

Tabelle 46: Die wesentlichen Merkmale der Dateisysteme

Des Weiteren können Sie die Größe der Zuordnungseinheit auf dem Datenträger (Cluster-Größe) festlegen. Kleine Zuordnungseinheiten haben den Vorteil, dass die Defragmentierung (siehe Datenträger defragmentieren auf Seite 418) gering bleibt, wobei sich dadurch aber auch der Schreibvorgang verzögern kann, da die Daten in wesentlich mehr Fragmente aufgeteilt werden müssen. Im Idealfall übernehmen Sie die Standardeinstellung, womit die Cluster-Größe vor dem Formatierungsprozess anhand der Festplattengröße automatisch ermittelt wird.

Sofern das Laufwerk bereits formatiert wurde und der Bereinigung wegen nur neu formatiert werden soll, so können Sie den Formatierungsprozess durch Auswahl der Option **SCHNELLFORMATIERUNG DURCHFÜHREN** um ein Vielfaches beschleunigen, da mit Auswahl dieser Option keine Überprüfung auf fehlerhafte Sektoren durchgeführt wird.

Und zu guter Letzt können Sie mit der Option **KOMPRIMIERUNG FÜR DATEIEN UND ORDNER AKTIVIEREN** bestimmen, ob das Laufwerk komprimiert werden soll, womit Sie zwar ein gutes Stück an wertvollem Speicherplatz einsparen können, dafür aber auch etwas an Zugriffsgeschwindigkeiten einbüßen müssen, da alle Dateien vor dem Öffnen erst entkomprimiert und nach dem Schließen wieder komprimiert werden müssen.

Wenn Sie die Formatierungsoptionen entsprechend ausgewählt haben, klicken Sie auf **OK**, um den Formatierungsprozess mit den angegebenen Optionen zu starten.

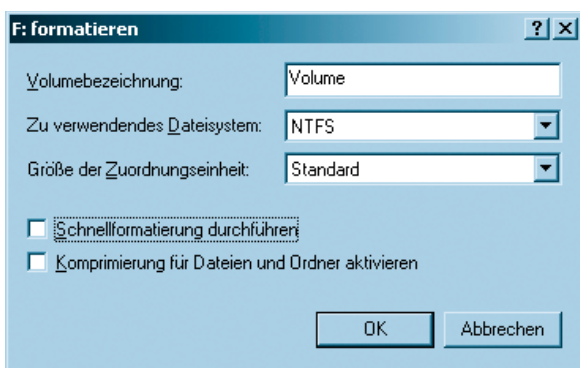


Abbildung 124: Eine Partition formatieren

Die Lösung per Eingabeaufforderung

Um eine Partition, ein Volume oder ein logisches Laufwerk von der Eingabeaufforderung aus zu formatieren, verwenden Sie den **Format**-Befehl, der standardmäßig im Befehlsumfang von Windows Server 2003 enthalten ist.

Der **Format**-Befehl erwartet den Laufwerksbuchstaben für das Laufwerk, das Sie formatieren möchten, als Parameter. Mit dem zusätzlichen Parameter **/FS:** (gefolgt von **FAT**, **FAT32** oder **NTFS**) können Sie das zu verwendende Dateisystem bestimmen. Und wenn Sie mit der Formatierung auch eine neue Datenträgerbezeichnung festlegen möchten, so können Sie die neue Bezeichnung mit dem Parameter **/V** übergeben.

Falls das Laufwerk bereits eine Datenträgerbezeichnung besitzt, so müssen Sie die Bezeichnung nach Eingabe des Formatbefehls eingeben, um den Formatierungsprozess zu bestätigen. Anschließend werden Sie noch einmal darauf aufmerksam gemacht, dass mit der Formatierung alle Daten, die auf dem Laufwerk gespeichert sind, verloren gehen. Um diese Interaktion zu vermeiden, können Sie die Formatierung mit dem Parameter **/Y** erzwingen, woraufhin die Aufforderung zur manuellen Bestätigung unterdrückt wird.

Um beispielsweise das Laufwerk *F* mit *NTFS* zu formatieren und die Datenträgerbezeichnung *Backup* zu übergeben, verwenden Sie folgende `Format`-Befehlszeile:

```
@Echo off
Format F: /FS:NTFS /V:Backup /Y
```

Listing 318: 2_03_015.bat

Die Lösung mit VBScript

Mit der `Format`-Methode der WMI-Klasse `Win32_Volume` können Sie die Formatierung einer Partition, eines Volumes oder eines logischen Laufwerks auch per VBScript durchführen. Hierzu führen Sie eine `ExecQuery`-Abfrage gegen `Win32_Volume` durch und selektieren mit Hilfe der `Where`-Klausel das entsprechende Laufwerk. Da die `ExecQuery`-Abfrage das Ergebnis in Form einer Kollektion zurückliefert, setzen Sie eine `For Each`-Schleife ein und wenden die `Format`-Methode gegen die von `ExecQuery` zurückgelieferten Elemente an. Der `Format`-Methode übergeben Sie dabei zumindest das Dateisystem (*FAT*, *FAT32*, *NTFS*) als Parameter.

Die folgenden Parameter können mit dem Aufruf der `Format`-Methode übergeben werden, wobei die angegebene Reihenfolge einzuhalten ist.

Parameter	Beschreibung
Filesystem	Das Dateisystem, das verwendet werden soll. Mögliche Werte sind <i>FAT</i> , <i>FAT32</i> und <i>NTFS</i> .
Quickformat (Optional)	Legt fest, ob die Schnellformatierung zugelassen (<i>True</i>) werden soll oder nicht (<i>False</i>).
Clustersize (Optional)	Die Cluster-Größe, die verwendet werden soll (z.B. <i>4096</i>).
Label (Optional)	Die Datenträgerbezeichnung (z.B. <i>Daten</i>)
EnableCompression (Optional)	Bestimmt die Komprimierung, wobei dieser Parameter allerdings nicht implementiert ist und daher auch nicht zur Verfügung steht.

Tabelle 47: Die Parameter der `Format`-Methode

```
Option Explicit

Dim oWMI
Dim colDrive
Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:\\." & strComputer & "\root\cimv2")
```

Listing 319: 2_03_015.vbs

```
Set colDrive = oWMI.ExecQuery ("Select * from Win32_Volume Where _  
    Name = 'F:\\'")  
For Each objItem in colDrive  
    Err = objItem.Format("NTFS")  
Next  
  
Set oWMI = Nothing
```

Listing 319: 2_03_015.vbs (Forts.)

76 Laufwerksbuchstaben und -pfade festlegen

Einem Laufwerk (Disketten-, CD-/DVD-Laufwerk, primäre Partitionen, logisches Laufwerk und dynamisches Volumes) ist in der Regel ein Laufwerksbuchstabe zugewiesen, worüber das Laufwerk angesprochen werden kann. Das Diskettenlaufwerk hat beispielsweise per Default den Laufwerksbuchstaben A, die Systempartition dagegen den Laufwerksbuchstaben C.

Wenn Sie nun weitere Festplatten eingebaut, Partitionen angelegt oder Volumes eingerichtet haben, so wird dem Laufwerk standardmäßig der nächst freie Laufwerksbuchstaben zugewiesen. Diesen Laufwerksbuchstaben können Sie im Nachhinein aber problemlos ändern (oder wahlweise auch ganz entfernen, womit das Laufwerk dann nicht mehr in der Laufwerksübersicht angezeigt wird) und die Anordnung der Laufwerke damit selbst bestimmen. Wenn Sie z.B. ein neues Volume eingerichtet haben, das den Laufwerksbuchstaben F besitzt, so können Sie dem Volume jederzeit einen anderen freien Laufwerksbuchstaben zuordnen und damit die Reihenfolge, in der das Laufwerk angezeigt wird, verändern.

Diese Aufgabe nehmen Sie unter grafischer Oberfläche mit Hilfe der DATENTRÄGERVERWALTUNG vor. Wählen Sie hierzu im Kontextmenü des ARBEITSPLATZ-Symbols den Eintrag VERWALTEN aus, woraufhin Sie in die COMPUTERVERWALTUNG gelangen. Dort erweitern Sie den Abschnitt DATENSPEICHER und markieren unterhalb davon DATENTRÄGERVERWALTUNG. Im unteren Bereich der rechten Fensterhälfte markieren Sie das entsprechende Laufwerk und wählen im Kontextmenü den Eintrag LAUFWERKBUCHSTABE UND -PFAD ÄNDERN... aus, worüber das gleichnamige Dialogfenster geöffnet wird. Klicken Sie auf ÄNDERN, wenn Sie einen anderen Laufwerksbuchstaben zuweisen möchten, oder betätigen Sie die Schaltfläche HINZUFÜGEN, wenn dem Laufwerk noch kein Laufwerksbuchstabe zugeordnet wurde. Wählen Sie dann den gewünschten Laufwerksbuchstaben aus, und übernehmen Sie die Einstellung mit OK.

Eine weitere Möglichkeit, ein Laufwerk anzusprechen, bieten so genannte Laufwerkspfade. Denn mit einem Laufwerkspfad können Sie ein neues Laufwerk in die bestehende Ordnerstruktur eines bereits vorhandenen NTFS-Laufwerks einbinden, womit Sie auf recht einfache Art und Weise die Speicherkapazität eines vorhandenen NTFS-Laufwerks vergrößern können, ohne dabei die Laufwerkstruktur zu verändern. Dies hat zusätzlich den Vorteil, dass somit der maximale Datenumfang eines Ordners fest-

gelegt ist. Damit kann beispielsweise verhindert werden, dass ein Temp-Ordner den kompletten freien Platz eines Laufwerkes konsumiert.

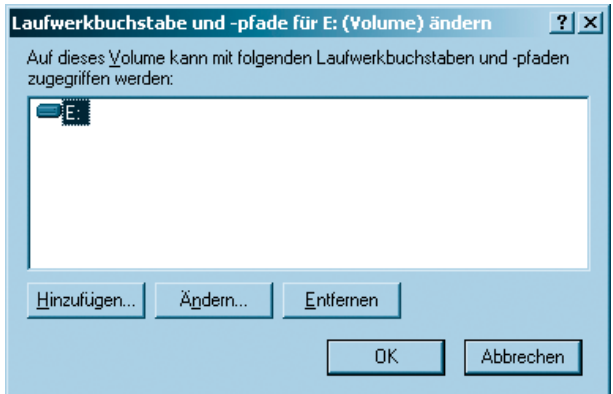


Abbildung 125: Einen Laufwerksbuchstaben festlegen

Wenn z.B. ein vorhandenes Laufwerk mehr als ausgelastet ist, so richten Sie einfach ein neues Laufwerk ein, binden das neue Laufwerk an einen leeren Ordner auf dem bisherigen Laufwerk (dieses muss mit NTFS formatiert sein) und weisen dem neuen Laufwerk damit einen Pfad zu, worüber das neue Laufwerk dann angesprochen werden kann. Die nachfolgende Abbildung zeigt, wie ein Laufwerk, das als Ordner auf einem anderen Laufwerk eingebunden ist, im WINDOWS EXPLORER angezeigt wird:

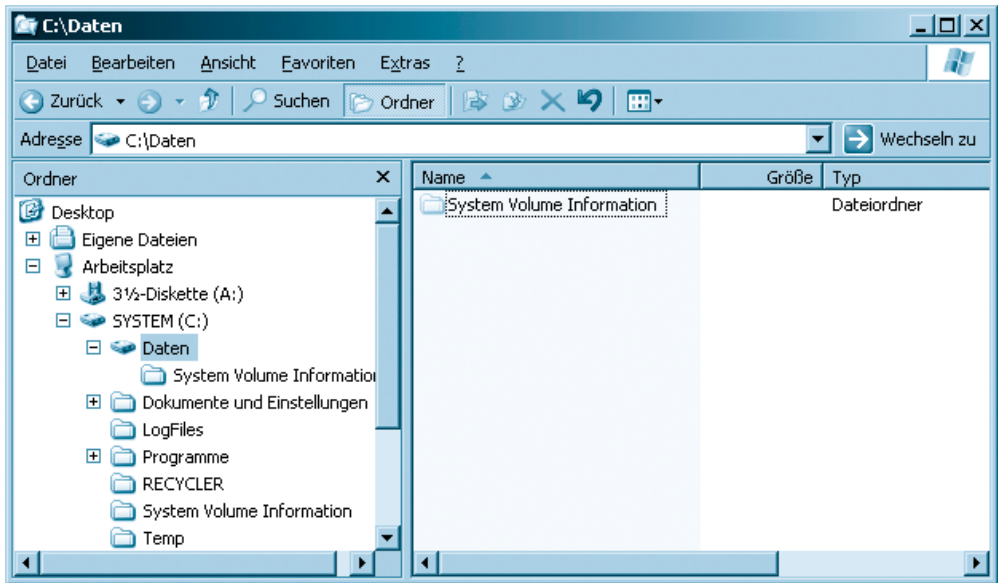


Abbildung 126: Ein Laufwerk wurde als Ordner eingebunden.

Um einen Laufwerkpfad für ein neues NTFS-Laufwerk zu bestimmen, rufen Sie wieder die DATENTRÄGERVERWALTUNG von Windows Server 2003 auf und markieren das NTFS-Laufwerk, für das Sie den Laufwerkpfad bestimmen möchten. Wählen Sie im Kontextmenü den Eintrag LAUFWERKBUCHSTABEN UND -PFADE ÄNDERN... aus, und klicken Sie auf HINZUFÜGEN. Aktivieren Sie die Option IN FOLGENDEM LEEREN NTFS-ORDNER BEREITSTELLEN:, und klicken Sie auf DURCHSUCHEN, um einen vorhandenen leeren Ordner auszuwählen oder einen neuen Ordner zu erstellen (z.B. C:\Daten). Markieren Sie dann den Ordner, und übernehmen Sie die Einstellung mit OK, woraufhin der Pfad zu dem neuen Laufwerk im Feld IN FOLGENDEM LEEREN NTFS-ORDNER BEREITSTELLEN: angezeigt wird. Schließen Sie das Dialogfenster LAUFWERKBUCHSTABEN ODER -PFADE ÄNDERN und auch die DATENTRÄGERVERWALTUNG.

Die Lösung per Eingabeaufforderung

Laufwerkbuchstaben und -pfade lassen sich natürlich auch von der Eingabeaufforderung aus festlegen und ändern. Hierfür verwenden Sie den Diskpart-Befehl, der fester Bestandteil des Befehlsumfangs von Windows Server 2003 ist.

Wenn Sie z.B. auf der Festplatte eine weitere Partition eingerichtet haben und der neuen Partition nun den Laufwerkbuchstaben X zuweisen möchten, so wechseln Sie durch Eingabe von Diskpart zunächst in die Diskpart-eigene Eingabeaufforderung und selektieren dann den Datenträger, auf dem sich die Partition befindet. Hierzu geben Sie `Select Disk=x` ein, wobei Sie das `x` durch die Nummer des Datenträgers (0 = erster Datenträger, 1 = zweiter Datenträger usw.) ersetzen. Anschließend setzen Sie den Fokus mit `Select Partition=x` auf die Partition, für die Sie den Laufwerkbuchstaben festlegen möchten, und ersetzen in diesem Fall das `x` durch die Nummer der Partition (1 = erste Partition, 2 = zweite Partition ...). Dann wenden Sie den Befehl `Assign Letter=X` an, womit Sie der Partition den Laufwerkbuchstaben X zuweisen, und beenden die Diskpart-eigene Eingabeaufforderung mit `Exit`.

Gleiches lässt sich durch Einsatz eines Diskpart-Skriptes auch automatisieren und damit ohne Interaktion mit der Diskpart-eigenen Eingabeaufforderung durchführen. Hierzu erstellen Sie lediglich eine Textdatei und nehmen die Diskpart-spezifischen Befehle auf, womit Sie zunächst den Datenträger selektieren, dann den Fokus auf die Partition setzen und zu guter Letzt mit `Assign Letter=X` den eigentlichen Laufwerkbuchstaben zuweisen. Anschließend führen Sie Diskpart mit dem Parameter `/s` aus und übergeben die Textdatei, in der die Diskpart-spezifischen Befehle hinterlegt sind.

```
@Echo off
Diskpart /s DriveLetter.txt > Result.txt
```

Listing 320: 2_03_016.bat

Nachfolgend finden Sie den Inhalt des Diskpart-Skriptes *DriveLetter.txt*, womit Sie zunächst den Datenträger 0 und die Partition 2 selektieren und dann mit `Assign Letter` den Laufwerkbuchstaben X zuweisen können:

```
Select Disk=0
Select Partition=2
Assign Letter=X
```

Listing 321: DriveLetter.txt

Auf ähnliche Art und Weise lassen sich von der Eingabeaufforderung aus auch Laufwerkpfade zuweisen. Wenn Sie beispielsweise die zweite Partition auf dem ersten Datenträger im Ordner `C:\Daten` der ersten Partition bereitstellen möchten, so wechseln Sie mit `Diskpart` wieder in die `Diskpart`-eigene Eingabeaufforderung und selektieren mit `Select Disk=0` zunächst den ersten Datenträger. Dann setzen Sie den Fokus mit `Select Partition=2` auf die zweite Partition und weisen mit `Assign Mount=C:\Daten` den Laufwerkpfad zu.

Wenn Sie diese Aufgabe ohne Interaktion mit der `Diskpart`-eigenen Eingabeaufforderung durchführen möchten, so rufen Sie `Diskpart` mit dem Parameter `/s` auf und übergeben den Namen des `Diskpart`-Skriptes, das folgenden Inhalt aufweisen muss:

```
Select Disk=0
Select Partition=2
Assign Mount=C:\Daten
```

Listing 322: DrivePath.txt

Die Lösung mit VBScript

Auch mit VBScript können Sie einem Laufwerk einen Laufwerkbuchstaben oder -pfad zuweisen. Hierzu stellen Sie eine Verbindung zur WMI-Klasse `Win32_Volume` her, selektieren das betreffende Laufwerk mit Hilfe der `Where`-Klausel, indem Sie den Wert der Eigenschaft `DriveLetter` (Laufwerkbuchstaben oder -pfad) oder `DeviceID` (Geräteerkennung) abfragen und weisen der Eigenschaft `DriveLetter` dann den neuen Laufwerkbuchstaben oder -pfad zu.

Wenn Sie beispielsweise dem Laufwerk `F` den neuen Laufwerkbuchstaben `X` zuweisen möchten, so führen Sie eine `ExecQuery`-Abfrage unter Verwendung der `Where`-Klausel durch und selektieren alle Datenträger, deren Laufwerkbuchstabe (`DriveLetter`) gleich `F:\` ist. Innerhalb einer `For Each`-Schleife weisen Sie der Eigenschaft `DriveLetter` dann den neuen Wert zu (in diesem Fall `X`;) und speichern die Änderung mit der `Put`-Methode ab. Beachten Sie hierbei, dass dem Laufwerkbuchstaben ein Doppelpunkt folgen muss.

```
Option Explicit
```

```
Dim oWMI
```

Listing 323: 2_03_016a.vbs


```

Dim colDrive
Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:\\." & strComputer & "\root\cimv2")
Set colDrive = oWMI.ExecQuery ("Select * from Win32_Volume _
    Where Name = 'F:\\'")
For Each objItem in colDrive
    Err = objItem.DriveLetter = "X:"
    objItem.Put_
Next

Set oWMI = Nothing

```

Listing 323: 2_03_016a.vbs (Forts.)

Die Zuweisung eines Laufwerkpfades nehmen Sie dagegen mit der `AddMountPoint`-Methode der WMI-Klasse `Win32_Volume` vor. Wenn Sie z.B. den zweiten Datenträger (Laufwerk *F*) im Ordner *C:\Daten* des ersten Datenträgers bereitstellen möchten, dann selektieren Sie das bereitzustellende Laufwerk wieder mit der `ExecQuery`-Abfrage unter Verwendung der `Where`-Klausel, wenden dann die `AddMountPoint`-Methode an und übergeben der `AddMountPoint`-Methode den Pfad des Bereitstellungspunktes als Parameter.

```

Option Explicit

Dim oWMI
Dim colDrive
Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:\\." & strComputer & "\root\cimv2")
Set colDrive = oWMI.ExecQuery ("Select * From Win32_Volume Where _
    Name = 'F:\\'")

For Each objItem in colDrive
    objItem.AddMountPoint("C:\\Daten\\")
Next

Set oWMI = Nothing

```

Listing 324: 2_03_016b.vbs

77 Datenträgerbezeichnungen festlegen

Mit Hilfe von Datenträgerbezeichnungen lassen sich die einzelnen Partitionen, Volumes und logischen Laufwerke bedeutend besser identifizieren als nur über den Laufwerkbuchstaben oder -pfad. Wenn Sie z.B. der Partition, auf der das Betriebssystem installiert ist, die Bezeichnung *System* zuweisen und der zweiten Partition, auf der sich die Benutzerdaten befinden, die Bezeichnung *Daten*, so können Sie die beiden Laufwerke in der Datenträgerübersicht anhand der Bezeichnung leichter unterscheiden.

Um eine Datenträgerbezeichnung für eine Partition, ein Volume oder ein logisches Laufwerk festzulegen, öffnen Sie zunächst den WINDOWS EXPLORER und rufen aus dem Kontextmenü des entsprechenden Laufwerks die EIGENSCHAFTEN auf. Auf der Registerkarte ALLGEMEIN können Sie nun die Datenträgerbezeichnung eingeben (oder wahlweise auch löschen), wobei Sie für ein mit NTFS formatiertes Laufwerk 32 Zeichen zur Verfügung haben, auf einem mit FAT oder FAT32 formatierten Laufwerk dagegen nur elf Zeichen. Geben Sie die gewünschte Bezeichnung ein, und klicken Sie anschließend auf ÜBERNEHMEN, um die geänderte Einstellung zu speichern.

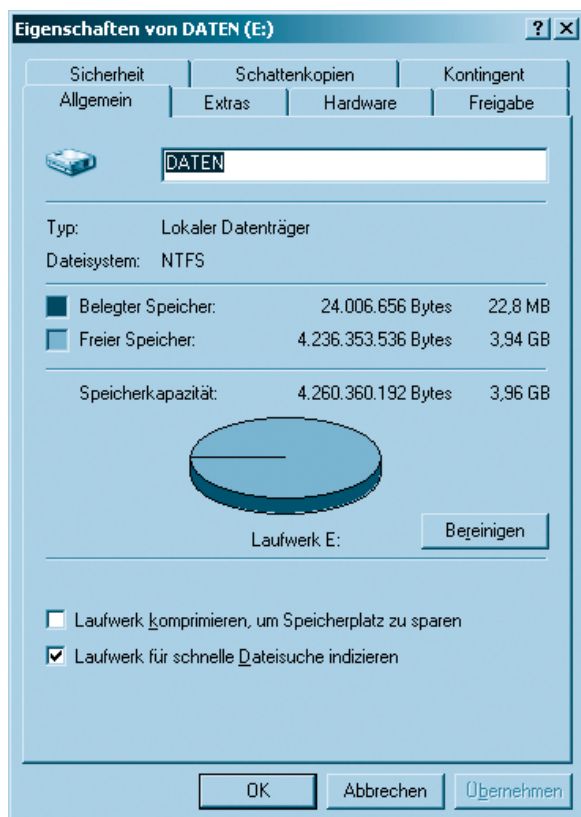


Abbildung 127: Die Datenträgerbezeichnung ändern

Die Lösung per Eingabeaufforderung

Datenträgerbezeichnungen lassen sich natürlich auch von der Eingabeaufforderung aus festlegen. Wenn Sie z.B. dem Laufwerk C:\ die Datenträgerbezeichnung *System* zuweisen möchten, so verwenden Sie im einfachsten Fall den `Label`-Befehl, der den Laufwerkbuchstaben oder den Laufwerkpfad (gefolgt von einem Doppelpunkt) und auch die neue Datenträgerbezeichnung als Parameter erwartet.

```
@Echo off  
Label C: System
```

Listing 325: 2_03_017a.bat

Den `Label`-Befehl können Sie natürlich auch zum Löschen einer Datenträgerbezeichnung verwenden, indem Sie lediglich den Laufwerkbuchstaben oder -pfad (inklusive Doppelpunkt) als Parameter übergeben. Sie werden daraufhin gefragt, ob Sie die Datenträgerbezeichnung löschen möchten. Durch Eingabe von ☐ wird die aktuelle Datenträgerbezeichnung entfernt.

Problem an der Sache ist, dass Sie das Löschen einer Datenträgerbezeichnung nicht automatisieren können, da der `Label`-Befehl die manuelle Bestätigung zum Entfernen der Datenträgerbezeichnung erwartet und Sie dies nicht erzwingen können. In solch einem Fall können Sie aber den `WMIC`-Befehl mit dem Alias `Volume` einsetzen und den Wert der Eigenschaft `Label` modifizieren, in der die Datenträgerbezeichnung hinterlegt ist.

Wenn Sie also aus einer Batch-Datei heraus eine Datenträgerbezeichnung löschen möchten, so wenden Sie den `WMIC Volume`-Befehl an, selektieren mit der `Where`-Klausel das entsprechende Laufwerk (der Laufwerkbuchstabe/-pfad muss mit einem doppelten Backslash angegeben werden) und überschreiben mit der `Set`-Anweisung den Wert der Eigenschaft `Label`.

```
@Echo off  
WMIC Volume Where "Caption='C:\\'" Set Label=""
```

Listing 326: 2_03_017b.bat

Die Lösung mit VBScript

Wenn Sie die Datenträgerbezeichnung einer Partition, eines Volumes oder eines logischen Laufwerks per VBScript ändern möchten, so stellen Sie im einfachsten Fall eine Verbindung zur WMI-Klasse `Win32_Volume` her, selektieren das entsprechende Laufwerk und weisen der Eigenschaft `Label` den Wert der neuen Datenträgerbezeichnung zu.

Das nachfolgende Listing legt beispielsweise die Datenträgerbezeichnung *System* für das Laufwerk C:\ fest. Zunächst folgt die Selektion des Laufwerks, wofür eine `Exec-`

Query-Abfrage gegen Win32_Volume durchgeführt wird. Dabei werden alle Objekte selektiert, deren Laufwerkbuchstabe (Eigenschaft Caption) gleich C:\\ ist. Innerhalb einer For Each-Schleife erfolgt dann die Zuweisung des Wertes System zur Eigenschaft Caption, die daraufhin mit der Put-Methode gespeichert wird.

Bei der Selektion der Laufwerke gilt es zu berücksichtigen, dass Laufwerkbuchstaben immer mit einem doppelten Backslash angegeben werden müssen, da die ExecQuery-Abfrage anderenfalls scheitern und keine Ergebnisse zurückliefern würde. Statt eines Laufwerkbuchstabens kann natürlich auch ein Laufwerkpfad abgefragt werden (z.B. C:\\MP\\).

Option Explicit

```
Dim oWMI
Dim strComputer
Dim colDrive
Dim objItem
```

```
strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\\" & strComputer & "\\root\\cimv2")
Set colDrive = oWMI.ExecQuery ("Select * from Win32_Volume _
    Where Caption = 'C:\\'")
```

```
For Each objItem in colDrive
    objItem.Label = "System"
    objItem.Put_
Next
```

```
Set oWMI = Nothing
```

Listing 327: 2_03_017.vbs

78 FAT und FAT32 nach NTFS konvertieren

Der Windows Server 2003 bietet die Möglichkeit an, einen mit *FAT* oder mit *FAT32* formatierten Datenträger nach *NTFS* zu konvertieren, ohne dass der Datenträger dabei neu formatiert werden muss und ohne dass die Konvertierung zu einem Datenverlust auf dem Datenträger führt.

Wenn Sie demnach einen Datenträger versehentlich mit *FAT* oder *FAT32* formatiert und bereits Daten darauf abgelegt haben, so müssen Sie den Datenträger nicht neu einrichten. Sie können das Laufwerk ohne großen Aufwand nach *NTFS* konvertieren und nach der Konvertierung alle *NTFS*-spezifischen Funktionen (Berechtigungen, Komprimierung, Verschlüsselung ...) nutzen. Die Daten, die bereits auf dem Datenträger abgelegt wurden, können Sie während der Konvertierung auf dem Datenträger belassen, da während der Konvertierung kein Formatierungsprozess stattfindet.

Die Dateisystemkonvertierung lässt sich allerdings nur in der Eingabeaufforderung durchführen. Die Datenträgerverwaltung von Windows Server 2003 enthält leider keine entsprechende Funktion, womit Sie die Konvertierung von *FAT* und *FAT32* nach *NTFS* auch unter grafischer Oberfläche durchführen können.

Achtung

Die Konvertierung des Dateisystems kann nur in eine Richtung (von *FAT* oder *FAT32* nach *NTFS*) vorgenommen werden. Eine Rückkonvertierung von *NTFS* nach *FAT* oder *FAT32* ist (zumindest ohne Zusatzprogramm) leider nicht möglich. Dies würde eine Neuformatierung des Datenträgers bedeuten. Setzen Sie die Dateisystemkonvertierung daher mit Bedacht ein, da sich die Konvertierung nicht mehr rückgängig machen lässt.

Die Lösung per Eingabeaufforderung

Um einen mit *FAT* oder mit *FAT32* formatierten Datenträger nach *NTFS* zu konvertieren, nutzen Sie in der Eingabeaufforderung den `Convert`-Befehl, der standardmäßig im Befehlsumfang von Windows Server 2003 enthalten ist. Der `Convert`-Befehl erwartet dabei die Angabe des Laufwerkbuchstabens oder des Laufwerkpfades, wie auch das Zielsystem (auch wenn es nur eines gibt), das Sie mit dem Parameter `/FS:NTFS` übergeben müssen.

Um beispielsweise einen mit *FAT32* formatierten Datenträger, der mit dem Laufwerkbuchstaben *F* gekennzeichnet ist, nach *NTFS* zu konvertieren, geben Sie in der Eingabeaufforderung den Befehl `Convert F: /FS:NTFS` ein. Falls der Datenträger über eine Datenträgerbezeichnung verfügt, so müssen Sie die Datenträgerbezeichnung nun eingeben, um den Konvertierungsprozess zu bestätigen. Der Datenträger wird – sofern es sich nicht um die Systempartition handelt – für momentane Datenträgerzugriffe gesperrt und daraufhin nach *NTFS* konvertiert. Kann der Datenträger nicht gesperrt werden, weil es sich bei dem Datenträger z.B. um die Systempartition handelt oder laufende Programme dort Dateien geöffnet haben, so wird die Konvertierung mit dem nächsten Systemstart durchgeführt.

Wenn Sie die Dateisystemkonvertierung aus einer Batch-Datei heraus aufrufen und damit automatisieren möchten, so müssen Sie vor dem Aufruf des `Convert`-Befehls entweder die Datenträgerbezeichnung löschen, oder Sie hinterlegen die Datenträgerbezeichnung in einer Textdatei und nutzen die Eingabeumleitung, damit der `Convert`-Befehl die Datenträgerbezeichnung aus der Textdatei lesen kann. Anderenfalls wartet der `Convert`-Befehl auf die Eingabe der Datenträgerbezeichnung, womit die Befehlsausführung unterbrochen wird.

Die nachfolgende `Convert`-Befehlszeile führt eine Dateisystemkonvertierung des Laufwerks *F* durch und liest die aktuelle Datenträgerbezeichnung aus der `Label.txt`, damit der `Convert`-Befehl ohne Benutzerinteraktion und damit automatisiert ausgeführt werden kann:

```
@Echo off
Convert F: /FS:NTFS < Label.txt
```

Listing 328: 2_03_018.bat

Die Lösung mit VBScript

VBScript bietet leider keine Möglichkeit an, womit Sie einen mit *FAT* oder mit *FAT32* formatierten Datenträger nach *NTFS* konvertieren können. Mit der *Run*-Methode des *Shell*-Objektes können Sie aber den *Convert*-Befehl der Eingabeaufforderung auch aus einem VBScript heraus aufrufen und die Dateisystemkonvertierung damit auch per VBScript starten.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Convert F: /FS:NTFS"

Set oShell = Nothing
```

Listing 329: 2_03_018.vbs

79 Datenträger defragmentieren

Die Defragmentierung eines Datenträgers dient der Verbesserung der Performance bei Lese- und Schreibzugriffen und zählt daher zu den Wartungsaufgaben, die regelmäßig auf einem Computer durchgeführt werden sollten. Denn mit der Defragmentierung werden alle Dateien, die in mehrere Fragmente aufgeteilt und in verteilter Form auf dem Datenträger abgelegt sind, wieder zusammengesetzt und auf dem Datenträger neu angeordnet. Damit muss der Lesekopf der Festplatte während eines Lese-/Schreibzugriffs nicht mehr an mehrere Positionen bewegt werden, und die Daten können damit schneller eingelesen bzw. zurückgeschrieben werden.

Achtung

Die Defragmentierung bewirkt (der Neuordnung der Daten wegen) natürlich unzählige Lese- und Schreibzugriffe und kann die Performance des Servers während der Defragmentierung arg beeinträchtigen. Und je nach Größe des Datenträgers und Grad der Fragmentierung kann sich die Ausführungszeit auch über mehrere Stunden hinweg erstrecken, weshalb Sie die Defragmentierung unbedingt in ein Wartungsfenster (z.B. Nacht- oder Wochenendstunden) verlegen sollten. In dieser Zeit sollten möglichst keine anderen Programme auf dem System laufen.

Bevor Sie nun eine Defragmentierung durchführen können, müssen Sie sicherstellen, dass der Datenträger über ausreichend freien Speicherplatz verfügt, da die Daten während der Defragmentierung zunächst sortiert, dann temporär in einen freien Speicherbereich verschoben und anschließend wieder neu angeordnet werden müssen. Für diese Neuordnung wird freier Speicherplatz von mindestens 15 % der Gesamtkapazität des Datenträgers benötigt, anderenfalls kann die Defragmentierung nicht durchgeführt werden. Dies kann zum Beispiel durch die Löschung nicht mehr benötigter Dateien geschehen. Durch ein Verschieben der Daten auf einen anderen Datenträger und Rückschieben nach der Defragmentierung werden diese Daten ebenfalls defragmentiert.

Wenn der erforderliche Speicherplatz sichergestellt ist, dann können Sie die Defragmentierung des Datenträgers starten. Hierzu öffnen Sie das STARTMENÜ, klicken auf ALLE PROGRAMME, erweitern ZUBEHÖR, dann SYSTEMPROGRAMME und wählen DEFRAGMENTIERUNG aus, oder aber Sie klicken im STARTMENÜ auf AUSFÜHREN und geben Dfng.msc ein, womit Sie die Defragmentierung auch direkt aufrufen können.

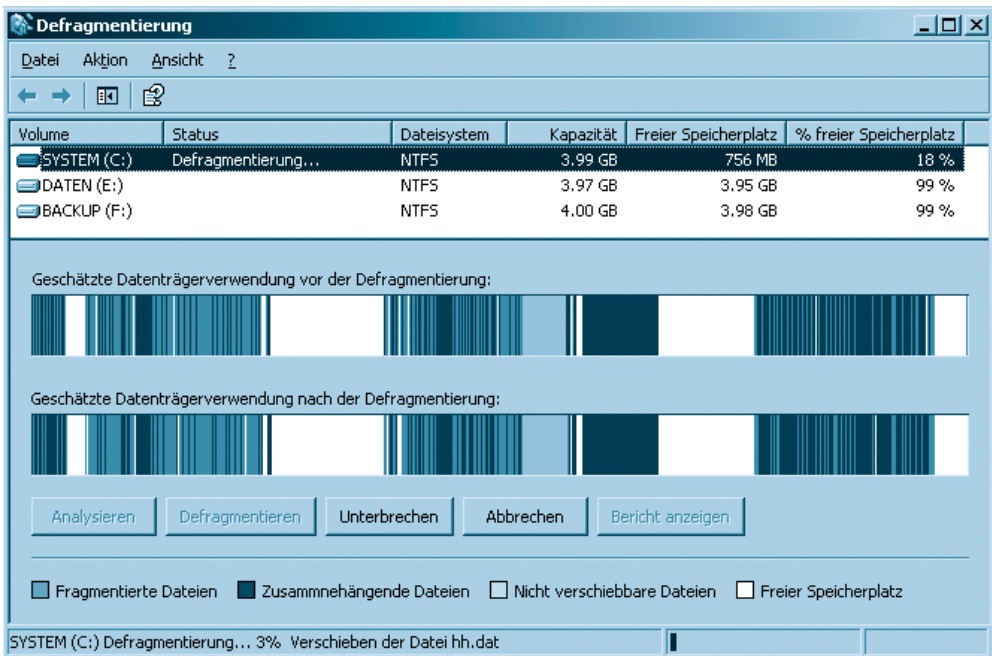


Abbildung 128: Eine Defragmentierung durchführen

Sie gelangen daraufhin in das Dialogfenster DEFRAGMENTIERUNG und finden im oberen Fensterbereich eine Übersicht der Datenträger vor, die Sie defragmentieren können. Markieren Sie den Datenträger, den Sie defragmentieren möchten, und klicken Sie auf DEFRAGMENTIEREN, um den Defragmentierungsprozess direkt zu starten. Statt die Defragmentierung sofort zu starten, können Sie auch erst eine Datenträgeranalyse durchführen und die Fragmentierung des Datenträgers ermitteln. Hierzu klicken Sie

auf ANALYSIEREN, woraufhin ein Report erzeugt wird, der eine Datenträgerstatistik unter Angabe der Gesamtkapazität, des freien Speicherbereichs, der am stärksten fragmentierten Dateien und auch der prozentualen Fragmentierungsquote ausgibt.

Die Lösung per Eingabeaufforderung

Die Defragmentierung eines Datenträgers lässt sich auch von der Eingabeaufforderung aus durchführen. Hierfür stellt der Windows Server 2003 den Defrag-Befehl bereit, der zumindest den Laufwerksbuchstaben oder den Laufwerkpfad des Datenträgers erwartet, der defragmentiert werden soll. Wenn Sie zeitgleich auch einen ausführlichen Report der Defragmentierung erzeugen möchten, so fügen Sie der Defrag-Befehlszeile den zusätzlichen Parameter /v an.

```
@Echo off
Defrag C: /v > Result.txt
```

Listing 330: 2_03_019.bat

Möchten Sie dagegen keine Defragmentierung, sondern nur eine Analyse durchführen, so wenden Sie den Defrag-Befehl unter Angabe des Laufwerksbuchstabens oder -pfades mit dem Parameter /a aus.

Die Lösung mit VBScript

Die Defrag-Methode der WMI-Klasse Win32_Volume erlaubt das Defragmentieren eines Laufwerkes auch aus einem VBScript heraus. Um diese Methode anwenden zu können, stellen Sie per GetObject zunächst eine Verbindung zum WMI-Namespacespace root\cimv2 des betreffenden Computersystems her und setzen den Impersonation-Level auf Impersonate. Anschließend wenden Sie die ExecQuery-Abfrage an und selektieren das Laufwerk mit Hilfe der Where-Klausel.

Die ExecQuery-Abfrage liefert das Ergebnis daraufhin als Kollektion zurück. Innerhalb einer For Each-Schleife können Sie dann die Defrag-Methode gegen das Laufwerkobjekt anwenden.

```
Option Explicit

Dim oWMI
Dim colDrive
Dim objItem
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:\\\" & strComputer & "\root\cimv2")
```

Listing 331: 2_03_019.vbs


```
Set colDrive = oWMI.ExecQuery ("Select * from Win32_Volume _
    Where Name = 'C:\\'")
For Each objItem in colDrive
    Err = objItem.Defrag()
Next
```

```
Set oWMI = Nothing
```

Listing 331: 2_03_019.vbs (Forts.)

80 Kontingentverwaltung für Datenträger aktivieren

Der Windows Server 2003 verfügt über ein so genanntes Verwaltungssystem für Datenträgerkontingente (*Disk Quota*), womit Sie als Administrator die Speicherplatznutzung der Partitionen, Volumes und logischen Laufwerke, die auf dem Server eingerichtet sind, sehr gut steuern können. Denn mit Hilfe von Datenträgerkontingenten können Sie für *NTFS*-Laufwerke bestimmen, wie viel Speicherplatz seitens der Benutzer (*Administratoren* sind standardmäßig davon ausgenommen) verwendet werden darf. Wird das Limit erreicht, so erhält der Benutzer entweder nur eine Warnung oder kann erst dann wieder Daten ablegen, wenn die Speicherplatznutzung reduziert oder das Kontingent angehoben wird. Welches Verhalten bei Erreichen des Kontingents in Kraft tritt, können Sie in den Eigenschaften der Kontingentverwaltung festlegen.

Da die Kontingentverwaltung für Datenträger per Default deaktiviert ist, müssen Sie diese Funktion zunächst von Hand aktivieren, um die Speicherplatznutzung steuern zu können. Hierzu öffnen Sie den *WINDOWS EXPLORER* und markieren das *NTFS*-Laufwerk, für das Sie die Kontingentverwaltung aktivieren möchten. Rufen Sie aus dem Kontextmenü des Laufwerks die *EIGENSCHAFTEN* auf, und wechseln Sie auf die Registerkarte *KONTINGENT*. Dort aktivieren Sie die Option *KONTINGENTVERWALTUNG AKTIVIEREN*, um die Funktion anzuschalten (siehe Abbildung 129).

Auf der gleichen Registerkarte können Sie auch die Standardeinstellungen für Kontingente einrichten, die mit der Aktivierung der Kontingentverwaltung für alle Benutzer gelten. Wenn Sie z.B. vermeiden möchten, dass Benutzer ihr jeweiliges Kontingent wahllos überschreiten können, so aktivieren Sie die Option *SPEICHERPLATZ BEI KONTINGENTÜBERSCHREITUNG VERWEIGERN*. Denn ist diese Option aktiviert, dann können die Kontingentinhaber erst dann wieder Daten speichern, wenn die Speicherplatznutzung reduziert oder das Kontingent erweitert wird.

Als Nächstes legen Sie die Kontingentgröße fest. Aktivieren Sie hierfür die Option *SPEICHERPLATZ BESCHRÄNKEN AUF*, und geben Sie an, wie viel Speicherplatz Sie den Kontingentinhabern zuweisen möchten. Die Kontingentgröße sollte einen realistischen Wert besitzen, der den verfügbaren Speicherplatz auf dem Datenträger und auch das durchschnittliche Datenvolumen der Benutzer und deren Anzahl berücksichtigt. Um die Benutzer rechtzeitig über das Erreichen des Limits zu informieren, legen Sie neben der eigentlichen Kontingentgröße auch eine Warnstufe fest, die mindestens 10 % unter dem eigentlichen Kontingent liegen sollte.

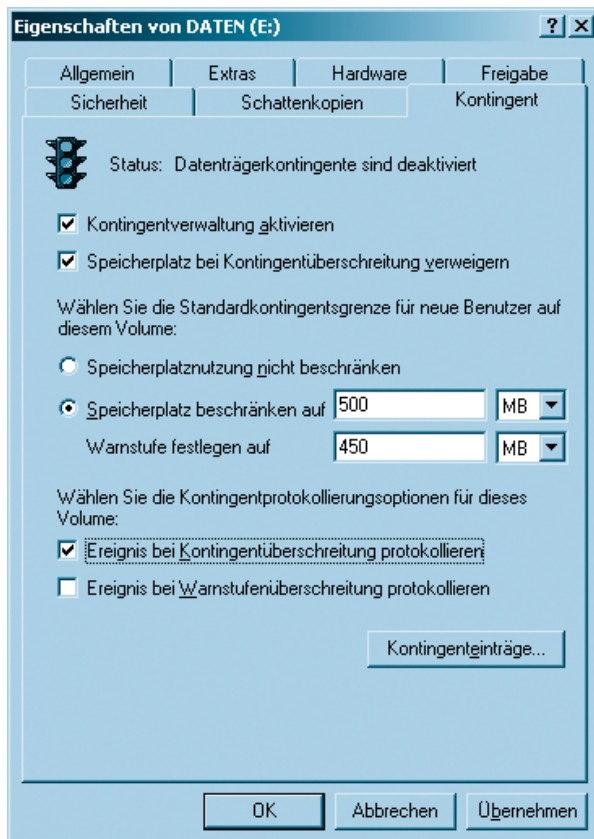


Abbildung 129: Die Kontingentverwaltung aktivieren

Und zu guter Letzt legen Sie noch fest, ob bei Überschreitung des Kontingents und der Warnstufe ein Ereignis im *Systemprotokoll* der EREIGNISANZEIGE generiert werden oder ob keine Ereignisprotokollierung erfolgen soll. Speichern Sie die geänderten Einstellungen dann mit einem Klick auf ÜBERNEHMEN.

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung lässt sich die Kontingentverwaltung für Datenträger ebenfalls aktivieren. Hierfür stellt der Windows Server 2003 den `Fsutil`-Befehl bereit, womit Sie unterschiedliche Dateisystemaufgaben durchführen können (z.B. Hardlink-, Reparsepoint- und Quota-Verwaltung).

Um beispielsweise die Kontingentverwaltung für das Laufwerk C:\ zu aktivieren, ohne dass der Speicherplatz bei Kontingentüberschreitung verweigert wird, öffnen Sie das Eingabeaufforderungsfenster und geben den Befehl `Fsutil Quota Track C:` ein. Wenn Sie die Einhaltung der Kontingente erzwingen und den Speicherplatz bei Überschreiten des Kontingents verweigern möchten, so geben Sie zusätzlich den Befehl `Fsutil Quota Enforce C:` ein.

```
@Echo off
Fsutil Quota Track C:
Fsutil Quota Enforce C:
```

Listing 332: 2_03_020.bat

Die Lösung mit VBScript

Selbst mit VBScript ist das Aktivieren der Kontingentverwaltung für Datenträger kein Problem, da Sie hierfür das `DiskQuotaControl`-Objekt nutzen können und der `QuotaState`-Eigenschaft nur den entsprechenden Wert zuweisen müssen, um die Kontingentverwaltung zu aktivieren, deren Einhaltung zu erzwingen oder die Funktion auch wieder zu deaktivieren.

Die nachfolgende Tabelle enthält die erlaubten Werte der `QuotaState`-Eigenschaft unter Angabe einer kurzen Beschreibung:

Wert	Beschreibung
0	Die Kontingentverwaltung für Datenträger ist deaktiviert.
1	Die Kontingentverwaltung für Datenträger ist aktiviert, die Einhaltung der Kontingente wird aber nicht erzwungen.
2	Die Kontingentverwaltung für Datenträger ist aktiviert und die Einhaltung der Kontingente wird erzwungen, womit weiterer Speicherplatz bei Überschreiten des Kontingents verweigert wird.

Tabelle 48: Die Werte der QuotaState-Eigenschaft

Wenn Sie beispielsweise die Kontingentverwaltung für das Laufwerk `C:\` aktivieren und die Einhaltung der Kontingentgrenze von 500 MByte erzwingen möchten, so erzeugen Sie im ersten Schritt eine neue Instanz des `DiskQuotaControl`-Objektes und wenden dann die `Initialize`-Methode gegen die Instanz des Objektes an, der Sie folgende Parameter übergeben:

- ▶ Laufwerk: Mit dem ersten Parameter übergeben Sie das Laufwerk (z.B. `C:\`), für das Sie die Kontingentverwaltung aktivieren möchten.
- ▶ Modus: Mit dem zweiten Parameter legen Sie fest, ob Sie die Instanz nur im Lese-modus (`False`) oder im Lese- und Schreibmodus (`True`) öffnen möchten. Um Änderungen an der Kontingentverwaltung vornehmen zu können, müssen Sie die Instanz auf jeden Fall im Lese- und Schreibmodus öffnen und deshalb den Wert `True` übergeben.

Anschließend weisen Sie der Eigenschaft `QuotaState` den entsprechenden Wert (siehe vorherige Tabelle) zu, den Sie zu Beginn des Skriptes als Konstante definieren können. Und zu guter Letzt legen Sie das Limit fest, indem Sie der Eigenschaft `DefaultQuotaLimit` die gewünschte Byte-Anzahl als Wert zuweisen.

```
Option Explicit

Dim colQuota
Dim strDrive
strDrive= "C:\\"
Const ENABLE = 2

Set colQuota = CreateObject("Microsoft.DiskQuota.1")
colQuota.Initialize strDrive, True
colQuota.QuotaState = ENABLE
colQuota.DefaultQuotaLimit = 500*1024*1024

Set colQuota = Nothing
```

Listing 333: 2_03_020.vbs

81 Neue Kontingenteinträge einrichten

Wenn Sie – wie im vorherigen Rezept beschrieben – die Kontingentverwaltung für Datenträger aktiviert und die Speicherplatznutzung definiert haben, so gilt das eingestellte Limit nun für alle Benutzer (außer den *Administratoren*), die Daten auf dem Laufwerk ablegen, auf dem Sie die Kontingentverwaltung aktiviert haben.

Nun kann es vorkommen, dass einzelne Benutzer etwas mehr Speicherplatz beantragen, weil sie vielleicht für mehrere Abteilungen tätig sind und daher ein überdurchschnittliches Datenvolumen haben. Diesen Benutzern können Sie nun mit Hilfe von Kontingenteinträgen mehr Speicherplatz zuweisen oder die Speicherplatzbegrenzung auch ganz aufheben, indem Sie eine Art Ausnahmeregel für die entsprechenden Benutzer erstellen.

Um einen solchen Kontingenteintrag zu erstellen, öffnen Sie wieder den WINDOWS EXPLORER und rufen aus dem Kontextmenü des entsprechenden Laufwerks die EIGENSCHAFTEN auf. Auf der Registerkarte KONTINGENT klicken Sie auf KONTINGENTEINTRÄGE..., woraufhin Sie in die Übersicht der Kontingente gelangen und – wenn Sie die Kontingentverwaltung gerade erst aktiviert haben – zumindest die folgenden Default-Einträge vorfinden:

Benutzer	Beschreibung
Vordefiniert\Administratoren	Die Mitglieder der Gruppe <i>Administratoren</i> haben standardmäßig keine Speicherplatzbegrenzung und dürfen unbegrenzt Speicher belegen.
NT-Autorität\Netzwerkdienst	Der <i>Netzwerkdienst</i> ist per Default ebenfalls von der Speicherplatzbegrenzung ausgenommen.
NT-Autorität\Lokaler Dienst	Der <i>lokale Dienst</i> darf ebenso unbegrenzten Speicherplatz belegen.

Tabelle 49: Die vordefinierten Kontingenteinträge

Sobald ein Benutzer Daten auf dem Laufwerk ablegt, auf dem Sie die Kontingentverwaltung aktiviert haben, so wird automatisch ein neuer Kontingenteintrag für den Benutzer erstellt. Prüfen Sie daher, ob bereits ein solcher Kontingenteintrag existiert, und – wenn ja – öffnen Sie diesen Eintrag durch einen Doppelklick. Sollte noch kein Kontingenteintrag vorhanden sein, so wählen Sie im Menü KONTINGENT den Eintrag NEUER KONTINGENTEINTRAG... aus, woraufhin Sie in das Dialogfenster BENUTZER WÄHLEN gelangen und den Benutzer auswählen müssen, für den Sie ein spezielles Kontingent definieren möchten.

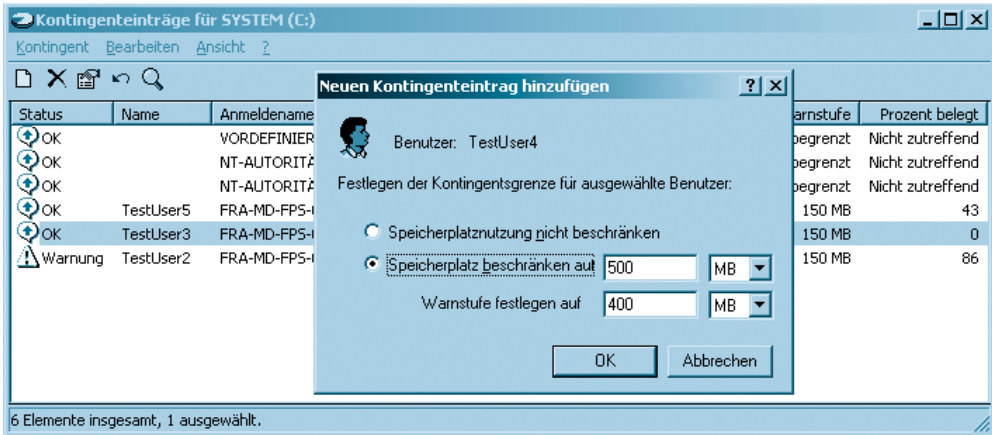


Abbildung 130: Einen neuen Kontingenteintrag hinzufügen

Nach Auswahl des Benutzers wird automatisch das Dialogfenster NEUEN KONTINGENTEINTRAG HINZUFÜGEN mit den Standard-Kontingentwerten angezeigt. Aktivieren Sie die Option SPEICHERPLATZNUTZUNG NICHT BESCHRÄNKEN, wenn Sie die Speicherplatzbegrenzung für den Benutzer ganz aufheben möchten, oder bestimmen Sie den zulässigen Speicherplatz, indem Sie das Limit und wahlweise auch die Warnstufe anpassen. Übernehmen Sie die Einstellungen dann mit einem Klick auf OK, um die Änderung zu speichern.

Die Lösung per Eingabeaufforderung

Mit dem `Fsutil`-Befehl lassen sich neue Kontingenteinträge auch von der Eingabeaufforderung aus erstellen. Der `Fsutil`-Befehl erwartet dabei die folgenden Angaben, die Sie mit dem Befehlsaufruf als Parameter übergeben müssen:

- ▶ **Aktion:** Der `Fsutil`-Befehl kann für unterschiedliche Dateisystemaufgaben verwendet werden, weshalb Sie mit dem ersten Parameter die durchzuführende Aktion `Quota Modify` übergeben müssen.
- ▶ **Laufwerk:** Mit dem zweiten Parameter übergeben Sie den Laufwerksbuchstaben oder -pfad des Laufwerks, für das Sie den neuen Kontingenteintrag erstellen möchten.

- ▶ Warnstufenwert: Als Nächstes legen Sie die Warnstufe fest und übergeben den Wert in der Größeneinheit *Bytes*.
- ▶ Kontingentwert: Mit dem vorletzten Parameter legen Sie das eigentliche Limit für den Benutzer fest, wobei auch diese Angabe in Bytes erfolgen muss.
- ▶ Benutzer: Und zu guter Letzt geben Sie den Benutzernamen (Format *Computername\Benutzername* oder *Domäne\Benutzername*) an, für den Sie den neuen Kontingenteintrag erzeugen möchten.

Die nachfolgende `Fsutil`-Befehlszeile erzeugt beispielsweise einen neuen Kontingenteintrag für den Benutzer *TestUser4* und ermöglicht dem Benutzer, auf Laufwerk C des lokalen Computersystems 200 MByte ($200 * 1024 * 1024 = 209715200$ Bytes) Speicherplatz zu nutzen, wobei der Benutzer ab einer Belegung von 150 MByte ($150 * 1024 * 1024 = 157286400$ Bytes) eine Warnung erhält:

```
@Echo off
Fsutil Quota Modify C: 157286400 209715200 FRA-MD-FPS-001\TestUser4
```

Listing 334: 2_03_021.bat

Die Lösung mit VBScript

Um einen neuen Kontingenteintrag per VBScript zu erstellen, können Sie die `AddUser`-Methode des `DiskQuotaControl`-Objektes verwenden. Wenn Sie z.B. einen neuen Kontingenteintrag für den Benutzer *TestUser4* erstellen und gleichzeitig auch das Limit auf 200 MByte setzen möchten, so erzeugen Sie im ersten Schritt eine neue Instanz des `DiskQuotaControl`-Objektes und wenden dann die `Initialize`-Methode gegen die Objektvariable an, der Sie das Laufwerk (z.B. C:\) und auch den Modus zum Öffnen (`True`) als Parameter übergeben.

Anschließend setzen Sie die `AddUser`-Methode ein und übergeben den Anmeldenamen des Benutzers (Format *Computername\Benutzername* oder *Domäne\Benutzername*), für den Sie den Kontingenteintrag erstellen möchten. Und um das Limit für den Benutzer festzulegen, weisen Sie der Eigenschaft `QuotaLimit` die Anzahl an Bytes als Wert zu.

```
Option Explicit

Dim colQuota
Dim oUser
Dim strDrive
strDrive= "C:\"

Set colQuota = CreateObject("Microsoft.DiskQuota.1")
```

Listing 335: 2_03_021.vbs

```
colQuota.Initialize strDrive, True

Set oUser = colQuota.AddUser("FRA-MD-FPS-001\TestUser4")
oUser.QuotaLimit = 209715200

Set colQuota = Nothing
```

Listing 335: 2_03_021.vbs (Forts.)

82 Auslastung der Datenträgerkontingente abfragen

Nach dem Aktivieren der Kontingentverwaltung können Sie sich jederzeit die Auslastung der Kontingente einzelner Benutzer anzeigen lassen. Wenn Sie beispielsweise wissen möchten, welche Benutzer bereits Daten auf dem Laufwerk abgelegt haben und wie viel Speicherplatz sie jeweils verwenden, so rufen Sie im WINDOWS EXPLORER wieder die EIGENSCHAFTEN des betreffenden Laufwerks auf und wechseln auf die Registerkarte KONTINGENTE. Dort klicken Sie auf KONTINGENTEINTRÄGE, um die Übersicht der Kontingenteinträge aufzurufen, worüber alle Kontingenteinträge unter Angabe des Status, des Benutzernamens, des belegten und verfügbaren Speicherplatzes wie auch der Warnstufe und der prozentualen Belegung angezeigt werden.

Status	Name	Anmeldename	Speicher belegt	Kontingentsgrenze	Warnstufe	Prozent belegt
OK		VORDEFINIERT\Administratoren	3,15 GB	Unbegrenzt	Unbegrenzt	Nicht zutreffend
OK		NT-AUTORITÄT\NETZWERKDIENT	1,37 MB	Unbegrenzt	Unbegrenzt	Nicht zutreffend
OK		NT-AUTORITÄT\LOKALER DIENST	219 KB	Unbegrenzt	Unbegrenzt	Nicht zutreffend
OK	TestUser3	FRA-MD-FPS-001\TestUser3	0 Bytes	200 MB	150 MB	0
OK	TestUser5	FRA-MD-FPS-001\TestUser5	87,92 MB	200 MB	150 MB	43
Warnung	TestUser2	FRA-MD-FPS-001\TestUser2	172,93 MB	200 MB	150 MB	86

6 Elemente insgesamt, 1 ausgewählt.

Abbildung 131: Die Kontingenteinträge überwachen

Die Lösung per Eingabeaufforderung

Die Kontingenteinträge der Benutzer, die bereits Daten auf dem Laufwerk abgelegt haben, können Sie natürlich auch von der Eingabeaufforderung aus überwachen. Um beispielsweise eine Aufstellung aller bestehenden Kontingenteinträge unter Angabe des Benutzernamens, der Änderungszeit, des verwendeten Kontingents wie auch der Warnstufe und der eigentlichen Kontingentgröße zu erzeugen, geben Sie in der Ein-

gabeaufforderung den Befehl `Fsutil Quota Query` ein und fügen den Laufwerksbuchstaben oder -pfad (gefolgt von einem Doppelpunkt) als Parameter an.

Die nachfolgende `Fsutil`-Befehlszeile z.B. gibt eine Aufstellung der Kontingenteinträge für das Laufwerk `C:\` des lokalen Computersystems am Bildschirm aus:

```
@Echo off
Fsutil Quota Query C:
```

Listing 336: 2_03_022a.bat

Wenn Sie stattdessen die Kontingentüberschreitungen für alle Laufwerke anzeigen lassen möchten, so verwenden Sie die folgende `Fsutil`-Befehlszeile:

```
@Echo off
Fsutil Quota Violations
```

Listing 337: 2_03_022b.bat

Die Lösung mit VBScript

Um die Auslastung der Datenträgerkontingente aus einem VBScript heraus abzufragen, nutzen Sie wieder das `DiskQuotaControl`-Objekt und lesen die Werte der Eigenschaften `LogonName` (Anmeldename), `QuotaLimit` (Speicherplatzbegrenzung) und `QuotaUsed` (verwendeter Speicherplatz) aus.

Hierzu stellen Sie im ersten Schritt wieder eine Verbindung zum `DiskQuotaControl`-Objekt her und wenden die `Initialize`-Methode an, der Sie das Laufwerk (z.B. `C:\`) und auch den Lesemodus (`False`) als Parameter übergeben. Innerhalb einer `For Each`-Schleife fragen Sie dann für jeden vorhandenen Kontingenteintrag die Werte der Eigenschaften ab und geben die zurückgelieferte Information per `WScript.Echo` am Bildschirm aus.

```
Option Explicit

Dim colQuota
Dim objItem
Dim strDrive
strDrive= "C:\"

Set colQuota = CreateObject("Microsoft.DiskQuota.1")
colQuota.Initialize strDrive, False

For Each objItem in colQuota
```

Listing 338: 2_03_022.vbs


```
Wscript.Echo "Username: " & objItem.LogonName
Wscript.Echo "Limit: " & objItem.QuotaLimit
Wscript.Echo "Used: " & objItem.QuotaUsed & VBCRLF
Next

Set colQuota = Nothing
```

Listing 338: 2_03_022.vbs (Forts.)

Netzwerk

Die Konfiguration des Netzwerkes lässt sich größtenteils schon während der Installation des Betriebssystems vornehmen. Dabei spielt es keine Rolle, ob Windows Server 2003 manuell von CD oder per Unattended Setup auf automatischen Wege eingerichtet wird. Aber selbst dann, wenn die Netzwerkeinrichtung mit der Installation des Betriebssystems erfolgt, müssen manchmal auch im Nachhinein noch Änderungen an der Netzwerkkonfiguration durchgeführt werden. Sei es, dass sich die IP-Adressen der DNS- oder WINS-Server ändern, von statischer IP-Adresse auf dynamische IP-Adresse (oder auch umgekehrt) gewechselt werden soll, neue TCP/IP-Filter eingerichtet werden müssen, Freigaben eingerichtet oder bearbeitet werden sollen oder einfach nur die Proxy-Einstellungen für Internetverbindungen angepasst werden müssen.

Die nachfolgenden Rezepte zeigen Ihnen daher, wie Sie die wichtigsten Aufgaben aus dem Netzwerkbereich unter grafischer Oberfläche lösen und wie Sie die gleichen Aufgaben auch von der Eingabeaufforderung aus und mit Hilfe von VBScript automatisieren können. Denn gerade im Bereich der Finalkonfiguration, die im Anschluss einer Betriebssysteminstallation folgt, bietet Letzteres sehr gute Möglichkeiten.

83 Welche Netzwerkkadapter sind installiert?

Bevor Sie mit der Konfiguration der Netzwerkeinstellungen beginnen, sollten Sie sich zunächst einen Überblick über die installierten Netzwerkkadapter verschaffen. Denn gerade im Serverbereich ist es keine Seltenheit, dass einzelne Computersysteme über mehrere Netzwerkkadapter verfügen, weil sie beispielsweise als Router fungieren und mehrere Netze miteinander verbinden oder weil sie unterschiedliche Serverdienste bereitstellen, die jeweils über gesonderte IP-Adressen erreichbar sind.

Um nun einen ersten Überblick über die installierten Netzwerkkadapter zu erhalten, wechseln Sie in die SYSTEMSTEUERUNG, rufen dort das Objekt SYSTEM auf, holen dann die Registerkarte HARDWARE in den Vordergrund und klicken anschließend auf GERÄTE-MANAGER. Im GERÄTE-MANAGER erweitern Sie den Abschnitt NETZWERKKADAPTER, woraufhin die physikalisch vorhandenen Netzwerkkarten angezeigt werden. Um auch die zusätzlichen Netzwerkkadapter wie WAN-Miniports, Direktanschlüsse etc. einzublenden, wählen Sie im Menü ANSICHT den Eintrag AUSGEBLENDETE GERÄTE ANZEIGEN aus. Die versteckten Adapter werden daraufhin ebenfalls angezeigt (siehe Abbildung 132).

Die Lösung per Eingabeaufforderung

Auch in der Eingabeaufforderung können Sie auf recht einfache Art und Weise eine Aufstellung der eingerichteten Netzwerkkarten erzeugen. Denn der Windows Server 2003 stellt hierfür den WMIC-Befehl mit dem Alias `NicConfig` bereit, womit Sie eine Verbindung zur WMI-Klasse `Win32_NetworkAdapterConfiguration` herstellen und die Eigenschaften der Netzwerkkarten abfragen können.

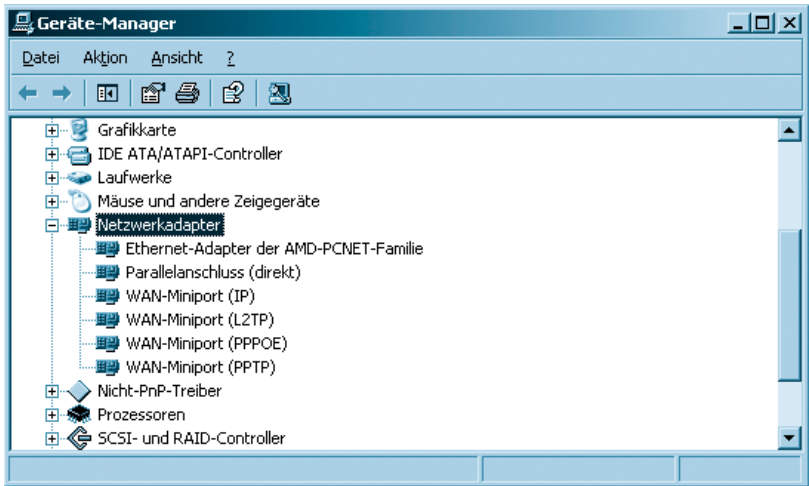


Abbildung 132: Die Übersicht der installierten Netzwerkkadpter

Die nachfolgende Tabelle zeigt die wichtigsten Eigenschaften der Win32_Network-AdapterConfiguration-Klasse, eine vollständige Aufstellung der Eigenschaften finden Sie in Teil III dieses Buches.

Eigenschaft	Beschreibung
Caption	Kurzbeschreibung des Netzwerkkadpters unter Angabe des Index und auch der Bezeichnung
DHCPEnabled	Enthält den Wert <i>True</i> , wenn die DHCP-Funktion für die Netzwerkkarte oder für den Netzwerkport aktiviert wurde und die IP-Adresse entweder automatisch konfiguriert oder dynamisch zugewiesen wird. Anderenfalls ist der Wert gleich <i>False</i> .
DHCPServer	IP-Adresse des DHCP-Servers
DNSDomain	Name der verwendeten Domäne
DNSDomainSuffixSearchOrder	Such-Reihenfolge der DNS-Domänensuffixe
DNSEnabledForWINSResolution	Enthält den Wert <i>True</i> , wenn die Namensauflösung zunächst mit DNS erfolgen soll und erst dann an den WINS-Service übergeben wird, wenn die Auflösung per DNS erfolglos war. Anderenfalls ist der Wert gleich <i>False</i> .
DNSHostname	Hostname des lokalen Computersystems, der auch in der DNS-Tabelle verwendet wird
DefaultIPGateway	IP-Adresse des Standardgateways
Description	Beschreibung der Netzwerkkarte

Tabelle 50: Die wichtigsten Eigenschaften der Win32_NetworkAdapterConfiguration-Klasse

Eigenschaft	Beschreibung
DomainDNSRegistrationEnabled	Enthält den Wert <i>True</i> , wenn die IP-Adresse der aktiven Netzwerkverbindung im DNS registriert werden soll, anderenfalls ist der Wert gleich <i>False</i> .
FullDNSRegistrationEnabled	Enthält den Wert <i>True</i> , wenn das Domänensuffix der aktiven Netzwerkverbindung im DNS registriert werden soll, anderenfalls ist der Wert <i>False</i> .
IPAddress	IP-Adresse(n) des lokalen Computersystems
IPEnabled	Enthält den Wert <i>True</i> , wenn das TCP/IP-Protokoll an den Netzwerkadapter gebunden und auch aktiviert ist. Anderenfalls ist der Wert <i>False</i> .
IPFilterSecurityEnabled	Enthält den Wert <i>True</i> , wenn die TCP/IP-Filterfunktion aktiviert ist, anderenfalls ist der Wert gleich <i>False</i> .
IPPortSecurityEnabled	Enthält den Wert <i>True</i> , wenn der Netzwerkverkehr durch eine Port-Security geschützt ist, anderenfalls ist der Wert gleich <i>False</i> .
IPSecPermitIPProtocols	Enthält die IP-Protokolle, die bei Aktivieren der Filterfunktion erlaubt werden.
IPSecPermitTCPPorts	Enthält die TCP-Ports, die bei Aktivieren der Filterfunktion zugelassen werden.
IPSecPermitUDPPorts	Enthält die UDP-Ports, die bei Aktivieren der Filterfunktion zugelassen werden.
IPSubnet	Enthält die Subnetzmaske, die im TCP/IP-Protokoll des Netzwerkadapters hinterlegt ist.
Index	Eindeutiger Index des Netzwerkadapters
MACAddress	MAC-Adresse des Netzwerkadapters
WINSEnableLMHostsLookup	Enthält den Wert <i>True</i> , wenn zur Auflösung der Hostnamen zur IP-Adresse eine lokale HOST-Datei verwendet werden soll, anderenfalls ist der Wert <i>False</i> .
WINSHostLookupFile	Pfad zur lokalen HOST-Datei, worüber die lokale Auflösung der Hostnamen zur IP-Adresse vorgenommen wird.
WINSPrimaryServer	IP-Adresse des primären WINS-Servers
WINSSecondaryServer	IP-Adresse des sekundären WINS-Servers

Tabelle 50: Die wichtigsten Eigenschaften der Win32_NetworkAdapterConfiguration-Klasse (Forts.)

Um beispielsweise eine vollständige Aufstellung der installierten Netzwerkkarten zu erzeugen, geben Sie in der Eingabeaufforderung den Befehl `WMIC NicConfig` ein, woraufhin für jede Netzwerkkarte die Werte aller Eigenschaften am Bildschirm ausgegeben werden. Wenn Sie dagegen nur den Namen der Netzwerkkarten abfragen möchten, so wenden Sie den `WMIC NicConfig`-Befehl an und legen mit der `Get`-Anweisung fest, dass nur die Werte der Eigenschaft `Caption` ausgegeben werden sollen.

```
@Echo off
WMIC NicConfig Get Caption
```

Listing 339: 2_04_001.bat

Die Lösung mit VBScript

Wenn Sie die Aufstellung der installierten Netzwerkkarten aus einem VBScript heraus erzeugen möchten, so nutzen Sie die gleiche Möglichkeit wie auch in der Eingabeaufforderung: Sie stellen eine Verbindung zur WMI-Klasse `Win32_NetworkAdapterConfiguration` her und lesen dann für jede Netzwerkkarte den Wert der Eigenschaft `Caption` aus.

Das nachfolgende Listing zeigt ein Beispiel dafür, wie Sie eine Aufstellung aller Netzwerkkarten unter Angabe des Namens erzeugen und die Informationen am Bildschirm ausgeben können. Hierzu wird im ersten Schritt eine Verbindung zum WMI-Namespace `root\cimv2` des betreffenden Computersystems (`strComputer`) hergestellt und der Personifizierung wegen der `Impersonation-Level` auf `Impersonate` gesetzt. Anschließend wird mit der `ExecQuery`-Abfrage gegen `Win32_NetworkAdapterConfiguration` eine Selektion aller Netzwerkkarten vorgenommen, die daraufhin in Form einer Kollektion zurückgeliefert werden. Innerhalb einer `For Each`-Schleife wird dann für jedes Element der Kollektion der Wert der Eigenschaft `Caption` ermittelt und per `WScript.Echo`-Anweisung am Bildschirm ausgegeben.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colNIC
Dim objItem

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colNIC = oWMI.ExecQuery ("SELECT * FROM _
    Win32_NetworkAdapterConfiguration")

For Each objItem in colNIC
    WScript.Echo objItem.Caption
Next

Set oWMI = Nothing
```

Listing 340: 2_04_001.vbs

84 Welche MAC-Adressen haben die Netzwerkkarten?

Jede Netzwerkkarte ist seitens des Herstellers mit einer eindeutigen Hardware-Adresse (engl. = *Media Access Control Address*) gekennzeichnet, die zur eigentlichen Adressierung der Datenpakete verwendet wird. Hierbei handelt es sich um einen 6 Byte langen Wert in hexadezimaler Schreibweise (z.B. *00:0C:29:4A:B6:32*), wovon die ersten 3 Byte den Hersteller-Code und die letzten 3 Byte die Netzwerkkartennummer darstellen. Mit dieser Konvention lässt sich die Eindeutigkeit der MAC-Adresse einer Netzwerkkarte gewährleisten.

Nun kann es in einigen Fällen erforderlich sein, die MAC-Adressen der installierten Netzwerkkarten zu ermitteln, weil z.B. Netzwerkprobleme aufgetreten sind und das Troubleshooting die Nachverfolgung und Analyse des Datenverkehrs erforderlich macht, weil Sicherheitsfilter auf MAC-Adressebene eingerichtet werden müssen oder weil Logfiles hinsichtlich des Netzwerkverkehrs ausgewertet werden sollen.

Doch gleich für welchen Zweck Sie die MAC-Adressen auch immer benötigen: Das Auslesen der MAC-Adressen nehmen Sie unter grafischer Oberfläche im einfachsten Fall mit Hilfe der SYSTEMINFORMATIONEN vor, da die SYSTEMINFORMATIONEN eine Aufstellung der Netzwerkadapter und auch deren wichtigste Konfigurationsparameter (unter anderem auch die der MAC-Adressen) beinhalten.

Um diese Aufstellung der Netzwerkadapter aufzurufen, öffnen Sie das STARTMENÜ, klicken auf AUSFÜHREN und geben *Msinfo32.exe* ein, woraufhin das Dialogfenster SYSTEMINFORMATIONEN geöffnet wird. Erweitern Sie in der linken Fensterhälfte den Abschnitt KOMPONENTEN, unterhalb davon NETZWERK, und markieren Sie darunter den Eintrag ADAPTER. Sie können der rechten Fensterhälfte nun die Netzwerkkarten und auch deren wichtigsten Konfigurationsparameter entnehmen, so auch die MAC-Adresse.

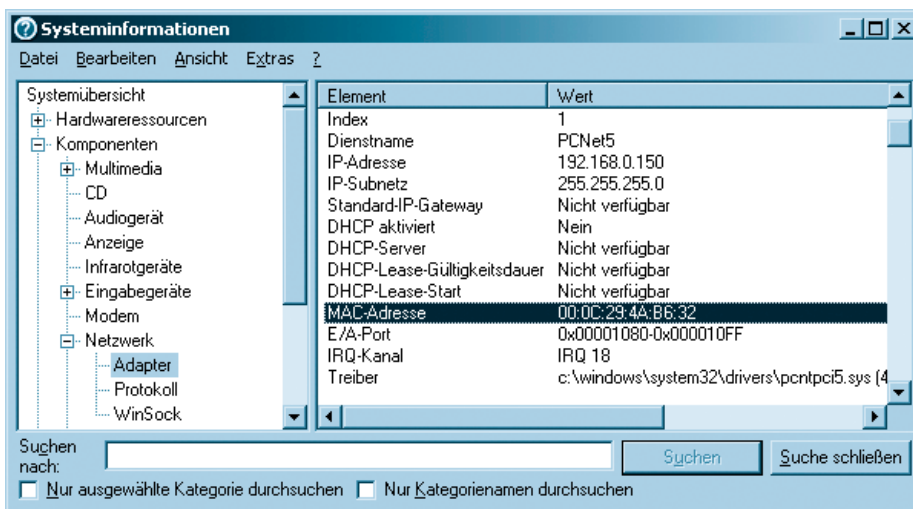


Abbildung 133: Die MAC-Adressen der Netzwerkkarten anzeigen lassen

Die Lösung per Eingabeaufforderung

Um die MAC-Adressen der installierten Netzwerkkarten von der Eingabeaufforderung aus abzufragen, können Sie zum einen den `IPConfig`-Befehl verwenden, zum anderen aber den `WMIC`-Befehl mit dem Alias `NicConfig`.

Wenn Sie alle Netzwerkkarten unter Angabe ihrer MAC-Adresse und der jeweiligen IP-Konfiguration am Bildschirm ausgeben möchten, so geben Sie in der Eingabeaufforderung einfach den folgenden `IPConfig`-Befehl ein.

```
@Echo off
IPConfig /All
```

Listing 341: 2_04_002a.bat

Möchten Sie dagegen nur den Namen der Netzwerkkarte und deren MAC-Adresse abfragen, so setzen Sie hierfür den `WMIC`-Befehl mit dem Alias `NicConfig` ein, womit Sie eine WMI-Abfrage gegen `Win32_NetworkAdapterConfiguration` durchführen und mit der `Get`-Anweisung nur die Werte der Eigenschaften `Caption` und `MACAddress` ausgeben lassen können.

```
@Echo off
WMIC NicConfig Get Caption, MACAddress
```

Listing 342: 2_04_002b.bat

Die Lösung mit VBScript

Auch mit VBScript ist das Auslesen der MAC-Adressen der installierten Netzwerkkarten kein Problem, da Sie auch per VBScript eine Abfrage gegen die WMI-Klasse `Win32_NetworkAdapterConfiguration` durchführen und die Werte der Eigenschaften `Caption` und `MACAddress` abfragen und ausgeben können.

Das nachfolgende Listing zeigt ein solches Beispiel. Nach dem Herstellen der Verbindung zum WMI-Namespace `root\cimv2` wird eine `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_NetworkAdapterConfiguration` durchgeführt, um die einzelnen Netzwerkkarten-Objekte zu selektieren. Für jedes zurückgelieferte Netzwerkkarten-Objekt wird dann der Wert der Eigenschaft `Caption` und `MacAddress` ermittelt und per `WScript.Echo`-Anweisung am Bildschirm ausgegeben.

```
Option Explicit

Dim oWMI
Dim colNic
```

Listing 343: 2_04_002.vbs


```

Dim strComputer
Dim objItem

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colNic = oWMI.ExecQuery("SELECT * FROM
Win32_NetworkAdapterConfiguration")

For Each objItem in colNic
    WScript.Echo objItem.Caption & ": " & objItem.MacAddress
Next

Set oWMI = Nothing

```

Listing 343: 2_04_002.vbs (Forts.)

85 Die aktuelle IP-Konfiguration ermitteln

Wenn Sie auf die Schnelle die aktuelle IP-Konfiguration der Netzwerkkarten (statische oder dynamische IP-Adresse, DHCP-Server, Subnetzmaske, Standardgateway ...) ermitteln müssen, so nehmen Sie unter grafischer Oberfläche wieder die SYSTEM-INFORMATIONEN zu Hilfe, da in den Systeminformationen alle benötigten Konfigurationsparameter sehr übersichtlich aufgelistet sind.

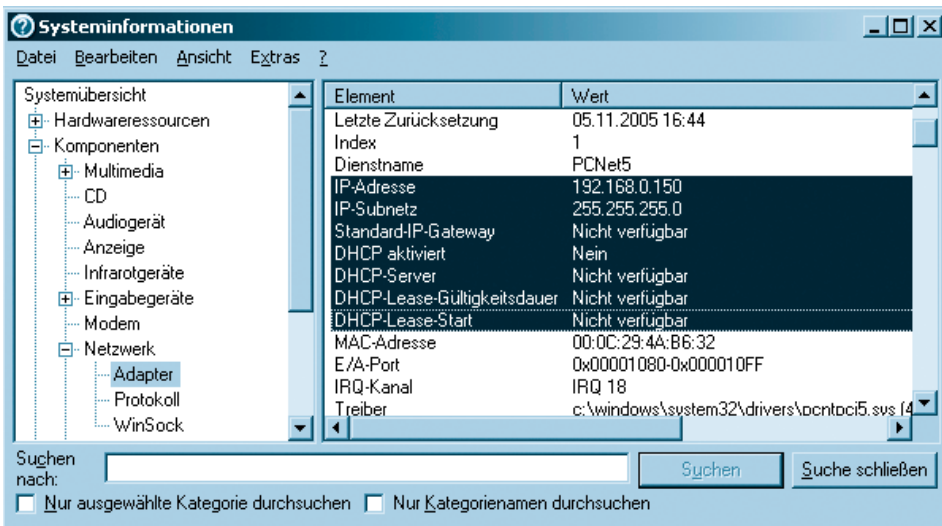


Abbildung 134: Die IP-Konfiguration der Netzwerkkarten anzeigen lassen

Öffnen Sie hierzu das STARTMENÜ, klicken Sie auf AUSFÜHREN, und geben Sie `Msiinfo32.exe` ein, um in das Dialogfenster SYSTEMINFORMATIONEN zu gelangen. Erweitern Sie in der linken Fensterhälfte den Eintrag KOMPONENTEN, dann NETZWERK, und markieren Sie unterhalb davon den Eintrag ADAPTER. In der rechten Fensterhälfte werden nun alle Netzwerkadapter unter Angabe der jeweiligen Konfigurationsparameter angezeigt. Darunter auch die IP-Konfiguration wie IP-Adresse, Subnetzmaske, Standardgateway und dergleichen mehr (siehe Abbildung 134).

Die Lösung per Eingabeaufforderung

Um die aktuelle IP-Konfiguration der installierten Netzwerkkarten von der Eingabeaufforderung aus abzufragen, können Sie zum einen den `IPConfig`-Befehl verwenden, zum anderen aber auch den `WMIC NicConfig`-Befehl. Letzteres hat den Vorteil, dass Sie explizit angeben können, welche Informationen ausgelesen werden sollen.

Wenn Sie sich auf die Schnelle einen Überblick über die aktuelle IP-Konfiguration verschaffen wollen, so geben Sie in der Eingabeaufforderung den Befehl `IPConfig /All` ein, woraufhin zunächst die domänen- oder Workgroup-relevanten Informationen wie Domänen- oder Workgroup-Name, DNS-Suffix, WINS-Proxy und DNS-Suffixsuchliste ausgegeben werden und dann pro Netzwerkadapter die statisch oder dynamisch festgelegte IP-Adresse, die Subnetzmaske, das Standardgateway und (sofern vorhanden) auch der DHCP-Server, die WINS- und DNS-Server aufgelistet wird.

```
@Echo off
IPConfig /All
```

Listing 344: 2_04_003a.bat

Eine weitere Möglichkeit, die aktuelle IP-Konfiguration auszugeben, bietet der `WMIC`-Befehl mit dem Alias `NicConfig`, womit Sie von der Eingabeaufforderung aus eine WMI-Abfrage durchführen und so die Konfigurationseinstellungen eines Netzwerkadapters abfragen können. Wenn Sie beispielsweise für alle Netzwerkadapter die Bezeichnung, die IP-Adresse, die Subnetzmaske, das Standardgateway und auch die DHCP-spezifischen Konfigurationseinstellungen am Bildschirm ausgeben möchten, so wenden Sie den `WMIC NicConfig`-Befehl an und legen mit der `Get`-Anweisung die auszulesenden Eigenschaften (siehe Tabelle 50: Die wichtigsten Eigenschaften der `Win32_NetworkAdapterConfiguration`-Klasse siehe Seite 432) fest.

```
@Echo off
WMIC NicConfig Get Caption, IPAddress, IPSubnet, DefaultIPGateway, _
    DHCPEnabled, DHCPServer, DHCPLeaseExpires, DHCPLeaseObtained
```

Listing 345: 2_04_003b.bat

Die Lösung mit VBScript

Wenn Sie die aktuelle IP-Konfiguration der installierten Netzwerkadapter aus einem VBScript heraus abfragen möchten, so nutzen Sie ebenfalls eine WMI-Abfrage. Hierzu stellen Sie im ersten Schritt eine Verbindung zum WMI-Namespace `root\cimv2` des betreffenden Computersystems her und selektieren dann die Netzwerkadapter-Objekte, indem Sie eine `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_NetworkAdapterConfiguration` durchführen. Danach können Sie für jedes Netzwerkadapter-Objekt die Werte der Eigenschaften (siehe Tabelle 50: Die wichtigsten Eigenschaften der `Win32_NetworkAdapterConfiguration`-Klasse auf Seite 432) ermitteln und mit `WScript.Echo` am Bildschirm ausgeben.

Das nachfolgende Listing gibt beispielsweise die Bezeichnung des Netzwerkadapters (Caption), die aktuelle IP-Adresse (IPAddress), die Subnetzmaske (IPSubnet), das Standardgateway (DefaultIPGateway), den Status der DHCP-Funktion (DHCPEnabled), die IP-Adresse des DHCP-Servers (DHCPServer) und auch Start (DHCPLeaseObtained) und Gültigkeitsdauer (DHCPLeaseExpires) einer dynamischen IP-Adresse am Bildschirm aus.

Option Explicit

```
Dim oWMI
Dim colNic
Dim strComputer
Dim objItem
Dim intCounter
```

```
strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colNic = oWMI.ExecQuery("SELECT * FROM
Win32_NetworkAdapterConfiguration")
```

```
For Each objItem in colNic
    WScript.Echo "Caption: " & objItem.Caption
    If IsArray(objItem.IPAddress) Then
        For intCounter = 0 To UBound(objItem.IPAddress)
            WScript.Echo "IP-Adresse: " & objItem.IPAddress(intCounter)
            WScript.Echo "IP-Subnet: " & objItem.IPSubnet(intCounter)
        Next
    End If

    If IsArray(objItem.DefaultIPGateway) Then
        For intCounter = 0 To UBound(objItem.DefaultIPGateway)
            WScript.Echo "IP-Gateway: " & objItem.DefaultIPGateway(intCounter)
        Next
    End If
Next
```

Listing 346: 2_04_003.vbs

```
End If
```

```
WScript.Echo "DHCP Enabled: " & objItem.DHCPEnabled
WScript.Echo "DHCP Server: " & objItem.DHCPServer
WScript.Echo "DHCP Lease Expires: " & objItem.DHCPLeaseExpires
WScript.Echo "DHCP Lease Obtained: " & objItem.DHCPLeaseObtained & VBCRLF
```

```
Next
```

```
Set oWMI = Nothing
```

Listing 346: 2_04_003.vbs (Forts.)

86 Eine statische IP-Adresse zuweisen

Die meisten Server-Systeme stellen spezielle Dienste und Funktionen bereit, auf die von unterschiedlichen Computersystemen aus zugegriffen wird. Daher ist es von Vorteil, dass die Server-Systeme immer unter der gleichen IP-Adresse erreichbar sind, weshalb sie vorzugsweise mit einer statischen IP-Adresse versehen werden. Das bedeutet, dass die IP-Adresse und auch die dazugehörige Subnetzmaske von Hand in den Einstellungen des TCP/IP-Protokolls hinterlegt und nicht automatisch konfiguriert oder per DHCP dynamisch zugewiesen werden, womit sie sich nach Ablauf der *Lease* (dem Gültigkeitszeitraum einer dynamischen IP-Adresse) ändern könnten.

Um nun eine statische IP-Adresse festzulegen, wechseln Sie zunächst in die Eigenschaften der NETZWERKUMGEBUNG. Dort markieren Sie die LAN-Verbindung, für die Sie die IP-Adresse vergeben möchten, und wählen im Kontextmenü den Eintrag EIGENSCHAFTEN aus. Anschließend markieren Sie den Eintrag INTERNETPROTOKOLL (TCP/IP) und klicken auf EIGENSCHAFTEN, um in die Einstellungen des Netzwerkprotokolls zu gelangen.

Aktivieren Sie die Option FOLGENDE IP-ADRESSE VERWENDEN, und geben Sie im Feld IP-ADRESSE die eindeutige IP-Adresse und im Feld SUBNETZMASKE die entsprechende Subnetzmaske ein.

Sie haben hierbei die Möglichkeit, noch weitere IP-Adressen für die Netzwerkkarte zu hinterlegen, und können damit veranlassen, dass der Server unter verschiedenen IP-Adressen erreichbar ist. Dies ist beispielsweise bei der Einrichtung virtueller Server (z.B. Webserver) eine sehr geläufige Methode, da die einzelnen Serverfunktionen jeweils unter einer anderen IP-Adresse angesprochen werden können.

Um weitere IP-Adressen zu hinterlegen, klicken Sie auf ERWEITERT und führen im oberen Bereich des Dialogfensters (in der Sektion IP-ADRESSEN) einen Klick auf die Schaltfläche HINZUFÜGEN aus und geben die IP-Adresse wie auch die Subnetzmaske ein.

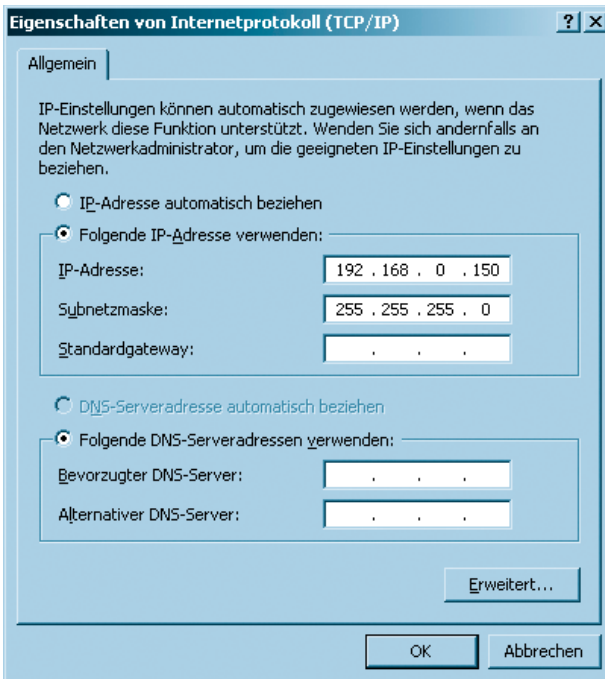


Abbildung 135: Eine statische IP-Adresse zuweisen

Die Lösung per Eingabeaufforderung

Um eine statische IP-Adresse von der Eingabeaufforderung aus zu vergeben, können Sie die `EnableStatic`-Methode der WMI-Klasse `Win32_NetworkAdapterConfiguration` verwenden, die Sie mit `WMIC NicConfig Call EnableStatic` auch im Eingabeaufforderungsfenster aufrufen können. Die `EnableStatic`-Methode erwartet dabei die Angabe der IP-Adresse und der Subnetzmaske in Form zweier Arrays, die Sie mit dem Methodenaufruf als Parameter übergeben müssen.

Wenn Sie beispielsweise der Netzwerkkarte mit der MAC-Adresse `00:0C:29:4A:B6:32` (die MAC-Adressen können Sie mit `WMIC NicConfig Get MACAddress` abfragen) die statische IP-Adresse `192.168.0.150` und die Subnetzmaske `255.255.255.0` zuweisen möchten, so rufen Sie in der Eingabeaufforderung den Befehl `WMIC NicConfig` auf, selektieren mit der `Where`-Klausel die betreffende Netzwerkkarte und rufen mit `Call EnableStatic` die eigentliche Methode auf, der Sie die Argumente IP-Adresse und Subnetzmaske als Array übergeben.

```
@Echo off
WMIC NicConfig WHERE "MACAddress='00:0C:29:4A:B6:32'" Call EnableStatic _
    ("192.168.0.150"),("255.255.255.0")
```

Listing 347: 2_04_004.bat

Die Lösung mit VBScript

Wenn Sie die Vergabe einer statischen IP-Adresse aus einem VBScript heraus vornehmen möchten, so nutzen Sie die gleiche Möglichkeit wie auch in der Eingabeaufforderung. Will heißen: Sie stellen eine Verbindung zur WMI-Klasse `Win32_NetworkAdapterConfiguration` her, selektieren die entsprechende Netzwerkkarte (z.B. anhand der MAC-Adresse) und wenden dann die `EnableStatic`-Methode an, der Sie die IP-Adresse und auch die Subnetzmaske als Array übergeben.

Das nachfolgende Listing zeigt, wie Sie der Netzwerkkarte mit der MAC-Adresse `00:0C:29:4A:B6:32` die statische IP-Adresse `192.168.0.150` und die Subnetzmaske `255.255.255.0` zuweisen können. Dabei wird zunächst eine Verbindung zum WMI-Namespace `root\cimv2` hergestellt und die zu konfigurierende Netzwerkkarte anhand der eindeutigen MAC-Adresse per `ExecQuery` selektiert. Innerhalb einer `For Each`-Schleife wird dann die `EnableStatic`-Methode gegen das Netzwerkkarten-Objekt (`objItem`) angewendet und die IP-Adresse (`arrIPAddress`) wie auch die Subnetzmaske (`arrSubNetMask`) als Argument übergeben.

Option Explicit

```
Dim oWMI
Dim strComputer
Dim colNIC
Dim objItem
Dim arrIPAddress
Dim arrSubNetMask
Dim intValue
```

```
strComputer = "."
arrIPAddress = Array("192.168.0.150")
arrSubNetMask = Array("255.255.255.0")
```

```
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colNIC = oWMI.ExecQuery ("SELECT * FROM _
    Win32_NetworkAdapterConfiguration WHERE MACAddress='00:0C:29:4A:B6:32'")
```

```
For Each objItem in colNIC
    intValue = objItem.EnableStatic(arrIPAddress, arrSubNetMask)
Next
```

```
Set oWMI = Nothing
```

Listing 348: 2_04_004.vbs

87 Standardgateway konfigurieren

Für den Fall, dass Sie im Netzwerk einen IP-Router (ein so genanntes *Gateway*) einsetzen, der mehrere Netzwerksegmente miteinander verbindet und für die Weiterleitung der Daten in die außen liegenden Netze verantwortlich ist, so müssen Sie bei Verwendung statischer IP-Adressen die IP-Adresse des Standardgateways in den Einstellungen des TCP/IP-Protokolls hinterlegen. Diese IP-Adresse wird zur Adressierung der Daten an das nächstgelegene Gateway verwendet, damit die Daten dann entsprechend in das Zielnetzwerk weitergeleitet werden können.

Um die IP-Adresse des Standardgateway zu hinterlegen, wechseln Sie in die Eigenschaften der NETZWERKUMGEBUNG, markieren dann die Verbindung, für die Sie die Einstellung vornehmen möchten, und öffnen aus dem Kontextmenü heraus die EIGENSCHAFTEN. Markieren Sie anschließend den Eintrag INTERNETPROTOKOLL (TCP/IP), und hinterlegen Sie im Feld STANDARDGATEWAY die IP-Adresse des Routers.

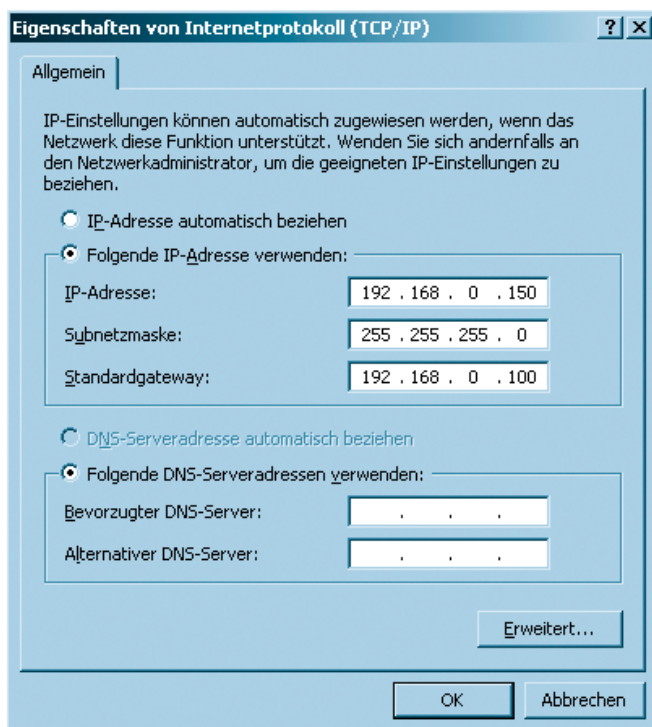


Abbildung 136: Das Standardgateway festlegen

Wenn Sie die IP-Adressen mehrerer Gateways hinterlegen möchten, so klicken Sie auf ERWEITERT, um in die erweiterten Einstellungen des TCP/IP-Protokolls zu gelangen, und führen in der Sektion STANDARDGATEWAYS (im mittleren Bereich des Dialogfensters) einen Klick auf HINZUFÜGEN aus, um je eine weitere IP-Adressen festzulegen. Windows Server 2003 nutzt dabei per Default eine automatische Schnittstellenmetrik

für das IP-Routing. Wenn beispielsweise mehrere Netzwerkkarten eingerichtet sind (z.B. eine 10- und eine 100-MBit-Netzwerkkarte) und jeweils das gleiche Gateway hinterlegt ist, so ermittelt das Betriebssystem automatisch die günstigste bzw. schnellste Verbindung zum nächsten IP-Router und weist damit der langsameren Netzwerkverbindung einen höheren Metrikwert zu. Die schnellere Netzwerkverbindung erhält einen niedrigeren Wert und hat damit eine höhere Priorität. Wenn Sie die Priorität jedoch selbst bestimmen möchten, so deaktivieren Sie die Option AUTOMATISCHE METRIK und legen im Feld SCHNITTSTELLENMETRIK den entsprechenden Wert fest.

Die Lösung per Eingabeaufforderung

Mit der `SetGateways`-Methode der WMI-Klasse `Win32_NetworkAdapterConfiguration` lässt sich das Standardgateway auch von der Eingabeaufforderung aus konfigurieren. Hierzu verwenden Sie den Befehl `WMIC NicConfig`, selektieren mit der `Where`-Klausel die betreffende Netzwerkkarte (z.B. anhand der MAC-Adresse oder des Namens) und rufen mit `Call` die `SetGateways`-Methode auf, der Sie die IP-Adresse des Standardgateways in Form eines Arrays als Parameter übergeben.

Um beispielsweise für die Netzwerkkarte mit der MAC-Adresse `00:0C:29:4A:B6:32` das Standardgateway mit der IP-Adresse `192.168.0.1` festzulegen, können Sie die nachfolgende `WMIC NicConfig`-Befehlszeile verwenden.

```
@Echo off
WMIC NicConfig WHERE "MACAddress='00:0C:29:4A:B6:32'" Call SetGateways _
("192.168.0.1")
```

Listing 349: 2_04_005a.bat

Möchten Sie zeitgleich die Metrik konfigurieren (standardmäßig wird *Automatische Metrik* verwendet), so übergeben Sie der `SetGateways`-Methode das Array mit den Metrik-Werten als optionale Parameter.

```
@Echo off
WMIC NicConfig WHERE "MACAddress='00:0C:29:4A:B6:32'" Call SetGateways _
("192.168.0.1"), ("10")
```

Listing 350: 2_04_005b.bat

Mit der `SetGateways`-Methode können Sie natürlich auch mehrere Standardgateways auf einmal hinterlegen. Wenn Sie einer Netzwerkkarte die Standardgateways `192.168.0.1` und `192.168.0.2` zuweisen möchten, so übergeben Sie der `SetGateways`-Methode einfach die durch Komma getrennten IP-Adressen als Parameter.


```
@Echo off
WMIC NicConfig WHERE "MACAddress='00:0C:29:4A:B6:32'" Call SetGateways _
("192.168.0.1","192.168.0.2")
```

Listing 351: 2_04_005c.bat

Die Lösung mit VBScript

Die `SetGateways`-Methode der WMI-Klasse `Win32_NetworkAdapterConfiguration` lässt sich natürlich auch per VBScript aufrufen, womit Sie die Konfiguration des Standardgateways auch aus einem VBScript heraus vornehmen können.

Wenn Sie z.B. der Netzwerkkarte mit der MAC-Adresse `00:0C:29:4A:B6:32` das Standardgateway `192.168.0.1` zuweisen und dabei die automatische Metrik verwenden möchten, so stellen Sie zunächst eine Verbindung zum WMI-Namespace `root\cimv2` her und selektieren die entsprechende Netzwerkkarte mit Hilfe einer `ExecQuery`-Abfrage, indem Sie mit der `Where`-Klausel die MAC-Adresse (`MACAddress`), den Namen (`Name`) oder beispielsweise den Index (`Index`) abfragen. Anschließend wenden Sie die `SetGateways`-Methode gegen das Netzwerkkarten-Objekt an und übergeben die IP-Adresse des Standardgateways als Argument.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colNIC
Dim objItem
Dim arrGateway
Dim intValue

strComputer = "."
arrGateway = Array("192.168.0.1")

Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colNIC = oWMI.ExecQuery ("SELECT * FROM _
    Win32_NetworkAdapterConfiguration WHERE MACAddress='00:0C:29:4A:B6:32'")

For Each objItem in colNIC
    intValue = objItem.SetGateways(arrGateway)
Next

Set oWMI = Nothing
```

Listing 352: 2_04_005.vbs

88 Dynamische IP-Adressen verwenden

Wenn Sie innerhalb des Netzwerkes einen DHCP-Server einsetzen und die IP-Adresse eines weiteren Computersystems nicht mehr statisch vergeben, sondern von nun an per DHCP dynamisch zuweisen möchten, so aktivieren Sie in den Einstellungen des TCP/IP-Protokolls die DHCP-Funktion, womit die statische IP-Adresse automatisch entfernt und eine dynamische IP-Adresse zugewiesen wird.

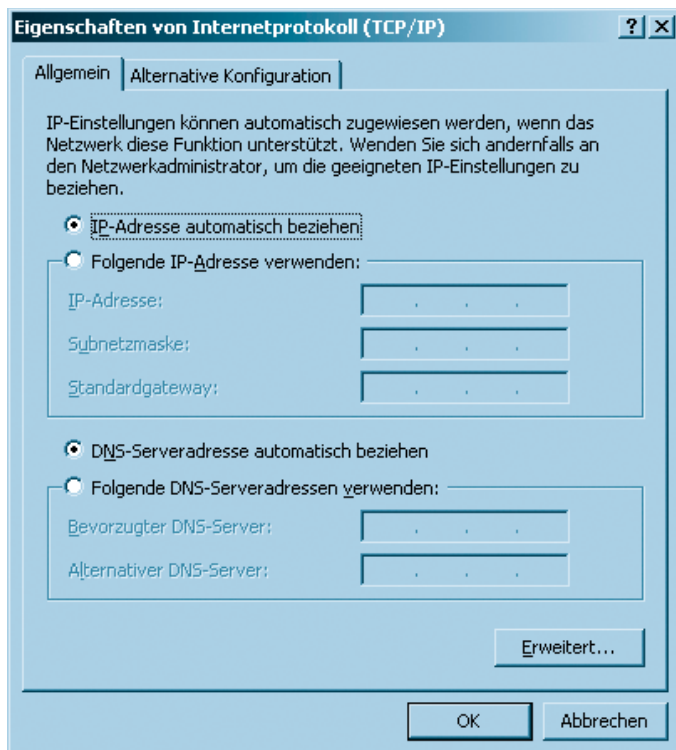


Abbildung 137: Dynamische IP-Adresse beziehen

Wechseln Sie hierzu in die Eigenschaften der NETZWERKUMGEBUNG, und markieren Sie die Verbindung, für die Sie die DHCP-Funktion aktivieren möchten. Rufen Sie aus dem Kontextmenü der Verbindung die EIGENSCHAFTEN auf, markieren Sie dann den Eintrag INTERNETPROTOKOLL (TCP/IP), und klicken Sie auf EIGENSCHAFTEN. Anschließend aktivieren Sie die Option IP-ADRESSE AUTOMATISCH BEZIEHEN, womit die statische IP-Adresse entfernt und die DHCP-Funktion aktiviert wird. Mit einem Klick auf OK wird dann ein Request an den DHCP-Server gesendet, mit der Bitte um Zuweisung einer noch freien IP-Adresse. Nach Erhalt dieser dynamischen IP-Adresse (dies lässt sich in der Eingabeaufforderung mit `IPConfig` ermitteln, siehe auch Rezept Dynamische IP-Adressen lösen und erneuern auf Seite 448) wird diese automatisch verwendet.

Die Lösung per Eingabeaufforderung

Das Aktivieren der DHCP-Funktion, womit die IP-Adresse dann automatisch zugewiesen wird, können Sie auch von der Eingabeaufforderung aus vornehmen. Hierfür stellt die WMI-Klasse `Win32_NetworkAdapterConfiguration` die `EnableDHCP`-Methode bereit, die Sie nur gegen die betreffende Netzwerkkarte anzuwenden brauchen.

Um die DHCP-Funktion für die Netzwerkkarte mit der MAC-Adresse `00:0C:29:4A:B6:32` zu aktivieren, geben Sie in der Eingabeaufforderung den WMIC `NicConfig`-Befehl ein, selektieren mit der `Where`-Klausel die entsprechende Netzwerkkarte und rufen dann mit `Call` die `EnableDHCP`-Methode auf.

```
@Echo off
WMIC NicConfig WHERE "MACAddress='00:0C:29:4A:B6:32'" Call EnableDHCP
```

Listing 353: 2_04_006.bat

Die Lösung mit VBScript

Gleiches lässt sich auch per VBScript realisieren. Dazu müssen Sie nur eine Verbindung zum WMI-Namespace `root\cimv2` des betreffenden Computersystems herstellen, die entsprechende Netzwerkkarte mit einer `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_NetworkAdapterConfiguration` selektieren und dann die `EnableDHCP`-Methode gegen das Netzwerkkarten-Objekt anwenden, woraufhin die DHCP-Funktion aktiviert wird.

Das nachfolgende Listing zeigt, wie Sie mit Hilfe der `ExecQuery`-Abfrage die Netzwerkkarte anhand ihrer MAC-Adresse selektieren und dann die `EnableDHCP`-Methode anwenden können, um die DHCP-Funktion zu aktivieren.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colNIC
Dim objItem
Dim intValue

strComputer = "."

Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colNIC = oWMI.ExecQuery ("SELECT * FROM _
    Win32_NetworkAdapterConfiguration WHERE MACAddress='00:0C:29:4A:B6:32'")
```

Listing 354: 2_04_006.vbs

```

For Each objItem in colNIC
    intValue = objItem.EnableDHCP
Next

```

```
Set oWMI = Nothing
```

Listing 354: 2_04_006.vbs (Forts.)

89 Dynamische IP-Adressen lösen und erneuern

Hin und wieder besteht die Notwendigkeit, die automatisch konfigurierte oder per DHCP zugewiesene IP-Adresse zu lösen und anschließend zu erneuern. Beispielsweise dann, wenn Sie gerade die DHCP-Funktion aktiviert haben, der DHCP-Server aber nicht schnell genug auf den Request geantwortet hat und nun eine automatisch konfigurierte – statt per DHCP zugewiesene – IP-Adresse verwendet wird. Oder wenn Sie die Erreichbarkeit und auch die Funktionalität des DHCP-Servers testen wollen und die gerade zugewiesene IP-Adresse ad hoc erneuern möchten.

Leider gibt es unter der grafischen Oberfläche keine Möglichkeit, womit Sie eine automatisch konfigurierte oder per DHCP zugewiesene IP-Adresse lösen und erneuern können. Dies geht nur in der Eingabeaufforderung oder per VBScript. Die nachfolgenden Abschnitte zeigen, wie sich eine automatische oder dynamische IP-Adresse erneuern lässt.

Die Lösung per Eingabeaufforderung

Um eine automatisch konfigurierte oder per DHCP zugewiesene IP-Adresse zu lösen und anschließend zu erneuern, können Sie in der Eingabeaufforderung den `IPConfig`- und auch den `WMIC`-Befehl verwenden.

Wenn der Netzwerkkarte beispielsweise eine automatisch konfigurierte IP-Adresse zugewiesen wurde, Sie aber eine dynamische IP-Adresse von einem DHCP-Server verwenden möchten, so können Sie die automatisch konfigurierte IP-Adresse mit dem Befehl `IPConfig /Release` wieder entfernen und mit `IPConfig /Renew` einen neuen Request an den DHCP-Server senden, der (sofern er verfügbar ist) dem Computersystem dann eine freie IP-Adresse zuweist. Anderenfalls (wenn der DHCP-Server nicht verfügbar ist) wird dem Computersystem wieder eine automatisch konfigurierte IP-Adresse aus dem 169er-Bereich zugewiesen (APIPA = Automatic Private IP Addressing).

Der `IPConfig /Release`- und `IPConfig /Renew`-Befehl wirkt sich dabei immer auf alle verfügbaren Netzwerkkarten mit aktivierter DHCP-Funktion aus. Wenn das Computersystem also mehrere Netzwerkkarten enthält und Sie die IP-Adresse nur eines bestimmten Adapters erneuern möchten, so fügen Sie dem `IPConfig /Release`- und `IPConfig /Renew`-Befehl den Namen der Netzwerkverbindung (diese können Sie sich mit `IPConfig` ohne Angabe weiterer Parameter anzeigen lassen) an.

Um für alle Netzwerkkarten, deren Verbindungsname mit *LAN* (z.B. *LAN-Verbindung*, *LAN-Verbindung1* ...) beginnt, die IP-Adresse zunächst zu lösen und anschließend wieder zu erneuern, können Sie die nachfolgenden `IPConfig`-Befehlszeilen verwenden.

```
@Echo off
IPConfig /Release LAN*
IPConfig /Renew LAN*
```

Listing 355: 2_04_007a.bat

Alternativ zum `IPConfig`-Befehl können Sie eine automatisch konfigurierte oder per DHCP zugewiesene IP-Adresse auch mit dem `WMIC NicConfig`-Befehl lösen und erneuern, indem Sie die `ReleaseDHCPLease`- und die `RenewDHCPLease`-Methode der WMI-Klasse `Win32_NetworkAdapterConfiguration` gegen die betreffende Netzwerkkarte anwenden.

Wenn Sie beispielsweise die automatisch konfigurierte oder dynamische IP-Adresse der Netzwerkkarte mit der MAC-Adresse `00:0C:29:4A:B6:32` zunächst lösen und anschließend erneuern möchten, so geben Sie in der Eingabeaufforderung den `WMIC NicConfig`-Befehl ein, selektieren mit der `Where`-Klausel die entsprechende Netzwerkkarte und führen dann einen `Call ReleaseDHCPLease` aus, um die IP-Adresse zu lösen. Zum Erneuern der IP-Adresse verwenden Sie die gleiche Befehlszeile, rufen jedoch statt der `ReleaseDHCPLease`-Methode die `RenewDHCPLease`-Methode auf.

```
@Echo off
WMIC NicConfig WHERE "MACAddress='00:0C:29:4A:B6:32'" Call ReleaseDHCPLease
WMIC NicConfig WHERE "MACAddress='00:0C:29:4A:B6:32'" Call RenewDHCPLease
```

Listing 356: 2_04_007b.bat

Die Lösung mit VBScript

Auch mit VBScript ist das Lösen und Erneuern einer automatisch konfigurierten oder einer dynamisch zugewiesenen IP-Adresse kein Problem, da die WMI-Klasse `Win32_NetworkAdapterConfiguration` hierfür die `ReleaseDHCPLease`- und `RenewDHCPLease`-Methode bereitstellt, die Sie nur gegen die betreffende Netzwerkkarte anwenden müssen.

Um beispielsweise die momentane IP-Adresse der Netzwerkkarte mit der MAC-Adresse `00:0C:29:4A:B6:32` zu lösen und anschließend wieder zu erneuern, stellen Sie im ersten Schritt eine Verbindung zum WMI-Namespace `root\cimv2` des entsprechenden Computersystems her und selektieren mit der `ExecQuery`-Abfrage gegen `Win32_NetworkAdapterConfiguration` die betreffende Netzwerkkarte. Anschließend wenden Sie die `ReleaseDHCPLease`-Methode gegen das Netzwerkkarten-Objekt an, um die momentane IP-Adresse zu lösen, und rufen dann die `RenewDHCPLease`-Methode auf, woraufhin die IP-Adresse erneuert wird.

Option Explicit

```
Dim oWMI
Dim strComputer
Dim colNIC
Dim objItem
Dim intValue
```

```
strComputer = "."
```

```
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colNIC = oWMI.ExecQuery ("SELECT * FROM _
    Win32_NetworkAdapterConfiguration WHERE MACAddress='00:0C:29:4A:B6:32'")
```

```
For Each objItem in colNIC
    intValue = objItem.ReleaseDHCPLease
    intValue = objItem.RenewDHCPLease
Next
```

```
Set oWMI = Nothing
```

Listing 357: 2_04_007.vbs

90 DNS-Einstellungen festlegen

Wenn Sie innerhalb des Netzwerkes einen DNS-Server (*Domain Name Service*) einsetzen, um von zentraler Stelle aus eine Auflösung der einzelnen Hostnamen zu den IP-Adressen vornehmen zu können, so müssen Sie die IP-Adresse des DNS-Servers in den DNS-Einstellungen der »DNS-Clients« hinterlegen, damit die clientseitigen DNS-Anfragen an den entsprechenden DNS-Server weitergeleitet und dort verarbeitet werden können. Diese Einstellungen nehmen Sie in den Eigenschaften des TCP/IP-Protokolls vor. Wechseln Sie also in die Netzwerkkumgebung, und öffnen Sie die Eigenschaften der Verbindung, für die Sie die DNS-Einstellungen vornehmen möchten. Markieren Sie anschließend INTERNETPROTOKOLL (TCP/IP), und klicken Sie auf EIGENSCHAFTEN. Im Feld BEVORZUGTER DNS-SERVER können Sie nun die IP-Adresse des primären DNS-Servers und im Feld ALTERNATIVER DNS-SERVER die IP-Adresse des sekundären DNS-Servers hinterlegen (siehe Abbildung 138).

Um weitere IP-Adressen zu hinterlegen und um weitere DNS-Einstellungen vornehmen zu können, führen Sie einen Klick auf die Schaltfläche ERWEITERT aus und wechseln dann auf die Registerkarte DNS.

Im Feld DNS-SERVERADRESSEN IN VERWENDUNGSREIHENFOLGE können Sie nun die IP-Adresse des primären DNS-Servers und bei Bedarf auch IP-Adressen zusätzlicher (sekundärer) DNS-Server hinterlegen. Hierbei spielt die Reihenfolge, in der Sie die DNS-Server aufnehmen, eine wichtige Rolle. Denn eine DNS-Anfrage wird zunächst

an den DNS-Server gesendet, der an oberster Stelle steht. Kann der Request nicht bearbeitet werden (weil der Server vielleicht nicht zur Verfügung steht), so wird die DNS-Anfrage an den nächsten DNS-Server geschickt. Dies hat unnötige Verzögerungen bei der Namensauflösung zur Folge, die es zu vermeiden gilt. Der primäre DNS-Server sollte also an erster Stelle stehen, positionieren Sie die Server-Adressen gegebenenfalls mit den Pfeil-Schaltflächen um.

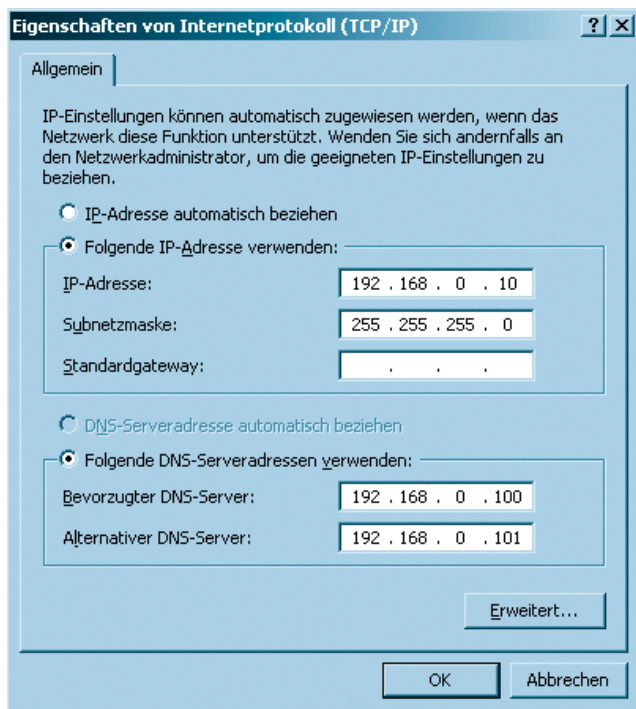


Abbildung 138: Die IP-Adresse des primären und sekundären DNS-Servers festlegen

Legen Sie anschließend die Einstellung für die Auflösung unvollständiger DNS-Namens fest. Aktivieren Sie die Option PRIMÄRE UND VERBINDUNGSSPEZIFISCHE DNS-SUFFIXE ANHÄNGEN, wenn einem unvollständigen DNS-Namen nur das primäre DNS-Suffix (das Suffix, das dem Namen des lokalen Computersystems angehängt ist) und bei Bedarf auch die übergeordneten Suffixe des primären DNS-Suffixes angehängt werden sollen, oder aktivieren Sie die Option DIESE DNS-SUFFIXE ANHÄNGEN (IN REIHENFOLGE), womit Sie die anzuhängenden DNS-Suffixe selbst bestimmen können.

Und zu guter Letzt bestimmen Sie das Verhalten zur automatischen DNS-Registrierung (sofern der DNS-Server diese Funktion unterstützt). Aktivieren Sie die Option ADRESSEN DIESER VERBINDUNG IN DNS REGISTRIEREN, wenn die IP-Adresse und auch der vollständige DNS-Name dieser Verbindung automatisch im DNS registriert werden soll, und aktivieren Sie zudem die Option DNS-SUFFIX DIESER VERBINDUNG IN DNS-REGISTRIERUNG VERWENDEN, wenn zeitgleich auch die verbindungsspezifischen DNS-Suffixe im DNS registriert werden sollen.

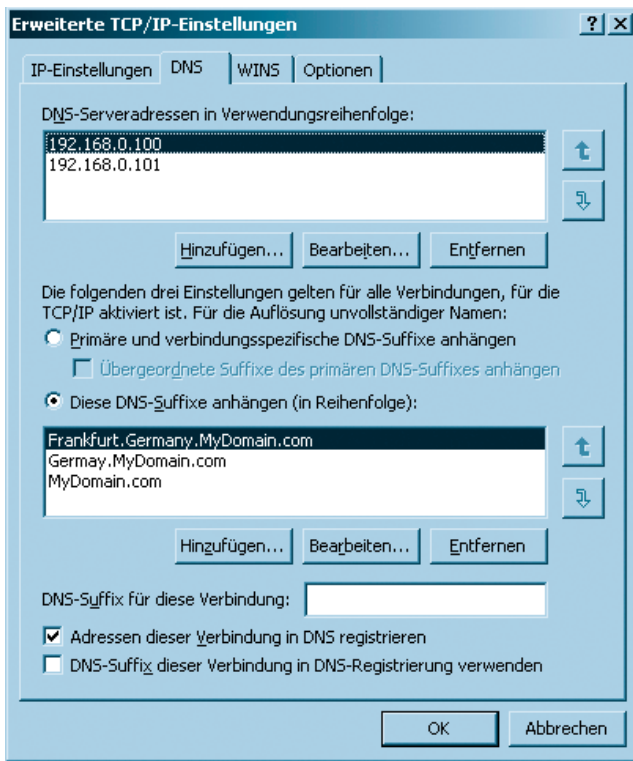


Abbildung 139: Die DNS-Einstellungen vornehmen

Die Lösung per Eingabeaufforderung

Mit dem WMIC-Befehl und dem Alias `NicConfig` können Sie die DNS-Einstellungen auch von der Eingabeaufforderung aus konfigurieren. Denn der `WMIC NicConfig`-Befehl erlaubt den Zugriff auf die WMI-Klasse `Win32_NetworkAdapterConfiguration`, die zwei Methoden zur Konfiguration der DNS-Einstellungen bereitstellt: die `EnableDNS`- und `SetDynamicDNSRegistration`-Methode.

Mit der `EnableDNS`-Methode haben Sie die Möglichkeit, die DNS-Funktionalität zu aktivieren, indem Sie die DNS-Serveradressen, die DNS-Suffixe und auch die Such-Reihenfolge für die DNS-Server und für die DNS-Suffixe festlegen. Die `EnableDNS`-Methode erwartet die Angabe folgender Parameter, die Sie mit dem Methodenaufruf übergeben müssen:

- ▶ **DNSDomain:** Mit dem ersten Parameter übergeben Sie den Namen der DNS-Domäne (z.B. "MyDomain.com").
- ▶ **DNSDomainSuffixSearchOrder:** Mit dem zweiten Parameter können Sie die DNS-Suffixe (in gleicher Reihenfolge) in der Such-Reihenfolge der DNS-Suffixe aufnehmen (z.B. "Frankfurt.Germany.MyDomain.com", "Germany.MyDomain.com", "MyDomain.com").

- ▶ **DNSHostname:** Dieser Parameter bestimmt den Hostnamen des lokalen Computersystems (z.B. "FRA-MD-SRV-001"), auf dem die DNS-Einstellungen vorgenommen werden.
- ▶ **DNSServerSearchOrder:** Der letzte Parameter ermöglicht das Festlegen der DNS-Server in der angegebenen Reihenfolge (z.B. "192.168.0.100", "192.168.0.101").

In Ergänzung zur `EnableDNS`-Methode können Sie mit der `SetDynamicDNSRegistration`-Methode festlegen, ob die Adressen und auch die DNS-Suffixe der Verbindungen im DNS registriert werden sollen oder nicht. Die `SetDynamicDNSRegistration`-Methode erwartet die folgenden beiden Parameter, die Sie mit dem Aufruf der Methode übergeben müssen:

- ▶ **DomainDNSRegistrationEnabled:** Mit dem Wert `True` wird im DNS, genauer gesagt unterhalb des vollständigen DNS-Namens für das lokale Computersystem, automatisch ein Eintrag mit der IP-Adresse der Verbindung erzeugt. Mit dem Wert `False` wird dieser Eintrag nicht erzeugt.
- ▶ **FullDNSRegistrationEnabled:** Mit dem Wert `True` wird unterhalb des Domänennamens der Verbindung und unterhalb des vollständigen DNS-Namens des lokalen Computersystems ein automatischer Eintrag mit der IP-Adresse der Verbindung erzeugt. Der Wert `False` (dies ist die Standardeinstellung) erzeugt diesen Eintrag nicht.

Die nachfolgende Batch-Datei zeigt ein Anwendungsbeispiel für die `EnableDNS`- und `SetDynamicDNSRegistration`-Methode. Mit der ersten WMIC-Befehlszeile wird das DNS-Suffix *Frankfurt.Germany.MyDomain.com* für die Verbindung festgelegt, die DNS-Suffixe *Frankfurt.Germany.MyDomain.com*, *Germany.MyDomain.com* und *MyDomain.com* in der Such-Reihenfolge aufgenommen, der Hostname *FRA-MD-SRV-001* hinterlegt und die Server *192.168.0.100* und *192.168.0.101* in der Such-Reihenfolge der DNS-Server aufgenommen.

Die zweite WMIC-Befehlszeile konfiguriert die automatische Registrierung im DNS, wobei die Adressen und auch die DNS-Suffixe der Verbindungen automatisch im DNS registriert werden.

```
@Echo off
WMIC NicConfig Call EnableDNS "Frankfurt.Germany.MyDomain.com", _
    ("Frankfurt.Germany.MyDomain.com", "Germany.MyDomain.com", "MyDomain.com"),
-
    "FRA-MD-SRV-001", ("192.168.0.100", "192.168.0.101")
WMIC NicConfig WHERE "MACAddress='00:0C:29:4A:B6:32'" _
    Call SetDynamicDNSRegistration "True", "True"
```

Listing 358: 2_04_008.bat

Die Lösung mit VBScript

Auf die gleiche Art und Weise, wie Sie die DNS-Einstellungen von der Eingabeaufforderung aus konfigurieren können, so können Sie die Einstellungen auch aus einem VBScript heraus vornehmen. Sprich: Sie nutzen WMI und stellen im ersten Schritt eine Verbindung zur WMI-Klasse Win32_NetworkAdapterConfiguration her und setzen dann die EnableDNS- und SetDynamicDNSRegistration-Methode (siehe vorheriger Abschnitt) ein, um die entsprechenden Einstellungen zu hinterlegen.

Option Explicit

```
Dim oWMI
Dim oNIC
Dim strComputer
Dim colNIC
Dim objItem
Dim strDNSDomain
Dim arrDNSSuffixSearchOrder
Dim strDNSHostname
Dim arrDNSServerSearchOrder
Dim intValue1
Dim intValue2

strComputer = "."
strDNSDomain = ""
arrDNSSuffixSearchOrder = Array("Frankfurt.Germany.MyDomain.com", _
    "Germany.MyDomain.com", "MyDomain.com")
strDNSHostname = "FRA-MD-SRV-001"
arrDNSServerSearchOrder = Array("192.168.0.100", "192.168.0.101")

Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}\\." & strComputer & "\root\cimv2")
Set oNIC = oWMI.Get("Win32_NetworkAdapterConfiguration")
intValue1 = oNIC.EnableDNS(strDNSHostname, strDNSDomain, _
    arrDNSServerSearchOrder, arrDNSSuffixSearchOrder)

Set colNIC = oWMI.ExecQuery("SELECT * FROM _
    Win32_NetworkAdapterConfiguration WHERE MACAddress='00:0C:29:4A:B6:32'")
For Each objItem in colNIC
    intValue2 = objItem.SetDynamicDNSRegistration("True", "True")
Next

Set oWMI = Nothing
Set oNIC = Nothing
```

91 WINS-Einstellungen konfigurieren

Für den Fall, dass Sie neben einem DNS-Server auch einen WINS-Server zur Auflösung der NetBIOS-Namen zu den jeweiligen IP-Adressen einsetzen, so müssen Sie die IP-Adresse des WINS-Servers ebenfalls in den Eigenschaften des TCP/IP-Protokolls hinterlegen.

Hierzu öffnen Sie auf den jeweiligen »WINS-Clients« wieder die Eigenschaften der Netzwerkumgebung, markieren die Netzwerkverbindung, für die Sie die WINS-Einstellungen hinterlegen wollen, und öffnen aus dem Kontextmenü heraus die EIGENSCHAFTEN der Netzwerkverbindung. In den Eigenschaften der Netzwerkverbindung markieren Sie den Eintrag INTERNETPROTOKOLL (TCP/IP) und klicken auf EIGENSCHAFTEN, um in die Einstellungen des TCP/IP-Protokolls zu gelangen. Dort klicken Sie auf die Schaltfläche ERWEITERT und holen anschließend die Registerkarte WINS in den Vordergrund.

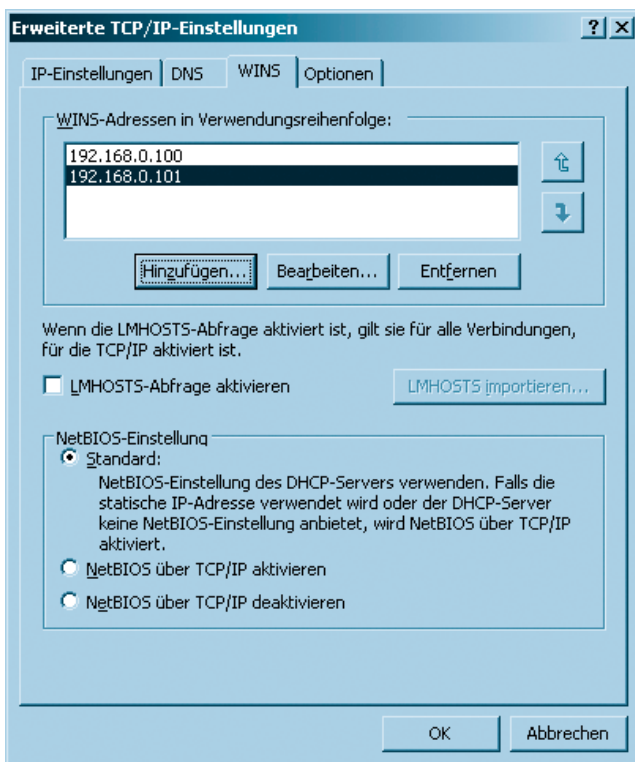


Abbildung 140: Die WINS-Einstellungen vornehmen

Im Feld WINS-ADRESSEN IN VERWENDUNGSREIHENFOLGE hinterlegen Sie nun die IP-Adresse des primären WINS-Servers und – falls Sie mehrere WINS-Server einsetzen – auch die des sekundären WINS-Servers. Mit der Option LMHOSTS-ABFRAGE AKTIVIEREN können Sie festlegen, dass die Auflösung der NetBIOS-Namen zu den IP-Adressen über die lokale und damit dezentrale LMHOSTS-Datei erfolgt. Bei Verwendung

eines zentralen WINS-Servers ist dies jedoch nicht (oder zumindest nur in Ausnahmefällen) erforderlich, da alle aufzulösenden NetBIOS-Namen und die dazugehörigen IP-Adressen in einer zentralen Tabelle auf dem WINS-Server hinterlegt sind. Deaktivieren Sie daher diese Option, wenn Sie einen WINS-Server einsetzen.

Und zu guter Letzt legen Sie fest, ob die NetBIOS over TCP/IP-Einstellungen des DHCP-Servers verwendet, grundsätzlich aktiviert oder aber deaktiviert werden sollen. Hierbei gilt zu berücksichtigen, dass NetBIOS over TCP/IP aktiviert sein muss, wenn NetBIOS-Namen verwendet und zudem via WINS aufgelöst werden. Bei Verwendung einer statischen IP-Adresse aktivieren Sie also die Option NETBIOS ÜBER TCP/IP AKTIVIEREN, anderenfalls (wenn Sie dynamische IP-Adressen per DHCP zuweisen) können Sie die Einstellungen auch vom DHCP-Server beziehen (STANDARD).

Die Lösung per Eingabeaufforderung

Die WINS-Einstellungen lassen sich nicht nur unter grafischer Oberfläche vornehmen, sondern auch von der Eingabeaufforderung aus. Denn mit dem WMIC-Befehl und dem Alias NicConfig können Sie eine Verbindung zur WMI-Klasse Win32_NetworkAdapterConfiguration herstellen und damit die folgenden drei Methoden aufrufen, die der Konfiguration der WINS-Einstellungen dienen:

- ▶ **EnableWINS:** Die `EnableWINS`-Methode erlaubt das Aktivieren der WINS-Clientfunktion und zeitgleich auch das Festlegen der zu verwendenden LMHOSTS-Datei, falls kein zentraler WINS-Server eingesetzt wird. Die `EnableWINS`-Methode erwartet dabei die Angabe von vier Parametern, die Sie mit dem Aufruf der Methode übergeben müssen. Mit dem ersten Parameter (`DNSEnabledForWINSResolution`) übergeben Sie den Wert `True`, um die Auflösung der NetBIOS-Namen ebenfalls über DNS vorzunehmen, anderenfalls übergeben Sie den Wert `False`. Mit dem zweiten Parameter (`WINSEnableLMHOSTSLookup`) legen Sie fest, ob zur Auflösung der NetBIOS-Namen zu den IP-Adressen eine lokale LMHOSTS-Datei verwendet (`True`) oder ob die Auflösung nur über einen zentralen WINS-Server erfolgen (`False`) soll. Wenn Sie mit dem zweiten Parameter `True` übergeben und damit die Nutzung einer lokalen LMHOSTS-Datei aktiviert haben, so übergeben Sie mit dem dritten Parameter (`WINSHostLookupFile`) den vollständigen Pfad zur LMHOSTS-Datei. Mit dem vierten und letzten Parameter (`WINSScopeID`) können Sie den Gültigkeitsbereich einschränken, indem Sie den Namen der Arbeitsgruppe übergeben, womit die Auflösung dann nur für die in der Arbeitsgruppe befindlichen Computersysteme vorgenommen wird.
- ▶ **SetWINSServer:** Mit der `SetWINSServer`-Methode können Sie die IP-Adresse des primären und auch die des sekundären WINS-Server übergeben. Die Methode erwartet die Angabe zweier Parameter, die Sie mit dem Aufruf der Methode übergeben müssen. Mit dem ersten Parameter (`WINSPrimaryServer`) übergeben Sie die IP-Adresse des primären WINS-Servers, mit dem zweiten Parameter (`WINSSecondaryServer`) dagegen die des sekundären WINS-Servers.

- **SetTCPIPNetBIOS:** Mit der SetTCPIPNetBIOS-Methode können Sie die NetBIOS-Einstellung festlegen, indem Sie mit dem Aufruf der Methode den Wert 0 (Standard: NetBIOS-Einstellungen des DHCP-Servers verwenden), den Wert 1 (NetBIOS über TCP/IP aktivieren) oder den Wert 2 (NetBIOS über TCP/IP deaktivieren) übergeben.

Wenn Sie beispielsweise auf einem Computersystem die WINS-Funktion aktivieren, dabei keine LMHOSTS-Datei verwenden, sondern die WINS-Server *192.168.0.100* (primär) und *192.168.0.101* (sekundär) hinterlegen wollen und zeitgleich auch NetBIOS über TCP/IP aktivieren möchten, so können Sie hier für die nachfolgenden drei WMIC NicConfig-Befehlszeilen verwenden:

```
@Echo off
WMIC NicConfig Call EnableWINS "True", "False"
WMIC NicConfig WHERE "IPEnabled='True'" _
    Call SetWINSServer "192.168.0.100", "192.168.0.101"
WMIC NicConfig WHERE "IPEnabled='True'" _
    Call SetTCPIPNetBIOS "1"
```

Listing 360: 2_04_009.bat

Die Lösung mit VBScript

Wenn Sie die clientseitigen WINS-Einstellungen per VBScript konfigurieren möchten, so setzen Sie ebenfalls WMI ein und wenden die im vorherigen Abschnitt beschriebene EnableWINS-, SetWINSServer- und gegebenenfalls auch die SetTCPIPNetBIOS-Methode an, womit Sie die WINS-Funktion aktivieren, die IP-Adressen der WINS-Server hinterlegen und auch die NetBIOS over TCP/IP-Funktion setzen können.

Das nachfolgende Listing zeigt ein Beispiel dafür, wie Sie mit der EnableWINS-Methode die WINS-Funktion auf dem Computersystem aktivieren und mit der SetWINSServer-Methode die IP-Adresse des primären WINS-Servers (*192.168.0.100*) und die des sekundären WINS-Servers (*192.168.0.101*) setzen können. Hierzu wird im ersten Schritt eine Verbindung zum WMI-Namespace *root\cimv2* und anschließend zur WMI-Klasse *Win32_NetworkAdapterConfiguration* hergestellt. Danach wird die EnableWINS-Methode mit dem Parameter *True* und *False* angewendet, womit die WINS-Clientfunktionalität aktiviert (*True*) wird, jedoch ohne Nutzung der LMHOSTS-Datei (*False*). Anschließend erfolgt eine Selektion aller Netzwerkadapter mit aktiviertem TCP/IP-Protokoll. Für jeden zurückgelieferten Netzwerkadapter wird dann die SetWINSServer-Methode ausgeführt, womit die IP-Adresse des primären und die des sekundären WINS-Servers hinterlegt wird.

```
Option Explicit
```

```
Dim oWMI
Dim strcomputer
```

Listing 361: 2_04_009.vbs

```

Dim NIC
Dim colNIC
Dim objItem
Dim intValue1
Dim intValue2
Dim strPrimaryServer
Dim strSecondaryServer

strComputer = "."
strPrimaryServer = "192.168.0.100"
strSecondaryServer = "192.168.0.101"

Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set NIC = oWMI.Get("Win32_NetworkAdapterConfiguration")
intValue1 = NIC.EnableWINS("True", "False")

Set colNIC = oWMI.ExecQuery("SELECT * FROM Win32_NetworkAdapterConfiguration
-
    WHERE IPEnabled='True'")
For Each objItem in colNIC
    intValue2 = objItem.SetWINSServer(strPrimaryServer, strSecondaryServer)
Next

Set oWMI = Nothing

```

Listing 361: 2_04_009.vbs (Forts.)

92 TCP/IP-Filter konfigurieren

Damit der Verbindungsaufbau zwischen zwei Computersystemen gezielt ablaufen kann und unterschiedliche Dienste gleichzeitig auf demselben Computer ausgeführt werden können, gibt es die so genannten Ports. Denn eine Socket-Verbindung zwischen zwei Computersystemen kann nur dann aufgebaut werden, wenn nicht nur die IP-Adresse angegeben wird, sondern auch der Port. Hinter jedem Port steckt ein so genannter *Listener*. Dabei handelt es sich um einen Prozess, der darauf wartet, dass ein anderer Computer Verbindung mit eben diesem Port aufnimmt.

Aus der Historie heraus sind nun bestimmte Dienste bestimmten Portnummern zugeordnet, wie beispielsweise das HTTP-Protokoll des Webserver dem Port 80. Diese Zuordnungen bezeichnet man als »Well-known Ports«.

Die folgende Tabelle gibt einen kurzen Überblick über die wichtigsten Ports und die Dienste, die normalerweise über diese Ports ausgeführt werden. Eine vollständige Aufstellung aller Ports und der dazugehörigen Dienste finden Sie auf der Internetseite www.iana.org. Des Weiteren stellt Microsoft auf der Internetseite www.microsoft.com eine Übersicht der Dienste von Windows Server 2003 und der dazugehörigen Ports zur Verfügung.

Port	Dienst	Beschreibung
20-21	FTP	Mit Hilfe von FTP können Sie Dateien von und zum Internet transferieren. Der Transfer ist effektiver als ein HTTP-Download, weshalb er gerne einem HTTP-Transfer vorgezogen wird.
23	Telnet	Mit Telnet sind Logins auf verschiedenen Server-Systemen möglich.
25	SMTP	Mailserver verwenden das Simple Mail Transfer Protocol zum Versenden von E-Mails.
53	DNS	Der DNS-Dienst dient zur Auflösung der DNS-Hostnamen zu den dazugehörigen IP-Adressen mit Hilfe eines DNS-Servers.
79	Finger	Der Finger-Dienst dient dazu, Informationen über einen Benutzer abzufragen.
80	HTTP	Das Hypertext Transfer Protocol wird von Webservern zur Übertragung von Webseiten eingesetzt.
110	POP3	Das Post Office Protocol dient zur Abholung von Mails von einer Mailbox.
135	RPC	Das Remote Procedure Call Protocol wird von einigen Systemen zur Verbindung mit Datei-, Druck-, Mail- und Datenbankservern eingesetzt.
137-139	NetBIOS	NetBIOS wird vor allem von Windows-Systemen zur Verbindung mit Datei- und Druckdiensten und zur Namensauflösung eingesetzt.
143	IMAP	Dieses Protokoll ist eine erweiterte und modernere Variante des POP3-Protokolls.
443	HTTPS	Dies ist der Port für verschlüsselte HTTP-Verbindungen zum Webserver per SSL (Secure Socket Layer), beispielsweise für die sichere Übermittlung von Kreditkartendaten.
445	SMB	Der Port 445 wird für den SMB-Transfer (z.B. Freigabe-Verbindungen) über TCP/IP verwendet.
8080	-	Häufig verwendeter Port für HTTP Proxy-Server

Tabelle 51: Die gängigsten Ports und die dazugehörigen Dienste

In den Eigenschaften des TCP/IP-Protokolls können Sie nun so genannte TCP/IP-Filter konfigurieren, womit Sie explizit festlegen können, welche eingehenden TCP- und UDP-Verbindungen akzeptiert und welche abgewiesen werden sollen. Damit können Sie den Datenverkehr sehr gut steuern und gewissermaßen ein Firewall-System auf Paketfilter-Basis einrichten.

Um die TCP/IP-Filterung zu aktivieren und einzelne Filter zu konfigurieren, rufen Sie zunächst die Eigenschaften der Netzwerkumgebung auf und markieren die Netzwerkverbindung, für die Sie die TCP/IP-Filter setzen möchten. Wählen Sie aus dem Kontextmenü der Netzwerkverbindung **EIGENSCHAFTEN** aus, um in die Einstellungen zu gelangen, und markieren Sie den Eintrag **INTERNETPROTOKOLL (TCP/IP)**. Klicken Sie auf **EIGENSCHAFTEN** und anschließend auf **ERWEITERTE**, um in die Protokolleigenschaften zu

gelangen, holen Sie dann die Registerkarte OPTIONEN in den Vordergrund, und führen Sie einen Klick auf die Schaltfläche EIGENSCHAFTEN aus, woraufhin das Dialogfenster TCP/IP-FILTERUNG aufgerufen wird.

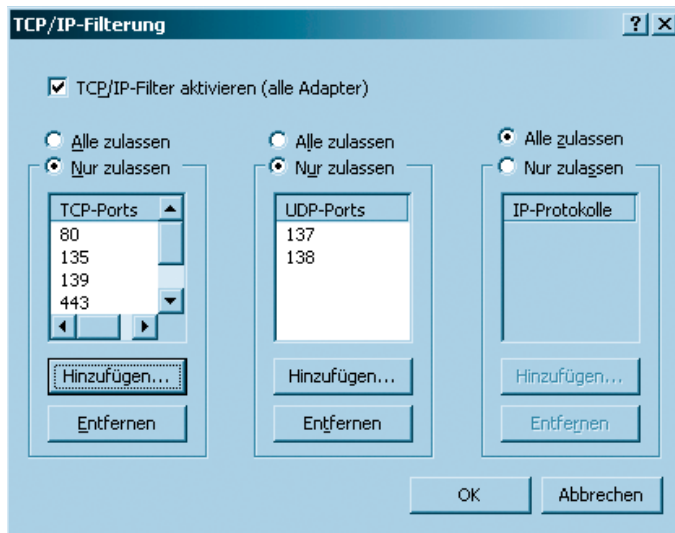


Abbildung 141: TCP- und UDP-Ports freigeben

Dort aktivieren Sie die Option TCP/IP-FILTER AKTIVIEREN (ALLE ADAPTER), um die Filter-Funktion einzuschalten. Danach können Sie – durch Anklicken der Option NUR ZULASSEN – für die TCP- und UDP-Ports wie auch für die einzelnen IP-Protokolle (mit Ausnahme des ICMP-Protokolls) die einzelnen Filter setzen, indem Sie die jeweiligen Ports und auch die Protokolle der Liste hinzufügen. Klicken Sie anschließend auf OK, um die Einstellungen zu übernehmen.

Exkurs

Die TCP/IP-Protokollfamilie

Bei TCP/IP handelt es sich nicht um ein allein stehendes Netzwerkprotokoll, sondern vielmehr um einen Zusammenschluss bzw. eine Gruppierung mehrerer Einzelprotokolle. Daher wird TCP/IP oft auch als Protokollfamilie oder Protokollstapel bezeichnet. TCP/IP setzt sich dabei aus den Protokollen IP (*Internet Protocol*), TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*), ARP (*Address Resolution Protocol*), ICMP (*Internet Control Message Protocol*) und IGMP (*Internet Group Management Protocol*) zusammen.

Das IP-Protokoll stellt ein verbindungsloses Protokoll dar, das als Bindeglied zwischen der physikalischen und der datenspezifischen Übertragungsebene fungiert und daher für die eigentliche Datenübertragung zuständig ist. Verbindungslos heißt in diesem Fall, dass das IP-Protokoll keine Garantie dafür übernimmt, dass die einzelnen IP-Pakete ihr Ziel auch wirklich erreichen und

zudem auch in der richtigen Reihenfolge empfangen werden. Die ordnungsgemäße Datenübertragung kann also nur über die darüber liegenden Protokolle sichergestellt werden. Ein solches darüber liegendes Protokoll ist z.B. das TCP-Protokoll. TCP ist verbindungsorientierter Natur und sorgt deshalb dafür, dass die Daten zum einen vollständig übertragen und zum anderen auch in der richtigen Reihenfolge an ihrem Ziel wieder zusammengesetzt werden. Ein weiteres darüber liegendes Protokoll ist das UDP-Protokoll, das im Gegensatz zu TCP jedoch verbindungslos ist und daher ebenfalls keine Garantie für die vollständige Datenübertragung übernimmt. TCP und UDP nutzen das IP-Protokoll für den eigentlichen Datentransfer auf der physikalischen Übertragungsebene.

Beim ARP-Protokoll handelt es sich weniger um ein Übertragungsprotokoll im herkömmlichen Sinne, sondern eher um eine Art Wartungsprotokoll. Denn ARP hat die Aufgabe, die MAC-Adresse des Zielsystems, an das die Daten übermittelt werden sollen, zu ermitteln. Hierzu wird eine Anfrage versendet, woraufhin das Computersystem mit der angegebenen IP-Adresse die entsprechende MAC-Adresse zurückschickt. Diese MAC-Adresse wird in der lokalen ARP-Tabelle (diese können Sie sich in der Eingabeaufforderung mit `Arp -a` anzeigen lassen) zwischengespeichert und mit dem nächsten Datentransfer zur Adressierung der Datenpakete verwendet.

Das ICMP-Protokoll stellt ebenfalls eine Art Wartungsprotokoll dar. Dieses Protokoll wird z.B. zur Ermittlung des aktuellen IP-Status und möglicher Übertragungsfehler verwendet. Bestes Beispiel für den Einsatz des ICMP-Protokolls ist der Befehl `Ping`, womit Sie die Erreichbarkeit eines bestimmten Zielsystems testen können, indem Sie gewissermaßen einige »Erreichbarkeitsanfragen« an das Zielsystem übermitteln und so dessen Verfügbarkeit überprüfen können.

Das IGMP-Protokoll dient dem Multicasting und damit der Datenübertragung an verschiedene Computersystemgruppen. Das IGMP-Protokoll kommt vorzugsweise beim Streaming zum Einsatz, wie es beispielsweise bei Web-Radio der Fall ist.

Die Lösung per Eingabeaufforderung

Die Aktivierung der TCP/IP-Filterung und auch das Setzen der einzelnen Filter sind selbst von der Eingabeaufforderung aus möglich. Denn die WMI-Klasse `Win32_NetworkAdapterConfiguration` stellt hierfür die `EnableIPFilterSec`- und `EnableIPSec`-Methode bereit, die Sie mit dem `WMIC`-Befehl unter Verwendung des Alias `NicConfig` auch in der Eingabeaufforderung aufrufen können.

- **EnableIPFilterSec:** Mit der `EnableIPFilterSec`-Methode haben Sie die Möglichkeit, die TCP/IP-Filterung ein- und auch wieder auszuschalten. Um die Funktion für alle Netzwerkadapter zu aktivieren, wenden Sie lediglich die Methode an und übergeben ihr den Wert `True`. Möchten Sie die Filterung dagegen deaktivieren, so übergeben Sie mit dem Aufruf der Methode den Wert `False`.

- **EnableIPSec:** Mit der `EnableIPSec`-Methode können Sie – sofern Sie mit `EnableIPFilterSec` die TCP/IP-Filterung aktiviert haben – die einzelnen TCP- und UDP-Ports wie auch die IP-Protokolle explizit zulassen. Die `EnableIPSec`-Methode erwartet dabei die Angabe von drei Parametern, die Sie mit dem Aufruf der Methode übergeben müssen. Mit dem ersten Parameter (`IPSecPermitIPProtocols`) übergeben Sie ein Array mit den Protokollen (z.B. *IP*, *TCP*, *UDP*, *ARP* ...), die explizit zugelassen werden sollen. Mit dem zweiten Parameter (`IPSecPermitTCPPorts`) übergeben Sie ein Array mit den TCP-Ports und mit dem dritten Parameter (`IPSecPermitUDPPorts`) ein Array mit den UDP-Ports, die freigeschaltet werden sollen.

Das nachfolgende Beispiel zeigt, wie Sie mit der `EnableIPFilterSec`-Methode die Funktion TCP/IP-FILTERUNG für alle Netzwerkadapter aktivieren, dabei alle Protokolle und nur die TCP-Ports *80*, *135*, *139*, *443*, *445* und UDP-Ports *137*, *138* zulassen können.

```
@Echo off
WMIC NicConfig Call EnableIPFilterSec "True"
WMIC NicConfig Where "IPEnabled='True'" _
    Call EnableIPSec (0), (80,135,139,443,445), (137,138)
```

Listing 362: 2_04_010a.bat

Um die TCP/IP-Filterung wieder zu deaktivieren, wenden Sie die `EnableIPFilterSec`-Methode an und übergeben ihr den Wert `False`. Damit werden automatisch die einzelnen Port- und Protokollfilter wieder aufgehoben.

```
@Echo off
WMIC NicConfig Call EnableIPFilterSec "False"
```

Listing 363: 2_04_010b.bat

Die Lösung mit VBScript

Um die TCP/IP-Filterung per VBScript zu aktivieren und zu konfigurieren, gehen Sie genauso vor wie auch in der Eingabeaufforderung. Sie nutzen WMI und stellen zunächst eine Verbindung zur WMI-Klasse `Win32_NetworkAdapterConfiguration` her, aktivieren dann mit der `EnableIPFilterSec`-Methode die TCP/IP-Filterung für alle Netzwerkadapter, indem Sie den Wert `True` als Parameter übergeben, und setzen dann mit der `EnableIPSec`-Methode die einzelnen Filter, wofür Sie je ein Array für die freizuschaltenden TCP-Ports, für die UDP-Ports und auch für die Protokolle übergeben.

Option Explicit

```
Dim oWMI
Dim strcomputer
Dim arrProtocols
Dim arrTCPPorts
Dim arrUDPPorts
Dim NIC
Dim colNIC
Dim objItem
Dim intValue1
Dim intValue2
```

```
strComputer = "."
arrProtocols = Array(0)
arrTCPPorts = Array(80,135,139,443,445)
arrUDPPorts = Array(137,139)
```

```
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set NIC = oWMI.Get("Win32_NetworkAdapterConfiguration")
intValue1 = NIC.EnableIPFilterSec("True")
```

```
Set colNIC = oWMI.ExecQuery("SELECT * FROM Win32_NetworkAdapterConfiguration
-
    WHERE IPEnabled='True'")
For Each objItem in colNIC
    intValue2 = objItem.EnableIPSec(arrTCPPorts, arrUDPPorts, arrProtocols)
Next
```

```
Set oWMI = Nothing
```

Listing 364: 2_04_010a.vbs

Wenn Sie die TCP/IP-Filterung wieder deaktivieren möchten, so wenden Sie lediglich die `EnableIPFilterSec`-Methode der WMI-Klasse `Win32_NetworkAdapterConfiguration`-Klasse an und übergeben mit dem Aufruf der Methode den Wert `False`.

Option Explicit

```
Dim oWMI
Dim strcomputer
Dim NIC
Dim intValue
```

Listing 365: 2_04_010b.vbs

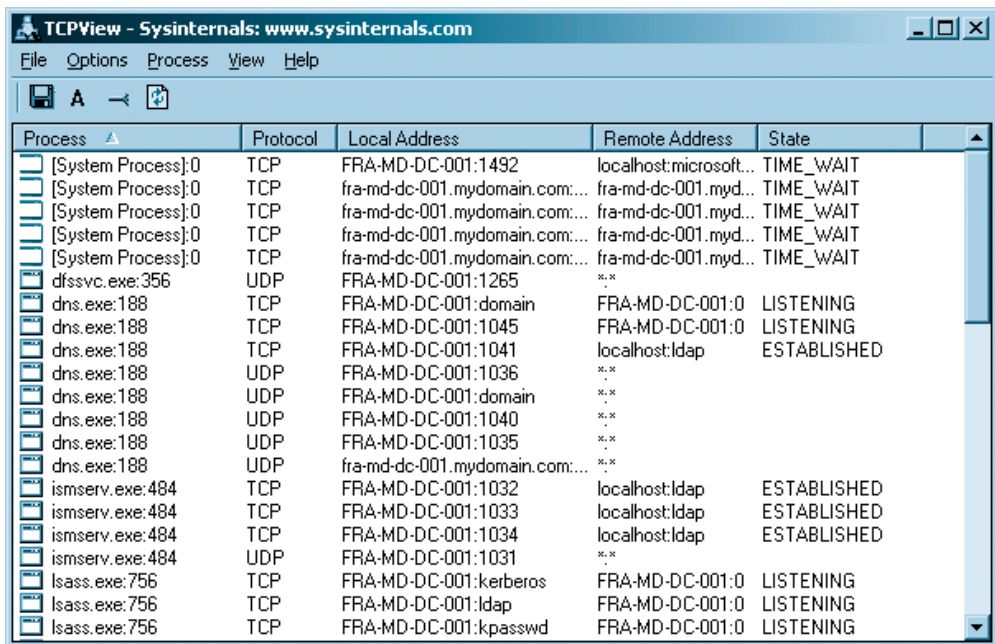
```
strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set NIC = oWMI.Get("Win32_NetworkAdapterConfiguration")
intValue = NIC.EnableIPFilterSec("False")

Set oWMI = Nothing
```

Listing 365: 2_04_010b.vbs (Forts.)

93 Aktive TCP/IP-Verbindungen abfragen

Wenn Sie auf die Schnelle herausfinden müssen, welche TCP- und UDP-Verbindungen gerade geöffnet sind und auf welchen Ports auf eingehende Verbindungen »gelauscht« wird, so bleibt nur der Einsatz eines externen Tools wie beispielsweise *TCPView* (www.sysinternals.com), da Windows Server 2003 unter grafischer Oberfläche leider keine eigenen Bordmittel bereitstellt.



Process	Protocol	Local Address	Remote Address	State
[System Process]:0	TCP	FRA-MD-DC-001:1492	localhost:microsoft...	TIME_WAIT
[System Process]:0	TCP	fra-md-dc-001.mydomain.com:...	fra-md-dc-001.myd...	TIME_WAIT
[System Process]:0	TCP	fra-md-dc-001.mydomain.com:...	fra-md-dc-001.myd...	TIME_WAIT
[System Process]:0	TCP	fra-md-dc-001.mydomain.com:...	fra-md-dc-001.myd...	TIME_WAIT
[System Process]:0	TCP	fra-md-dc-001.mydomain.com:...	fra-md-dc-001.myd...	TIME_WAIT
dfssvc.exe:356	UDP	FRA-MD-DC-001:1265	*,*	
dns.exe:188	TCP	FRA-MD-DC-001:domain	FRA-MD-DC-001:0	LISTENING
dns.exe:188	TCP	FRA-MD-DC-001:1045	FRA-MD-DC-001:0	LISTENING
dns.exe:188	TCP	FRA-MD-DC-001:1041	localhost:ldap	ESTABLISHED
dns.exe:188	UDP	FRA-MD-DC-001:1036	*,*	
dns.exe:188	UDP	FRA-MD-DC-001:domain	*,*	
dns.exe:188	UDP	FRA-MD-DC-001:1040	*,*	
dns.exe:188	UDP	FRA-MD-DC-001:1035	*,*	
dns.exe:188	UDP	fra-md-dc-001.mydomain.com:...	*,*	
ismserv.exe:484	TCP	FRA-MD-DC-001:1032	localhost:ldap	ESTABLISHED
ismserv.exe:484	TCP	FRA-MD-DC-001:1033	localhost:ldap	ESTABLISHED
ismserv.exe:484	TCP	FRA-MD-DC-001:1034	localhost:ldap	ESTABLISHED
ismserv.exe:484	UDP	FRA-MD-DC-001:1031	*,*	
lsass.exe:756	TCP	FRA-MD-DC-001:kerberos	FRA-MD-DC-001:0	LISTENING
lsass.exe:756	TCP	FRA-MD-DC-001:ldap	FRA-MD-DC-001:0	LISTENING
lsass.exe:756	TCP	FRA-MD-DC-001:kpasswd	FRA-MD-DC-001:0	LISTENING

Abbildung 142: Die Übersicht der aktiven TCP- und UDP-Verbindungen

Wenn Sie die aktiven TCP- und UDP-Verbindungen des Öfteren abfragen müssen und dies vorzugsweise unter grafischer Oberfläche tun möchten, so laden Sie zunächst das kostenlose Tool *TCPView* von der zuvor genannten Internetseite herunter und extrahieren die Dateien in einen beliebigen Ordner auf der lokalen Festplatte.

Anschließend starten Sie das Tool durch einen Doppelklick auf *TCPView.exe*, woraufhin alle aktiven TCP- und UDP-Verbindungen und auch die geöffneten Ports recht übersichtlich dargestellt werden.

Die Lösung per Eingabeaufforderung

Mit dem *Netstat*-Befehl, der standardmäßig zum Befehlsumfang von Windows Server 2003 gehört, können Sie sich jederzeit die aktiven TCP- und UDP-Verbindungen wie auch die geöffneten TCP- und UDP-Ports anzeigen lassen. Des Weiteren ermöglicht der *Netstat*-Befehl das Abfragen der Ethernet-Statistiken und auch das Ausgeben der eingerichteten IP-Routen.

Wenn Sie beispielsweise wissen möchten, welche TCP- und UDP-Verbindungen gerade hergestellt und damit aktiv sind, so geben Sie in der Eingabeaufforderung lediglich *Netstat* ein. Sie erhalten daraufhin eine Aufstellung aller aktiven Verbindungen, die Sie per Ausgabeumleitung auch in eine Datei ausgeben können.

```
@Echo off  
Netstat > Result.txt
```

Listing 366: 2_04_011a.bat

Wenn Sie neben den aktiven Verbindungen auch eine Aufstellung der geöffneten Ports, die auf eingehende Verbindungen warten, erzeugen möchten, so wenden Sie zusätzlich den Parameter *-a* an, woraufhin Sie eine Übersicht der aktiven Verbindungen und auch die der empfangsbereiten Ports erhalten. Und mit dem zusätzlichen Parameter *-n* können Sie bestimmen, dass die Ports der Windows-Basisdienste (LDAP, Domain, Kerberos ...) nicht unter Angabe des Dienstnamens ausgegeben werden, sondern ausschließlich deren Portnummer angezeigt wird.

```
@Echo off  
Netstat -a > Result.txt
```

Listing 367: 2_04_011b.bat

Um herauszufinden, welche Systemdienste und welche Applikationen aktive Verbindungen nutzen bzw. auf eingehende Verbindungen warten, so wenden Sie den Befehl *Netstat -b* an, woraufhin nicht nur die aktiven Verbindungen und empfangsbereiten Ports, sondern auch die dazugehörigen Prozesse unter Angabe der ausführbaren Datei und der Prozess-ID ausgegeben werden.

```
@Echo off  
Netstat -b > Result.txt
```

Listing 368: 2_04_011c.bat

Die Lösung mit VBScript

Die einfachste Möglichkeit, die aktiven Verbindungen und geöffneten Ports aus einem VBScript heraus abzufragen, stellt die Ausführung des `Netstat`-Befehls der Eingabeaufforderung dar, da Sie diesen Befehl mit der `Run`-Methode des `Shell`-Objektes auch per VBScript aufrufen können.

Das folgende Listing zeigt, wie Sie die aktiven Verbindungen und geöffneten Ports abfragen und das Ergebnis per Ausgabeumleitung in die Datei *Result.txt* umleiten können.

```
Option Explicit
```

```
Dim oShell
```

```
Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Netstat -a > Result.txt"
```

```
Set oShell = Nothing
```

Listing 369: 2_04_011.vbs

94 Welche Freigaben sind eingerichtet?

Wenn Sie auf die Schnelle unter grafischer Oberfläche herausfinden möchten, welche Freigaben auf einem bestimmten Computersystem eingerichtet sind, so nehmen Sie die Management-Konsole FREIGELEGEBENE ORDNER zu Hilfe und lassen sich die Freigaben einfach anzeigen. Öffnen Sie hierzu das Kontextmenü des ARBEITSPLATZ-Symbols, und wählen Sie VERWALTEN aus, um in die COMPUTERVERWALTUNG zu gelangen. Erweitern Sie im Konsolenstamm der COMPUTERVERWALTUNG den Eintrag FREIGELEGEBENE ORDNER, und markieren Sie den Eintrag FREIGABEN, die eingerichteten Freigaben werden daraufhin in der rechten Fensterhälfte unter Angabe des Freigabenamens, des Pfades, des Typs und der derzeit aktiven Verbindungen aufgelistet (siehe Abbildung 143).

Die Lösung per Eingabeaufforderung

Die eingerichteten Freigaben lassen sich natürlich auch von der Eingabeaufforderung aus abfragen. Hierfür verwenden Sie im einfachsten Fall den Befehl `Net Share`, der ohne Angabe weiterer Parameter eine Aufstellung aller eingerichteten Freigaben erzeugt und dabei den Namen der Freigabe, den Pfad zur freigegebenen Ressource und (sofern vergeben) auch die Kurzbeschreibung der Freigabe am Bildschirm ausgibt.

```
@Echo off
Net Share
```

Listing 370: 2_04_012a.bat

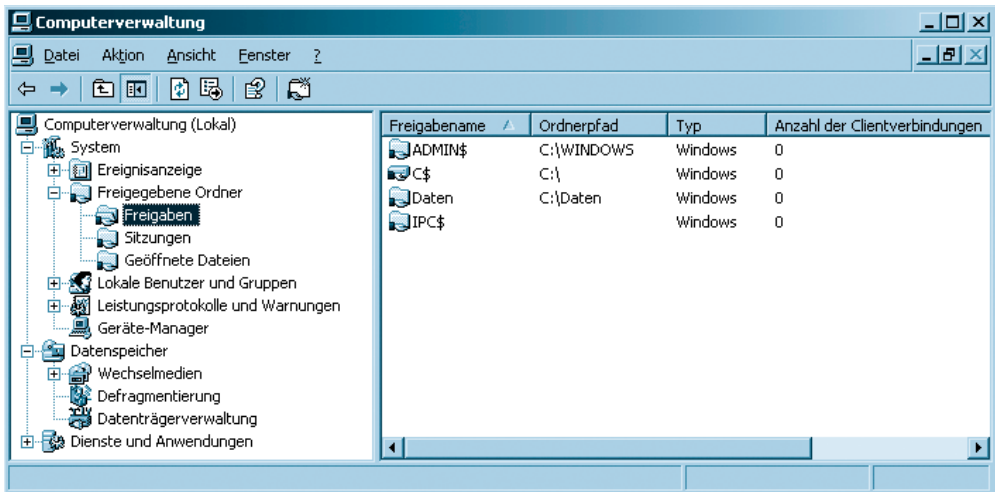


Abbildung 143: Die eingerichteten Freigaben anzeigen lassen

Wenn Sie dem `Net Share`-Befehl den Namen der Freigabe als Parameter übergeben, so können Sie sich die erweiterten Informationen zur jeweiligen Freigabe anzeigen lassen. Dabei werden nicht nur der Name der Freigabe, der Pfad zur Ressource und auch die Kurzbezeichnung ausgegeben, sondern auch die zulässige Anzahl der gleichzeitigen Zugriffe, die aktuell mit der Freigabe verbundenen Benutzer sowie die Offline-Einstellung (Caching-Option).

```
@Echo off
Net Share Daten
```

Listing 371: 2_04_012b.bat

Eine weitere Möglichkeit, die eingerichteten Freigaben abzufragen, bietet der `WMIC`-Befehl in Verbindung mit dem Alias `Share`, womit Sie von der Eingabeaufforderung aus eine WMI-Abfrage gegen die WMI-Klasse `Win32_Share` durchführen und die Eigenschaften einer Freigabe (siehe nachfolgende Tabelle) auslesen können.

Eigenschaft	Beschreibung
AccessMask	Bit-Muster für die Freigabeberechtigungen. Diese Eigenschaft wird jedoch nicht mehr verwendet und besitzt daher keinen Wert
AllowMaximum	Enthält den Wert <i>True</i> , wenn die Anzahl der gleichzeitigen Verbindungen zur Freigabe nicht limitiert ist und die Höchstanzahl zugelassen wird. Anderenfalls ist der Wert gleich <i>False</i>
Caption	Kurzbeschreibung der Freigabe (einzeilig)
Description	Kurzbeschreibung der Freigabe

Tabelle 52: Die Eigenschaften der WMI-Klasse »Win32_Share«

Eigenschaft	Beschreibung
InstallDate	Datum und Uhrzeit, wann die Freigabe eingerichtet wurde
MaximumAllowed	Legt die maximale Anzahl der gleichzeitigen Verbindungen zur Freigabe fest
Name	Name der Freigabe
Path	Pfad zur freigegebenen Ressource
Status	Aktueller Status der Freigabe (<i>OK, Error, Degraded, Unknown, Pred Fail, Starting, Stopping, Service, Stressed, Nonrecover, NoContact, LostComm</i>)
Type	Typ der Freigabe (<i>0 = Disk Drive, 1 = Print Queue, 2 = Device, 3 = IPC, 2147483648 = Disk Drive Admin, 2147483649 = Print Queue Admin, 2147483650 = Device Admin, 2147483651 = IPC Admin</i>)

Tabelle 52: Die Eigenschaften der WMI-Klasse »Win32_Share« (Forts.)

Um beispielsweise eine Auflistung aller eingerichteten Freigaben unter Angabe aller Eigenschaften am Bildschirm auszugeben, geben Sie in der Eingabeaufforderung den Befehl WMIC Share ein.

```
@Echo off
WMIC Share
```

Listing 372: 2_04_012c.bat

Möchten Sie dagegen nur bestimmte Eigenschaften am Bildschirm ausgeben, so können Sie die Ausgabe der Freigabe-Eigenschaften mit der `Get`-Anweisung auf bestimmte Eigenschaften limitieren. Um für alle Freigaben nur den Namen der Freigabe und auch den Pfad zur Ressource aufzulisten, geben Sie in der Eingabeaufforderung die folgende WMIC-Befehlszeile ein:

```
@Echo off
WMIC Share Get Name, Path
```

Listing 373: 2_04_012d.bat

Die Lösung mit VBScript

Um eine Aufstellung der bereits eingerichteten Freigaben per VBScript zu erstellen, gehen Sie ähnlich vor wie in der Eingabeaufforderung. Sie stellen eine Verbindung zur WMI-Klasse `Win32_Share` her und fragen die Werte der gewünschten Eigenschaften (siehe Tabelle 52: auf Seite 467) ab.

Das nachfolgende Listing gibt beispielsweise für alle eingerichteten Freigaben die Werte der Eigenschaften `Name`, `Path`, `Caption`, `AllowMaximum`, `MaximumAllowed`, `Status`

und Type am Bildschirm aus. Hierzu wird im ersten Schritt eine Verbindung zum WMI-Namespace `root\cimv2` des betreffenden Computersystems hergestellt und mit der `ExecQuery`-Methode eine Selektion aller Freigaben vorgenommen.

Die `ExecQuery`-Methode liefert das Ergebnis (die Auflistung der Freigaben) in Form einer Kollektion zurück, und die Werte der Eigenschaften können daraufhin innerhalb einer `For Each`-Schleife abgefragt und mit der `WScript.Echo`-Anweisung am Bildschirm ausgegeben werden.

Option Explicit

```
Dim oWMI
Dim strComputer
Dim colShare
Dim objItem

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colShare = oWMI.ExecQuery("SELECT * FROM Win32_Share")

For Each objItem in colShare
    WScript.Echo "Name: " & objItem.Name
    WScript.Echo "Path: " & objItem.Path
    WScript.Echo "Caption: " & objItem.Caption
    WScript.Echo "Allow Maximum: " & objItem.AllowMaximum
    WScript.Echo "Maximum Allowed: " & objItem.MaximumAllowed
    WScript.Echo "Status: " & objItem.Status
    WScript.Echo "Type: " & objItem.Type & VBCRLF
Next

Set oWMI = Nothing
```

Listing 374: 2_04_012.vbs

95 Neue Ordnerfreigaben einrichten

Mit der Voraussetzung, dass in den Eigenschaften der LAN-Verbindung die DATEI- UND DRUCKERFREIGABE FÜR MICROSOFT-NETZWERKE eingerichtet und auch der Systemdienst SERVER gestartet ist, so können Sie jederzeit neue Freigaben einrichten und damit bestimmte Ressourcen (Ordner, Laufwerke, Drucker, Scanner) anderen Netzwerkteilnehmern zur Verfügung stellen.

Wenn Sie beispielsweise auf einem der Server-Systeme einen Ordner namens *Software* eingerichtet haben und darin die Installationsdateien für diverse Applikationen aufbewahren, diesen Ordner nun via Netzwerkverbindung auch von anderen Computersystemen aus zugänglich machen wollen, so brauchen Sie den Ordner nur ent-

sprechend freizugeben. Andere Netzwerkteilnehmer können sich dann mit dieser Freigabe verbinden und (je nach Berechtigung) den Inhalt des Ordners einsehen, bearbeiten und auch löschen.

Um einen Ordner freizugeben, können Sie zum einen den WINDOWS EXPLORER nutzen und aus dem Kontextmenü des betreffenden Ordners heraus den Eintrag FREIGABE UND SICHERHEIT aufrufen, womit Sie in die Freigabe-Einstellungen gelangen und die Freigabe einrichten können. Sie können aber auch die Management-Konsole FREIGE- GEBENE ORDNER verwenden, um die Freigaben mit Hilfe eines Assistenten einzurichten. Letzteres hat den Vorteil, dass sich Freigaben nicht nur sehr einfach einrichten, sondern zudem auch sehr leicht verwalten lassen.

Wenn Sie eine neue Freigabe einrichten und z.B. den Ordner C:\Software für den gemeinsamen Zugriff zur Verfügung stellen möchten, so wählen Sie im Kontextmenü des Arbeitsplatz-Symbols VERWALTEN aus, um die COMPUTERVERWALTUNG zu starten. In der COMPUTERVERWALTUNG erweitern Sie erst SYSTEM, dann FREIGE GEBENE ORDNER und markieren unterhalb davon den Eintrag FREIGABEN. Im Kontextmenü von FREIGABEN wählen Sie NEUE FREIGABE... aus, um den ORDNERFREIGABE-ASSISTENT aufzurufen.

Schließen Sie den Willkommensbildschirm mit einem Klick auf WEITER, und wählen Sie im nächsten Schritt den Ordner aus, den Sie freigeben möchten. Legen Sie daraufhin den Namen für die Freigabe fest, wobei per Default der Name des Ordners auch als Name für die Freigabe verwendet wird. Da Freigaben grundsätzlich in der Netzwerkumgebung als verfügbare Netzwerkressource angezeigt werden, können Sie die Freigabe durch Anfügen des \$-Zeichens verstecken und damit verhindern, dass die Freigabe im WINDOWS EXPLORER und auch in der Netzwerkumgebung angezeigt wird. Der Zugriff auf die Freigabe ist dann nur denen vorbehalten, die von der Freigabe in Kenntnis gesetzt sind und den Freigabe-Pfad kennen.

Klicken Sie auf WEITER, um die Berechtigungen für den Zugriff auf die Freigabe festzulegen. Sie können zwischen den folgenden Optionen wählen, wobei per Default die Option ALLE BENUTZER HABEN SCHREIBGESCHÜTZTEN ZUGRIFF verwendet wird.

- ▶ Alle Benutzer haben schreibgeschützten Zugriff.
- ▶ Administratoren haben Vollzugriff, andere Benutzer haben schreibgeschützten Zugriff.
- ▶ Administratoren haben Vollzugriff, andere Benutzer haben schreibgeschützten Zugriff und Schreibzugriff.
- ▶ Benutzerdefinierte Freigabe- und Ordnerberechtigungen verwenden.

Legen Sie die gewünschte Berechtigung fest, und klicken Sie anschließend auf FERTIG STELLEN, um die Freigabe einzurichten und die zuvor markierte Berechtigungsoption zu setzen.

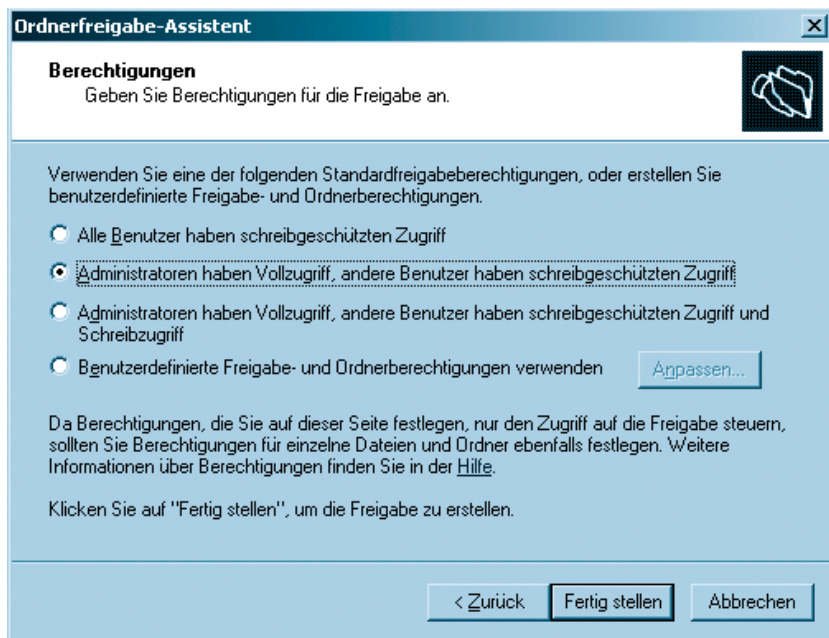


Abbildung 144: Die Berechtigungen für die neue Freigabe definieren

Die Lösung per Eingabeaufforderung

Neue Freigaben lassen sich auch von der Eingabeaufforderung aus einrichten. Hierfür stellt der Windows Server 2003 den `Net Share`-Befehl und den `WMIC Share`-Befehl bereit. Wenn Sie beispielsweise mit `Net Share` eine neue Freigabe einrichten möchten, so geben Sie in der Eingabeaufforderung den Befehl `Net Share Freigabename=Pfad` ein, wobei Sie `Freigabename` durch den Namen der Freigabe und `Pfad` durch den Pfad der freizugebenden Ressource ersetzen. Des Weiteren können Sie der `Net Share`-Befehlszeile die folgenden optionalen Parameter anfügen, um die Freigabe nicht nur anzulegen, sondern zeitgleich auch zu konfigurieren:

- ▶ `/Grant`: Der Parameter `/Grant`: erlaubt die Angabe der Freigabeberechtigung. Der Parameter erwartet zunächst die Angabe des Benutzers oder der Gruppe wie auch die entsprechende Berechtigung selbst. Mögliche Werte für die Berechtigung sind `Read`, `Change` und `Full` (z.B. `/Grant:Administratoren, Full`).
- ▶ `/Users`: Mit dem Parameter `/Users`: können Sie die maximale Anzahl der gleichzeitigen Verbindungen zur Freigabe bestimmen, indem Sie dem Parameter `/Users`: die gewünschte Anzahl anfügen (z.B. `/Users:5`).
- ▶ `/Unlimited`: Der Parameter `/Unlimited` gibt an, dass die Anzahl gleichzeitiger Verbindungen nicht limitiert werden soll. Mit diesem Parameter wird die Höchstanzahl an gleichzeitigen Verbindungen zugelassen (dieser Parameter kann nicht mit `/Users:x` verwendet werden).

- ▶ **/Remark:** Der Parameter `/Remark:` erlaubt die Angabe einer kurzen Bezeichnung für die Freigabe (z.B. `/Remark: "Freigabe für Software-Installationsdateien"`).
- ▶ **/Cache:** Der Parameter `/Cache:` erlaubt die Festlegung der Offline-Einstellung (Caching-Option). Mögliche Werte sind `Manual`, `Documents`, `Programs`, `No` (z.B. `/Cache:Manual`).

Die nachfolgende `Net Share`-Befehlszeile gibt den Ordner `C:\Software` frei und verwendet dabei den Freigabenamen `Software`. Gleichzeitig wird der Gruppe `Administratoren` ein Vollzugriff gewährt, die maximale Anzahl der gleichzeitigen Verbindungen auf 5 limitiert und die Kurzbeschreibung `Software-Installationsdateien` vergeben:

```
@Echo off
Net Share Software=C:\Software /Grant:Administratoren,Full _
/users:5 /Remark:"Software-Installationsdateien"
```

Listing 375: 2_04_013a.bat

Gleiches lässt sich auch mit dem `WMIC`-Befehl in Verbindung mit dem Alias `Share` realisieren, indem Sie mit `Call` die `Create`-Methode zum Erstellen einer neuer Freigabe aufrufen und dabei folgende Parameter übergeben (die Reihenfolge muss dabei unbedingt eingehalten werden):

- ▶ **Access:** Dieser Parameter gilt zwar den Berechtigungen, wird jedoch nicht mehr verwendet. Übergeben Sie daher als ersten Parameter die Zeichenfolge `"`.
- ▶ **Description:** Mit diesem Parameter können Sie eine Kurzbezeichnung für die Freigabe festlegen (z.B. `"Freigabe für Software-Installationsdateien"`).
- ▶ **MaximumAllowed:** Der Parameter `MaximumAllowed` ermöglicht die Angabe der maximalen Anzahl gleichzeitiger Verbindungen zur Freigabe (z.B. `"5"`).
- ▶ **Name:** Mit diesem Parameter legen Sie den Namen der Freigabe fest (z.B. `"Software"`).
- ▶ **Password:** Ermöglicht das Festlegen des Kennwortes bei Verwendung einer kennwortgeschützten Freigabe (Windows 95/98/ME), bei Verwendung der Freigabesicherheit auf Benutzerebene, wie dies unter Windows Server 2003 der Fall ist, wird dieser Parameter jedoch ignoriert. Übergeben Sie daher die Zeichenfolge `"`.
- ▶ **Path:** Mit diesem Parameter legen Sie den Pfad zur Ressource fest, die freigegeben werden soll (z.B. `"C:\Software"`).
- ▶ **Type:** Erlaubt das Festlegen des Freigabetyps. Mögliche Werte sind `0` = Disk Drive, `1` = Print Queue, `2` = Device, `3` = IPC, `2147483648` = Disk Drive Admin, `2147483649` = Print Queue Admin, `2147483650` = Device Admin, `2147483651` = IPC Admin.

Die nachfolgende `WMIC`-Befehlszeile gibt den Ordner `C:\Software` frei und verwendet dabei den Freigabenamen `Software`. Des Weiteren wird die maximale Anzahl an gleichzeitigen Verbindungen zur Freigabe auf 5 limitiert und der Freigabetyp `0` (= *Disk Drive*) bestimmt.

```
@Echo off
WMIC Share Call Create "", "Software-Installationsdateien", "5", _
    "Software", "", "C:\Software", 0
```

Listing 376: 2_04_013b.bat

Die Lösung mit VBScript

Mit der Create-Methode der WMI-Klasse Win32_Share lässt sich eine Ressource auch per VBScript freigeben. Hierzu stellen Sie im ersten Schritt eine Verbindung zum WMI-Namespace `root\cimv2` her und erzeugen eine Referenz zur WMI-Klasse Win32_Share. Anschließend wenden Sie die Create-Methode an, um die neue Freigabe zu erstellen. Der Create-Methode übergeben Sie die folgenden Parameter (diese wurden bereits im vorherigen Abschnitt anhand des WMIC-Befehls erläutert), wobei Sie auch hier wieder auf die angegebene Reihenfolge achten müssen:

- ▶ Path
- ▶ Name
- ▶ Type
- ▶ MaximumAllowed
- ▶ Description

Um beispielsweise den Ordner `C:\Software` freizugeben, dabei den Freigabenamen *Software* zu verwenden, die Anzahl gleichzeitiger Verbindungen auf 5 zu limitieren und die Kurzbeschreibung *Software-Installationsdateien* zu verwenden, können Sie das folgende Listing verwenden.

```
Option Explicit

Dim oWMI
Dim oShare
Dim strComputer
Dim intValue

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set oShare = oWMI.Get("Win32_Share")
intValue = oShare.Create ("C:\Software", "Software", "0", "5", _
    "Software-Installationsdateien")

Set oWMI = Nothing
```

Listing 377: 2_04_013.vbs

96 Bestehende Freigaben ändern

Eine bestehende Freigabe können Sie natürlich auch im Nachhinein noch ändern. Sei es, dass Sie eine Kurzbeschreibung hinzufügen oder ändern wollen, die maximale Anzahl der gleichzeitigen Verbindungen zur Freigabe modifizieren möchten oder einfach nur die Offline-Einstellungen bearbeiten müssen. Lediglich der Freigabename und auch der Pfad zur Freigabe lassen sich nur durch Löschen und Neuanlegen der Freigabe verändern.

Um eine bestehende Freigabe zu ändern, rufen Sie zunächst die Computerverwaltung (hierfür wählen Sie im Kontextmenü des Arbeitsplatz-Symbols den Eintrag **Verwalten** aus) auf und wechseln dann – innerhalb der Computerverwaltung – in die Management-Konsole **Freigegebene Ordner**. Im Ordner **Freigaben** markieren Sie die Freigabe, die Sie bearbeiten möchten, und führen einen Doppelklick auf die Freigabe aus oder wählen im Kontextmenü den Eintrag **Eigenschaften** aus.

Auf der Registerkarte **ALLGEMEIN** können Sie nun die Kurzbeschreibung der Freigabe modifizieren, die Zugriffsbegrenzung verändern und – mit einem Klick auf **OFFLINE-EINSTELLUNGEN...** – auch festlegen, ob Dateien und Programme für den Offline-Zugriff verfügbar gemacht werden sollen oder nicht.

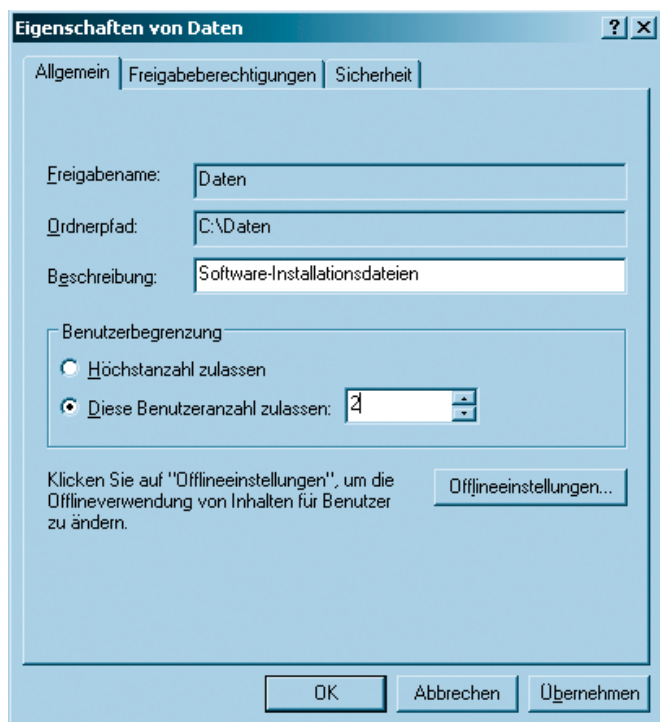


Abbildung 145: Eine bestehende Freigabe ändern

Die Lösung per Eingabeaufforderung

Mit dem `Net Share`-Befehl können Sie eine Freigabe auch von der Eingabeaufforderung aus bearbeiten. Wenn Sie beispielsweise die Zugriffsbegrenzung der Freigabe *Software* auf 2 limitieren und die Kurzbeschreibung *Software-Installationsdateien* hinzufügen oder überschreiben möchten, so wenden Sie den `Net Share`-Befehl unter Angabe der zu bearbeitenden Freigabe (*Software*) an und übergeben mit `/Users:` die maximale Anzahl der gleichzeitigen Verbindungen und mit `/Remark` die gewünschte Kurzbeschreibung.

```
@Echo off
Net Share Software /Users:2 /Remark:"Software-Installationsdateien"
```

Listing 378: 2_04_014a.bat

Alternativ dazu können Sie auch den `WMIC`-Befehl mit dem Alias `Share` verwenden, um eine bestehende Freigabe zu verändern. Hierzu wenden Sie den `WMIC Share`-Befehl an, selektieren mit der `Where`-Klausel die entsprechende Freigabe und rufen dann die `SetShareInfo`-Methode auf, der Sie folgende Parameter übergeben:

- ▶ **Access:** Der erste Parameter gilt dem Bit-Muster der Berechtigungen, der jedoch nicht mehr verwendet wird. Übergeben Sie daher als ersten Parameter die Zeichenfolge `" "`.
- ▶ **Description:** Mit dem zweiten Parameter können Sie die Kurzbeschreibung für die Freigabe festlegen, die in Anführungszeichen gesetzt sein muss.
- ▶ **MaxiumAllowed:** Mit dem dritten und letzten Parameter haben Sie die Möglichkeit, die maximale Anzahl der gleichzeitigen Verbindungen zur Freigabe zu beschränken. Wenn die Anzahl gleichzeitiger Verbindungen nicht limitiert werden soll, so übergeben Sie die Zeichenfolge `" "`.

Die nachfolgende `WMIC Share`-Befehlszeile zeigt ein Beispiel dafür, wie Sie die Kurzbeschreibung *Software-Installationsdateien* setzen und die maximale Anzahl der gleichzeitigen Verbindungen auf 2 limitieren.

```
@Echo off
WMIC Share WHERE "Name='Software'" Call SetShareInfo " ", _
    "Software-Installationsdateien", "2"
```

Listing 379: 2_04_014b.bat

Die Lösung mit VBScript

Um eine bestehende Freigabe per VBScript zu modifizieren, nutzen Sie die `SetShareInfo`-Methode der WMI-Klasse `Win32_Share`, der Sie die gleichen Argumente übergeben wie auch der `WMIC Share`-Befehlszeile (siehe vorheriger Abschnitt).

Wenn Sie beispielsweise die Kurzbeschreibung *Software-Installationsdateien* für die Freigabe *Software* festlegen und dabei auch die maximale Anzahl der gleichzeitigen Verbindungen auf 2 limitieren möchten, so stellen Sie erst eine Verbindung zum WMI-Namespace `root\cimv2` her, selektieren dann per `ExecQuery` die entsprechende Freigabe und wenden dann die `SetShareInfo`-Methode unter Angabe der Argumente `Access`, `Description` und `MaximumAllowed` an.

Option Explicit

```
Dim oWMI
Dim colShare
Dim strComputer
Dim objItem
Dim intValue
```

```
strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colShare = oWMI.ExecQuery("SELECT * FROM Win32_Share _
    WHERE Name='Software'")
```

```
For Each objItem in colShare
    intValue = objItem.SetShareInfo "", "Software-Installationsdateien", "2"
Next
```

```
Set oWMI = Nothing
```

Listing 380: 2_04_014.vbs

97 Freigabeberechtigungen bearbeiten

Der Zugriff auf eine Freigabe, der vom gleichen Computersystem aus stattfindet, auf dem die Freigabe selbst eingerichtet ist, setzt lediglich eine entsprechende NTFS-Berechtigung für den jeweiligen Benutzer (oder für das Mitglied einer Gruppe) voraus. Wenn der Zugriff dagegen über eine Netzwerkverbindung hinweg erfolgt (dies ist bei Freigabe-Zugriffen natürlich die Regel), so muss eine explizite Freigabeberechtigung gesetzt sein, da der Zugriff anderenfalls aufgrund fehlender Berechtigungen scheitert.

Mit dem Einrichten einer neuen Freigabe wird per Default der Gruppe *Jeder* eine Leseberechtigung zugewiesen, womit alle authentifizierte Benutzer den Inhalt einer Freigabe lesen können. Um nun weitere Benutzer und Gruppen für den Freigabezugriff zu berechtigen und beispielsweise Änderungs- oder gar Vollzugriffsrechte zu erteilen, nehmen Sie die entsprechenden Benutzer und Gruppen in der Freigabe-ACL auf und legen die entsprechenden Berechtigungen fest.

Hierzu öffnen Sie die Management-Konsole FREIGELEGEBENE ORDNER, die Bestandteil der COMPUTERVERWALTUNG ist, markieren im Ordner FREIGABEN die entsprechende Freigabe und öffnen aus dem Kontextmenü heraus die EIGENSCHAFTEN. In den EIGENSCHAFTEN der Freigabe wechseln Sie auf die Registerkarte FREIGABEBERECHTIGUNGEN und fügen den Benutzer oder die Gruppe der ACL hinzu. Anschließend markieren Sie den soeben hinzugefügten Eintrag und legen durch Auswahl der Berechtigungen VOLLZUGRIFF, ÄNDERN oder LESEN fest, ob die Berechtigungen zugelassen oder explizit verweigert werden sollen.

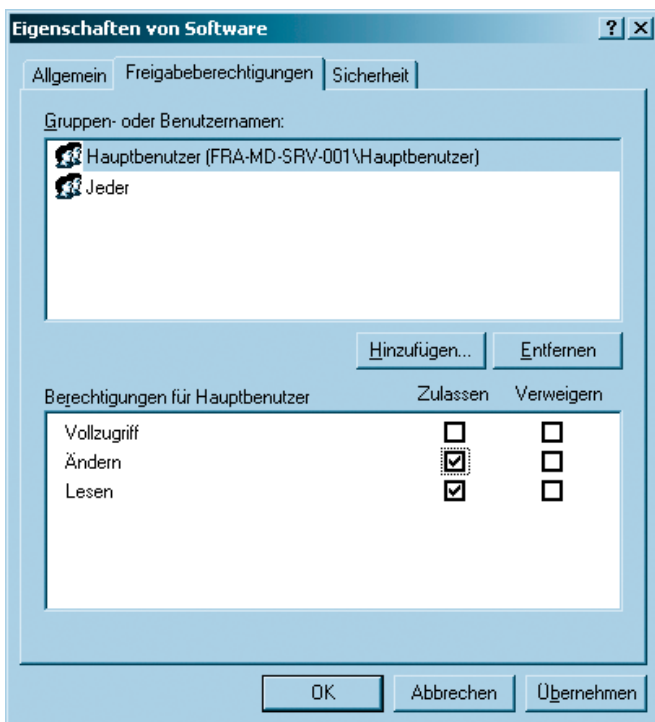


Abbildung 146: Der Gruppe »Hauptbenutzer« ein Änderungsrecht erteilen

Achtung

Für den Fall, dass Sie für einen Benutzer oder für eine Gruppe sowohl NTFS-Berechtigungen als auch Freigabeberechtigungen konfiguriert haben und sich die Rechte unterscheiden, so gilt automatisch die größere Restriktion. Will heißen: Wenn Sie z.B. der Gruppe *Hauptbenutzer* für den Ordner *C:\Software* die NTFS-Berechtigung *Ändern* und parallel dazu die Freigabeberechtigung *Vollzugriff* erteilt haben, so dürfen die Mitglieder Gruppe *Hauptbenutzer* den Inhalt der Freigabe trotz der vollen Zugriffsrechte auf Freigabeebene nur ändern, da die NTFS-Berechtigung eine geringere Berechtigung darstellt und somit eine größere Priorität hat.

Auf die gleiche Art und Weise lassen sich die einzelnen Berechtigungen natürlich auch jederzeit wieder ändern und sogar entfernen. Markieren Sie hierzu den Benutzer oder die Gruppe in der Freigabe-ACL, und aktivieren Sie die entsprechende Zulassung oder Verweigerung, oder nehmen Sie – falls Sie eine Berechtigung komplett entfernen möchten – den ACL-Eintrag mit einem Klick auf ENTFERNEN heraus.

Die Lösung per Eingabeaufforderung

Für den Fall, dass Sie die Berechtigungen für eine bereits eingerichtete Freigabe von der Eingabeaufforderung aus modifizieren möchten, so müssen Sie hierfür auf ein externes Tool zurückgreifen, da Sie mit dem `Net Share`-Befehl die Berechtigungen zwar während der Einrichtung der Freigabe setzen, im Nachhinein jedoch nicht mehr verändern können.

Sie können beispielsweise den `Subinac1`-Befehl aus den *Windows Server 2003 Resource Kit-Tools* (siehe Rezept *Wer ist der Besitzer der Datei oder des Ordners?* in der Kategorie *Dateien und Ordner*) verwenden, um die Freigabeberechtigungen zu ändern. Der `Subinac1`-Befehl erwartet dabei die Angabe des Objekttyps `/Share`, den Pfad zur Freigabe (Format `\\Computername\Freigabe`) und auch die Aktion `/Grant` (erlauben) bzw. `/Deny` (verweigern) mit dem Benutzer-/Gruppennamen und der Berechtigung (F = Vollzugriff, C = Ändern, R = Lesen).

Die nachfolgende `Subinac1`-Befehlszeile fügt der Freigabe-ACL die Gruppe *Hauptbenutzer* hinzu und weist der Gruppe den Vollzugriff zu, der mit `/Grant` erlaubt wird.

```
@Echo off
Subinac1 /Share \\FRA-MD-DC-001\Software /Grant=Hauptbenutzer=F
```

Listing 381: 2_04_015a.bat

Um einen Benutzer oder eine Gruppe wieder aus der Freigabe-ACL zu entfernen, übergeben Sie der `Subinac1`-Befehlszeile die Aktion `/Revoke` und fügen den entsprechenden Benutzer (bzw. die Gruppe) an.

```
@Echo off
Subinac1 /Share \\FRA-MD-DC-001\Software /Revoke=Hauptbenutzer
```

Listing 382: 2_04_015b.bat

Die Lösung mit VBScript

Mit der `GetAccessMask`-Methode der WMI-Klasse `Win32_Share` können Sie die Freigabeberechtigungen zwar abfragen, aber leider nicht verändern. Dies geht nur durch Aufruf eines externen Tools, beispielsweise dem `Subinac1`-Befehl aus dem *Windows Server 2003 Resource Kit* (siehe vorheriger Abschnitt), den Sie mit der `Run`-Methode des `Shell`-Objektes auch aus einem VBScript heraus aufrufen können.

Das nachfolgende Listing fügt z.B. die Gruppe *Hauptbenutzer* zur ACL der Freigabe *\\FRA-MD-DC-001\Software* hinzu und gewährt der Gruppe volle Zugriffsrechte.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Subinac1 /Share \\FRA-MD-DC-001\Software _
    /Grant=Hauptbenutzer=F"

Set oShell = Nothing
```

Listing 383: 2_04_015a.vbs

Mit dem folgenden Listing können Sie die soeben hinzugefügte Gruppe wieder aus der ACL entfernen, womit den Mitgliedern der Gruppe *Hauptbenutzer* dann die vollen Zugriffsrechte wieder entzogen werden.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Subinac1 /Share \\FRA-MD-DC-001\Software _
    /Revoke=Hauptbenutzer"

Set oShell = Nothing
```

Listing 384: 2_04_015b.vbs

98 Auf welche Freigabe wird gerade zugegriffen?

Hin und wieder ist es interessant zu wissen, ob Benutzer mit einer bestimmten Freigabe verbunden sind und gerade auf die freigegebene(n) Ressource(n) zugreifen. Beispielsweise dann, wenn die maximale Anzahl gleichzeitiger Verbindungen eingeschränkt ist und Sie herausfinden müssen, wie viele Benutzer die Ressource gerade nutzen. Oder wenn Sie eine Freigabe ändern oder gar entfernen und die aktiven Verbindungen nicht gewaltsam beenden möchten, sondern die entsprechenden Benutzer zum Beenden der Verbindung auffordern wollen.

Um nun unter grafischer Oberfläche herauszufinden, wie viele (und welche) Benutzer im Einzelnen mit den eingerichteten Freigaben verbunden sind, so nehmen Sie die Management-Konsole FREIGELEGEBENE ORDNER zu Hilfe und lassen sich die aktiven Verbindungen einfach anzeigen. Öffnen Sie hierzu das Kontextmenü des ARBEITSPLATZ-Symbols, und wählen Sie VERWALTEN aus, um die COMPUTERVERWALTUNG zu öffnen. In

der COMPUTERVERWALTUNG erweitern Sie (im Konsolenstamm der linken Fensterhälfte) den Abschnitt FREIGELEGEBENE ORDNER und unterhalb davon FREIGABEN. Sie sehen nun die Anzahl der Verbindungen zu den vorhandenen Freigaben.

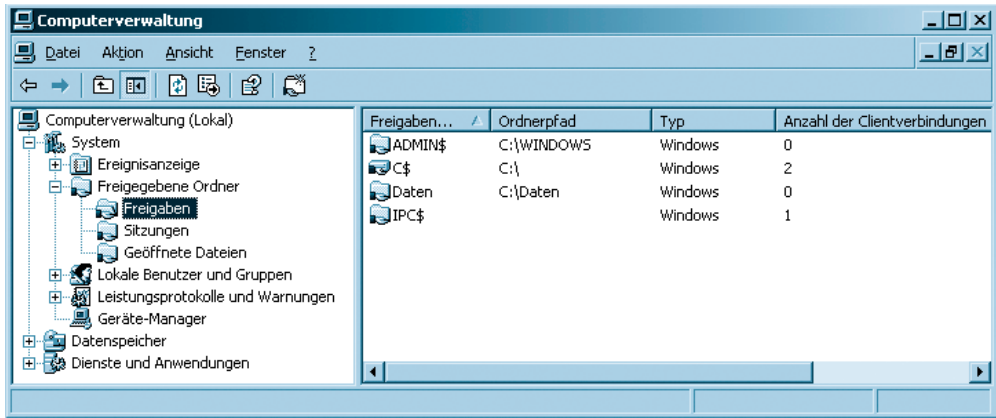


Abbildung 147: Die aktiven Verbindungen zu den einzelnen Freigaben anzeigen lassen

Wenn Sie zudem ermitteln möchten, welche Benutzer im Einzelnen mit den Freigaben verbunden sind, so markieren Sie im Konsolenstamm den Eintrag Sitzungen, woraufhin die aktiven Verbindungen unter Angabe des Benutzernamens, der IP-Adresse des Remote-Systems, die Anzahl der geöffneten Dateien wie auch die Verbindungs- und Leerlaufzeit angezeigt werden. Und mit einem Klick auf GEÖFFNETE DATEIEN können Sie sich die Dateien anzeigen lassen, die seitens des auf die Freigabe zugreifenden Benutzers geöffnet wurden.

Die Lösung per Eingabeaufforderung

Die Anzahl der aktiven Verbindungen zu einer Freigabe und auch die IP-Adresse und den Benutzernamen dessen, der gerade auf die Freigabe zugreift, können Sie natürlich auch von der Eingabeaufforderung aus abfragen. Hierzu stellt der Windows Server 2003 den Befehl `Net Session` bereit, der alle aktiven Verbindungen unter Angabe der IP-Adresse des Remote-Systems, des Benutzernamens, des Client-Typs, der Anzahl geöffneten Dateien und auch der Leerlaufzeit am Bildschirm ausgibt.

```
@Echo off
Net Session
```

Listing 385: 2_04_016.bat

Um eine Aufstellung aller geöffneten Dateien zu erhalten, können Sie in der Eingabeaufforderung zusätzlich den Befehl `Net Files` verwenden.

Die Lösung mit VBScript

Leider bietet VBScript keine entsprechenden Möglichkeiten an, womit Sie sich die aktiven Verbindungen zu einer Freigabe anzeigen lassen können. Mit der `Run`-Methode des `Shell`-Objektes können Sie jedoch den `Net Session`-Befehl der Eingabeaufforderung auch aus einem VBScript heraus aufrufen und so die gleiche Möglichkeit nutzen, die bereits im vorherigen Abschnitt beschrieben ist.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Net Session"

Set oShell = Nothing
```

Listing 386: 2_04_016.vbs

99 Aktive Freigabeverbindungen trennen

In einigen Fällen kann es erforderlich sein, aktive Verbindungen zu einer Freigabe zu trennen. Zum Beispiel dann, wenn Sie die maximale Anzahl gleichzeitiger Verbindungen zu einer Freigabe limitiert haben, die zulässigen Verbindungen allesamt belegt sind und Sie sich aus administrativen Gründen ad hoc mit der Freigabe verbinden müssen, die Anzahl der Verbindungen aber nicht erhöhen dürfen. Oder wenn Sie die Freigabe bearbeiten oder gar aufheben müssen, just zu diesem Zeitpunkt aber noch Benutzer mit der Freigabe verbunden sind.

Mit Hilfe der Management-Konsole FREIGELEGEBENE ORDNER können Sie sich nun die aktiven Verbindungen anzeigen lassen und bei Bedarf auch schließen. Hierbei sei allerdings angemerkt, dass das Beenden einer aktiven Verbindung zu Datenverlusten und auch zu korrupten Dateien führen kann, da geöffnete Dateien recht »unsanft« geschlossen werden.

Um eine Verbindung zu trennen, wählen Sie im Kontextmenü des ARBEITSPLATZ-Symbols VERWALTEN aus, um die COMPUTERVERWALTUNG zu öffnen. In der COMPUTERVERWALTUNG erweitern Sie den Eintrag FREIGELEGEBENE ORDNER und markieren unterhalb davon den Ordner SITZUNGEN, worüber alle aktiven Verbindungen zu den Freigaben des lokalen Computersystems angezeigt werden. Markieren Sie den betreffenden Eintrag in der rechten Fensterhälfte, und wählen Sie dann im Kontextmenü SITZUNG SCHLIESSEN aus (siehe Abbildung 148).

Die Lösung per Eingabeaufforderung

Um eine aktive Verbindung von der Eingabeaufforderung aus zu trennen, können Sie den `Net Session`-Befehl verwenden, der standardmäßig im Befehlsumfang von Windows Server 2003 enthalten ist.

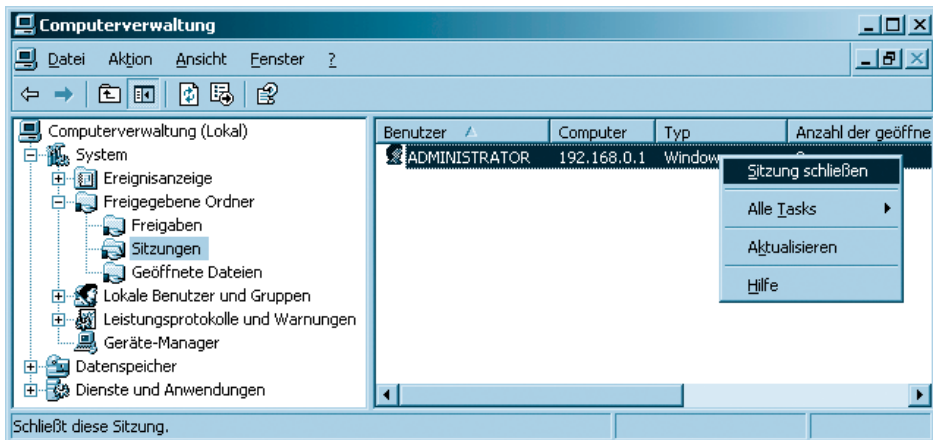


Abbildung 148: Eine aktive Verbindung zu einer Freigabe beenden

Wenden Sie zunächst den Befehl `Net Session` (ohne Angabe weiterer Parameter) an, um alle aktiven Verbindungen aufzulisten, und geben Sie anschließend `Net Session \\Computernamen /Delete` ein (wobei Sie *Computernamen* durch den Hostnamen oder die IP-Adresse ersetzen), um die Verbindung zu trennen.

```
@Echo off
Net Session \\192.168.0.1 /Delete
```

Listing 387: 2_04_017.bat

Möchten Sie alle Verbindungen auf einmal trennen, so geben Sie keinen Computernamen an. Geben Sie hierzu in der Eingabeaufforderung den Befehl `Net Session /Delete` ein.

Die Lösung mit VBScript

VBScript stellt leider keine entsprechenden Möglichkeiten bereit, womit Sie eine aktive Freigabeverbindung trennen können. Mit der `Run`-Methode des `Shell`-Objektes können Sie jedoch den `Net Session`-Befehl der Eingabeaufforderung aus einem VBScript heraus aufrufen und das Trennen einer Verbindung dennoch vornehmen.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Net Session \\192.168.0.1 /Delete"

Set oShell = Nothing
```

Listing 388: 2_04_017.vbs

100 Netzlaufwerke verbinden und trennen

Für den Fall, dass Sie ein oder auch mehrere Freigaben eingerichtet haben und des Öfteren von einem anderen Computersystem aus auf diese Freigabe(n) zugreifen müssen, so können Sie der Freigabe einen Laufwerksbuchstaben zuweisen und ein so genanntes Netzlaufwerk einrichten. Dieses Netzlaufwerk erscheint dann in den Übersichten der Laufwerke (z.B. im WINDOWS EXPLORER) wie ein eigenständiges Laufwerk, und Sie können schnell und komfortabel darauf zugreifen. Des Weiteren können Sie die Freigabe auch dauerhaft an einen Laufwerksbuchstaben binden, womit das Netzlaufwerk dann auch nach der nächsten Anmeldung an Windows Server 2003 und selbst nach einem Computerneustart noch zur Verfügung steht. Dies eignet sich insbesondere für das Einrichten von Benutzer-Heimlaufwerken oder für das zentrale Bereitstellen von An- und Abmeldeskripten.

Um ein Netzlaufwerk einzurichten, starten Sie den WINDOWS EXPLORER und wählen im Menü EXTRAS den Eintrag NETZLAUFWERK VERBINDEN... aus. Sie gelangen daraufhin in das Dialogfenster NETZLAUFWERK VERBINDEN und müssen im Feld LAUFWERK zunächst einen noch freien Laufwerksbuchstaben auswählen. Anschließend geben Sie im Feld ORDNER den Pfad zu der gewünschten Freigabe an, die an den Laufwerksbuchstaben gebunden werden soll. Sie können – wenn Sie den genauen Pfad kennen – den Pfad entweder direkt eingeben oder mit einem Klick auf DURCHSUCHEN... in der Netzwerkumgebung auch danach suchen.

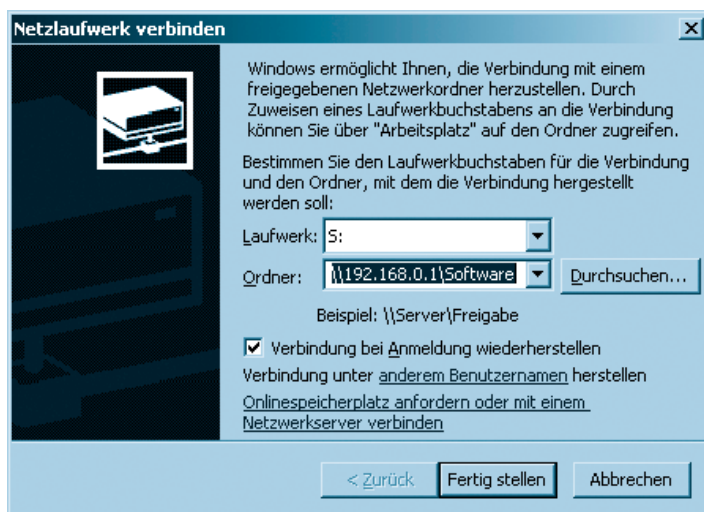


Abbildung 149: Ein Netzlaufwerk verbinden

Wenn Sie die Freigabe angegeben haben, dann können Sie mit der Option VERBINDUNG BEI ANMELDUNG WIEDERHERSTELLEN bestimmen, ob das Netzlaufwerk nur während der aktuellen Windows-Sitzung oder dauerhaft zur Verfügung stehen soll. Des Weiteren können Sie durch einen Klick auf den Link ANDEREM BENUTZERNAMEN die Anmeldeinformationen hinterlegen, die für den Zugriff auf die Freigabe erforderlich sind.

Hierbei sei angemerkt, dass lediglich der Benutzername gespeichert wird. Das Kennwort müssen Sie – sofern Sie sich abmelden oder das Computersystem neu starten – mit dem nächsten Zugriff auf das Netzlaufwerk neu eingeben. Klicken Sie anschließend auf FERTIG STELLEN, um die Erstellung des Netzlaufwerks zu beenden.

Wenn Sie ein Netzlaufwerk eingerichtet haben und die Bindung der Freigabe an den Laufwerksbuchstaben wieder lösen möchten, so starten Sie den WINDOWS EXPLORER und öffnen den Menüpunkt EXTRAS. Dort klicken Sie auf NETZLAUFWERK TRENNEN... Sie gelangen daraufhin in ein Dialogfenster, worüber alle eingerichteten Netzlaufwerke grafisch angezeigt werden. Markieren Sie das betreffende Netzlaufwerk, das Sie entfernen möchten, und klicken Sie anschließend auf OK, die Bindung wird damit aufgehoben.

Die Lösung per Eingabeaufforderung

Mit dem `Net Use`-Befehl können Sie Netzlaufwerke auch von der Eingabeaufforderung aus erstellen und trennen. Zum Erstellen eines neuen Netzlaufwerks übergeben Sie dem `Net Use`-Befehl zumindest den Laufwerksbuchstaben (gefolgt von einem Doppelpunkt) wie auch den Pfad der zu verbindenden Freigabe (wenn Sie statt des Laufwerksbuchstabens einen `*` verwenden, so nutzt Windows Server 2003 automatisch den nächsten freien Laufwerksbuchstaben). Mit dem Parameter `/User:` können Sie zusätzlich den Benutzernamen zur Authentifizierung übergeben und mit `/Persistent:Yes|No` zeitgleich auch festlegen, ob das Netzlaufwerk dauerhaft zur Verfügung gestellt werden soll oder nur während der aktiven Windows-Sitzung.

Wenn Sie beispielsweise der Freigabe `\\192.168.0.1\Software` den Laufwerksbuchstaben `S:` zuweisen, zur Authentifizierung den Benutzernamen `AdmInstall` verwenden und das Netzlaufwerk dauerhaft bereitstellen möchten, so können Sie hierfür die nachfolgende `Net Use`-Befehlszeile verwenden.

```
@Echo off
Net Use S: \\192.168.0.1\Software /User:AdmInstall /Persistent:Yes
```

Listing 389: 2_04_018a.bat

Um ein Netzlaufwerk wieder zu trennen, übergeben Sie dem `Net Use`-Befehl entweder den Laufwerksbuchstaben oder aber den Pfad zur Freigabe und fügen der Befehlszeile den Parameter `/Delete` hinzu.

```
@Echo off
Net Use S: /Delete
```

Listing 390: 2_04_018b.bat

Die Lösung mit VBScript

Wenn Sie aus einem VBScript heraus ein neues Netzlaufwerk erstellen möchten, so können Sie hierfür die `MapNetworkDrive`-Methode des `WSHNetwork`-Objektes verwenden, der Sie zumindest den Laufwerksbuchstaben wie auch die zu verbindende Freigabe als Argument übergeben müssen. Die `MapNetworkDrive`-Methode erlaubt dabei die Angabe weiterer optionaler Argumente, womit Sie zeitgleich auch das Benutzerprofil modifizieren und auch die Authentifizierungsparameter übergeben können. Folgende Argumente sind möglich:

- ▶ `strDriveLetter`: Mit dem ersten Parameter übergeben Sie den Laufwerksbuchstaben (gefolgt von einem Doppelpunkt), der für das neue Netzlaufwerk verwendet werden soll (z.B. `S:`).
- ▶ `strShare`: Mit dem zweiten Parameter übergeben Sie den Pfad der Freigabe (z.B. `\\192.168.0.1\Software`).
- ▶ `blnProfile`: Mit diesem optionalen Parameter können Sie veranlassen, dass das Netzlaufwerk im Benutzerprofil automatisch als Home-Laufwerk (künftiger Basisordner für die benutzerspezifischen Daten) deklariert wird. Wenn das Benutzerprofil angepasst werden soll, so übergeben Sie den Wert `True`, anderenfalls legen Sie `False` (dies ist die Standardeinstellung) fest.
- ▶ `strUserName`: Sofern der Zugriff auf die Freigabe die Angabe eines anderen Benutzernamens erfordert und nicht unter Verwendung des Benutzers stattfinden soll, der das Skript gerade ausführt, so übergeben Sie mit `strUserName` den entsprechenden Benutzernamen.
- ▶ `strPassword`: Mit diesem Parameter übergeben Sie das Kennwort des angegebenen Benutzernamens.

Das nachfolgende Listing zeigt ein Beispiel dafür, wie Sie die Freigabe `\\192.168.0.1\Software` an den Laufwerksbuchstaben `S:` binden und die Authentifizierung unter Verwendung des Benutzernamens `AdmInstall` und des Kennwortes `12345678` vornehmen können.

Option Explicit

```
Dim oNetwork
Dim strDriveLetter
Dim strShare
Dim blnProfile
Dim strUserName
Dim strPassword

strDriveLetter = "S:"
strShare = "\\192.168.0.1\Software"
blnProfile = False
```

Listing 391: 2_04_018a.vbs

```

strUserName = "AdmInstall"
strPassword = "12345678"

Set oNetwork = CreateObject("WScript.Network")
oNetwork.MapNetworkDrive strDriveLetter, strShare, blnProfile, _
    strUserName, strPassword
Set oNetwork = Nothing

```

Listing 391: 2_04_018a.vbs (Forts.)

Um ein bestehendes Netzlaufwerk wieder zu trennen, verwenden Sie die `RemoveNetworkDrive`-Methode des `WSHNetwork`-Objektes, der Sie zumindest den Laufwerksbuchstaben als Argument übergeben. Folgende Argumente sind möglich:

- ▶ `strDriveLetter`: Mit dem ersten Parameter übergeben Sie wieder den Laufwerksbuchstaben (gefolgt von einem Doppelpunkt).
- ▶ `blnForce`: Mit dem optionalen Parameter `blnForce` können Sie das Trennen des Netzlaufwerkes erzwingen (`True`). Standardmäßig wird `False` verwendet.
- ▶ `blnProfile`: Der optionale Parameter `blnProfile` ermöglicht die automatische Anpassung des Benutzerprofils (`True`), standardmäßig wird `False` verwendet.

Mit den nachfolgenden Code-Zeilen können Sie das zuvor erstellte Netzlaufwerk `S:` wieder trennen:

```

Option Explicit

Dim oNetwork
Dim strDriveLetter
Dim blnForce

strDriveLetter = "S:"
blnForce = True

Set oNetwork = CreateObject("WScript.Network")
oNetwork.RemoveNetworkDrive strDriveLetter, blnForce
Set oNetwork = Nothing

```

Listing 392: 2_04_018b.vbs

101 Freigaben entfernen

Die Freigabe einer Ressource lässt sich natürlich jederzeit wieder aufheben, wobei Sie jedoch vorher sicherstellen sollten, dass keine Benutzer mehr mit der Freigabe verbunden sind und auf die freigegebene Ressource zugreifen. Denn wenn Sie eine Freigabe aufheben, mit der noch Benutzer verbunden sind, so wird die Netzwerkverbindung zwi-

schen dem Benutzer und der Freigabe ad hoc getrennt, womit geöffnete Dateien recht »unsanft« geschlossen werden. Dies kann zu korrupten Dateien oder gar zu Datenverlusten führen.

Um eine Freigabe zu entfernen, öffnen Sie das Kontextmenü des ARBEITSPLATZ-Symbols auf dem DESKTOP und wählen VERWALTEN aus, um die COMPUTERVERWALTUNG zu öffnen. In der COMPUTERVERWALTUNG erweitern Sie den Eintrag FREIGELEGEBENE ORDNER und markieren unterhalb davon den Eintrag FREIGABEN. Markieren Sie die Freigabe, die Sie entfernen möchten, und wählen Sie im Kontextmenü der Freigabe den Eintrag FREIGABE AUFHEBEN aus. Für den Fall, dass noch Benutzer mit der Freigabe verbunden sind, so erhalten Sie nun einen entsprechenden Hinweis darauf und müssen das Aufheben der Freigabe und damit auch das Trennen der Verbindungen zur Freigabe noch einmal von Hand bestätigen.

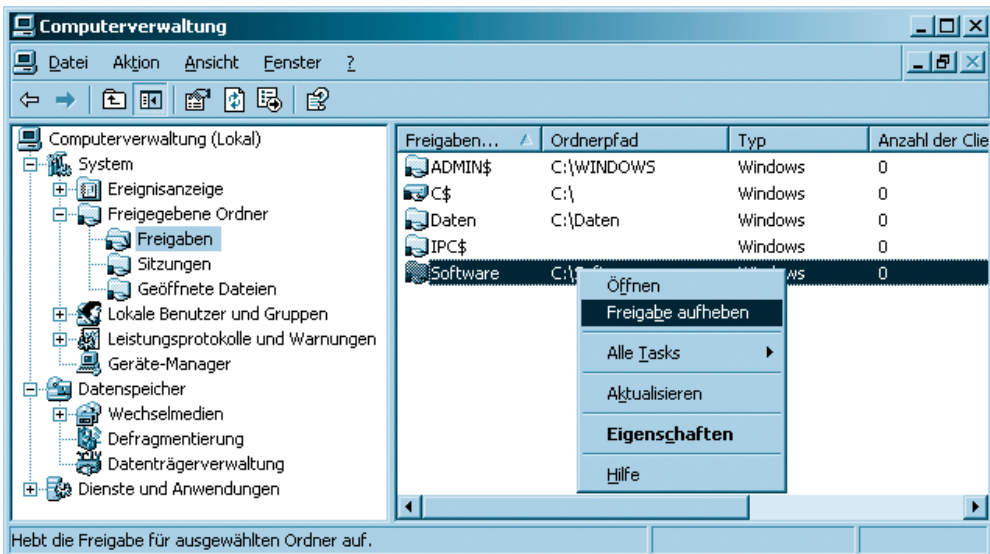


Abbildung 150: Eine vorhandene Freigabe wieder aufheben

Die Lösung per Eingabeaufforderung

Um eine Freigabe von der Eingabeaufforderung aus zu entfernen, können Sie zum einen den `Net Share`-Befehl einsetzen, zum anderen aber auch den `WMIC`-Befehl mit dem `Alias Share`. Wenn Sie den `Net Share` verwenden möchten, so übergeben Sie mit dem Befehlsaufruf den Namen der Freigabe wie auch den Parameter `/Delete`, woraufhin die angegebene Freigabe sofort aufgehoben wird.

```
@Echo off
Net Share Software /Delete
```

Listing 393: 2_04_019a.bat

Möchten Sie die Freigabe dagegen mit dem WMIC-Befehl entfernen, so wenden Sie den Befehl `WMIC Share an`, selektieren mit der `Where`-Klausel die entsprechende Freigabe und rufen dann mit `Call` die `Delete`-Methode auf. Die Freigabe wird daraufhin sofort entfernt.

```
@Echo off
WMIC Share WHERE "Name='Software'" Call Delete
```

Listing 394: 2_04_019b.bat

Die Lösung mit VBScript

Wenn Sie eine vorhandene Freigabe aus einem VBScript heraus entfernen möchten, so nutzen Sie WMI und stellen zunächst eine Verbindung zum WMI-Namespace `root\cimv2` des betreffenden Computersystems her. Anschließend selektieren Sie mit der `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_Share` das entsprechende Freigabe-Objekt und wenden innerhalb einer `For Each`-Schleife die `Delete`-Methode gegen das Freigabe-Objekt an, womit die Freigabe unmittelbar danach aufgehoben wird.

```
Option Explicit

Dim oWMI
Dim colShare
Dim strComputer
Dim objItem
Dim intValue

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colShare = oWMI.ExecQuery("SELECT * FROM Win32_Share WHERE _
    Name='Software'")

For Each objItem in colShare
    intValue = objItem.Delete
Next

Set oWMI = Nothing
```

Listing 395: 2_04_019.vbs

102 Freigaben im Active Directory veröffentlichen

Freigaben ermöglichen zwar eine schnelle und komfortable Bereitstellung vorhandener Ressourcen, allerdings gestaltet sich das netzwerkweite Auffinden der Freigaben meist als recht aufwändig, da es in der Regel keine zentrale Übersicht der vorhandenen Freigaben gibt.

Mit der Voraussetzung, dass Sie über eine Active Directory-Domäne verfügen, so können Sie die Freigaben im Active Directory veröffentlichen und das Auffinden der Freigaben damit wesentlich einfacher gestalten. Denn mit dem Veröffentlichen einer Freigabe wird im Active Directory-Verzeichnis ein Freigabe-Objekt erzeugt, wonach Sie entweder über die Netzwerkumgebung, per LDAP-Abfrage oder auch innerhalb von ACTIVE DIRECTORY-BENUTZER UND -COMPUTER suchen können. Mit dem Veröffentlichen einer Freigabe können Sie zeitgleich auch ein oder mehrere Schlüsselwörter hinterlegen, die ebenfalls in den Suchvorgang einbezogen werden können.

Um eine bereits vorhandene Freigabe im Active Directory zu veröffentlichen, rufen Sie auf dem Domänen-Controller die Management-Konsole ACTIVE DIRECTORY-BENUTZER UND -COMPUTER auf und wählen im Kontextmenü des Containers oder der Organisationseinheit, in der Sie das Freigabe-Objekt erzeugen möchten, NEU und anschließend FREIGELEGEBENER ORDNER aus. Sie gelangen daraufhin in das Dialogfenster NEUES OBJEKT – FREIGELEGEBENER ORDNER und können nun die freigabespezifischen Informationen wie Name und Pfad der Freigabe eingeben.

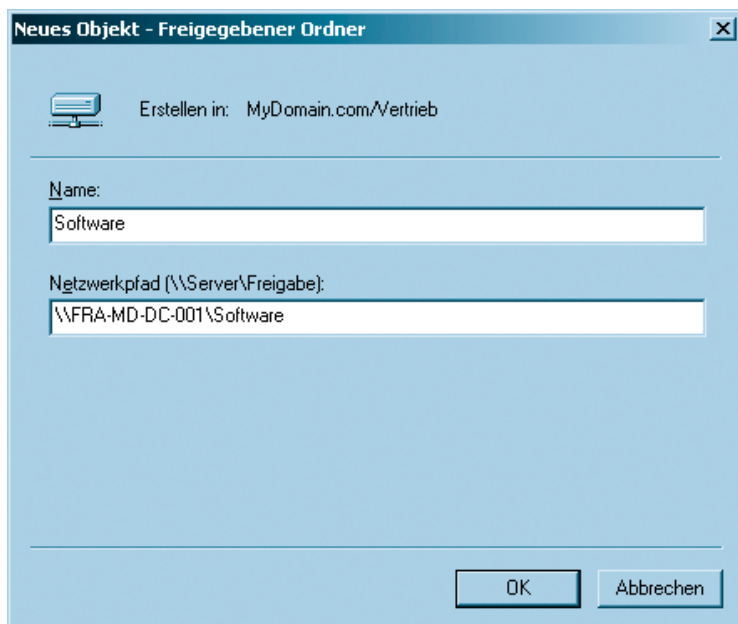


Abbildung 151: Einen freigegebenen Ordner veröffentlichen

Wenn Sie den Namen und auch den Pfad für die Freigabe hinterlegt haben, so schließen Sie das Dialogfenster mit einem Klick auf OK, woraufhin das Objekt dann erstellt wird. Um das Auffinden der Freigabe noch weiter zu vereinfachen, können Sie dem soeben erstellen Freigabe-Objekt noch entsprechende Schlüsselwörter zuweisen, die für die spätere Suche nach der Freigabe als Suchbegriffe verwendet werden können. Öffnen Sie hierzu die Eigenschaften des Freigabe-Objektes im Active Directory, und klicken Sie auf SCHLÜSSELWÖRTER. Geben Sie nun einen oder wahlweise auch mehrere aussagekräftige Suchbegriffe ein, und fügen Sie die Wörter durch einen Klick auf HINZUFÜGEN der Liste der Schlüsselwörter hinzu. Übernehmen Sie die Einstellung dann mit einem Klick auf OK, womit das Dialogfenster geschlossen und die Veröffentlichung der Freigabe abgeschlossen ist.

Die Lösung per Eingabeaufforderung

Leider gibt es keine Möglichkeit, womit sich eine Freigabe von der Eingabeaufforderung aus im Active Directory veröffentlichen lässt. Hier bleibt nur die Ausführung des im nächsten Abschnitt beschriebenen Skriptes.

Die Lösung mit VBScript

Mit Hilfe von ADSI lassen sich vorhandene Freigaben auf recht einfache und Weise im Active Directory veröffentlichen. Um beispielsweise die Freigabe `\\FRA-MD-DC-001\Software` im Active Directory unter der Bezeichnung *Software* zu publizieren und die Schlüsselwörter *Software*, *Installation*, *Setup* und *Distribution* zu hinterlegen, stellen Sie im ersten Schritt mittels `GetObject` eine LDAP-Verbindung zum Ziel-Container oder zur Ziel-Organisationseinheit her und erzeugen dann mit der `Create`-Methode ein neues Objekt vom Volume unter Angabe der Objektbezeichnung (`CN=Software`).

Anschließend setzen Sie mit der `Put`-Methode die Werte der Eigenschaften `UNCName` (Pfad zur Freigabe), `Description` (Beschreibung) und `Keywords` (Schlüsselwörter) und wenden dann die `SetInfo`-Methode an, um das Objekt und auch die Eigenschaften zu speichern.

Option Explicit

```
Dim oOU
Dim oShare
Dim strUNCName
Dim strDescription
Dim arrKeywords
```

```
strUNCName = "\\FRA-MD-DC-001\Software"
strDescription = "Software-Installationsdateien"
arrKeywords = Array("Software", "Installation", "Setup", "Distribution")
```

Listing 396: 2_04_020.vbs

```
Set oOU = GetObject("LDAP://OU=Vertrieb,DC=MyDomain,DC=com")
Set oShare = oOU.Create("Volume", "CN=Software")
oShare.Put "UNCName", strUNCName
oShare.Put "Description", strDescription
oShare.Put "Keywords", arrKeywords
oShare.SetInfo

Set oOU = Nothing
Set oShare = Nothing
```

Listing 396: 2_04_020.vbs (Forts.)

Registrierung

Die Registrierung von Windows Server 2003 gilt als zentrales Repository (Aufbewahrungsort) der Konfigurationseinstellungen und somit auch als Herzstück der Systemkonfiguration. Denn in der Registrierung sind sämtliche Konfigurationsparameter zum Betriebssystem selbst, zur verwendeten Hard- und Software wie auch zu den einzelnen Benutzereinstellungen hinterlegt.

Die meisten Änderungen, die Sie über die zahlreichen Dialogboxen der grafischen Oberfläche vornehmen können, werden in der Registrierung gespeichert. Und wenn die entsprechende Einstellung vom Betriebssystem benötigt wird, so wird die entsprechende Option wieder aus der Registrierung gelesen. Dies passiert natürlich alles vollkommen im Hintergrund, weshalb zunächst einmal kein Bedarf besteht, sich näher mit der Registrierung zu befassen. Geschweige denn, eine eigene Kategorie mit zahlreichen Rezepten zu erstellen. Doch weit gefehlt! Denn in der Registrierung sind wesentlich mehr Konfigurationseinstellungen hinterlegt, als Sie über die grafische Oberfläche im Zugriff haben. Ein direkter Blick in die Registrierung kann also in vielen Fällen weitaus mehr Informationen liefern als zunächst angenommen.

Und mit gekonnten Handgriffen lassen sich viele Konfigurationseinstellungen auch direkt in der Registrierung ändern oder per Ex- und Import auch auf andere Computersysteme übertragen. Dieses Verfahren hat sich gerade in der Systemadministration und auch im Troubleshooting-Bereich sehr bewährt, da dies eine Unmenge an Konfigurationsarbeit und damit jede Menge Zeit einspart. Vorausgesetzt natürlich, man weiß, wo die jeweiligen Konfigurationseinstellungen in der Registrierung zu finden sind. Durch den Vergleich der Registrierung vor und nach einer erfolgten Änderung über die grafische Oberfläche lassen sich zudem die durchgeführten Änderungen protokollieren, z.B. für eine Systemdokumentation.

Und gerade diejenigen, die sich noch nicht allzu sehr mit der Registrierung befassen haben, tun sich anfangs mit dem Finden der richtigen Registrierungseinstellungen schwer. Daher soll direkt zu Beginn auf die Registry-Referenz verwiesen werden, in der zum einen die Struktur der Registrierung beschrieben wird, zum anderen aber auch sehr viele Registrierungseinstellungen. In vielen Fällen gelangen Sie durch Eingabe eines simplen Stichwortes zu den gewünschten Informationen und erfahren so, in welchem Schlüssel die jeweilige Betriebssystemeinstellung hinterlegt ist und welche Werte erlaubt sind. Sie finden die Registry-Referenz im *Windows Server 2003 Resource Kit* (Microsoft Press, ISBN: 0735614717) wie auch auf der Internetseite <http://www.microsoft.com>.

Achtung

Zu guter Letzt noch ein wichtiger Hinweis: In der Registrierung werden sämtliche Konfigurationseinstellungen hinterlegt, die für den Betrieb von Windows Server 2003 wie auch seitens der installierten Hard- und Software zwingend benötigt werden. Die Registrierung erlaubt daher keine fehlerhaften Einstellungen und quittiert solche nicht selten mit Systeminstabilitäten oder gar einem totalen Systemabsturz. Wenn Sie Änderungen an den Registrierungseinstellungen vornehmen, so müssen Sie zweifel- und fehlerhafte Einstellungen unbedingt ausschließen können und auf jeden Fall vorher ein Backup der Registrierung (siehe Registrierung sichern und wiederherstellen auf Seite 549) erstellen. Falls Windows Server 2003 selbst im abgesicherten Modus nicht mehr starten kann, weil die Registrierung fehlerhaft ist, so benötigen Sie zur Wiederherstellung der Registrierung ein aktuelles Backup wie auch die Recovery-Konsole. Nehmen Sie die Änderung an der Registrierung deshalb zunächst auf einem Testsystem vor, auf dem die Recovery-Konsole installiert ist.

103 Existiert der Schlüssel bereits?

Um unter der grafischen Oberfläche herauszufinden, ob ein bestimmter Schlüssel bereits existiert, rufen Sie einfach den Registrierungs-Editor auf, wechseln in den Schlüssel, der den zu überprüfenden Schlüssel enthalten soll, und schauen nach, ob der gewünschte Schlüssel darin enthalten ist.

Wenn Sie beispielsweise herausfinden möchten, ob der Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\SETUP existiert, dann rufen Sie den Registrierungs-Editor auf, wechseln in den Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION und prüfen dessen untergeordnete Schlüssel. Ist der Schlüssel SETUP in der Schlüsselstruktur vorhanden, so existiert der Schlüssel. Anderenfalls müsste er erst angelegt werden.

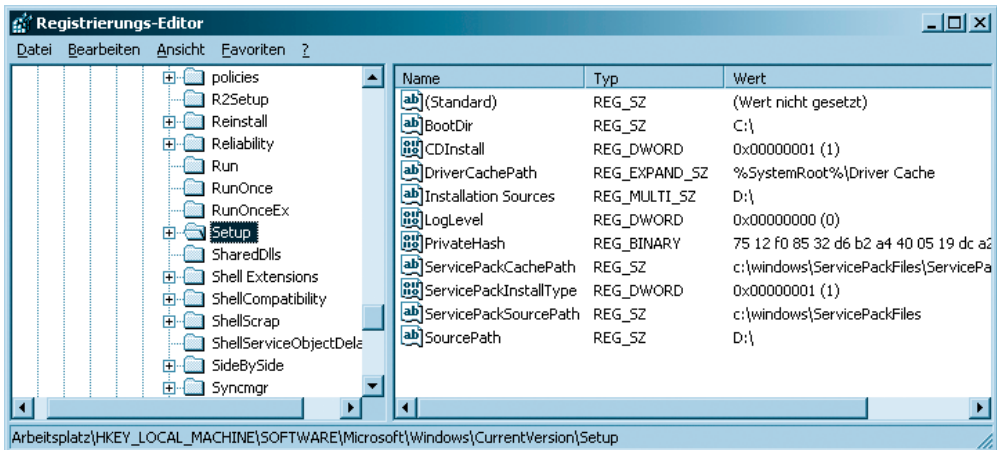


Abbildung 152: Der Setup-Schlüssel ist bereits vorhanden.

Die Lösung per Eingabeaufforderung

Wenn Sie in der Eingabeaufforderung das Vorhandensein eines bestimmten Schlüssels abfragen wollen, dann verwenden Sie hierfür den Befehl `Reg Query`, dem Sie lediglich den zu überprüfenden Schlüssel als Parameter übergeben müssen. `Reg Query` versucht daraufhin, diesen Schlüssel zu öffnen, die darin enthaltenen Objekte (Schlüssel, Werte und Daten) auszulesen und die zurückgelieferten Daten am Bildschirm auszugeben.

```
@Echo off
Reg Query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Setup
```

Listing 397: 2_05_001.bat

Die Lösung mit VBScript

Um das Vorhandensein eines bestimmten Schlüssels mit VBScript zu überprüfen, nutzen Sie die `EnumKey`-Methode der WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`), womit Sie die untergeordneten Schlüssel eines bestimmten Schlüssels auslesen können.

Im ersten Schritt stellen Sie mit `GetObject` eine Verbindung zu der WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`) her und setzen den `Impersonation-Level` auf `Impersonate`. Anschließend führen Sie eine `EnumKey`-Abfrage gegen den Schlüssel aus, dessen untergeordnete Schlüssel Sie auslesen möchten. Der `EnumKey`-Methode übergeben Sie dabei folgende Parameter:

- ▶ **HKLM**: Der hexadezimale Wert des Hauptzweiges, der den auszulesenden Schlüssel enthält (siehe nachfolgende Tabelle), standardmäßig wird `&H80000002` verwendet. Der Überschaubarkeit wegen wird die Abkürzung des Hauptschlüssels als Konstante verwendet.
- ▶ **strKey**: Der Schlüssel (jedoch ohne Angabe des Hauptzweiges), dessen untergeordnete Schlüssel ausgelesen werden sollen
- ▶ **strArray**: Das Array, in das die ausgelesenen Schlüssel als Elemente übertragen werden sollen

Hauptzweig	Hexadezipimaler Wert
HKEY_CLASSES_ROOT	&H80000000
HKEY_CURRENT_USER	&H80000001
HKEY_LOCAL_MACHINE	&H80000002
HKEY_USERS	&H80000003
HKEY_CURRENT_CONFIG	&H80000005

Tabelle 53: Die hexadezimalen Werte der Hauptzweige

Die `EnumKey`-Methode liest daraufhin die untergeordneten Schlüssel des mit `strKey` übergebenen Schlüssels aus und überträgt die gefundenen Schlüssel nach `strArray`. Mit Hilfe einer `For Each`-Schleife können Sie nun für jedes Element überprüfen, ob der Name gleich `Setup` ist. Ist die Bedingung wahr, so können Sie die `For Each`-Schleife mit `Exit For` vorzeitig verlassen, da aufgrund der Eindeutigkeit der Schlüssel keine weiteren Ergebnisse geliefert werden können.

Option Explicit

```
Dim oReg
Dim strComputer
Dim strKey
Dim strSubKey
Dim strArray
```

```
Const HKLM = &H80000002
strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows\CurrentVersion"
```

```
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\default:StdRegProv")
oReg.EnumKey HKLM, strKey, strArray
```

```
For Each strSubKey In strArray
    If strSubKey = "Setup" then
        WScript.Echo "Schluessel existiert"
        Exit For
    End if
Next
```

```
Set oReg = Nothing
```

Listing 398: 2_05_001.vbs

104 Einzelne Schlüssel auflisten

Möchten Sie unter grafischer Oberfläche herausfinden, welche untergeordneten Schlüssel ein bestimmter Schlüssel hat, so brauchen Sie sich lediglich die Schlüsselstruktur im Registrierungs-Editor anzeigen zu lassen.

Um herauszufinden, welche Schlüssel sich unter `HKLM\SYSTEM\CURRENTCONTROLSET\SERVICES` befinden, starten Sie den Registrierungs-Editor und öffnen den genannten Schlüssel. Die darin enthaltene Schlüsselstruktur wird automatisch angezeigt, wobei Sie natürlich auch die unteren Ebenen mit einem Klick auf das Pluszeichen einblenden können.

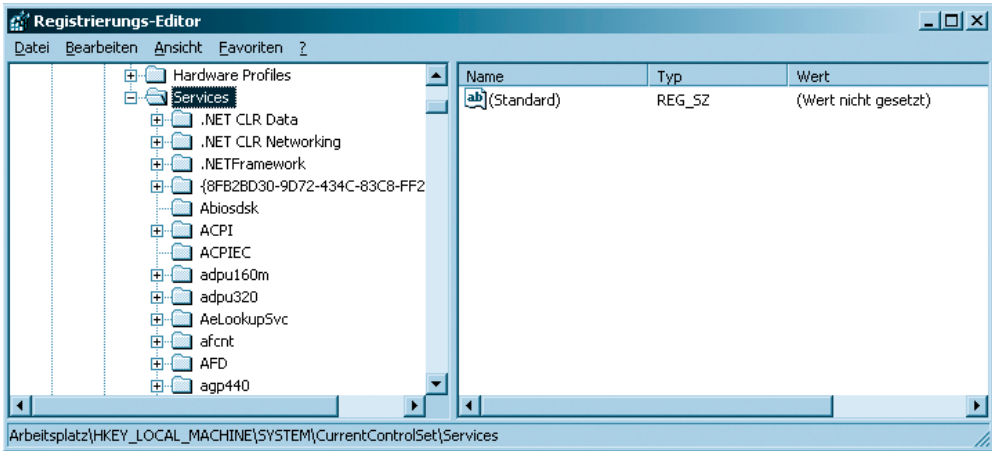


Abbildung 153: Die untergeordneten Schlüssel eines bestimmten Schlüssels anzeigen lassen

Die Lösung per Eingabeaufforderung

Beim Auflisten der untergeordneten Schlüssel eines bestimmten Schlüssels gehen Sie genauso vor wie bei der Überprüfung eines einzelnen Schlüssels (siehe Existiert der Schlüssel bereits? auf Seite 495). Das nachfolgende Reg Query-Beispiel liest die untergeordneten Schlüssel der ersten Ebene von HKLM\SYSTEM\CurrentControlSet\Services aus und schreibt das Ergebnis per Ausgabeumleitung in die Datei *Datei.txt*.

```
@Echo off
Reg Query HKLM\SYSTEM\CurrentControlSet\Services > Datei.txt
```

Listing 399: 2_05_002.bat

Ohne Angabe weiterer Parameter liest der Reg Query-Befehl allerdings nur die untergeordneten Schlüssel der ersten Ebene aus. Um auch die untergeordneten Schlüssel weiterer Ebenen auszulesen, müssen Sie der Reg Query-Befehlszeile den Parameter `/s` anhängen.

Die Lösung mit VBScript

Beim Auslesen untergeordneter Schlüssel gehen Sie ähnlich wie im Rezept Existiert der Schlüssel bereits? auf Seite 495 vor: Sie stellen eine Verbindung zu der WMI-Klasse StdRegProv (WMI-Namespace root\default) her und setzen die EnumKey-Methode gegen den auszulesenden Schlüssel ein. Der EnumKey-Methode übergeben Sie dabei den hexadezimalen Wert des Hauptzweiges, den auszulesenden Schlüssel (jedoch ohne Angabe des Hauptzweiges) wie auch das zu verwendende Array als Parameter. Die untergeordneten Schlüssel werden daraufhin ausgelesen und als Elemente in das Array übertragen. Mit Hilfe einer For Each-Schleife können Sie dann

auf jedes Element zugreifen und dieses durch Einsatz weiterer Methoden entweder am Bildschirm ausgeben, in eine Datei schreiben oder an den Drucker senden.

Das folgende Skript zeigt ein Beispiel dafür, wie Sie mit der `EnumKey`-Methode die untergeordneten Schlüssel von `HKLM\SYSTEM\CURRENTCONTROLSET\SERVICES` auslesen und in eine Datei schreiben können:

Zunächst folgt die Definition der Variablen und Konstanten, damit das Skript überschaubar bleibt und leichter angepasst werden kann. Anschließend folgen die Referenzierung des `FileSystem`-Objektes und der Aufruf der `OpenTextFile`-Methode, der Sie folgende Parameter übergeben müssen:

- ▶ `strFile`: Die Datei, die geöffnet werden soll
- ▶ `Write`: Konstante für den Dateizugriffsmodus (1 = Lesen, 2 = Schreiben, 8 = Anhängen)
- ▶ `True/False`: Der Wert `True` legt fest, dass die Datei erstellt werden soll, sofern sie noch nicht existiert. Die Verwendung des Wertes `False` dagegen öffnet die Datei nur, wenn sie schon existiert.

Danach wird mit der `GetObject`-Methode eine Verbindung zur WMI-Klasse `StdRegProv` hergestellt und der `Impersonation-Level` auf `Impersonate` gesetzt. Im Anschluss daran folgt der eigentliche Aufruf der `EnumKey`-Methode, womit die untergeordneten Schlüssel des als Parameter übergebenen Schlüssels in das Array übertragen werden. Mit Hilfe der `For Each`-Schleife werden die Elemente dann ausgelesen und mit `oFile.WriteLine (strSubKey)` in die Datei geschrieben.

Option Explicit

```
Dim oReg
Dim oFS
Dim oFile
Dim strComputer
Dim strKey
Dim strSubKey
Dim strArray
Dim strFile
```

```
Const HKLM = &H80000002
Const Write = 2
strComputer = "."
strKey = "SYSTEM\CurrentControlSet\Services"
strFile = "C:\Temp\Datei.txt"
```

```
Set oFS=CreateObject("Scripting.FileSystemObject")
Set oFile=oFS.OpenTextFile(strFile, Write, True)
Set oReg=GetObject("winmgmts:{impersonationLevel=_
```

```

        impersonate!\\\" & strComputer & "\root\default:StdRegProv")
oReg.EnumKey HKLM, strKey, strArray

For Each strSubKey In strArray
    oFile.WriteLine strSubKey
Next

Set oReg = Nothing
Set oFS = Nothing
Set oFile = Nothing

```

Listing 400: 2_05_002a.vbs (Forts.)

Das obige Skript gibt allerdings nur die untergeordneten Schlüssel der ersten Ebene aus. Wenn Sie die `EnumKey`-Methode dazu veranlassen möchten, auch die tiefer verzweigten Schlüssel mit auszulesen, dann müssen Sie stattdessen eine rekursive Abfrage verwenden, womit die `EnumKey`-Methode für jeden untergeordneten Schlüssel neu aufgerufen wird:

```

Option Explicit

Dim oReg
Dim oFS
Dim oFile
Dim strComputer
Dim strKey
Dim strFile

Const HKLM = &H80000002
Const Write = 2
strComputer = "."
strKey = "SYSTEM\CurrentControlSet\Services"
strFile = "C:\Temp\Datei.txt"

Set oFS=CreateObject("Scripting.FileSystemObject")
Set oFile=oFS.OpenTextFile(strFile, Write, True)
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate!\\\" & strComputer & "\root\default:StdRegProv")
ListKeys HKLM, strKey

Set oReg = Nothing
Set oFS = Nothing
Set oFile = Nothing

```

Listing 401: 2_05_002b.vbs

```
Sub ListKeys(HKLM, strKey)

    oFile.WriteLine "HKLM\" & strKey & "\" & strSubKey

    Dim strArray
    Dim strSubKey
    Dim intIdx

    On Error Resume Next
    intIdx = oReg.EnumKey(HKLM, strKey, strArray)
    If intIdx = 0 And IsArray(strArray) Then
        For Each strSubKey in strArray
            ListKeys HKLM, strKey & "\" & strSubKey
        Next
    End If

End Sub
```

Listing 401: 2_05_002b.vbs (Forts.)

105 Neue Schlüssel erstellen

Um einen neuen Schlüssel unter grafischer Oberfläche anzulegen, starten Sie einfach den Registrierungs-Editor, wechseln in den Schlüssel, der den neuen Schlüssel enthalten soll, und wählen im Menü BEARBEITEN den Eintrag NEU und anschließend SCHLÜSSEL aus. Alternativ können Sie den neuen Schlüssel auch über das Kontextmenü des übergeordneten Schlüssels aufrufen.

Registrierung

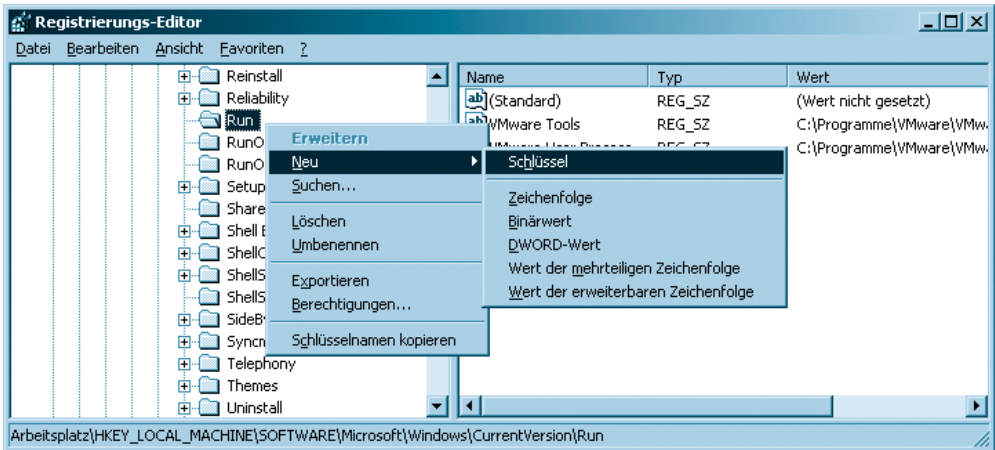


Abbildung 154: Einen neuen Schlüssel erstellen

Daraufhin wird ein neuer Schlüssel mit einer Standardbezeichnung (NEUER SCHLÜSSEL #1) erstellt. Diese Standardbezeichnung brauchen Sie nur noch mit der eigentlichen Schlüsselbezeichnung zu überschreiben.

Die Lösung per Eingabeaufforderung

Um neue Schlüssel in der Eingabeaufforderung anzulegen, nutzen Sie den Befehl `Reg Add`, dem Sie lediglich den anzulegenden Schlüssel als Parameter übergeben müssen. `Reg Add` versucht daraufhin, den angegebenen Schlüssel anzulegen. Sofern der Schlüssel bereits vorhanden ist, müssen Sie das Überschreiben des vorhandenen Schlüssels von Hand bestätigen. Dies können Sie jedoch umgehen, indem Sie die Befehlszeile um den Parameter `/f` erweitern, womit das Überschreiben eines bereits vorhandenen Schlüssels erzwungen wird:

```
@Echo off
Reg Add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Backup /f
```

Listing 402: 2_05_003.bat

Die Lösung mit VBScript

Auch mit VBScript können Sie neue Schlüssel erstellen. Hierzu stellt die WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`) die `CreateKey`-Methode bereit, der Sie lediglich den hexadezimalen Wert des Hauptzweiges und auch den neuen Schlüssel (ohne Angabe des Hauptzweiges) als Parameter übergeben müssen. Die `CreateKey`-Methode stellt daraufhin eine Verbindung zu dem Hauptzweig her und legt den angegebenen Schlüssel an. Und zwar unabhängig davon, ob der Schlüssel bereits existiert oder nicht.

```
Option Explicit

Dim oReg
Dim strComputer
Dim strKey

Const HKLM = &H80000002
strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Backup"

Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\default:StdRegProv")
oReg.CreateKey HKLM,strKey

Set oReg = Nothing
```

Listing 403: 2_05_003.vbs

106 Schlüssel umbenennen

Für den Fall, dass Sie beim Anlegen eines neuen Schlüssels eine falsche Bezeichnung eingegeben oder sich möglicherweise nur leicht vertippt haben, so müssen Sie den Schlüssel unter grafischer Oberfläche nicht löschen und neu anlegen, sondern brauchen ihn nur umzubenennen. Denn der Registrierungs-Editor ist mit einer Funktion ausgestattet, mit deren Hilfe Sie die Bezeichnungen jederzeit ändern können:

Um den Schlüssel `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN\BACKUP`, den Sie im Rezept Neue Schlüssel erstellen auf Seite 500 probenhalber angelegt haben, nach `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN\DATENSICHERUNG` umzubenennen, starten Sie einfach den Registrierungs-Editor, markieren den Schlüssel `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN\BACKUP` und wählen im Menü BEARBEITEN oder im Kontextmenü des Schlüssels den Eintrag UMBENENNEN aus. Daraufhin können Sie die Bezeichnung des Schlüssels ändern.

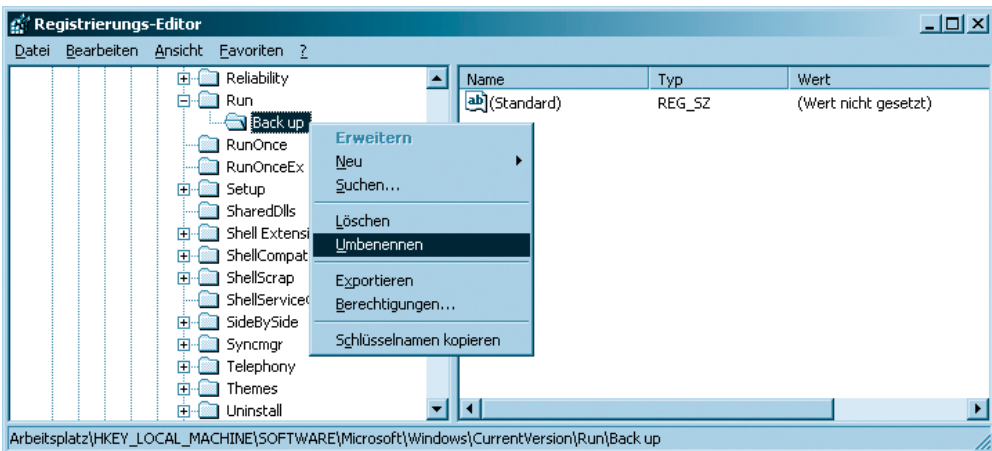


Abbildung 155: Einen vorhandenen Schlüssel umbenennen

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung lässt sich ein vorhandener Schlüssel leider nicht ganz so einfach umbenennen, wie dies mit Hilfe des Registrierungs-Editors der Fall ist, da es hierfür keinen entsprechenden Befehl gibt. Mit dem Befehl `Reg Copy` können Sie aber einen vorhandenen Schlüssel mitsamt Inhalt in einen neuen Schlüssel kopieren, wobei Sie die Bezeichnung des neuen Schlüssels natürlich frei bestimmen können. Und wenn Sie nach dem Kopiervorgang den ursprünglichen Schlüssel mit `Reg Delete` aus der Registrierung löschen, haben Sie die fehlende Möglichkeit des Umbenennens bestenfalls improvisiert.

Kopieren Sie also den Schlüssel, den Sie umbenennen möchten, zunächst mit `Reg Copy` in den gewünschten Zielschlüssel. Übergeben Sie dem Befehl dabei den zu kopierenden Schlüssel wie auch den eigentlichen Zielschlüssel als Parameter. Mit dem zusätzlichen Parameter `/s` geben Sie an, dass nicht nur der Schlüssel, sondern

auch dessen kompletter Inhalt kopiert werden soll. Und für den Fall, dass der Zielschlüssel bereits existieren sollte, können Sie das Überschreiben des Zielschlüssels mit dem Parameter /f erzwingen.

Anschließend löschen Sie mit `Reg Delete` den ursprünglichen Schlüssel, wobei Sie dem Befehl den zu löschenden Schlüssel als Parameter übergeben. Um eine Rückfrage zu unterdrücken, die Sie manuell bestätigen müssten, ergänzen Sie die Befehlszeile um den Parameter /f, womit Sie das Löschen erzwingen können. Beachten Sie dabei, dass Schlüssel in Anführungszeichen gesetzt werden müssen, sofern sie in ihrer Bezeichnung ein oder auch mehrere Leerzeichen enthalten.

```
@Echo off
Reg Copy "HKLM\Software\Microsoft\Windows\Current_
    Version\Run\Back up" HKLM\Software\Micro_
    soft\Windows\CurrentVersion\Run\Backup /s /f
Reg Delete "HKLM\Software\Microsoft\Windows\Current_
    Version\Run\Back up" /f
```

Listing 404: 2_05_004.bat

Die Lösung mit VBScript

Um einen vorhandenen Schlüssel per VBScript umzubenennen, können Sie zwar den Inhalt des Schlüssels rekursiv auslesen, dann mit der `CreateKey`-Methode den neuen Schlüssel erzeugen und die zuvor ausgelesenen untergeordneten Schlüssel, Werte und Daten wieder neu hineinschreiben. Weitaus einfacher lässt sich ein Schlüssel aber mit Hilfe des `Reg Copy`-Befehls der Eingabeaufforderung umbenennen, den Sie mit der `Run`-Methode des `Shell`-Objektes auch aus einem VBScript heraus aufrufen können.

Das nachfolgende Listing führt beispielsweise den Inhalt der Batch-Datei `RenameKey.bat` aus, womit die `Reg Copy`-Befehlszeilen des vorherigen Listings (`2_05_004.bat`) ausgeführt werden.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c RenameKey.bat"
Set oShell = Nothing
```

Listing 405: 2_05_004.vbs

107 Schlüssel löschen

Einzelne Schlüssel können Sie natürlich jederzeit aus der Registrierung löschen, wobei Sie vorher sicherstellen sollten, dass dieser Schlüssel nicht mehr benötigt wird. Und für den Fall, dass der zu löschende Schlüssel weitere Schlüssel und Werte enthält, so müssen Sie auch für die untergeordneten Objekte sicherstellen können, dass sie nicht mehr benötigt werden. Denn wenn Sie einen Schlüssel löschen, der noch weitere Objekte enthält, so werden auch die untergeordneten Objekte ohne jegliche Rückfrage aus der Registrierung entfernt.

Um einen vorhandenen Schlüssel aus der Registrierung zu löschen, nutzen Sie unter grafischer Oberfläche wieder den Registrierungs-Editor. Wenn Sie beispielsweise den Schlüssel BACKUP, den Sie im Rezept Neue Schlüssel erstellen auf Seite 500 vielleicht zu Testzwecken angelegt haben, wieder aus der Registrierung löschen wollen, dann rufen Sie den Registrierungs-Editor auf, wechseln in den Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN, markieren den Schlüssel BACKUP und drücken die **[Entf]**-Taste. Alternativ können Sie auch das Kontextmenü des Schlüssels BACKUP öffnen und dort LÖSCHEN auswählen. Oder aber Sie wählen im Menü BEARBEITEN den Eintrag LÖSCHEN aus.

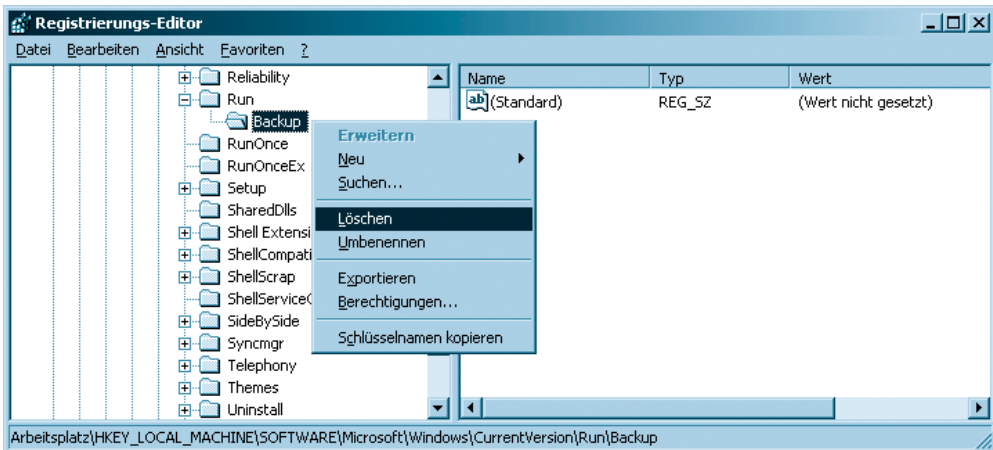


Abbildung 156: Einen vorhandenen Schlüssel löschen

Die Lösung per Eingabeaufforderung

Wenn Sie in der Eingabeaufforderung einen vorhandenen Schlüssel aus der Registrierung löschen möchten, dann nutzen Sie hierfür den Befehl `Reg Delete`, dem Sie lediglich den zu löschenden Schlüssel als Parameter übergeben müssen. `Reg Delete` versucht daraufhin, den angegebenen Schlüssel aus der Registrierung zu löschen, wobei Sie diesen Löschvorgang manuell bestätigen müssen. Um dies zu vermeiden, fügen Sie den Parameter `/f` an, womit Sie das Löschen des Schlüssels erzwingen können. Die Rückfrage, ob der Schlüssel auch wirklich gelöscht werden soll, bleibt damit aus.

```
@Echo off
Reg Delete HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Backup /f
```

Listing 406: 2_05_005.bat

Die Lösung mit VBScript

Um einen vorhandenen Schlüssel mit VBScript zu löschen, verwenden Sie die `DeleteKey`-Methode, die durch die WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`) bereitgestellt wird. Die `DeleteKey`-Methode erwartet dabei die Angabe des hexadezimalen Wertes des Hauptzweiges wie auch den zu löschenden Schlüssel selbst als Parameter.

```
Option Explicit

Dim oReg
Dim strComputer
Dim strKey

Const HKLM = &H80000002
strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Backup"

Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\" & strComputer & _
    "\root\default:StdRegProv")
oReg.DeleteKey HKLM,strKey

Set oReg = Nothing
```

Listing 407: 2_05_005a.vbs

Die `DeleteKey`-Methode versucht daraufhin, eine Verbindung zu dem angegebenen Schlüssel herzustellen und den Schlüssel inklusive der darin enthaltenen Werte und Daten zu löschen. Enthält der zu löschende Schlüssel jedoch weitere untergeordnete Schlüssel, so bricht der Vorgang ab, und der angegebene Schlüssel wird nicht gelöscht. Grund: Mit Hilfe der `DeleteKey`-Methode können Sie nur leere Schlüssel löschen. Werte und Daten dürfen zwar enthalten sein, jedoch keine weiteren Unterschlüssel.

Wenn Sie mit der `DeleteKey`-Methode einen Schlüssel löschen möchten, der weitere Schlüssel enthält, so müssen Sie rekursiv löschen. Das heißt, in obigem Skript würden Sie die Zeile `oReg.DeleteKey HKLM, strKey` durch `DeleteKeys HKLM, strKey` ersetzen und am Ende des Skriptes die folgende Subprozedur für das rekursive Löschen einfügen:

```

Sub DeleteKeys(HKLM, strKey)

    Dim strArray
    Dim strSubKey
    Dim intIdx

    intIdx = oReg.EnumKey(HKLM, strKey, strArray)

    On Error Resume Next

    If intIdx = 0 And IsArray(strArray) Then
        For Each strSubKey in strArray
            DeleteKeys HKLM, strKey & "\" & strSubKey
        Next
    End If

    oReg.DeleteKey HKLM, strKey

End Sub

```

Listing 408: 2_05_005b.vbs

108 Existiert der Wert bereits?

Einzelne Werte, gleich welchen Typs, müssen in der Registrierung genauso eindeutig sein wie Schlüssel. Bevor Sie also neue Zeichenfolgen, Binärwerte, DWORD-Werte oder Werte der mehrteiligen oder erweiterbaren Zeichenfolgen erstellen, sollten Sie zunächst überprüfen, ob der betreffende Wert bereits existiert. Und diese Überprüfung nehmen Sie unter grafischer Oberfläche – wie soll es auch anders sein – wieder mit dem Registrierungs-Editor vor.

Um herauszufinden, ob die Zeichenfolge AUTOADMINLOGON bereits im Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON enthalten ist, starten Sie einfach den Registrierungs-Editor, wechseln in den genannten Schlüssel und prüfen dessen Inhalt. Ist der Wert AUTOADMINLOGON vom Typ REG_SZ in der rechten Fensterhälfte enthalten, so existiert die Zeichenfolge bereits (siehe Abbildung 157).

Die Lösung per Eingabeaufforderung

Um in der Eingabeaufforderung das Vorhandensein eines bestimmten Wertes zu überprüfen, nutzen Sie wieder den Befehl `Reg Query`, dem Sie zumindest einen gültigen Schlüssel als Parameter übergeben müssen. `Reg Query` stellt daraufhin eine Verbindung zu dem angegebenen Schlüssel her, liest dessen Inhalt (Schlüssel, Werte und Daten) aus und gibt das Ergebnis anschließend am Bildschirm aus.

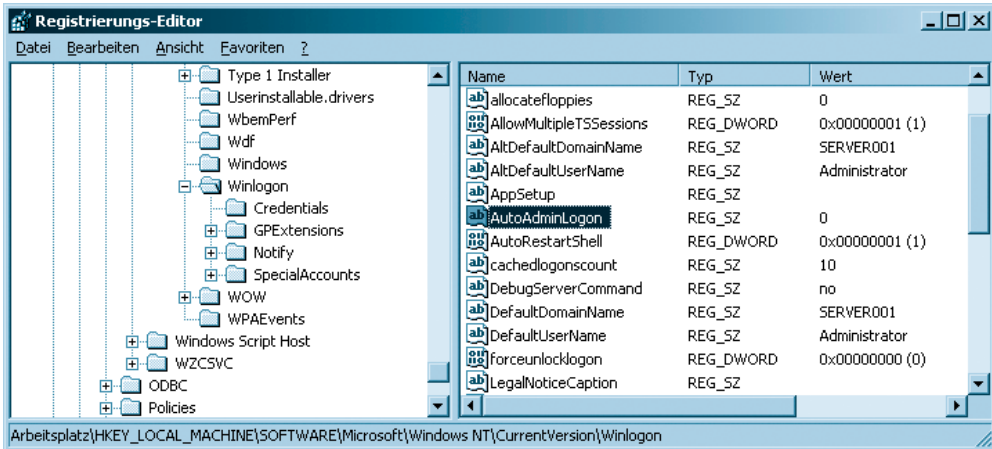


Abbildung 157: Der gesuchte Wert ist bereits vorhanden.

Statt nun den kompletten Schlüssel auszulesen, können Sie die Abfrage auch auf einen einzelnen Wert beschränken. Hierzu verwenden Sie den Parameter `/v`, gefolgt von der Bezeichnung des auszulesenden Wertes (z.B. `/v AutoAdminLogon`).

```
@Echo off
Reg Query "HKLM\SOFTWARE\Microsoft\Windows _
NT\CurrentVersion\Winlogon" /v AutoAdminLogon
```

Listing 409: 2_05_006.bat

Die Lösung mit VBScript

Wenn Sie mit VBScript das Vorhandensein eines bestimmten Wertes überprüfen möchten, dann verwenden Sie hierfür die `EnumValues`-Methode, die durch die WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`) bereitgestellt wird. Die `EnumValues`-Methode erwartet dabei folgende Angaben, die Sie als Parameter übergeben müssen:

- ▶ **HKLM**: Konstante für den hexadezimalen Wert des Hauptzweiges, der den Schlüssel mit den auszulesenden Werten enthält
- ▶ **strKey**: Schlüssel, dessen Werte ausgelesen werden sollen
- ▶ **strArray**: Das Array, in das die ausgelesenen Werte als Elemente übertragen werden
- ▶ **intType**: Rückgabe des dezimalen Wertes des Datentyps, der ausgelesen wurde (siehe nachfolgende Tabelle)

Datentyp	Dezimaler Wert
REG_SZ	1
REG_EXPAND_SZ	2
REG_BINARY	3
REG_DWORD	4
REG_MULTI_SZ	7

Tabelle 54: Die dezimalen Werte der Datentypen

Um nun das Vorhandensein der Zeichenfolge `AUTOADMINLOGON` des Schlüssels `HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON` zu überprüfen, stellen Sie mit `GetObject` eine Verbindung zur WMI-Klasse `StdRegProv` her und setzen die `EnumValues`-Methode gegen den zuvor genannten Schlüssel ein. Daraufhin werden alle Werte in das Array übertragen. Mit Hilfe einer `For Each`-Schleife können Sie dann auf die einzelnen Elemente zugreifen und die Bezeichnung der Werte per `IF`-Abfrage überprüfen. Ist die Bezeichnung gleich `AutoAdminLogon`, so ist die Bedingung erfüllt, und die `For Each`-Schleife kann mit `Exit For` vorzeitig verlassen werden, da weitere Ergebnisse aufgrund ihrer Eindeutigkeit nicht gefunden werden können.

Option Explicit

Dim oReg
Dim strComputer
Dim strKey
Dim strArray
Dim strValue
Dim intType

Const HKLM = &H80000002
strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows _
NT\CurrentVersion\Winlogon"

Set oReg=GetObject("winmgmts:{impersonationLevel=_
impersonate}!\\" & strComputer & _
"\root\default:StdRegProv")
On Error Resume Next
oReg.EnumValues HKLM, strKey, strArray, intType

For Each strValue In strArray
If strValue = "AutoAdminLogon" then
WScript.Echo "Wert existiert"
Exit For

Listing 410: 2_05_006.vbs


```

End if
Next

Set oReg = Nothing

```

Listing 410: 2_05_006.vbs (Forts.)

109 Einzelne Werte auflisten

In einigen Situationen ist es sehr hilfreich, sich die Werte eines bestimmten Schlüssels einmal auflisten zu lassen. Wenn Sie beispielsweise wissen möchten, welche Software-Komponenten bei Systemstart automatisch aufgerufen werden, die aber nicht in einer der Autostart-Gruppen enthalten sind, so brauchen Sie sich nur die Zeichenfolgen des Schlüssels HKLM\SOFTWARE\ MICROSOFT\WINDOWS\CURRENT-VERSION\RUN auflisten zu lassen. Denn in jeder Zeichenfolge ist der Pfad zu der Software-Komponente hinterlegt, die mit dem Systemstart automatisch aufgerufen wird.

Unter grafischer Oberfläche nehmen Sie hierfür wieder den Registrierungs-Editor zu Hilfe. Wechseln Sie einfach in den entsprechenden Schlüssel, die darin enthaltenen Zeichenfolgen werden automatisch in der rechten Fensterhälfte angezeigt.

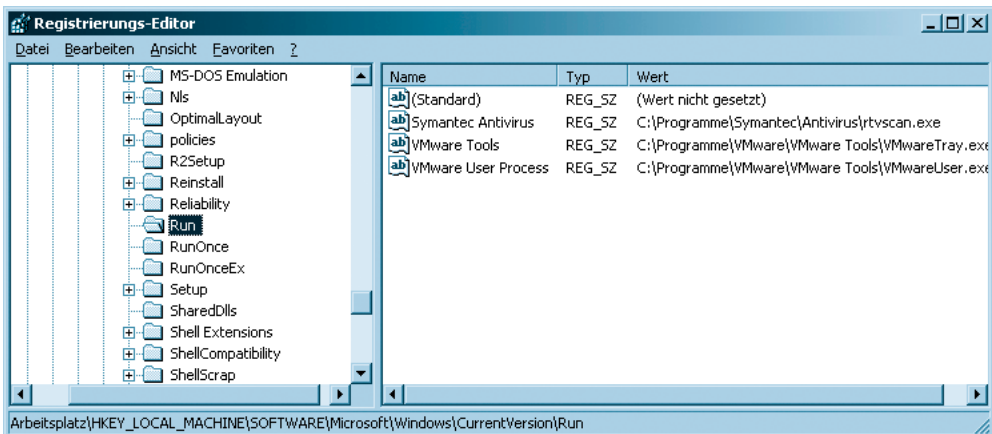


Abbildung 158: Die Werte im Run-Schlüssel auflisten

Die Lösung per Eingabeaufforderung

Möchten Sie die Werte eines bestimmten Schlüssels von der Eingabeaufforderung aus auflisten, so nutzen Sie hierfür den Befehl `Reg Query`, dem Sie lediglich den Schlüssel, dessen Inhalt ausgelesen werden soll, als Parameter übergeben müssen. `Reg Query` stellt daraufhin eine Verbindung zu dem angegebenen Schlüssel her, fragt dessen Inhalt ab und gibt das Ergebnis (untergeordnete Schlüssel, Bezeichnung des Wertes, Datentyp und Daten) am Bildschirm aus.

```
@Echo off
Reg Query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

Listing 411: 2_05_007a.bat

Leider haben Sie hierbei keine Möglichkeiten, die untergeordneten Schlüssel außer Acht zu lassen oder die Abfrage auf bestimmte Datentypen zu beschränken. Letzteres ermöglicht aber der Befehl `RegFind`, der Bestandteil der *Windows 2000 Resource Kit-Tools* ist. Sollten Sie im Besitz dieser *Resource Kit-Tools* sein, so können Sie die Abfrage unter anderem auf einen bestimmten Datentyp beschränken. Möchten Sie beispielsweise nur die Zeichenfolgen des zuvor genannten Schlüssels auslesen, so führen Sie einfach folgende Befehlszeile aus:

```
@Echo off
Regfind -p HKEY_LOCAL_MACHINE\SOFTWARE\Micro_
      soft\Windows\CurrentVersion\Run -t REG_SZ
```

Listing 412: 2_05_007b.bat

Die Lösung mit VBScript

Die Werte eines bestimmten Schlüssels lassen sich natürlich auch mit VBScript auslesen. Hierfür stellt die WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`) die `EnumValues`-Methode bereit, die folgende Angaben als Parameter erwartet und daraufhin die Werte des angegebenen Schlüssels in ein Array überträgt:

- ▶ `HKLM`: Die Konstante für den hexadezimalen Wert des Hauptzweiges, der die auszulesenden Werte enthält
- ▶ `strKey`: Der Schlüssel, dessen Werte ausgelesen werden sollen
- ▶ `strArray`: Das Array, in das die ausgelesenen Werte als Elemente übertragen werden
- ▶ `intType`: Rückgabe des dezimalen Wertes des Datentyps, der ausgelesen wurde (siehe Tabelle 54: Die dezimalen Werte der Datentypen auf Seite 508)

Das nachfolgende Skript liest die Werte des Schlüssels `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN` aus und schreibt die Bezeichnungen der Werte in eine Datei namens *Datei.txt*, die im Ordner `C:\Temp` abgelegt wird.

Im ersten Schritt werden die Konstanten und Variablen deklariert, damit das Skript ohne große Änderungen an neue Anforderungen angepasst werden kann. Anschließend wird mit der `GetObject`-Methode eine Verbindung zur WMI-Klasse `StdRegProv` hergestellt, der `Impersonation-Level` auf `Impersonate` gesetzt und eine Referenzierung des `FileSystem-Objektes` durchgeführt.

Danach folgt der eigentliche Aufruf der `EnumValues`-Methode gegen den `RUN`-Schlüssel, wobei die zuvor genannten Parameter übergeben werden. Die `EnumValues`-

Methode liest daraufhin die Werte des angegebenen Schlüssels aus und überträgt diese nach `strArray`. Mit Hilfe einer `For Each`-Schleife können die Elemente dann ausgelesen und in die Datei geschrieben werden.

Option Explicit

```
Dim oReg
Dim oFS
Dim oFile
Dim strComputer
Dim strFile
Dim strKey
Dim strArray
Dim strValue
Dim intType

Const HKLM = &H80000002
Const Write = 2
strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
strFile = "C:\Temp\Datei.txt"

Set oFS=CreateObject("Scripting.FileSystemObject")
Set oFile=oFS.OpenTextFile(strFile, Write, True)
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate!}\\\" & strComputer & _
    "\root\default:StdRegProv")
On Error Resume Next
oReg.EnumValues HKLM, strKey, strArray, intType

For Each strValue In strArray
    oFile.WriteLine (strValue)
Next

Set oReg = Nothing
Set oFS = Nothing
Set oFile = Nothing
```

Listing 413: 2_05_007.vbs

110 Neue Werte erstellen

Wenn Sie unter grafischer Oberfläche einen neuen Wert in die Registrierung schreiben möchten, dann nehmen Sie wieder den Registrierungs-Editor zu Hilfe. Wechseln Sie in den Schlüssel, der um den neuen Wert erweitert werden soll, und wählen Sie im Kontextmenü des Schlüssels den Eintrag **NEU** aus, woraufhin Sie den Datentyp des neuen Wertes auswählen müssen:

- ▶ Zeichenfolge
- ▶ Binärwert
- ▶ DWORD-Wert
- ▶ Wert der mehrteiligen Zeichenfolge
- ▶ Wert der erweiterbaren Zeichenfolge

Welchen Datentyp Sie auswählen müssen, können Sie der Beschreibung zum Wert entnehmen. Bei Registrierungseinstellungen zum Betriebssystem selbst finden Sie einen Großteil der Beschreibungen in der *Windows Server 2003 Registry Reference*, die Bestandteil des *Windows Server 2003 Resource Kits* ist oder auch auf der Internetseite <http://www.microsoft.com> publiziert wurde.

Der Windows Server 2003 unterstützt allerdings mehr Datentypen, als diese vom Registrierungs-Editor zur Auswahl angeboten werden. Die nachfolgende Tabelle zeigt, welche Datentypen im Einzelnen unterstützt werden:

Datentyp	Beschreibung
REG_SZ	Zeichenfolge fester Länge
REG_MULTI_SZ	Mehrteilige Zeichenfolge, deren Werte durch ein bestimmtes Zeichen (Komma, Leerzeichen ...) voneinander getrennt werden
REG_EXPAND_SZ	Erweiterbare Zeichenfolge
REG_BINARY	Binärdaten, die im Registrierungs-Editor im hexadezimalen Format oder als Binärwert angezeigt werden
REG_DWORD	Ganzzahliger Wert (32 Bit), der im Registrierungs-Editor im dezimalen, hexadezimalen oder binären Format angezeigt wird
REG_DWORD_LITTLE_ENDIAN	Ganzzahliger Wert (32 Bit), der im Registrierungs-Editor im dezimalen, hexadezimalen oder binären Format angezeigt wird, dabei steht das niedrigstwertige Byte am Anfang. Dieses Format entspricht dem Typen REG_DWORD.
REG_DWORD_BIG_ENDIAN	Ganzzahliger Wert (32 Bit), der im Registrierungs-Editor im dezimalen, hexadezimalen oder binären Format angezeigt wird, dabei steht das höchstwertige Byte am Ende
REG_LINK	Zeichenfolge, die einen Link zwischen applikations- oder systemspezifischen Daten und Werten der Registrierung bezeichnet
REG_FULL_RESOURCE_DESCRIPTOR	Verschachtelte Arrays zur Speicherung einer Ressourcenliste, die im Registrierungs-Editor im hexadezimalen Format als Binärwert angezeigt wird

Tabelle 55: Die Datentypen

Um beispielsweise im Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENT-VERSION\RUN eine neue Zeichenfolge mit der Bezeichnung *Calc* anzulegen, starten Sie den Registrierungs-Editor, wechseln in den zuvor genannten Pfad und wählen im Kontextmenü des Schlüssels RUN den Eintrag NEU und anschließend ZEICHENFOLGE aus. Daraufhin wird eine neue Zeichenfolge mit der Bezeichnung NEUER WERT #1 angelegt. Überschreiben Sie NEUER WERT #1 mit *Calc*, und führen Sie einen Doppelklick auf die neue Zeichenfolge aus, um die eigentliche Zeichenfolge zu hinterlegen. Geben Sie *C:\Windows\System32\calc.exe* ein, und schließen Sie das Dialogfenster wieder. Nach einem Neustart des Computersystems wird im Anschluss an die Windows-Anmeldung automatisch der Taschenrechner gestartet.

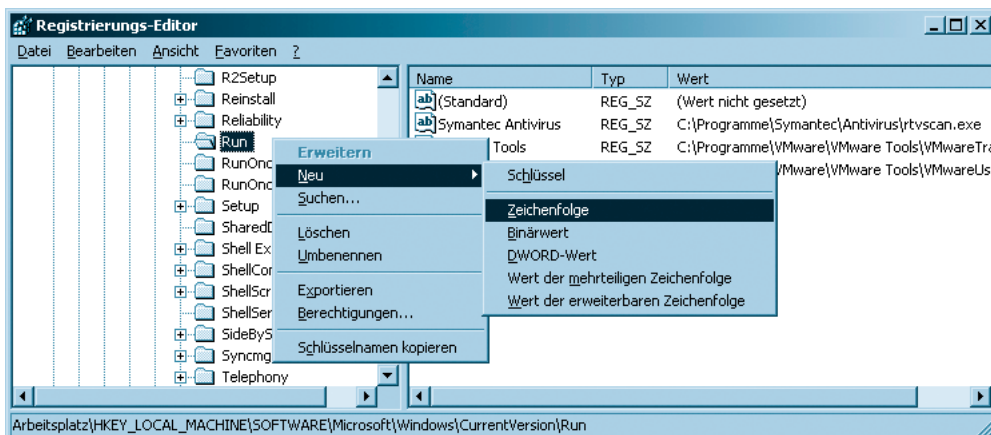


Abbildung 159: Eine neue Zeichenfolge erstellen

Die Lösung per Eingabeaufforderung

Neue Werte lassen sich natürlich auch per Eingabeaufforderung erstellen. Hierzu verwenden Sie den Befehl `Reg Add`, der standardmäßig im Befehlsumfang von Windows Server 2003 enthalten ist. Zum Anlegen neuer Werte erwartet der Befehl `Reg Add` dabei folgende Angaben, die Sie mit dem Befehlsaufruf als Parameter übergeben müssen:

- ▶ Schlüssel (/s): Der Schlüssel, der um einen neuen Wert ergänzt werden soll
- ▶ Bezeichnung (/v): Die Bezeichnung des neuen Wertes
- ▶ Datentyp (/t): Der Datentyp des neuen Wertes (siehe Rezept Neue Werte erstellen auf Seite 513). Mögliche Werte sind `REG_SZ`, `REG_EXPAND_SZ`, `REG_MULTI_SZ`, `REG_DWORD`, `REG_DWORD_LITTLE_ENDIAN`, `REG_DWORD_BIG_ENDIAN`, `REG_BINARY` und `REG_FULL_RESOURCE_DESCRIPTOR`.
- ▶ Daten (/d): Die eigentlichen Daten, die dem neuen Wert zugewiesen werden sollen

Der `Reg Add`-Befehl stellt daraufhin eine Verbindung zu dem angegebenen Schlüssel her und versucht, den neuen Wert anzulegen. Falls der Wert bis dato noch nicht in

der Registrierung existiert, so wird dieser neu angelegt. Existiert der Wert dagegen schon, so müssen Sie das Überschreiben des vorhandenen Wertes von Hand bestätigen, was Sie jedoch mit dem zusätzlichen Parameter /f erzwingen können.

```
@Echo off
Reg Add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run _
/v Calc /t REG_SZ /d "c:\Windows\System32\calc.exe" /f
```

Listing 414: 2_05_008.bat

Die Lösung mit VBScript

Um neue Werte per VBScript in die Registrierung zu schreiben, stellt die WMI-Klasse StdRegProv (WMI-Namespace root\default) für jeden Datentyp eine eigene Methode zur Verfügung:

Datentyp	Methode
REG_SZ	SetStringValue
REG_MULTI_SZ	SetMultiStringValue
REG_EXPAND_SZ	SetExpandedStringValue
REG_DWORD	SetDWORDValue
REG_BINARY	SetBinaryValue

Tabelle 56: Die Methoden zum Erstellen neuer Werte

Möchten Sie beispielsweise im Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN eine neue Zeichenfolge mit der Bezeichnung *Calc* erstellen, so stellen Sie mit der *GetObject*-Methode eine Verbindung zur WMI-Klasse *StdRegProv* her und führen anschließend die *SetStringValue*-Methode aus, der Sie folgende Parameter übergeben:

- ▶ **HKLM:** Die Konstante für den hexadezimalen Wert des Hauptzweiges, der um einen neuen Wert ergänzt werden soll
- ▶ **strKey:** Der Schlüssel (jedoch ohne Angabe des Hauptzweiges), der um einen neuen Wert ergänzt werden soll
- ▶ **strValueName:** Die Bezeichnung des neuen Wertes
- ▶ **strValue:** Die Daten, die dem neuen Wert zugewiesen werden sollen

Das nachfolgende Skript zeigt ein solches Beispiel: Zunächst werden die Konstanten und Variablen definiert. Anschließend wird mit der *GetObject*-Methode eine Verbindung zur WMI-Klasse *StdRegProv* hergestellt und die *SetStringValue*-Methode gegen den Schlüssel ausgeführt, der um den per Parameter übergebenen Wert ergänzt werden soll. Und soll statt einer Zeichenfolge ein anderer Datentyp angelegt werden, so brauchen Sie nur die entsprechende Methode einzusetzen.

Option Explicit

```
Dim oReg
Dim strComputer
Dim strKey
Dim strValueName
Dim strValue
```

```
Const HKLM = &H80000002
strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
strValueName = "Calc"
strValue = "C:\Windows\System32\Calc.exe"
```

```
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")
On Error Resume Next
oReg.SetStringValue HKLM, strKey, strValueName, strValue
```

```
Set oReg = Nothing
```

Listing 415: 2_05_008.vbs

111 Werte umbenennen

Ähnlich wie Sie einzelne Schlüssel umbenennen können (siehe Rezept Schlüssel umbenennen auf Seite 502), so können Sie auch vorhandene Werte umbenennen. Wenn Sie beispielsweise einen Schlüssel um einen neuen Wert ergänzt und sich dabei aus Versehen verschrieben haben, müssen Sie den Wert nicht löschen und neu anlegen. Es genügt, wenn Sie den Wert einfach umbenennen.

Hierzu starten Sie den Registrierungs-Editor, und wechseln Sie in den Schlüssel, dessen Wert Sie umbenennen möchten. Markieren Sie den entsprechenden Wert, und wählen Sie im Kontextmenü den Eintrag UMBENENNEN aus. Daraufhin wird der Bearbeitungsmodus für die Wertbezeichnung geöffnet, und Sie können die aktuelle Bezeichnung durch eine neue Bezeichnung ersetzen (siehe Abbildung 160).

Die Lösung per Eingabeaufforderung

Um von der Eingabeaufforderung aus einen bestimmten Wert umzubenennen, bleibt nur das manuelle Auslesen des Wertes mit Hilfe des `Reg Query`-Befehls, das erneute Zurückschreiben mit `Reg Add` und das anschließende Löschen des alten Wertes mit `Reg Delete`, da es zum Umbenennen einzelner Werte leider keinen entsprechenden Befehl gibt.

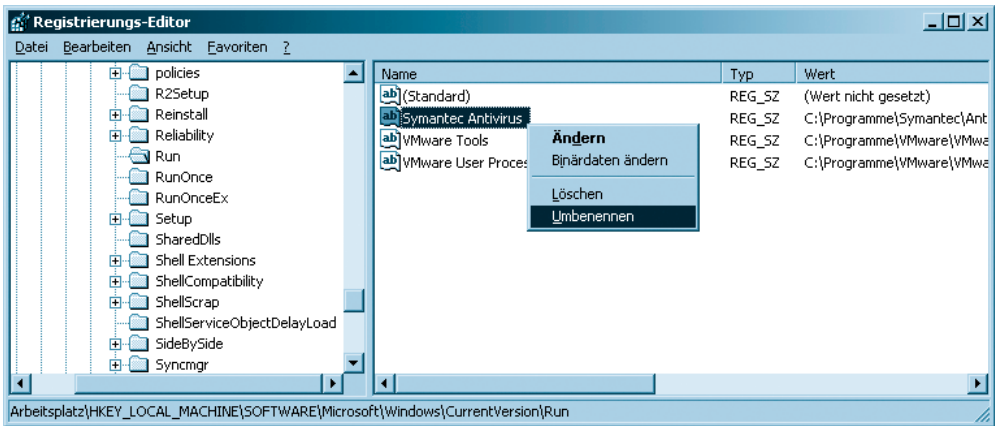


Abbildung 160: Einen vorhandenen Wert umbenennen

Im nachfolgenden Abschnitt finden Sie jedoch ein Listing, womit Sie die Aufgabe per VBScript automatisieren und damit weitaus schneller durchführen können.

Die Lösung mit VBScript

Wenn Sie einen vorhandenen Wert per VBScript umbenennen möchten, so lesen Sie den Wert (inkl. Daten) zunächst aus und schreiben ihn dann unter Angabe einer neuen Bezeichnung wieder in die Registrierung. Je nachdem, welchen Datentyp Sie hierbei bearbeiten möchten, nutzen Sie die entsprechende Get-/Set-Methode der WMI-Klasse StdRegProv (siehe auch Rezepte Existiert der Wert bereits? auf Seite 507 und Neue Werte erstellen auf Seite 514). Anschließend löschen Sie den ursprünglichen Wert aus der Registrierung heraus.

Um beispielsweise die Zeichenfolge *Symantec Antivirus*, die im Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN abgelegt ist, nach *Symantec Antivirus* umzubenennen und die hinterlegten Daten beizubehalten, setzen Sie erst einmal die EnumValues-Methode der WMI-Klasse StdRegProv (WMI-Namespace root\default) ein, um die Werte des Schlüssels unter Angabe ihres jeweiligen Datentyps in ein Array einzulesen. Anschließend prüfen Sie für jedes Element des Arrays, ob dessen Name gleich dem des umzubenennenden Wertes entspricht. Ist dies der Fall, so prüfen Sie anschließend den Datentyp des Wertes, wofür Sie den Wert der EnumValues-Ausgabevariable arrValueTypes abfragen.

Innerhalb einer Select Case-Anweisung können Sie dann die entsprechende Set- und Get-Anweisung einsetzen, um den Wert und dessen Daten auszulesen und unter Angabe der neuen Bezeichnung wieder in die Registrierung zu schreiben. Zu guter Letzt wenden Sie die DeleteValue-Methode an, um den ursprünglichen Wert aus der Registrierung zu löschen.

Option Explicit

```
Dim oReg
Dim strComputer
Dim strKey
Dim strValue
Dim strOldValueName
Dim strNewValueName
Dim arrEntryNames
Dim arrValueTypes
Dim arrValue
Dim intCounter
```

```
Const HKLM = &H80000002
Const REG_SZ = 1
Const REG_Expand_SZ = 2
Const REG_Binary = 3
Const REG_DWORD = 4
Const REG_Multi_SZ = 7
```

```
strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
strOldValueName = "Symantec Antivirus"
strNewValueName = "Symantec Antivirus"
```

```
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\default:StdRegProv")
oReg.EnumValues HKLM, strKey, arrEntryNames, arrValueTypes
```

```
For intCounter = 0 to UBound(arrEntryNames)
    If arrEntryNames(intCounter) = strOldValueName Then
        Select Case arrValueTypes(intCounter)
            Case REG_SZ
                oReg.GetStringValue HKLM, strKey, arrEntryNames(intCounter), _
                    strValue
                oReg.SetStringValue HKLM, strKey, strNewValueName, strValue
            Case REG_Expand_SZ
                oReg.GetExpandedStringValue HKLM, strKey, _
                    arrEntryNames(intCounter), strValue
                oReg.SetExpandedStringValue HKLM, strKey, strNewValueName, _
                    strValue
            Case REG_Binary
                oReg.GetBinaryValue HKLM, strKey, arrEntryNames(intCounter), _
                    strValue
                oReg.SetBinaryValue HKLM, strKey, strNewValueName, strValue
            Case REG_DWORD
```

```
oReg.GetDWORDValue HKLM, strKey, arrEntryNames(intCounter), _
    strValue
oReg.SetDWORDValue HKLM, strKey, strNewValueName, strValue
Case REG_Multi_SZ
oReg.GetMultiStringValue HKLM, strKey, _
    arrEntryNames(intCounter), _arrValue
oReg.SetMultiStringValue HKLM, strKey, strNewValueName, arrValue
End Select
End If
Next

oReg.DeleteValue HKLM, strKey, strOldValueName
Set oReg = Nothing
```

Listing 416: 2_05_009.vbs (Forts.)

112 Werte löschen

Einzelne Werte lassen sich natürlich ohne Probleme wieder aus der Registrierung entfernen. Wobei Sie vorher aber sicherstellen müssen, dass der entsprechende Wert nicht mehr benötigt wird und auch wirklich gelöscht werden kann. Denn anderenfalls laufen Sie Gefahr möglicher Systeminstabilitäten oder gar eines kompletten Systemabsturzes.

Um einen vorhandenen Wert aus der Registrierung zu löschen, nehmen Sie unter grafischer Oberfläche wieder den Registrierungs-Editor zu Hilfe. Wechseln Sie in den Schlüssel, der den zu löschenden Wert enthält, markieren Sie den Wert, und wählen Sie im Kontextmenü den Eintrag LÖSCHEN aus. Sie werden daraufhin gefragt, ob Sie den Wert auch wirklich löschen möchten. Klicken Sie auf JA, um den Wert inklusive der hinterlegten Daten unwiderruflich aus der Registrierung zu entfernen.

Registrierung

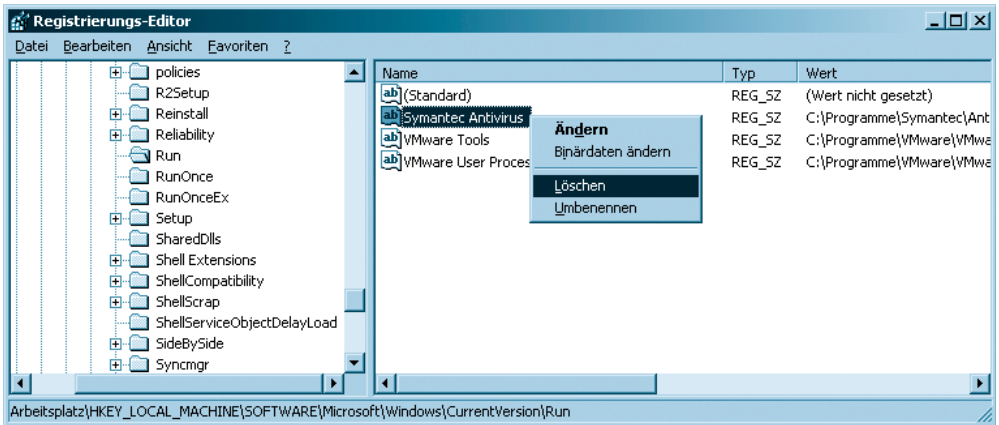


Abbildung 161: Einen vorhandenen Wert aus der Registrierung löschen

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung verwenden Sie zum Löschen einzelner Werte den Befehl `Reg Delete`, der standardmäßig im Befehlsumfang von Windows Server 2003 enthalten ist. `Reg Delete` erwartet die Angabe des Schlüssels, der den zu löschenden Wert enthält, wie auch den eigentlichen Wert selbst. Diese beiden Angaben müssen Sie dem Befehl `Reg Delete` als Parameter übergeben.

`Reg Delete` stellt daraufhin eine Verbindung zu dem angegebenen Schlüssel her und prüft, ob der per Parameter übergebene Wert vorhanden ist. Ist das der Fall, so fragt `Reg Delete` noch einmal nach, ob der Wert auch wirklich gelöscht werden soll, was Sie manuell bestätigen müssen. Mit dem Parameter `/f` können Sie das Löschen des Wertes jedoch erzwingen, die lästige Nachfrage, die eine Unterbrechung der Befehlsausführung bedeutet, entfällt damit.

```
@Echo off
Reg Delete HKLM\SOFTWARE\Microsoft\Windows\Current_
    Version\Run /v "Symantec Antivirus" /f
```

Listing 417: 2_05_010.bat

Die Lösung mit VBScript

Wenn Sie mit VBScript einzelne Werte aus der Registrierung löschen möchten, so steht Ihnen hierfür die `DeleteValue`-Methode zur Verfügung, die seitens der WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`) bereitgestellt wird. Die `DeleteValue`-Methode erwartet dabei folgende Parameter, die Sie mit dem Methodenaufruf als Parameter übergeben müssen:

- ▶ `HKLM`: Die Konstante für den hexadezimalen Wert des Hauptzweiges
- ▶ `strKey`: Der Schlüssel (jedoch ohne Angabe des Hauptzweiges), der den zu löschenden Wert enthält
- ▶ `strValueName`: Die Bezeichnung des Wertes, der gelöscht werden soll

Das folgende Skript zeigt ein Beispiel dafür, wie Sie mit der `DeleteValue`-Methode den Wert *Symantec Antivirus* aus dem Schlüssel `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` löschen können. Zunächst wird mit der `GetObject`-Methode eine Verbindung zur WMI-Klasse `StdRegProv` hergestellt. Danach folgt der eigentliche Aufruf der `DeleteValue`-Methode und das Übergeben der erforderlichen Parameter (`HKLM`, `strKey`, `strValueName`).

```
Option Explicit
```

```
Dim oReg
```

Listing 418: 2_05_010.vbs

```
Dim strComputer
Dim strKey
Dim strValueName

Const HKLM = &H80000002
strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
strValueName = "Symantec Antivirus"

Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")
On Error Resume Next
oReg.DeleteValue HKLM, strKey, strValueName

Set oReg = Nothing
```

Listing 418: 2_05_010.vbs (Forts.)

113 Welche Daten enthält der Wert?

Es gibt eine ganze Reihe von Betriebssystemeinstellungen, die Sie nur in der Registrierung vornehmen können, da die grafische Oberfläche keine entsprechenden Möglichkeiten zur Ansicht oder zur Änderung bietet. In einigen Fällen müssen Sie also direkt in der Registrierung nachsehen, welche Optionen gesetzt sind.

Registrierung

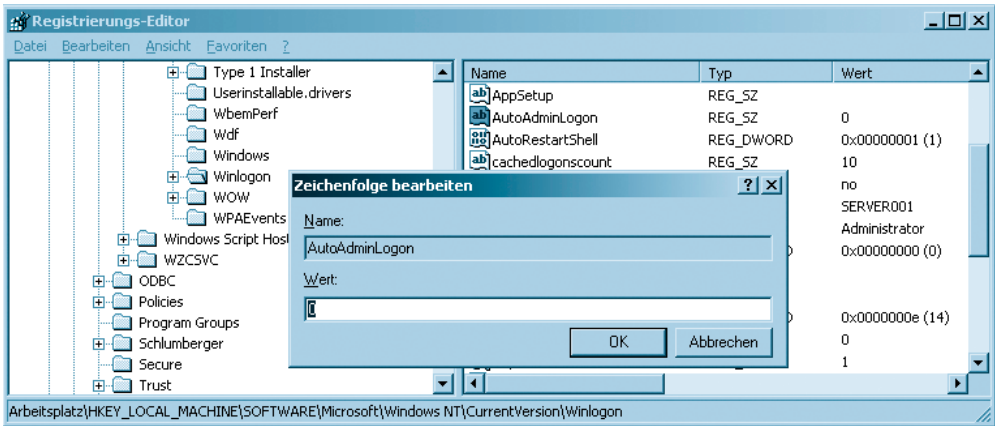


Abbildung 162: Die Daten eines Wertes anzeigen lassen

Wenn Sie z.B. wissen möchten, ob die automatische Windows-Anmeldung (*AutoAdminLogon*) aktiviert ist, dann starten Sie unter grafischer Oberfläche den Registrierungs-Editor, wechseln in den Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON, prüfen, ob die Zeichenfolge AUTOADMINLOGON vor-

handen ist und – wenn ja – welche Daten hinterlegt sind. Enthält die Zeichenfolge eine 0, so ist die Funktion deaktiviert, bei einer 1 dagegen ist die automatische Windows-Anmeldung aktiviert.

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus die Daten eines bestimmten Wertes auslesen möchten, so nutzen Sie hierfür den Befehl `Reg Query`, dem Sie lediglich den Schlüssel, der den auszulesenden Wert enthält, wie auch den Wert selbst als Parameter übergeben müssen. `Reg Query` stellt daraufhin eine Verbindung zu dem angegebenen Schlüssel her, liest die Daten des betreffenden Wertes aus und gibt das Ergebnis (Schlüssel, Wert, Datentyp, Daten) am Bildschirm aus.

Das nachfolgende Beispiel führt eine `Reg Query`-Abfrage gegen den Wert `AutoAdminLogon` aus, der sich im Schlüssel `HKLM\SOFTWARE\MICROSOFT NT\CURRENTVERSION\WINLOGON` befindet:

```
@Echo off
Reg Query "HKLM\SOFTWARE\Microsoft\Windows _
    NT\CurrentVersion\Winlogon" /v AutoAdminLogon
```

Listing 419: 2_05_011.bat

Die Lösung mit VBScript

Auch mit VBScript lassen sich die Daten bestimmter Werte auslesen. Hierzu stellt die WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`) für jeden Datentyp eine entsprechende Methode bereit, die Sie gegen den betreffenden Wert einsetzen und so deren Inhalt auslesen können.

Datentyp	Methode
Zeichenfolge	GetStringValue
Mehrteilige Zeichenfolge	GetMultiStringValue
Erweiterbare Zeichenfolge	GetExpandedStringValue
DWORD-Wert	GetDWORDValue
Binärwert	GetBinaryValue

Tabelle 57: Die Methoden zum Auslesen der Datentypen

Die Datentypen lassen sich mit Hilfe der `EnumValues`-Methode, genauer gesagt über die Ausgabevariablen `intType` ermitteln (siehe Die Lösung mit VBScript auf Seite 510). Um beispielsweise die Daten der Zeichenfolge `AUTOADMINLOGON` (Schlüssel `HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON`) auszulesen, stellen Sie mit der `GetObject`-Methode zunächst eine Verbindung zu der WMI-Klasse `StdReg-`

Prov her und setzen dann die GetStringValue-Methode gegen den genannten Wert ein. Mit dem Aufruf der GetStringValue-Methode übergeben Sie gleichzeitig die erforderlichen Parameter:

- ▶ HKLM: Die Konstante für den hexadezimalen Wert des Hauptzweiges, der den auszulesenden Wert enthält
- ▶ strKey: Der Schlüssel (jedoch ohne Angabe des Hauptzweiges), der den auszulesenden Wert enthält
- ▶ strValueName: Die Bezeichnung des Wertes, dessen Daten ausgelesen werden sollen
- ▶ strValue: Variable für die ausgelesenen Daten, worüber das Ergebnis abgefragt werden kann

Das nachfolgende Skript zeigt ein Beispiel, wie Sie mit der GetStringValue-Methode die Daten des zuvor genannten Wertes auslesen und am Bildschirm ausgeben können. Dieses Beispielskript können Sie ohne große Änderungen auch zum Auslesen anderer Werte einsetzen. Sie müssen lediglich die entsprechende Methode verwenden (die Syntax bleibt dabei immer gleich), den entsprechenden Hauptzweig und Schlüssel angeben und die Variable strValue abfragen, um an die gewünschte Information zu kommen.

Option Explicit

```
Dim oReg
Dim strComputer
Dim strKey
Dim strValue
Dim strValueName
```

```
Const HKLM = &H80000002
strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows _
    NT\CurrentVersion\Winlogon"
strValueName = "AutoAdminLogon"
```

```
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")
On Error Resume Next
oReg.GetStringValue HKLM, strKey, strValueName, strValue
WScript.Echo strValueName & ": " & strValue
```

```
Set oReg = Nothing
```

Listing 420: 2_05_011.vbs

114 Daten auflisten

Statt komplette Werte auszulesen, können Sie sich natürlich auch nur die Daten, die den Werten jeweils zugewiesen sind, anzeigen lassen. Wenn Sie z.B. ausschließlich die Daten des Schlüssels `HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENT-VERSION\WINLOGON` abfragen wollen, so nehmen Sie unter grafischer Oberfläche wieder den Registrierungs-Editor zu Hilfe. Wechseln Sie in den zuvor genannten Schlüssel, und richten Sie die Spaltenbreite der rechten Fensterhälfte so aus, dass Sie die vollständigen Daten der Zeichenfolgen sehen können. Alternativ können Sie die Zeichenfolgen auch per Doppelklick öffnen, um die Daten auszulesen.

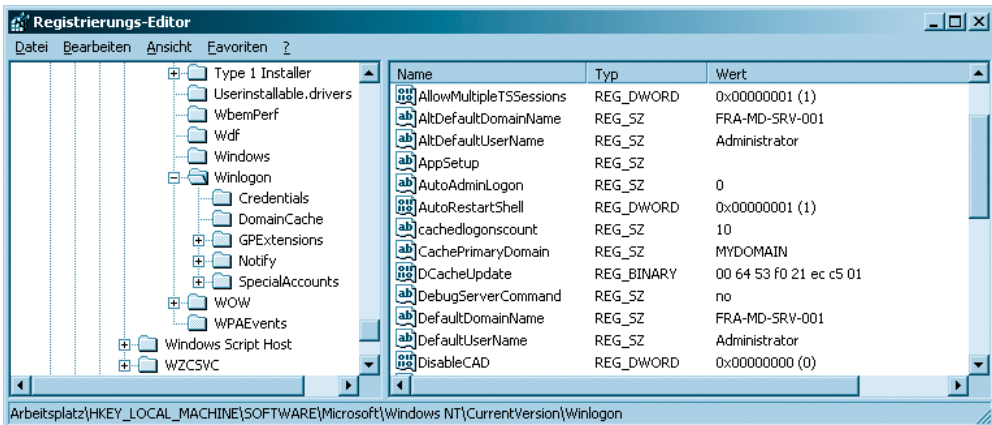


Abbildung 163: Die Daten des Winlogon-Schlüssels auslesen

Die Lösung per Eingabeaufforderung

Wenn Sie die Daten eines bestimmten Schlüssels von der Eingabeaufforderung aus abfragen möchten, so verwenden Sie hierfür den Befehl `Reg Query`, womit Sie die Inhalte (Schlüssel, Werte, Datentypen und Daten) eines angegebenen Schlüssels auslesen und am Bildschirm ausgeben können. Einziger Nachteil hierbei: `Reg Query` gibt grundsätzlich nur die kompletten Inhalte aus und lässt sich nicht auf die explizite Ausgabe der Daten beschränken, sofern Sie nicht auch die jeweiligen Werte als Parameter übergeben.

Das nachfolgende Beispiel führt eine `Reg Query`-Abfrage gegen den Schlüssel `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon` aus und schreibt das Ergebnis in eine Textdatei namens `Datei.txt`, die im Ordner `C:\Temp` abgelegt wird:

```
@Echo off
Reg Query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" _
> C:\Temp.Datei.txt
```

Listing 421: 2_05_012.bat

Mit Hilfe der *Unix-Tools für Win32* können Sie den Inhalt der Ausgabedatei aber automatisiert nachbearbeiten und in vielen Fällen unerwünschte Informationen herauslöschen. In der Kategorie *Dateien und Ordner* finden Sie eine ganze Reihe fertiger Lösungen zu diesem Thema.

Die Lösung mit VBScript

Wenn Sie per VBScript die Daten eines bestimmten Schlüssels auslesen möchten, so setzen Sie zunächst die `EnumValues`-Methode ein, um erst die Namen der Werte des entsprechenden Schlüssels auszulesen bzw. in ein Array zu übertragen. Anschließend können Sie für jeden Wert, der als Element in dem Array hinterlegt ist, dessen Datentyp prüfen und per `Select Case`-Anweisung die entsprechende `Get`-Methode einsetzen, um die eigentlichen Daten auszulesen. Denn das Problem beim Auslesen von Daten mit VBScript ist, dass Sie für jeden Datentyp eine andere `Get`-Methode einsetzen müssen.

Das nachfolgende Skript zeigt ein Beispiel dafür, wie Sie mit VBScript alle Daten des Schlüssels `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN` auslesen und das Ergebnis in eine Datei schreiben können: Zunächst folgt wieder das Deklarieren der benötigten Variablen und Konstanten, damit das Skript halbwegs überschaubar bleibt und auch leichter an neue Anforderungen angepasst werden kann. Danach wird eine Verbindung zum `FileSystem`-Objekt hergestellt, womit die Ausgabedatei (`C:\Temp\Datei.txt`) zum Schreiben (`Write`) geöffnet und später auch per `oFile.WriteLine` mit Daten gefüllt werden kann.

Im Anschluss daran erfolgt das Herstellen einer Verbindung zur WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`), wodurch die `EnumValues`- und die verschiedenen `Get`-Methoden bereitgestellt werden. Zeitgleich wird das Computersystem (`strComputer`) bestimmt und der `Impersonation-Level` auf `Impersonate` gesetzt, damit der Zugriff nicht aufgrund der Berechtigungen scheitert. Dann folgt der Einsatz der `EnumValues`-Methode gegen den Schlüssel `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN`, dem zwei Arrays übergeben werden. Das erste Array (`strArrValueNames`) nimmt die Bezeichnungen der Werte auf, das zweite Array (`strValueTypes`) dagegen die Datentypen der jeweiligen Werte.

Mit der `For Counter = 0 to Ubound(strArrValueNames)`-Anweisung wird die Anzahl der Durchläufe für die `For Next`-Schleife bestimmt. Die Schleife wird exakt so oft durchlaufen, wie auch Werte in `strArrValueNames` enthalten sind. Damit kann sichergestellt werden, dass die nachfolgenden Anweisungen auch für jedes Element von `strArrValueNames` durchgeführt werden und die Schleife nicht zu früh oder zu spät verlassen wird. Mit der `For Next`-Schleife wird dann für jedes Element von `strArrValueNames` der Datentyp (`strArrValueType`) geprüft, woraufhin die Daten dann mit der entsprechenden `Get`-Methode ausgelesen und per `oFile.WriteLine strValue` in die Ausgabedatei geschrieben werden.

Option Explicit

```
Dim oReg
Dim oFS
Dim oFile
Dim strComputer
Dim strKey
Dim strArrValueName
Dim strArrValueType
Dim strValue
Dim strValues
Dim strFile
Dim Counter

Const HKLM = &H80000002
Const REG_SZ = 1
Const REG_EXPAND_SZ = 2
Const REG_BINARY = 3
Const REG_DWORD = 4
Const REG_MULTI_SZ = 7
Const Write = 2

strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon"
strFile = "C:\Temp\Datei.txt"

Set oFS=CreateObject("Scripting.FileSystemObject")
Set oFile=oFS.OpenTextFile(strFile, Write, True)
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\default:StdRegProv")
oReg.EnumValues HKLM,strKey, strArrValueName, strArrValueType

For Counter=0 To UBound(strArrValueName)
    Select Case strArrValueType(Counter)
        Case REG_SZ
            oReg.GetStringValue HKLM, strKey, strArrValueName(Counter),strValue
            oFile.WriteLine strValue
        Case REG_EXPAND_SZ
            oReg.GetExpandedStringValue HKLM, strKey, _
                strArrValueName(Counter),strValue
            oFile.WriteLine strValue
        Case REG_MULTI_SZ
            oReg.GetMultiStringValue HKLM, strKey, _
                strArrValueName(Counter),strValues
            For Each strValue in strValues
                oFile.WriteLine strValue
```

```
Next
Case REG_BINARY
    oReg.GetBinaryValue HKLM, strKey, strArrValueName(Counter),strValues
    For Each strValue in strValues
        oFile.WriteLine strValue
    Next
Case REG_DWORD
    oReg.GetDWORDValue HKLM, strKey, strArrValueName(Counter),strValue
    oFile.WriteLine strValue
End Select
Next

Set oReg = Nothing
Set oFS = Nothing
Set oFile = Nothing
```

Listing 422: 2_05_012.vbs (Forts.)

115 Daten ändern

Einzelne Daten können Sie mit Hilfe des Registrierungs-Editors jederzeit ändern. Dies ist gerade dann sinnvoll, wenn Sie Änderungen an der Systemkonfiguration vornehmen möchten, wofür die grafische Oberfläche keine entsprechenden Möglichkeiten bietet oder wenn Sie bestimmte Systemeinstellungen auch auf andere Computersysteme übertragen möchten.

Registrierung

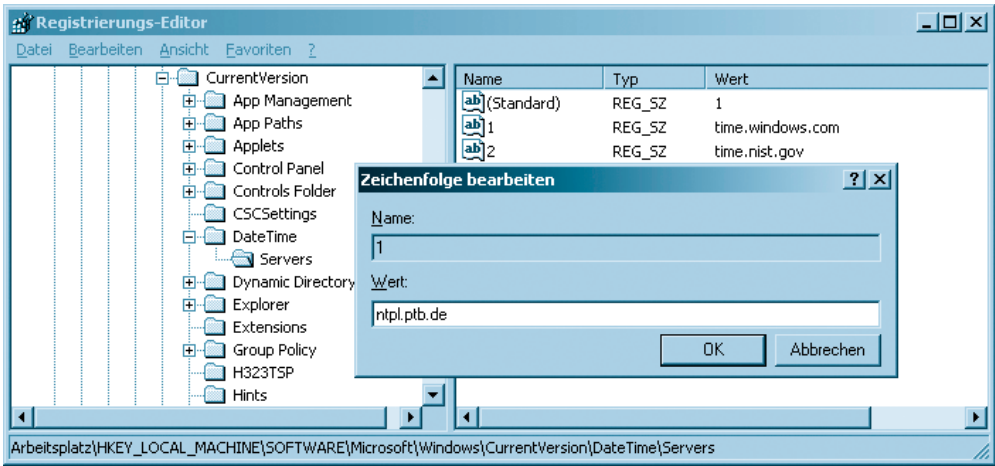


Abbildung 164: Die Daten eines vorhandenen Wertes editieren

Möchten Sie beispielsweise die Liste der Internetzeitserver bearbeiten, die Sie auf der Registerkarte INTERNETZEIT des Systemsteuerungsobjektes DATUM UND ZEIT finden, dann brauchen Sie lediglich die Einträge des Schlüssels HKLM\SOFTWARE\MICRO-

SOFT\WINDOWS\CURRENTVERSION\ DATETIME\SERVERS zu editieren. Denn jede Zeichenfolge enthält die Adresse eines Internetzeitervers, die Sie durch andere Adressen ersetzen können.

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung steht Ihnen leider kein entsprechender Befehl zur Verfügung, womit Sie vorhandene Daten, die bereits in die Registrierung geschrieben wurden, einfach ändern können. Mit dem Befehl `Reg Add` können Sie aber einem vorhandenen Wert neue Daten zuweisen, womit Sie die vorhandenen Daten einfach durch neue Daten überschreiben.

Wenn Sie beispielsweise dem Wert 1, der sich im Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\DATETIME\SERVERS befindet, neue Daten zuweisen und damit die vorhandenen Daten überschreiben möchten, so setzen Sie den `Reg Add`-Befehl gegen den Wert 1 ein und übergeben mit dem Befehlsaufruf von `Reg Add` folgende Parameter:

- ▶ Schlüssel: Der Schlüssel, dessen Werte bzw. Daten geändert werden sollen
- ▶ Wert (/v): Bezeichnung des Wertes, dessen Daten geändert werden sollen
- ▶ Datentyp (/t): Typ der Daten, die in die Registrierung geschrieben werden sollen (dieser Parameter wird nur benötigt, falls der Wert bei Nichtvorhandensein neu angelegt werden soll). Mögliche Datentypen sind `REG_SZ`, `REG_EXPAND_SZ`, `REG_MULTI_SZ`, `REG_DWORD`, `REG_DWORD_LITTLE_ENDIAN`, `REG_DWORD_BIG_ENDIAN`, `REG_BINARY` und `REG_FULL_RESOURCE_DESCRIPTOR` (siehe Rezept Neue Werte erstellen auf Seite 513)
- ▶ Daten (/d): Daten, die in die Registrierung geschrieben werden sollen

Der `Reg Add`-Befehl versucht daraufhin, eine Verbindung zu dem angegebenen Wert herzustellen und die per Parameter übergebenen Daten in die Registrierung zu schreiben. Falls der Wert noch nicht existiert, wird der Wert (inklusive Daten) neu angelegt. Existiert der Wert dagegen und wurden bereits Daten zugewiesen, so versucht `Reg Add`, die Daten zu überschreiben, was Sie jedoch von Hand bestätigen müssen. Diese Rückfrage können Sie aber vermeiden, wenn Sie zusätzlich den Parameter `/f` angeben, womit Sie das Überschreiben vorhandener Daten erzwingen können.

```
@Echo off
Reg Add HKLM\SOFTWARE\Microsoft\Windows\Current_
    Version\DateTime\Servers /v 1 /t REG_SZ _
    /d ntpl.ptb.de /f
```

Listing 423: 2_05_013.bat

Die Lösung mit VBScript

Wenn Sie per VBScript vorhandene Daten der Registrierung ändern möchten, so überschreiben Sie auch in diesem Fall die vorhandenen Daten durch neue Daten. Welche Methode (SetStringValue, SetExpandedStringValue, SetDWORDValue ...) Sie hierfür verwenden, hängt dabei vom jeweiligen Datentyp ab (siehe Rezept Neue Werte erstellen auf Seite 514).

Möchten Sie beispielsweise dem Wert 1, der im Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\DATETIME\SERVERS abgelegt ist, den Hostnamen eines neuen Zeitserverns zuweisen, so verwenden Sie hierfür die SetStringValue-Methode der WMI-Klasse StdRegProv (WMI-Namespace root\default), da Sie in diesem Fall den Inhalt einer Zeichenfolge ändern werden. Die Methode erwartet die Angabe folgender Parameter:

- ▶ HKLM: Konstante für den hexadezimalen Wert des Hauptzweiges, dessen Daten geändert werden sollen
- ▶ strKey: Schlüssel (ohne Angabe des Hauptzweiges), unter dem die zu ändernden Daten abgelegt sind
- ▶ strValueName: Wert, dessen Daten geändert werden sollen
- ▶ strValue: Die neuen Daten, die in die Registrierung geschrieben werden sollen

Die SetStringValue-Methode stellt daraufhin eine Verbindung zu dem angegebenen Schlüssel her und schreibt Wert und Daten in die Registrierung. Sofern der Wert noch nicht existiert, wird dieser ohne jegliche Rückfrage neu angelegt. Ist der Wert dagegen schon vorhanden, so werden Wert und Daten mit den per Parameter übergebenen Informationen einfach überschrieben.

Option Explicit

```
Dim oReg
Dim strComputer
Dim strKey
Dim strValueName
Dim strValue
```

```
Const HKLM = &H80000002
strComputer = "."
strKey = "Software\Microsoft\Windows\Current_
    Version\DateTime\Servers"
strValueName = "1"
strValue = "ntp1.ptb.de"
```

```
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}\\\" & strComputer & _
```

Listing 424: 2_05_013.vbs

```
"\root\default:StdRegProv")
On Error Resume Next
oReg.SetStringValue HKLM, strKey, strValueName, strValue

Set oReg = Nothing
```

Listing 424: 2_05_013.vbs (Forts.)

116 Daten löschen

Ähnlich wie Sie Schlüssel und Werte löschen können, so können Sie auch einzelne Daten aus der Registrierung entfernen. Und um die Daten eines Wertes zu löschen, nutzen Sie unter grafischer Oberfläche wieder den Registrierungs-Editor. Wechseln Sie in den entsprechenden Schlüssel, und öffnen Sie den betreffenden Wert durch einen Doppelklick. Löschen Sie den Eintrag des Wertes heraus, und übernehmen Sie die Änderung mit einem Klick auf OK.

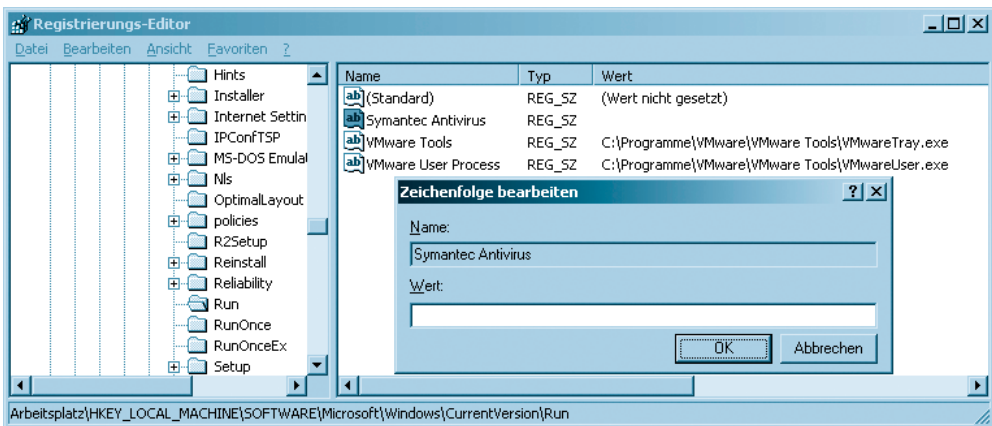


Abbildung 165: Die Daten eines Wertes löschen

Die Lösung per Eingabeaufforderung

Um bestimmte Registrierungseinstellungen (Schlüssel, Werte) von der Eingabeaufforderung aus zu löschen, nutzen Sie normalerweise den Befehl `Reg Delete`. Leider bietet Ihnen dieser Befehl aber keine Möglichkeit, um ausschließlich die Daten eines bestimmten Wertes entfernen können. Wenn Sie jedoch den Wert, dessen Daten Sie löschen wollen, mit `Reg Add` neu in die Registrierung schreiben und dabei keine Daten übergeben, so können Sie die Daten auch ohne entsprechende Option seitens des `Reg Delete`-Befehls löschen.

Wenn Sie zum Beispiel die Daten der Zeichenfolge *Symantec Antivirus* (Schlüssel `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN`) entfernen möchten, ohne den kompletten Wert aus der Registrierung zu entfernen, so überschreiben Sie den Wert ein-

fach und weisen dem Wert dabei keine Daten zu. Setzen Sie den `Reg Add`-Befehl gegen den Schlüssel ein, der den betreffenden Wert enthält, und übergeben Sie folgende Parameter:

- ▶ Schlüssel: Der Schlüssel, der den entsprechenden Wert enthält
- ▶ Wert (/v): Die Bezeichnung des Wertes, der überschrieben werden soll
- ▶ Daten (/d): Die Daten, die Sie im Normalfall dem Wert zuweisen würden. Um – wie in diesem Beispiel beschrieben – die Daten eines Wertes zu löschen, verwenden Sie den Parameter ohne Angabe von Daten.

Die folgenden Befehlszeilen überschreiben den Wert *Symantec Antivirus* mit einem Nullwert, womit Sie das Löschen der Daten improvisieren können:

```
@Echo off
Reg Add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run _
/v "Symantec Antivirus" /d "" /f
```

Listing 425: 2_05_014.bat

Die Lösung mit VBScript

Wenn Sie die Daten eines bestimmten Wertes per VBScript löschen möchten, so müssen Sie den gleichen Umweg gehen wie in der Eingabeaufforderung. Sprich: Sie müssen den Wert erneut in die Registrierung schreiben, wobei Sie dem Wert aber keine Daten übergeben.

Um neue Werte in die Registrierung zu schreiben, stellt die WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`) für jeden Datentyp eine entsprechende Methode bereit (siehe Rezept Neue Werte erstellen auf Seite 514). Möchten Sie beispielsweise die Daten der Zeichenfolge *Symantec Antivirus* löschen, die sich im Schlüssel `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN` befindet, so setzen Sie die `SetStringValue`-Methode gegen den zuvor genannten Schlüssel ein und übergeben mit dem Methodenaufruf folgende Parameter:

- ▶ `HKLM`: Die Konstante für den hexadezimalen Wert des Hauptzweiges, der die entsprechende Zeichenfolge enthält
- ▶ `strKey`: Der Schlüssel (jedoch ohne Angabe des Hauptzweiges), der die betreffende Zeichenfolge enthält
- ▶ `strValueName`: Die Bezeichnung des Wertes, der in die Registrierung geschrieben werden soll
- ▶ `strValue`: Die Daten, die dem Wert zugewiesen werden sollen. Da in diesem Beispiel die Daten eines Wertes gelöscht werden sollen, wird die Variable `strValue` nur mit einem Nullwert angegeben.

Das nachfolgende Skript zeigt ein solches Beispiel. Der Variablen `strValue` werden dabei keine Daten zugewiesen, da die `SetStringValue`-Methode zum Löschen von

Daten gewissermaßen zweckentfremdet wird. Mit dem Aufruf der `SetStringValue`-Methode wird der Wert vom Datentyp `REG_SZ` (= Zeichenfolge) erneut in die Registrierung geschrieben,

```
Option Explicit
Dim oReg
Dim strComputer
Dim strKey
Dim strValueName
Dim strValue

Const HKLM = &H80000002
strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
strValueName = "Symantec Antivirus"
strValue = ""

Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")
On Error Resume Next
oReg.SetStringValue HKLM, strKey, strValueName, strValue

Set oReg = Nothing
```

Listing 426: 2_05_014.vbs

117 Besitzer festlegen

Um die Besitzrechte eines bestimmten Objektes auf einen anderen Benutzer oder auf eine andere Gruppe zu übertragen, gehen Sie ähnlich wie beim Anpassen der Berechtigungen vor. Sofern Sie Mitglied der Gruppe *Administratoren* sind oder zumindest das Privileg *Besitzer festlegen* haben, so starten Sie den Registrierungs-Editor, markieren den Schlüssel, für den Sie die Besitzrechte übertragen möchten, und wählen im Kontextmenü (oder im Menü BEARBEITEN) den Eintrag *BERECHTIGUNGEN...* aus. Anschließend klicken Sie auf *ERWEITERT*, um das Dialogfenster mit den erweiterten Berechtigungen zu öffnen, und holen die Registerkarte *BESITZER* in den Vordergrund.

Dort klicken Sie auf *WEITERE BENUTZER ODER GRUPPEN...*, anschließend auf *ERWEITERT* und dann auf *JETZT SUCHEN*, womit die Benutzer- und Gruppenkonten aufgelistet werden. Markieren Sie den Benutzer oder die Gruppe, für die Sie die Besitzrechte übergeben möchten, und schließen Sie die letzten beiden Dialogfenster jeweils mit einem Klick auf *OK*, so dass Sie wieder auf der Registerkarte *BESITZER* der erweiterten Berechtigungen sind.

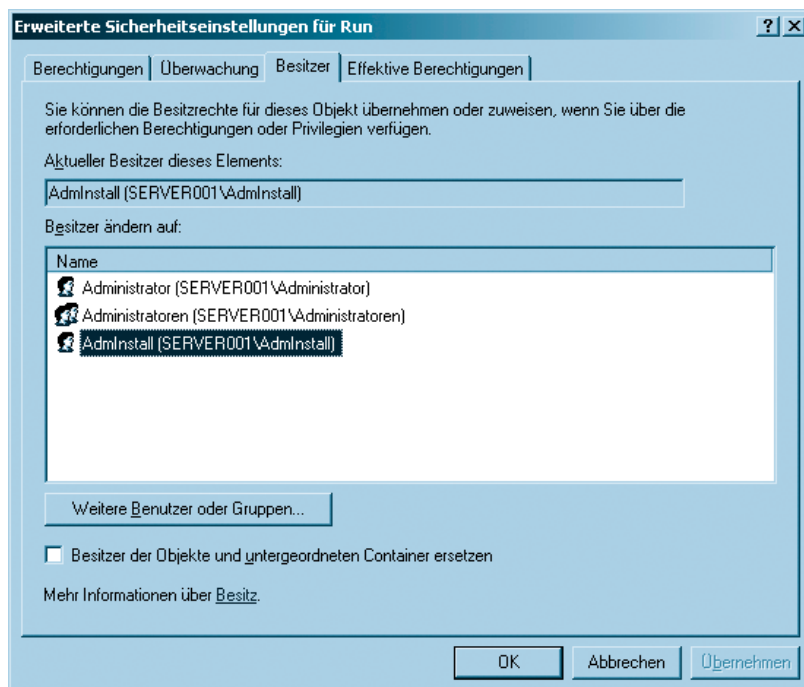


Abbildung 166: Die Besitzrechte auf einen anderen Benutzer übertragen

Der neue Besitzer bzw. die neue Besitzergruppe erscheint nun in der Liste. Mit einem Klick auf **ÜBERNEHMEN** übergeben Sie die Besitzrechte vollständig an den ausgewählten Benutzer bzw. die Gruppe, was Sie im Feld **AKTUELLER BESITZER DIESES ELEMENTS** sehr leicht überprüfen können.

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus den Besitzer eines Schlüssels festlegen möchten, so verwenden Sie hierfür das Tool `Subinacl.exe`, das Bestandteil des *Windows Server 2003 Resource Kits* ist, aber auch als neuere Version von der Internetseite <http://www.microsoft.com> heruntergeladen werden kann. Der Aufruf von `Subinacl.exe` erwartet dabei folgende Angaben, die Sie mit dem Befehlsaufruf als Parameter übergeben müssen:

- ▶ **Objekt-Typ:** Verwenden Sie `/KeyReg`, wenn Sie die Besitzrechte nur für den angegebenen Schlüssel ändern möchten, oder `/SubKeyReg`, wenn die Besitzrechte auch für die untergeordneten Objekte (Schlüssel, Werte, Daten) geändert werden sollen.
- ▶ **Aktion:** Verwenden Sie `/SetOwner`, um den Besitzer des angegebenen Schlüssels zu ändern.
- ▶ **Benutzer:** Der Benutzer- oder Gruppenname, dem die Besitzrechte übertragen werden sollen

Um z.B. die Besitzrechte des Schlüssels `HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN` auf den Benutzer *AdmInstall* zu übertragen, verwenden Sie folgende Befehlszeile:

```
@Echo off
SubInAcl /KeyReg HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\
Windows\CurrentVersion\Run /SetOwner=AdmInstall
```

Listing 427: 2_05_015.bat

Die Lösung mit VBScript

Das `SubInAcl`-Tool (siehe vorheriger Abschnitt) können Sie mit Hilfe der `Run`-Methode des `Shell`-Objektes natürlich auch aus einem VBScript heraus aufrufen. Wenn Sie den Besitzer des Schlüssels `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN` auf *AdmInstall* ändern möchten, so erstellen Sie mit `CreateObject` einen Verweis auf das `Shell`-Objekt und wenden die `Run`-Methode unter Angabe des zu verwendenden Befehlszeileninterpreters (`%Comspec%`) und der `Subinac1`-Befehlszeile an, der Sie mit `/KeyReg` den Schlüssel und mit `/SetOwner` den neuen Besitzer als Parameter übergeben.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c SubInAcl /KeyReg HKLM\SOFTWARE\Micro_
soft\Windows\CurrentVersion\Run /SetOwner=AdmInstall"
Set oShell = Nothing
```

Listing 428: 2_05_015.vbs

118 Registrierungsrelevante Ereignisse überwachen

Die erfolgreichen und auch fehlgeschlagenen Lese- und Schreibzugriffe, die auf die Objekte der Registrierung erfolgen, können Sie natürlich mit Hilfe der Ereignisanzeige überwachen und in das Auditing mit einbeziehen. Allerdings ist die Überwachung der Registrierung per Default nicht aktiviert, weshalb Sie die Überwachung erst noch konfigurieren müssen. Dies nehmen Sie direkt in der Registrierung vor, indem Sie in die erweiterten Einstellungen der Berechtigungen für das betreffende Objekt wechseln und dort die entsprechenden Überwachung festlegen.

Um beispielsweise sicherheitsrelevante Ereignisse für den Schlüssel `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN` mit Hilfe der Ereignisanzeige auswerten zu können, starten Sie den Registrierungs-Editor, markieren den genannten Schlüssel und wählen im Kontextmenü (oder im Menü `BEARBEITEN`) den Eintrag

BERECHTIGUNGEN... aus. Sie gelangen daraufhin in die Berechtigungsübersicht, klicken Sie dort auf ERWEITERT, und wechseln Sie anschließend auf die Registerkarte ÜBERWACHUNG.

Klicken Sie auf HINZUFÜGEN, anschließend auf ERWEITERT und dann auf JETZT SUCHEN, um in die detaillierte Übersicht der Überwachungsmöglichkeiten zu gelangen. Dort können Sie nun explizit festlegen, welche erfolgreichen und/oder fehlgeschlagenen Aktionen Sie für den betreffenden Schlüssel aufzeichnen möchten.

Achtung

Auf die Registrierung erfolgen ständige Lese- und Schreibzugriffe durch das Betriebssystem, durch Software-Komponenten und auch durch die Benutzer. Es macht daher wenig Sinn, wenn Sie die Überwachung aller erfolgreichen und fehlgeschlagenen Aktionen für alle Hauptzweige aktivieren, da das Sicherheitsprotokoll der Ereignisanzeige wahrscheinlich binnen weniger Sekunden voll wäre. Setzen Sie die Überwachungsrichtlinie daher mit Bedacht ein.

Zugriff:	Erfolgreich	Fehlgeschlagen
Vollzugriff	☐	☐
Wert abfragen	☐	☐
Wert festlegen	☒	☒
Unterschlüssel erstellen	☒	☒
Unterschlüssel auflisten	☐	☐
Benachrichtigen	☐	☐
Verknüpfung erstellen	☒	☒
Löschen	☒	☒
DAC schreiben	☐	☐
Besitzer festlegen	☒	☒
Lesekontrolle	☐	☐

Abbildung 167: Die Überwachungsrichtlinien für den Run-Schlüssel festlegen

Nachdem Sie die Überwachung konfiguriert haben, können Sie die erfolgreichen und/oder fehlgeschlagenen Ereignisse hinsichtlich der Zugriffe auf die Registrierungsobjekte mit Hilfe der Ereignisanzeige überprüfen. Starten Sie die Ereignisanzeige aus dem STARTMENÜ (Menü VERWALTUNG), und markieren Sie in der linken Fensterhälfte SICHERHEIT, um das Sicherheitsprotokoll zu öffnen. Je nach Protokollierung werden Sie nun eine ganze Reihe von Einträgen vorfinden. Wenn Sie die Einträge einsehen möchten, die auf Registrierungszugriffe hinweisen, dann sortieren Sie einfach die Spalte EREIGNIS und öffnen die Einträge mit der Ereignis-ID 560 und 562. Dies sind zumindest die relevantesten Einträge. Durch einen Doppelklick können Sie das jeweilige Ereignis öffnen. In der Ereignismeldung selbst ist das Objekt benannt, auf das erfolgreich/fehlerhaft zugegriffen wurde.

Achtung

Die Protokollierung der Zugriffe auf die Registrierung setzt voraus, dass in den Gruppenrichtlinien die ÜBERWACHUNG FÜR OBJEKTZUGRIFFSVERSUCHE aktiviert ist. Falls Sie also im Registrierungs-Editor die Überwachung konfiguriert haben und keine Einträge im Sicherheitsprotokoll finden können, dann müssen Sie erst die entsprechende Richtlinie aktivieren. Rufen Sie hierzu GPEDIT.MSC auf, und wechseln Sie in die Richtlinie COMPUTERKONFIGURATION\WINDOWS-EINSTELLUNGEN\SICHERHEITSEINSTELLUNGEN\LOKALE RICHTLINIEN\ ÜBERWACHUNGSRICHTLINIE. Dort aktivieren Sie die Überwachungsrichtlinie für erfolgreiche und/oder fehlerhafte Objektzugriffsversuche.

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung lassen sich die registrierungsrelevanten Ereignisse mit dem WMIC-Befehl unter Angabe des Alias NTEvent abfragen, da Sie hiermit eine Verbindung zur WMI-Klasse Win32_NTLogevent herstellen und die Eigenschaften der einzelnen Ereignisse auslesen können.

Die nachfolgende WMIC-Befehlszeile selektiert beispielsweise das Sicherheitsprotokoll (LogFile = 'Security') und alle darin enthaltenen Ereignisse mit der Ereignis-ID 560 und 562 (EventCode = '560' or EventCode = '562') und gibt die Werte der Eigenschaften TimeGenerated (Datum und Uhrzeit), SourceName (Quelle), Category (Kategorie), EventCode (Ereignis-ID), User (Benutzername) und ComputerName am Bildschirm aus:

```
@Echo off
WMIC NTEvent Where "LogFile = 'Security' AND EventCode = '560' OR _
    EventCode = '562'" Get TimeGenerated, SourceName, Category, _
    EventCode, User, ComputerName
```

Listing 429: 2_05_016.bat

Die Lösung mit VBScript

Um die registrierungsrelevanten Ereignisse per VBScript abzufragen, gehen Sie ähnlich vor wie in der Eingabeaufforderung. Will heißen: Sie stellen eine Verbindung zur WMI-Klasse Win32_NTLogEvent her, selektieren die Ereignisse mit der ID 560 und 562 und fragen die gewünschten Informationen ab.

Das nachfolgende Listing gibt beispielsweise alle Ereignisse mit der Ereignis-ID 560 und 562 des Sicherheitsprotokolls unter Angabe des Datums und der Uhrzeit der Ereigniserstellung (TimeGenerated), der Quelle (SourceName), der Kategorie (Category), der Ereignis-ID (EventCode), des Benutzernamens (User), des Computernamens (ComputerName) und der Beschreibung (Message) am Bildschirm aus. Dazu wird im ersten Schritt eine ExecQuery-Abfrage gegen Win32_NTLogEvent durchgeführt, womit alle Ereignisse mit der Ereignis-ID 560 und 562 selektiert werden. Anschließend werden die Werte der Eigenschaften ausgelesen und per WScript.Echo-Anweisung am Bildschirm ausgegeben.

Option Explicit

```
Dim oWMI
Dim strComputer
Dim colEvent
Dim objItem
```

```
strComputer = "."
```

```
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
```

```
Set colEvent = oWMI.ExecQuery ("Select * from Win32_NTLogEvent Where _
    Logfile = 'Security' AND EventCode = '560' OR EventCode = '562'")
```

```
For Each objItem in colEvent
```

```
    WScript.Echo "Date/Time: " & objItem.TimeGenerated
```

```
    WScript.Echo "Source: " & objItem.SourceName
```

```
    WScript.Echo "Category: " & objItem.Category
```

```
    WScript.Echo "Event-ID: " & objItem.EventCode
```

```
    WScript.Echo "User: " & objItem.User
```

```
    WScript.Echo "Computername: " & objItem.ComputerName
```

```
    WScript.Echo "Message:" & objItem.Message
```

```
    WScript.Echo "-----" & VBCRLF
```

```
Next
```

```
Set oWMI = Nothing
```

Listing 430: 2_05_016.vbs

119 Registrierungseinstellungen exportieren

In manchen Situationen kann es sehr hilfreich sein, einige Registrierungseinstellungen zu exportieren. Beispielsweise dann, wenn Sie Änderungen an der Systemkonfiguration vorgenommen haben und diese Änderung direkt in die Registrierung eines anderen Computersystems übertragen wollen. Oder wenn Sie Änderungen an der Registrierung vornehmen wollen, die ursprünglichen Registrierungseinstellungen aber im Notfall schnell wieder restaurieren möchten.

Um bestimmte Registrierungseinstellungen unter grafischer Oberfläche zu exportieren, nehmen Sie wieder den Registrierungs-Editor zu Hilfe. Markieren Sie einfach den Hauptzweig, oder wechseln Sie in den Schlüssel, dessen Inhalt (Schlüssel, Werte und Daten) Sie exportieren möchten, und wählen Sie im Kontextmenü EXPORTIEREN aus. Daraufhin wird ein Dialogfenster geöffnet, worüber Sie den Pfad und auch den Dateinamen für die Exportdatei festlegen können. Die Registrierungseinstellungen werden dann in die Exportdatei kopiert und können bei Bedarf mit Hilfe der Import-Funktion wieder in die Registrierung eingefügt werden.

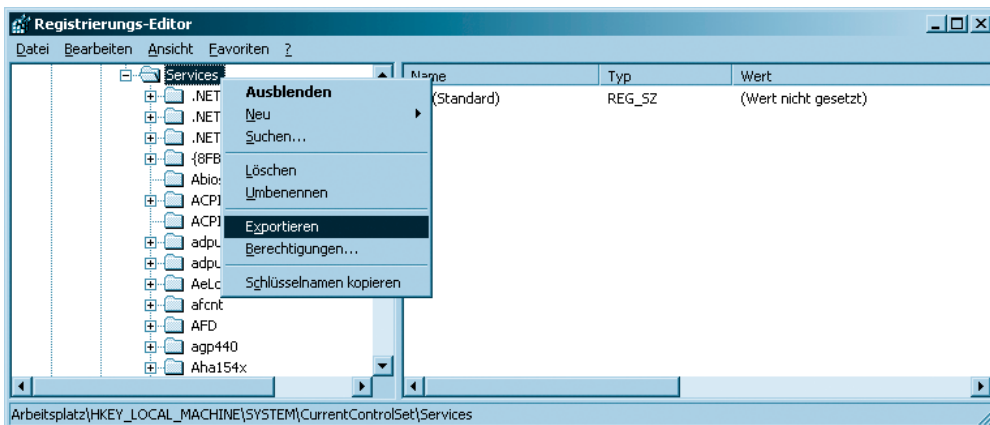


Abbildung 168: Registrierungseinstellungen exportieren

Die Lösung per Eingabeaufforderung

Auch in der Eingabeaufforderung haben Sie die Möglichkeit, komplette Hauptzweige oder auch Teile davon zu exportieren. Hierzu nutzen Sie den Befehl `Reg Export`, dem Sie lediglich den zu exportierenden Hauptzweig oder Schlüssel und die entsprechende Exportdatei als Parameter übergeben müssen.

Um den kompletten Inhalt des Schlüssels `HKLM\SYSTEM\CURRENTCONTROLSET\SERVICES` in die Datei `C:\Temp\Services.Reg` zu exportieren, nutzen Sie folgende Befehlszeilen, wobei die `@Echo off`-Anweisung lediglich der Ausschaltung der Befehlsanzeigefunktion dient:

```
@Echo off
Reg Export HKLM\SYSTEM\CurrentControlSet\Services _
C:\Temp\Services.Reg
```

Listing 431: 2_05_017.bat

Die Lösung mit VBScript

Um bestimmte Registrierungseinstellungen mit VBScript zu exportieren, könnten Sie natürlich mit den verschiedenen Methoden der WMI-Klasse `StdRegProv` eine rekursive Abfrage gegen die entsprechenden Schlüssel, Werte und Daten durchführen und das Ergebnis in eine Datei schreiben. Da ein Exportieren von Registrierungseinstellungen aber meist auch ein späteres Importieren der Registrierungseinstellungen bedeutet, ist es weitaus komfortabler (und auch sicherer), den Export mit Hilfe des `Reg Export`-Befehls durchzuführen.

Wenn Sie aus einem VBScript heraus einen Befehl der Eingabeaufforderung starten möchten, dann nutzen Sie hierfür die `Run`-Methode des `Shell`-Objektes, wobei Sie den zu verwendenden Befehlszeileninterpreter (`cmd.exe`) wie auch den eigentlichen Befehl (`Reg Export`) als Parameter übergeben.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Reg Export _
HKLM\SYSTEM\CurrentControlSet\Services _
C:\Temp\Services.Reg"

Set oShell = Nothing
```

Listing 432: 2_05_017.vbs

120 Registrierungseinstellungen importieren

Registrierungseinstellungen lassen sich nicht nur exportieren, sondern auch wieder importieren. Wenn Sie die zuvor exportierten Registrierungseinstellungen wieder in die Registrierung importieren möchten, reicht ein simpler Doppelklick auf die `Reg`-Datei für gewöhnlich aus. Bedenken Sie aber, dass die Einstellungen genau in den Registrierungspfad geschrieben werden, von wo aus sie ursprünglich auch exportiert wurden. Die aktuellen Registrierungseinstellungen werden damit mit denen aus der Importdatei ohne jegliche Rückfrage überschrieben!

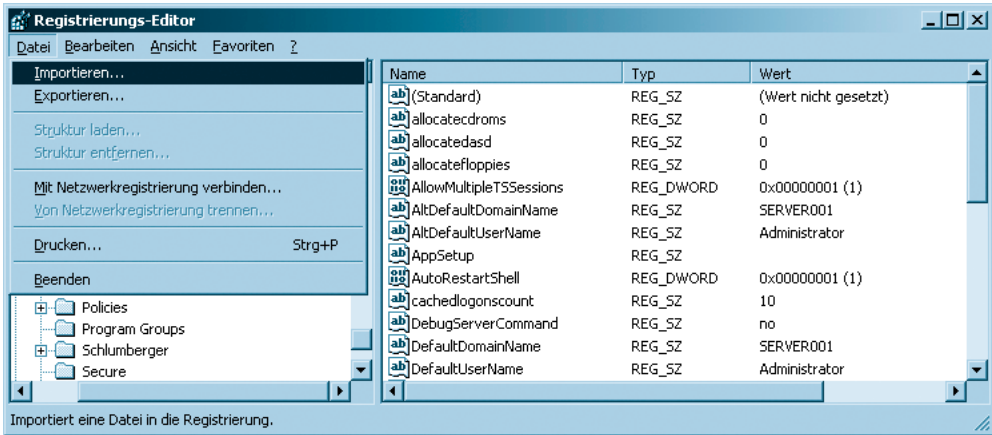


Abbildung 169: Registrierungseinstellungen wieder importieren

Sie können die Registrierungseinstellungen aber auch mit Hilfe der Import-Funktion des Registrierungs-Editors wieder in die Registrierung überführen. Hierzu starten Sie einfach den Registrierungs-Editor und wählen im Menü DATEI den Eintrag IMPORTIEREN... aus. Anschließend wählen Sie die Reg-Datei aus, die Sie importieren möchten, und klicken auf ÖFFNEN. Die aktiven Registrierungseinstellungen werden nun mit denen aus der Reg-Datei überschrieben.

Die Lösung per Eingabeaufforderung

Den Import der zuvor exportierten Registrierungseinstellungen nehmen Sie mit dem Befehl `Reg Import` vor, der ebenfalls zum Befehlsumfang von Windows Server 2003 gehört.

`Reg Import` erwartet lediglich die Angabe der Datei mit den zu importierenden Registrierungseinstellungen als Parameter, woraufhin die aktiven Registrierungseinstellungen ohne jegliche Rückfrage mit denen aus der Importdatei überschrieben werden. Um die Daten, die Sie im vorherigen Rezept exportiert haben, wieder zu importieren, nutzen Sie folgende Befehlszeilen:

```
@Echo off
Reg Import C:\Temp\Services.reg
```

Listing 433: 2_05_018.bat

Die Lösung mit VBScript

Um die zuvor exportierten Registrierungseinstellungen mit VBScript wieder zu importieren, nutzen Sie ebenfalls die `Run`-Methode des `Shell`-Objektes, wobei Sie diesmal aber den Befehl `Reg Import` ausführen.

Stellen Sie mit der `CreateObject`-Methode eine Verbindung zum `Shell`-Objekt her, und führen Sie die `Run`-Methode aus. Der `Run`-Methode übergeben Sie dabei den zu verwendenden Befehlszeileninterpreter durch Angabe der Systemvariablen `%Comspec%`, gefolgt von dem Parameter `/c` und der eigentlichen Befehlszeile `Reg Import C:\Temp\Services.reg`:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Exec "%comspec% /k Reg Import C:\Temp\Services.Reg"

Set oShell = Nothing
```

Listing 434: 2_05_018.vbs

121 Einen Dump der Registrierung erzeugen

Wenn Sie aus Datensicherungsgründen, der Dokumentation wegen oder zu Troubleshooting-Zwecken ein Abbild der Registrierung benötigen, so können Sie mit Hilfe der Export- und auch der Druck-Funktion des Registrierungs-Editors einen Dump der Registrierung (oder auch eines Teils davon) erzeugen.

Möchten Sie beispielsweise einen vollständigen Report des Hauptzweiges `HKEY_LOCAL_MACHINE` erstellen, so starten Sie den Registrierungs-Editor und vergeben zunächst eine Leseberechtigung (siehe Kategorie *Berechtigungen*) auf den untergeordneten Schlüssel mit der Bezeichnung `SECURITY`, da dieser Schlüssel standardmäßig für Benutzer (auch für die Administratoren) ausgeblendet ist. Damit erhalten Sie gleichzeitig auch Zugriff auf den `SAM`-Schlüssel. Anschließend markieren Sie den Hauptzweig, wählen im Menü `DATEI` den Eintrag `DRUCKEN` und den gewünschten Drucker aus und klicken auf `DRUCKEN`.

Wenn Sie den Dump dagegen als eine Art Datensicherung verwenden möchten, so dass Sie den Dump (oder Teile davon) später wieder in die Registrierung einfließen lassen können, so drucken Sie den Hauptzweig nicht aus, sondern exportieren dessen Einstellungen in eine *Reg*-Datei (siehe auch Rezept Registrierungseinstellungen exportieren auf Seite 537).

Achtung

Wenn Sie einen kompletten Dump der Registrierung erstellen möchten, so sollten Sie die Hauptzweige nacheinander ausdrucken oder exportieren. Denn wenn Sie die oberste Ebene des Konsolenstammes (`ARBEITSPLATZ`) markieren und die Druck- oder Export-Funktion aufrufen, werden lediglich die Hauptzweige `HKEY_LOCAL_MACHINE` und `HKEY_USERS` ausgedruckt oder exportiert. Damit werden zwar auch alle Registrierungseinstellungen berücksichtigt, allerdings nicht mehr in der gewohnten Struktur, wie sie im Registrierungs-Editor ersichtlich ist.

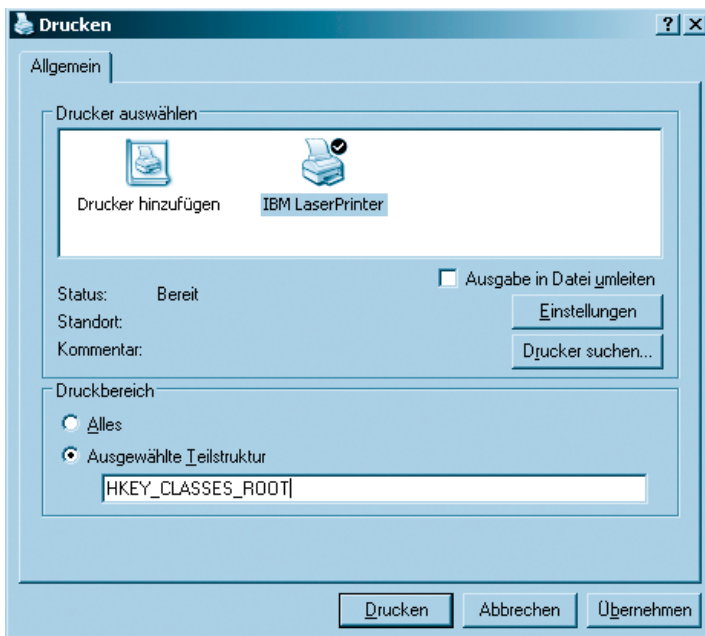


Abbildung 170: Einen Report des HKCR-Hauptzweiges erstellen

Die Lösung per Eingabeaufforderung

Möchten Sie den Dump der Registrierung dagegen von der Eingabeaufforderung aus erstellen, so nutzen Sie hierfür idealerweise den Befehl `Reg Export`, womit Sie den Inhalt eines kompletten Hauptzweiges in eine externe Datei schreiben können. Für einen vollständigen Dump der Registrierung führen Sie diesen Befehl gegen alle Hauptzweige aus und schreiben die Ergebnisse in die entsprechenden *Reg*-Dateien:

```
@Echo off
Reg Export HKCR C:\Temp\HKCR.reg
Reg Export HKCU C:\Temp\HKCU.reg
Reg Export HKLM C:\Temp\HKLM.reg
Reg Export HKU C:\Temp\HKU.reg
Reg Export HKCC C:\Temp\HKCC.reg
```

Listing 435: 2_05_019.bat

Die Lösung mit VBScript

Um per VBScript einen Dump der Registrierung zu erstellen, können Sie zwar die Methoden der WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`) für rekursive Abfragen einsetzen, weitaus komfortabler (und auch sicherer) ist jedoch der Einsatz des Befehls `Reg Export`, da Sie die Daten bei Bedarf wieder mit `Reg Import` in die Registrierung einfließen lassen können.

Wenn Sie einen kompletten Dump der fünf Hauptzweige erzeugen möchten, so stellen Sie mit der `GetObject`-Methode zunächst eine Verbindung zum `Shell`-Objekt her. Anschließend führen Sie mit der `Run`-Methode den Befehl `Reg Export` aus, wobei Sie den jeweiligen Hauptzweig wie auch die entsprechende Exportdatei als Parameter übergeben:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Reg Export HKCR C:\Temp\HKCR.Reg"
oShell.Run "%comspec% /c Reg Export HKCU C:\Temp\HKCU.Reg"
oShell.Run "%comspec% /c Reg Export HKLM C:\Temp\HKLM.Reg"
oShell.Run "%comspec% /c Reg Export HKU C:\Temp\HKU.Reg"
oShell.Run "%comspec% /c Reg Export HKCC C:\Temp\HKCC.Reg"

Set oShell = Nothing
```

Listing 436: 2_05_019.vbs

122 Favoriten hinzufügen

In einigen Fällen kann es sehr hilfreich sein, bestimmte Schlüssel in der Favoritenliste des Registrierungs-Editors aufzunehmen. Wenn Sie beispielsweise hin und wieder auf tief verzweigte Schlüssel zugreifen möchten, sich die Pfade aber nicht merken wollen, dann können Sie mit Hilfe der Favoriten-Funktion eine Art *Lesezeichen* setzen. Fügen Sie einfach den entsprechenden Schlüssel der Favoritenliste hinzu, und wählen Sie beim nächsten Zugriff auf den Schlüssel nur noch den Favoriten aus. Der Registrierungs-Editor öffnet den Schlüssel dann von selbst, das lästige Durchklicken bis zum gewünschten Schlüssel entfällt somit. Dies funktioniert allerdings nur mit Schlüsseln, einzelne Werte können Sie in der Favoritenliste nicht aufnehmen.

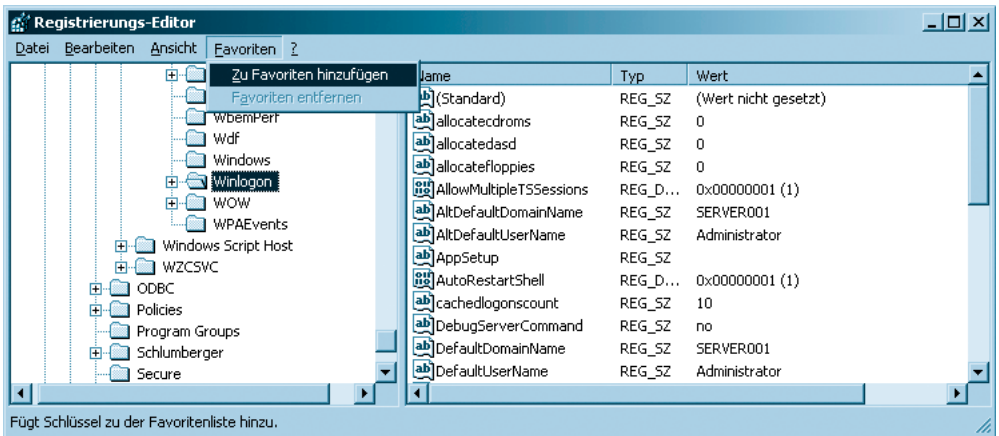


Abbildung 171: Neue Favoriten hinzufügen

Um den Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON in der Favoritenliste abzulegen, öffnen Sie einfach den Registrierungs-Editor, wechseln in den angegebenen Schlüssel, markieren den Schlüssel WINLOGON und wählen im Menü FAVORITEN den Eintrag ZU FAVORITEN HINZUFÜGEN aus. Daraufhin wird ein kleines Dialogfenster geöffnet, worüber Sie den Favoritennamen (z.B. WINLOGON) festlegen können.

Die Lösung per Eingabeaufforderung

Bei der Favoriten-Funktion handelt es sich um ein spezielles Feature des Registrierungs-Editors, weshalb diese Funktion normalerweise nur unter der grafischen Oberfläche zum Einsatz kommt. Da aber alle Favoriten, die Sie im Registrierungs-Editor anlegen, im Schlüssel HKCU\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\APPLETS\REGEDIT\FAVORITES als Zeichenfolgen hinterlegt sind, können Sie mit dem Reg Add-Befehl auch in der Eingabeaufforderung neue Favoriten erstellen und diese beispielsweise per Batch-Datei auf andere Computersysteme übertragen.

Der Reg Add-Befehl erwartet dabei folgende Angaben, die Sie mit dem Befehlsaufruf als Parameter übergeben müssen:

- ▶ Schlüssel: Schlüssel, der um den neuen Wert ergänzt werden soll
- ▶ Bezeichnung (/v): Bezeichnung des neuen Wertes
- ▶ Typ (/t): Datentyp des neuen Wertes (standardmäßig wird REG_SZ verwendet)
- ▶ Daten (/d): Eigentlicher Inhalt der Zeichenfolge

Die folgenden Befehlszeilen zeigen ein Beispiel für das Anlegen des neuen Favoriten WINLOGON, der auf den Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\WINLOGON verweist:

```
@Echo off
Reg Add HKCU\SOFTWARE\Microsoft\Windows\Current_
    Version\Applets\Regedit\Favorites /v Winlogon _
    /t REG_SZ /d "HKLM\SOFTWARE\Microsoft\Windows _
    NT\CurrentVersion\Winlogon"
```

Listing 437: 2_05_020.bat

Die Lösung mit VBScript

Auch mit VBScript lassen sich neue Favoriten erstellen, da Sie lediglich den Schlüssel HKCU\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\APPLETS\REGEDIT\FAVORITES um eine neue Zeichenfolge erweitern müssen. Und zum Erstellen neuer Werte, sprich: neuer Zeichenfolgen, verwenden Sie die SetValue-Methode, die durch die WMI-Klasse StdRegProv (WMI-Namespace root\default) bereitgestellt wird.

Die SetValue-Methode erwartet dabei die folgenden Angaben, die Sie beim Aufruf der Methode als Parameter übergeben müssen:

- ▶ HKCU: Konstante für den hexadezimalen Wert des Hauptzweiges HKEY_CURRENT_USER
- ▶ strKey: Schlüssel, der um die neue Zeichenfolge erweitert werden soll
- ▶ strValueName: Bezeichnung für die neue Zeichenfolge
- ▶ strValue: Die eigentliche Zeichenfolge selbst

Um nun einen neuen Favoriten zu kreieren, stellen Sie zunächst mit der `GetObject`-Methode eine Verbindung zur WMI-Klasse `StdRegProv` her und setzen den `Impersonation-Level` auf `Impersonate`. Anschließend führen Sie die `SetStringValue`-Methode aus und übergeben die erforderlichen Parameter. Die `SetStringValue`-Methode schreibt daraufhin die neue Zeichenfolge in den genannten Schlüssel. Sofern die Zeichenfolge bereits existiert, wird die ursprüngliche Zeichenfolge ohne jegliche Rückfrage mit den neuen Daten überschrieben.

Option Explicit

```
Dim oReg
Dim strComputer
Dim strKey
Dim strValueName
Dim strValue
```

```
Const HKCU = &H80000001
strComputer = "."
strKey = "Software\Microsoft\Windows\CurrentVersion\
    Applets\Regedit\Favorites"
strValueName = "Winlogon"
strValue = "HKLM\SOFTWARE\Microsoft\CurrentVersion\
    Windows NT\Winlogon"
```

```
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")
On Error Resume Next
oReg.SetStringValue HKCU, strKey, strValueName, strValue
```

```
Set oReg = Nothing
```

Listing 438: 2_05_020.vbs

123 Favoriten entfernen

Favoriten lassen sich nicht nur erstellen, sondern auch jederzeit wieder löschen. Hierzu rufen Sie unter grafischer Oberfläche einfach den Registrierungs-Editor auf und wählen im Menü FAVORITEN den Eintrag FAVORITEN ENTFERNEN aus. Daraufhin wird ein kleines Dialogfenster mit den bereits erstellten Favoriten angezeigt. Wählen

Sie den Favoriten aus, den Sie löschen möchten, und klicken Sie auf OK. Die Zeichenfolge des Favoriten wird daraufhin aus dem Schlüssel HKCU\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\APPLETS\REGEDIT\FAVORITES gelöscht und taucht damit auch nicht mehr in der Liste der Favoriten auf.

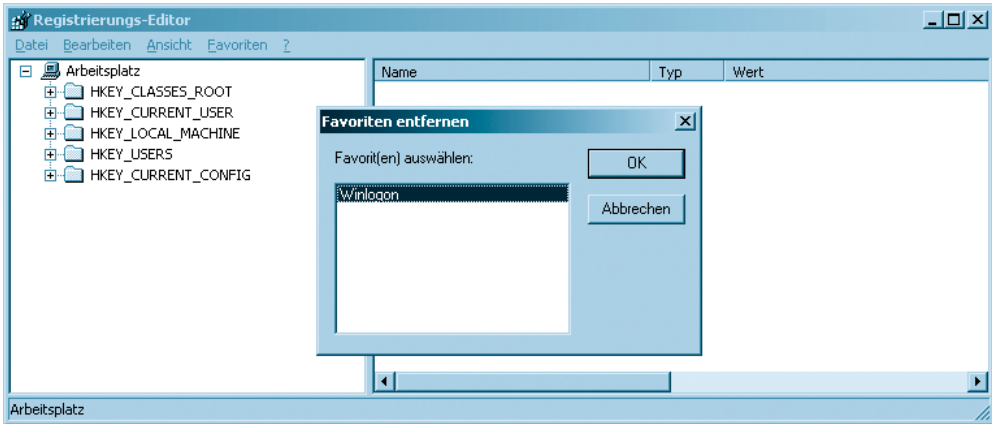


Abbildung 172: Favoriten wieder entfernen

Die Lösung per Eingabeaufforderung

Um vorhandene Favoriten per Eingabeaufforderung zu entfernen, nutzen Sie den Befehl `Reg Delete`, womit Sie die zuständige Zeichenfolge des Favoriten aus dem Schlüssel `HKCU\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\APPLETS\REGEDIT\FAVORITES` löschen können.

`Reg Delete` erwartet dabei lediglich die Angabe des Schlüssels, der die zu löschende Zeichenfolge enthält, wie auch die Zeichenfolge selbst, die Sie mit dem Parameter `/v` übergeben. Der `Reg Delete`-Befehl versucht daraufhin, eine Verbindung zu dem angegebenen Schlüssel herzustellen und die Zeichenfolge zu löschen. Ohne Angabe weiterer Parameter fragt `Reg Delete` noch einmal nach, ob die angegebene Zeichenfolge auch wirklich gelöscht werden soll. Dies können Sie jedoch mit dem Parameter `/f` unterbinden, da dieser Parameter das Löschen der übergebenen Zeichenfolge erzwingt.

```
@Echo off
Reg Delete HKCU\SOFTWARE\Microsoft\Windows\Current_
    Version\Applets\Reg edit\Favorites /v Winlogon /f
```

Listing 439: 2_05_021.bat

Die Lösung mit VBScript

Möchten Sie vorhandene Favoriten des Registrierungs-Editors wieder mit VBScript löschen, so löschen Sie die betreffende Zeichenfolge mit der `DeleteValue`-Methode, die seitens der WMI-Klasse `StdRegProv` (WMI-Namespace `root\default`) bereitgestellt wird. Die `DeleteValue`-Methode erwartet dabei folgende Angaben, die Sie mit dem Methodenaufruf als Parameter übergeben müssen:

- ▶ `HKCU`: Konstante für den hexadezimalen Wert des Hauptzweiges `HKEY_CURRENT_USER`
- ▶ `strKey`: Schlüssel, der die zu löschende Zeichenfolge enthält
- ▶ `strValueName`: Die Zeichenfolge, die gelöscht werden soll

Um den Favoriten mit der Bezeichnung `Winlogon` zu löschen, stellen Sie zunächst mit der `GetObject`-Methode eine Verbindung zur WMI-Klasse `StdRegProv` her. Anschließend rufen Sie die `DeleteValue`-Methode auf, wobei Sie die erforderlichen Parameter übergeben. Die `DeleteValue`-Methode löscht den angegebenen Wert daraufhin ohne jegliche Rückfrage aus der Registrierung.

Option Explicit

```
Dim oReg
Dim strComputer
Dim strKey
Dim strValueName
```

```
Const HKCU = &H80000001
strComputer = "."
strKey = "Software\Microsoft\Windows\CurrentVersion\Applets\
    Regedit\Favorites"
strValueName = "Winlogon"
```

```
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")
```

```
On Error Resume Next
oReg.DeleteValue HKCU,strKey,strValueName
```

```
Set oReg = Nothing
```

Listing 440: 2_05_021.vbs

124 Registrierungsdateien auflisten

Die Registrierung setzt sich bekanntlich aus dem Inhalt mehrerer Dateien zusammen. Wenn Sie einmal auf die Schnelle herausfinden müssen, welche Dateien dies im Einzelnen sind, dann brauchen Sie sich nur den Inhalt des Schlüssels HKLM\SYSTEM\CURRENTCONTROLSET\CONTROL\HIVELIST anzusehen. Denn dort sind alle Hauptzweige unter Angabe der zuständigen Dateien aufgelistet. Wundern Sie sich aber bitte nicht, dass Sie nur die Hauptzweige HKLM und HKU wiederfinden. Denn die Inhalte der Hauptzweige HKCR, HKCU und HKCC spiegeln nur bestimmte Unterschlüssel der beiden Hauptzweige HKLM und HKU wider.

Unter grafischer Oberfläche nehmen Sie wieder den Registrierungs-Editor zu Hilfe. Wechseln Sie einfach in den genannten Schlüssel, und sehen Sie sich die Hauptzweige und Dateien in der rechten Fensterhälfte an.

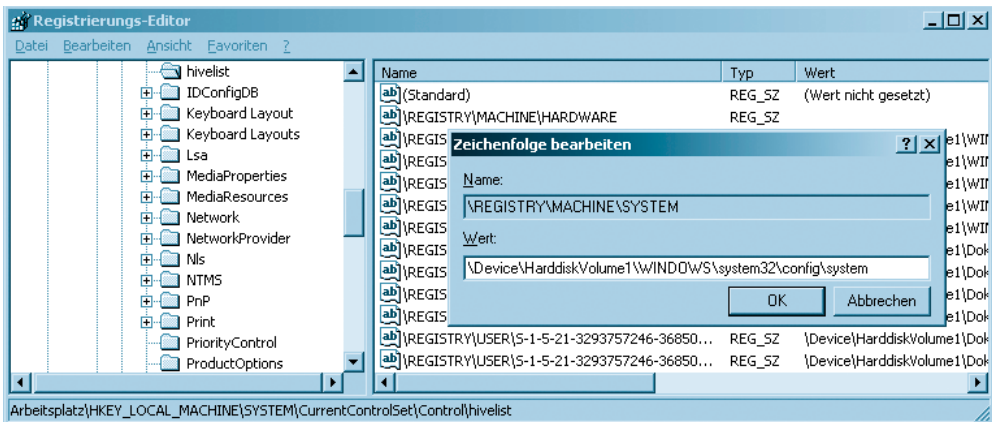


Abbildung 173: Die Registrierungsdateien anzeigen lassen

Die Lösung per Eingabeaufforderung

Auch in der Eingabeaufforderung können Sie sich die Registrierungsdateien anzeigen lassen, indem Sie mit dem Befehl `Reg Query` eine Abfrage gegen den Schlüssel HKLM\SYSTEM\CURRENTCONTROLSET\CONTROL\HIVELIST durchführen und dessen Inhalt (Werte, Wert-Typen, Daten) einfach auslesen.

`Reg Query` erwartet lediglich die Angabe des Schlüssels, dessen Inhalt Sie auslesen möchten, stellt daraufhin eine Verbindung zu dem angegebenen Schlüssel her und gibt dessen Inhalt am Bildschirm aus. Mit Hilfe der Ausgabeumleitung können Sie das Ergebnis auch in eine Datei schreiben lassen:

```
@Echo off
Reg Query HKLM\SYSTEM\CurrentControlSet\Control\
Hivelist > Datei.txt
```

Listing 441: 2_05_022.bat

Die Lösung mit VBScript

Wenn Sie die Registrierungsdateien dagegen mit VBScript abfragen möchten, so lesen Sie mit der `EnumValues`-Methode erst die Bezeichnungen der Zeichenfolgen des Schlüssels `HKLM\SYSTEM\CURRENTCONTROLSET\CONTROL\HIVELIST` aus und fragen anschließend mit der `GetStringValues`-Methode die eigentlichen Daten ab.

Beide Methoden werden durch die WMI-Klasse `StdRegProv` bereitgestellt. Das heißt, Sie müssen mit der `GetObject`-Methode erst eine Verbindung zur WMI-Klasse `StdRegProv` herstellen und den `Impersonation-Level` auf `Impersonate` setzen. Anschließend können Sie die `EnumValues`-Methode aufrufen, wobei Sie folgende Parameter übergeben müssen:

- ▶ `HKLM`: Konstante für den hexadezimalen Wert des Hauptzweiges `HKEY_LOCAL_MACHINE`
- ▶ `strKey`: Schlüssel, dessen Werte ausgelesen werden sollen
- ▶ `strArray`: Array, in das die ausgelesenen Informationen übertragen werden
- ▶ `intType`: Rückgabe des dezimalen Wertes des Datentyps, der ausgelesen wurde (siehe Tabelle 54: Die dezimalen Werte der Datentypen auf Seite 508)

Daraufhin werden die Zeichenfolgen, die sich in dem angegebenen Schlüssel befinden, in das Array übertragen. Mit Hilfe einer `For Each`-Schleife können Sie nun auf die einzelnen Elemente zugreifen und für jedes Element eine `GetStringValue`-Abfrage starten, für die Sie die folgenden Parameter übergeben müssen:

- ▶ `HKLM`: Konstante für den hexadezimalen Wert des Hauptzweiges `HKEY_LOCAL_MACHINE`
- ▶ `strKey`: Schlüssel, dessen Zeichenfolgen ausgelesen werden sollen
- ▶ `strValueName`: Name der Zeichenfolge, dessen Daten ausgelesen werden sollen
- ▶ `strValue`: Variable, worüber die ausgelesenen Daten zurückgegeben werden

Das nachfolgende Skript zeigt ein solches Beispiel, das die ausgelesenen Informationen in eine Datei schreibt: Zu Beginn werden die Variablen und Konstanten definiert, damit das Skript halbwegs überschaubar bleibt und ohne großen Aufwand angepasst werden kann. Dann folgt die Referenzierung des `FileSystemObject`-Objektes und der WMI-Klasse `StdRegProv`. Anschließend wird die `EnumValues`-Methode gegen den Schlüssel `HKLM\SYSTEM\CURRENTCONTROLSET\CONTROL\HIVELIST` ausgeführt, woraufhin die Bezeichnungen der Zeichenfolgen in das Array übertragen werden. Mit der `For Each`-Schleife wird dann für jedes Element die Bezeichnung ausgelesen und in die Datei geschrieben, anschließend folgt eine `GetStringValue`-Methode gegen die Zeichenfolge, wobei auch hier wieder das Ergebnis in die Datei `C:\Temp\Datei.txt` geschrieben wird.

Option Explicit

```
Dim oReg
Dim oFS
Dim oFile
Dim strComputer
Dim strKey
Dim strArray
Dim strValue
Dim strValueName
Dim strFile
```

```
Const HKLM = &H80000002
Const REG_SZ = 1
Const Write = 2
strComputer = "."
strKey = "SYSTEM\CurrentControlSet\Control\hivelist"
strFile = "C:\Temp\Datei.txt"
```

```
Set oFS=CreateObject("Scripting.FileSystemObject")
Set oFile=oFS.OpenTextFile(strFile, Write, True)
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")
oReg.EnumValues HKLM, strKey, strArray, REG_SZ
```

```
For Each strValueName in strArray
    oFile.WriteLine strValueName
    oReg.GetStringValue HKLM, strKey, strValueName, strValue
    oFile.WriteLine strValue
Next
```

```
Set oReg = Nothing
Set oFS = Nothing
Set oFile = Nothing
```

Listing 442: 2_05_022.vbs

125 Registrierung sichern und wiederherstellen

Die Registrierung enthält sämtliche Konfigurationseinstellungen zum Betriebssystem und auch zur verwenden Hard- und Software. Fehlende und auch fehlerhafte Einträge können daher zu Instabilitäten und gar zu einem totalen Systemabsturz führen, weshalb Sie die Registrierung regelmäßig sichern sollten. Insbesondere dann, wenn Sie manuelle Änderungen an der Registrierung vornehmen wollen.

Nun gibt es mehrere Möglichkeiten, wie Sie die Registrierung sichern und im Notfall auch wieder restaurieren können. Für welche Methode Sie sich entscheiden, hängt in erster Linie davon ab, ob Sie die gesamte Registrierung sichern möchten oder nur Teile davon. Auch ist entscheidend, ob das Server-System für eine vollständige Datensicherung kurzzeitig heruntergefahren werden kann oder ob die Sicherung im laufenden Windows-Betrieb stattfinden muss (was bei Server-Systemen eher die Regel ist).

Die nachfolgende Aufstellung zeigt die verschiedenen Möglichkeiten zum Sichern und Wiederherstellen der Registrierung auf und nennt jeweils die Vor- und Nachteile der einzelnen Möglichkeiten.

- ▶ **Exportieren der Hauptzweige:** Mit dem Export der Hauptzweige (siehe Registrierungseinstellungen exportieren auf Seite 537) oder wahlweise auch der untergeordneten Schlüssel können Sie die Registrierungseinstellungen, wie Sie sie im Registrierungs-Editor vorfinden, in eine gesonderte Textdatei exportieren und bei Bedarf auch wieder in die Registrierung einfließen lassen. Diese Methode eignet sich vorwiegend zur schnellen Sicherung bestimmter Registrierungseinstellungen und wird vorzugsweise dann eingesetzt, wenn bestimmte Registrierungseinstellungen von Hand geändert und die Originaleinträge vorher gesichert werden sollen. Um die exportierten Daten wieder zu importieren, muss Windows Server 2003 zumindest im abgesicherten Modus startfähig sein.
- ▶ **Kopieren der Registrierungsdateien:** Das manuelle Kopieren der Registrierungsdateien (siehe Registrierungsdateien auflisten auf Seite 547) stellt die sicherste und auch zuverlässigste Methode der Sicherung dar, da Sie die Dateien, in denen die Registrierungseinstellungen hinterlegt sind, auf ein anderes Medium kopieren und so im Notfall auch vollständig wieder restaurieren können. Selbst dann, wenn Windows Server 2003 auch im abgesicherten Modus nicht mehr starten kann. Diese Methode hat jedoch den Nachteil, dass Sie das Betriebssystem kurzzeitig herunterfahren und die Recovery Console aufrufen müssen, da Sie die Daten anderenfalls nicht kopieren können, weil während des laufenden Windows-Betriebs unterschiedliche Prozesse darauf zugreifen und die Dateien damit gesperrt sind.
- ▶ **Sichern des Systemstatus mit NTBackup:** Das Windows Server 2003-eigene Datensicherungstool namens NTBACKUP bietet Ihnen die Möglichkeit, den aktuellen Systemstatus des Betriebssystems zu sichern, womit Sie unter anderem auch eine vollständige Sicherung der Registrierung erstellen können. Im Gegensatz zur vorherigen Methode wird hierbei die Schattenkopie-Technik eingesetzt, womit die Registrierungsdateien auch während einer aktiven Windows-Sitzung gesichert werden können. Nachteil hierbei ist jedoch, dass neben der Registrierung auch weitere Betriebssystemdaten (Active Directory-Datenbank, Startdateien, Sysvol-Ordner, COM+-Klassenbibliothek) mitgesichert werden, die Sie hierbei nicht ausklammern können. Die Datensicherung dauert damit wesentlich länger und ist auch weitaus umfangreicher, als es zur Sicherung der Registrierung erforderlich wäre.

Um beispielsweise den Inhalt eines Hauptzweiges oder den eines untergeordneten Schlüssels auf die Schnelle zu sichern, können Sie die entsprechenden Registrierungseinstellungen exportieren. Öffnen Sie hierzu den Registrierungs-Editor, markieren den Hauptzweig oder den Schlüssel und wählen im Kontextmenü den Eintrag EXPORTIEREN aus.

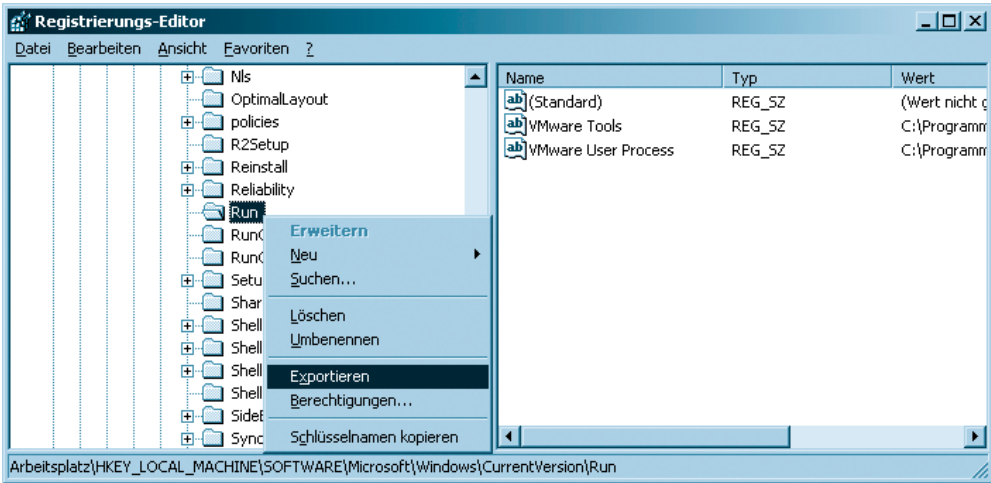


Abbildung 174: Den Inhalt eines Schlüssels exportieren

Sie gelangen daraufhin in ein Dialogfenster, worüber Sie den Pfad und auch den Dateinamen für die Exportdatei angeben können. Legen Sie den Pfad und auch den Namen der Datei fest, übernehmen Sie die Dateinamenerweiterung *Reg*, und klicken Sie auf OK, um den Inhalt des zuvor markieren Hauptzweiges oder Schlüssels zu exportieren.

Wenn Sie den Inhalt der Exportdatei zu einem späteren Zeitpunkt wieder in die Registrierung einfügen möchten, so brauchen Sie unter grafischer Oberfläche nur einen Doppelklick auf die *Reg*-Datei vorzunehmen. Die Einstellungen werden daraufhin automatisch in die Registrierung übertragen.

Eine weitere Möglichkeit, die Registrierung zu sichern, bietet das Windows Server 2003-eigene Datensicherungstool mit der Bezeichnung NTBACKUP, womit Sie auf recht einfache Art und Weise den Systemstatus und damit auch die Registrierung sichern können. Um beispielsweise eine vollständige Sicherung des aktuellen Systemstatus vorzunehmen und die Sicherung im Ordner *E:\Backup* abzulegen, öffnen Sie das Startmenü, klicken auf AUSFÜHREN und geben *NTBackup* ein, woraufhin das Datensicherungstool gestartet wird. Klicken Sie im Willkommensbildschirm auf den Link ERWEITERTEN MODUS, um das Dialogfenster SICHERUNGSPROGRAMM aufzurufen, holen Sie dann die Registerkarte SICHERN in den Vordergrund, und markieren Sie in der linken Fensterhälfte unterhalb von Arbeitsplatz den Eintrag SYSTEM STATE.

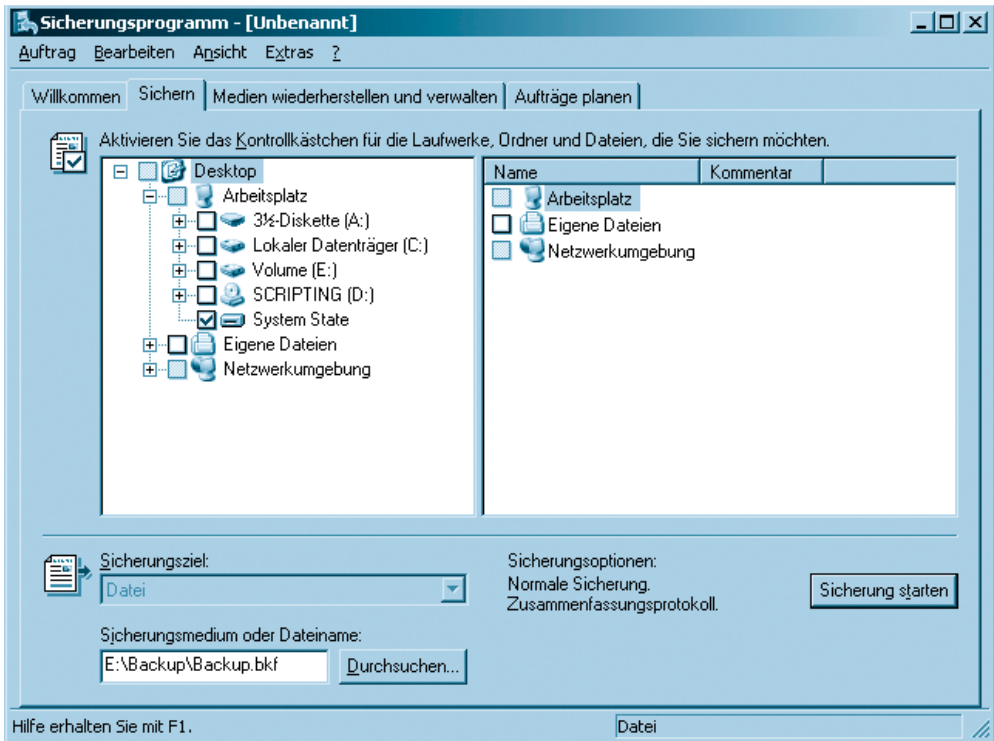


Abbildung 175: Den aktuellen Systemstatus sichern

Klicken Sie anschließend auf DURCHSUCHEN, um den Pfad für die Backup-Datei festzulegen, und klicken Sie anschließend auf SICHERUNG STARTEN, um weitere Einstellungen vornehmen zu können. Denn im Dialogfenster INFORMATIONEN ZUM SICHERUNGS-AUFTRAG können Sie eine kurze Beschreibung zur Datensicherung festlegen und auch bestimmen, ob die Sicherung einer bereits vorhandenen Sicherung angehängt werden oder diese ersetzen soll. Mit einem Klick auf ERWEITERT können Sie zudem die Sicherungsart (beim Sichern des Systemstatus können Sie nur NORMAL oder KOPIEREN verwenden) auswählen und auch die Überprüfung der Sicherung aktivieren.

Nehmen Sie Ihre Einstellungen vor, und klicken Sie anschließend auf STARTEN, um den aktuellen Systemstatus in den festgelegten Pfad zu sichern. Dieser Vorgang kann (je nach Laufwerksgeschwindigkeit) bis zu einer Viertelstunde dauern und nimmt (z.B. nach einer Standardinstallation von Windows Server 2003) einen ungefähren Plattenplatz von guten 400 MByte ein.

Für den Fall, dass es nun zu möglichen Instabilitäten aufgrund fehlerhafter Registrierungseinträge kommt, so können Sie den soeben gesicherten Systemstatus mit wenig Aufwand wieder restaurieren und den ursprünglichen Zustand, der vor der Datensicherung existierte, wiederherstellen. Hierzu rufen Sie NTBACKUP auf, wechseln auf die Registerkarte MEDIEN WIEDERHERSTELLEN UND VERWALTEN und markieren die Sicherung, die Sie restaurieren möchten. Im Feld DATEIEN WIEDERHERSTELLEN IN wählen Sie aus,

ob alle Systemstatusdateien in den ursprünglichen Bereich restauriert werden sollen oder stattdessen in einen alternativen Pfad oder gar in einen einzelnen Ordner. Wählen Sie die Option **URSPRÜNGLICHER BEREICH** aus, womit die aktuellen Systemstatusdateien durch die aus der Sicherung ersetzt werden können, ohne dass Sie Windows Server 2003 hierfür herunterfahren müssen. Denn wenn Sie die Registrierungsdateien in einen alternativen Pfad oder in einen einzelnen Ordner restaurieren, so benötigen Sie die Recovery Console, um die Originaldateien überschreiben zu können, da sie während der aktiven Windows-Sitzung geöffnet und damit gesperrt sind.

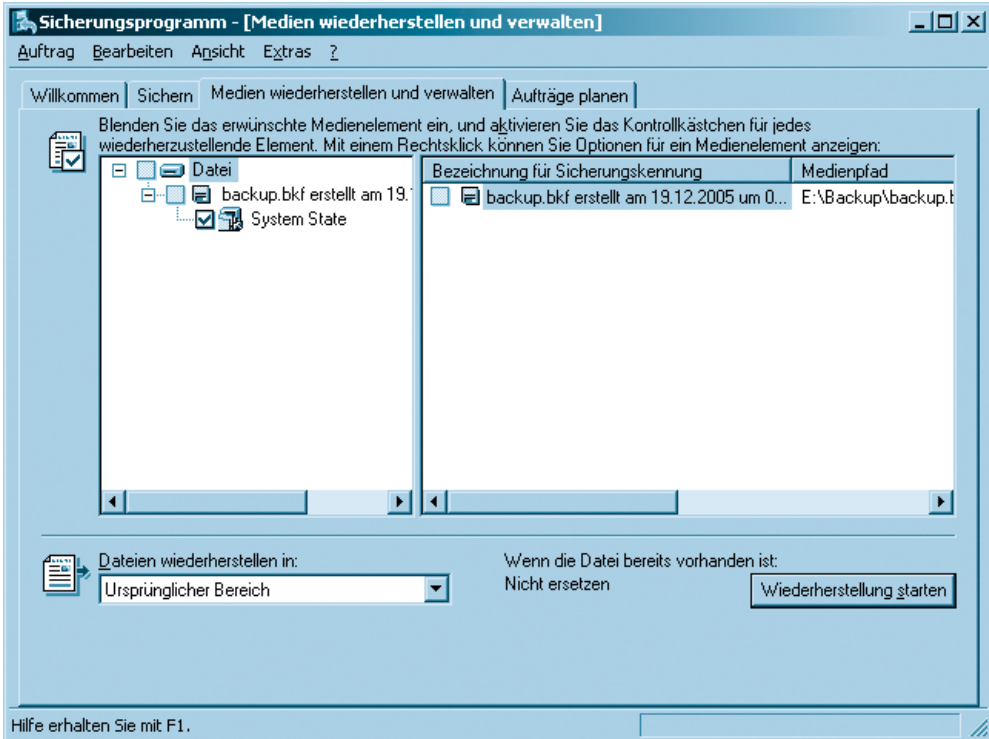


Abbildung 176: Den Systemstatus wieder restaurieren

Die Lösung per Eingabeaufforderung

Wenn Sie nur bestimmte Teile der Registrierung von der Eingabeaufforderung aus sichern möchten, so können Sie hierfür den `Reg Export`-Befehl verwenden, womit Sie zum einen komplette Hauptzweige, zum anderen aber auch einzelne Schlüssel in eine gesonderte Textdatei exportieren und später per Import wieder in die Registrierung einfließen lassen können.

Um beispielsweise den Inhalt des Schlüssels `HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN` (inklusive der untergeordneten Schlüssel, Werte und Daten) in die Datei `RunKey.reg` zu exportieren und damit zu sichern, geben Sie in der Eingabeaufforderung folgende `Reg Export`-Befehlszeile ein:

```
@Echo off
Reg Export HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run RunKey.reg
```

Listing 443: 2_05_023a.bat

Wenn Sie Änderungen an dem zuvor genannten Schlüssel vorgenommen haben und die Sicherung, sprich: den Inhalt der Datei *RunKey.reg*, wieder restaurieren müssen, so geben Sie in der Eingabeaufforderung den folgenden `Reg Import`-Befehl ein:

```
@Echo off
Reg Import RunKey.reg
```

Listing 444: 2_05_023b.bat

Um stattdessen eine vollständige Sicherung der Registrierung vorzunehmen, können Sie in der Eingabeaufforderung den `NTBackup`-Befehl einsetzen, womit Sie – wie auch unter grafischer Oberfläche – eine Datensicherung des aktuellen Systemstatus vornehmen können. Allerdings können Sie in der Eingabeaufforderung lediglich die Sicherung starten. Das Wiederherstellen der gesicherten Daten kann nur mit der `NTBackup`-Version unter grafischer Oberfläche vorgenommen werden (siehe vorheriger Abschnitt).

Wenn Sie beispielsweise von der Eingabeaufforderung aus den Systemstatus sichern möchten, hierbei den Auftragsnamen *SystemState 122005* verwenden, die Sicherungsart *Copy* nutzen (in Verwendung mit dem Parameter `SystemState` sind nur die Sicherungsarten *Normal* und *Copy* zulässig) und das Backup im Pfad *E:\Backup\Systemstate 122005.bkf* ablegen möchten und nach Abschluss der Datensicherung eine Überprüfung der Sicherung durchführen wollen, so können Sie hierfür die folgende `NTBackup`-Befehlszeile verwenden:

```
@Echo off
NTBackup Backup Systemstate /J "SystemState 122005" /F _
"E:\Backup\Systemstate 122005.bkf" /M Copy /V:Yes
```

Listing 445: 2_05_023c.bat

Die Lösung mit VBScript

Gleiches lässt sich natürlich auch per VBScript realisieren. Um einen vollständigen Hauptzweig oder einzelne Schlüssel per VBScript in eine gesonderte Textdatei zu exportieren, könnten Sie den Inhalt des Hauptzweiges oder des Schlüssels zwar mit den Methoden des `StdRegProviders` rekursiv auslesen und mit der `Write`-Methode des `File-System`-Objektes in eine Textdatei schreiben. Weitaus einfacher und schneller exportieren Sie den Inhalt eines Hauptzweiges oder eines Schlüssels jedoch mit dem `Reg Export`-

Befehl der Eingabeaufforderung, den Sie mit der `Run`-Methode des `Shell`-Objektes auch aus einem VBScript heraus aufrufen können.

Das nachfolgende Listing zeigt ein Beispiel dafür, wie Sie den `Reg Export`-Befehl per VBScript aufrufen und den Inhalt des Schlüssels `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` in die Datei *RunKey.reg* exportieren können.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Reg Export HKLM\SOFTWARE\Microsoft\Windows\_
    CurrentVersion\Run RunKey.reg"

Set oShell = Nothing
```

Listing 446: 2_05_023a.vbs

Um die exportierten Einstellungen wieder in die Registrierung einfließen zu lassen, verwenden Sie – wie auch in der Eingabeaufforderung – den `Reg Import`-Befehl und übergeben die Datei mit den Registrierungseinstellungen als Parameter.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Reg Import RunKey.reg"

Set oShell = Nothing
```

Listing 447: 2_05_023b.vbs

Und last but not least, können Sie mit Hilfe von VBScript natürlich auch eine Datensicherung mit `NTBackup` starten und damit den Systemstatus sichern, womit die Registrierung ebenfalls gesichert wird. Hierzu wenden Sie wieder den `NTBackup`-Befehl unter Angabe der erforderlichen Parameter an, den Sie ebenfalls mit der `Run`-Methode des `Shell`-Objektes aus einem VBScript heraus aufrufen können.

Da einige Parameter die Angabe der Werte in Anführungszeichen erwarten, hinterlegen Sie die `NTBackup`-Befehlszeile und die dazugehörigen Parameter in einer gesonderten Batch-Datei, die Sie anschließend mit der `Run`-Methode aufrufen. Das nachfolgende Listing führt beispielsweise den Inhalt der Batchdatei *Backup.bat* aus, die den gleichen Inhalt hat wie das Listing *2_05_023c.bat* auf Seite 554.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Backup.bat"

Set oShell = Nothing
```

Listing 448: 2_05_023c.vbs

126 Berechtigungen innerhalb der Registrierung bearbeiten

Die Hauptzweige und Schlüssel, die in der Registrierung hinterlegt sind, verfügen über eine so genannte Zugriffskontrollliste (siehe *Dateien und Ordner*), worüber Sie die Zugriffe auf die einzelnen Hauptzweige und Schlüssel (und damit auch auf die untergeordneten Werte und Daten) explizit festlegen können. Wenn Sie über administrative Berechtigungen verfügen oder zumindest das Privileg zum Ändern der Zugriffskontrolllisten haben, so können Sie neue Benutzer und Gruppen zur Zugriffskontrollliste hinzufügen und eine Leseberechtigung oder gar einen Vollzugriff erteilen, spezielle Berechtigungen vergeben und die Zugriffe natürlich auch explizit verweigern. Auch können Sie die Standardberechtigungen, die per Default nach der Installation des Betriebssystems gelten, verändern.

Achtung

Beim Ändern der Berechtigungen ist allerdings größte Vorsicht geboten. Denn wenn Sie Berechtigungen entfernen (insbesondere die des Systems und auch die des Administrators), so gefährden Sie damit die Stabilität des Betriebssystems und riskieren möglicherweise einen kompletten Systemabsturz. Denn das Betriebssystem benötigt jederzeit Zugriff auf die Einträge, die in der Registrierung hinterlegt sind. Wenn diese Zugriffe fehlen, können Teile des Betriebssystems nicht mehr ausgeführt werden. Und wenn Sie die Berechtigungen für den Administrator einschränken, so laufen Sie Gefahr, die Zugriffskontrollliste später nicht mehr ändern zu können, da hierfür die entsprechenden Berechtigungen vorhanden sein müssen. Gehen Sie also mit größter Sorgfalt vor, wenn Sie die Berechtigungen innerhalb der Registrierung bearbeiten.

Um beispielsweise den Benutzer *AdmInstall* in der Zugriffskontrollliste des Hauptzweiges HKEY_LOCAL_MACHINE aufzunehmen und die Berechtigung zum Erstellen neuer Schlüssel und Werte zu erteilen, starten Sie den Registrierungs-Editor, markieren dann den Hauptzweig, dessen Zugriffskontrollliste Sie bearbeiten möchten, und fügen den Benutzer mit einem Klick auf HINZUFÜGEN der Zugriffskontrollliste hinzu.

Anschließend klicken Sie auf ERWEITERT, markieren den soeben hinzugefügten Benutzer und führen dann einen Klick auf die Schaltfläche BEARBEITEN aus, um in die Einstellungen der speziellen Berechtigungen zu gelangen. Dort aktivieren Sie in der

Spalte ZULASSEN die Optionen WERT FESTLEGEN und UNTERSCHLÜSSEL ERSTELLEN. Wenn Sie anschließend auf OK klicken und die geänderten Einstellungen damit übernehmen, so gelten die soeben gesetzten Berechtigungen für den Inhalt des zuvor markieren Hauptzweiges sowie für alle untergeordneten Schlüssel. Wenn Sie die Berechtigung ausschließlich für den Hauptzweig, jedoch nicht für die untergeordneten Schlüssel setzen und damit die Vererbung an untergeordnete Elemente unterbinden möchten, so wählen Sie im Feld ÜBERNEHMEN FÜR den Eintrag NUR DIESER SCHLÜSSEL aus und klicken anschließend auf OK.

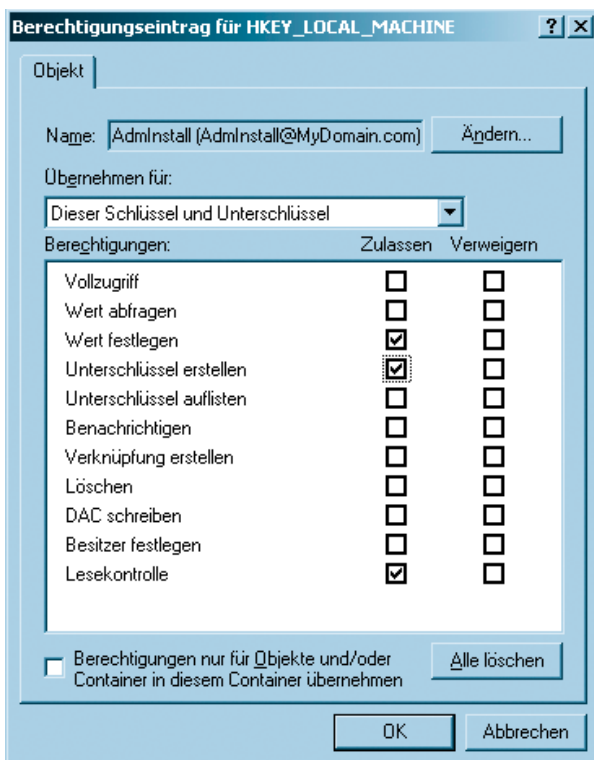


Abbildung 177: Spezielle Berechtigungen für einen Benutzer vergeben

Auf die gleiche Art und Weise, wie Sie einem Benutzer Berechtigungen erteilen können, so können Sie die Berechtigungen natürlich auch explizit verweigern, indem Sie in der Spalte VERWEIGERN das entsprechende Häkchen setzen. Hierbei müssen Sie berücksichtigen, dass eine Verweigerung eine höhere Priorität hat als ein Zulassen. Wenn Sie einem Benutzer beispielsweise volle Zugriffsrechte gewähren, zeitgleich aber die Berechtigung zum Löschen verweigern, so darf der Benutzer trotz der vollen Zugriffsrechte keine Einträge entfernen.

Und wenn Sie einen Benutzer in der Zugriffskontrollliste eines Hauptzweiges oder eines untergeordneten Schlüssels aufgenommen haben, die Änderung im Nachhinein aber wieder rückgängig machen möchten, so brauchen Sie nur die Zugriffskontroll-

liste zu öffnen, den Benutzer zu markieren und den Eintrag mit einem Klick auf ENT-FERNEN aus der Zugriffskontrollliste zu verbannen. Der Benutzer hat daraufhin, sofern er kein Mitglied der in der Zugriffskontrollliste enthaltenen Gruppen ist, keinerlei Berechtigungen für den Hauptzweig oder den untergeordneten Schlüssel.

Die Lösung per Eingabeaufforderung

Windows Server 2003 beinhaltet leider keinen entsprechenden Befehl, womit Sie die Berechtigungen von der Eingabeaufforderung aus modifizieren können. Hier bleibt nur der Einsatz eines zusätzlichen Tools wie beispielsweise *Subinac1.exe*, das Bestandteil des *Windows Server 2003 Resource Kits* ist und das Sie kostenlos von der Internetseite *www.microsoft.com* herunterladen können.

Der *Subinac1*-Befehl erwartet zunächst den Objekttyp als Parameter. Verwenden Sie */Keyreg*, wenn Sie die Berechtigungen für einen Hauptzweig bearbeiten möchten, oder */SubKeyreg*, um die Berechtigungen für einen untergeordneten Schlüssel zu modifizieren. Des Weiteren müssen Sie dem *Subinac1*-Befehl den Hauptzweig oder den Pfad des untergeordneten Schlüssels übergeben, dessen Berechtigungen Sie ändern wollen. Und zu guter Letzt übergeben Sie den Parameter */Grant=Benutzer=Berechtigung*, um eine neue Berechtigung zu erteilen, oder den Parameter */Deny=Benutzer=Berechtigung*, um eine Berechtigung explizit zu verweigern. Der Platzhalter *Berechtigung* muss dabei durch das jeweilige Berechtigungsflag ersetzt werden. Folgende Flags sind möglich:

Berechtigung	Flag
Vollzugriff	F
Lesen	R
Lesekontrolle	A
Wert abfragen	Q
Wert festlegen	S
Unterschlüssel erstellen	C
Unterschlüssel auflisten	E
Benachrichtigen	Y
Verknüpfung erstellen	L
Löschen	D
DAC schreiben	W
Besitzer festlegen	O

Tabelle 58: Die Berechtigungsflags des *Subinac1*-Befehls

Wenn Sie beispielsweise den Benutzer *AdminInstall* der Zugriffskontrollliste des Hauptzweiges *HKEY_LOCAL_MACHINE* hinzufügen und dem Benutzer volle Zugriffsrechte erteilen möchten, so wenden Sie den *Subinac1*-Befehl an und legen mit */Keyreg*

fest, dass Sie die Zugriffskontrollliste eines Hauptzweiges ändern möchten. Anschließend übergeben Sie den Namen des Hauptzweiges, dessen Zugriffskontrollliste Sie modifizieren möchten, und legen mit `/Grant=FRA-MD-MS-001\AdmInstall=F` fest, dass der Benutzer `FRA-MD-001\AdmInstall` hinzugefügt und die vollen Zugriffsrechte erteilt werden sollen.

```
@Echo off
Subinac1 /Keyreg HKEY_LOCAL_MACHINE /Grant=FRA-MD-MS-001\AdmInstall=F
```

Listing 449: 2_05_024a.bat

Wenn Sie nicht die Zugriffskontrollliste eines Hauptzweiges, sondern die eines untergeordneten Schlüssels bearbeiten wollen, so verwenden Sie statt des Objekttyps `/Keyreg` den Objekttyp `/SubKeyreg`.

```
@Echo off
Subinac1 /Subkeyreg HKEY_LOCAL_MACHINE\SOFTWARE _
/Grant=FRA-MD-MS-001\AdmInstall=F
```

Listing 450: 2_05_024b.bat

Um einen Benutzer wieder aus der Zugriffskontrollliste herauszunehmen, können Sie ebenfalls den `Subinac1`-Befehl verwenden. Wenn Sie beispielsweise den Benutzer `AdmInstall` aus der Zugriffskontrollliste des Hauptzweiges `HKEY_LOCAL_MACHINE` herauslöschen möchten, so wenden Sie den `Subinac1`-Befehl unter Angabe des Objekttyps `/Keyreg` und des entsprechenden Hauptzweiges an und übergeben den Parameter `/Revoke=Benutzer`, wobei Sie `Benutzer` durch den Namen des Benutzers (Format `Computer-/Domänenname\Benutzername`) ersetzen.

```
@Echo off
Subinac1 /Keyreg HKEY_LOCAL_MACHINE /Revoke=FRA-MD-MS-001\AdmInstall
```

Listing 451: 2_05_024c.bat

Gleiches lässt sich natürlich auch für einen untergeordneten Schlüssel durchführen, wobei Sie dem `Subinac1`-Befehl jedoch nicht den Objekttyp `/Keyreg`, sondern den Objekttyp `/SubKeyreg` übergeben müssen.

```
@Echo off
Subinac1 /Subkeyreg HKEY_LOCAL_MACHINE\SOFTWARE _
/Revoke=FRA-MD-MS-001\AdmInstall
```

Listing 452: 2_05_024d.bat

Die Lösung mit VBScript

Um die Berechtigungen per VBScript zu modifizieren, bleibt nur der Einsatz der `Run`-Methode des `Shell`-Objektes, womit Sie den `Subinacl`-Befehl (siehe vorheriger Abschnitt), der Bestandteil des *Windows Server 2003 Resource Kits* ist, auch aus einem VBScript heraus aufrufen können.

Wenn Sie z.B. den Benutzer *AdmInstall* in der Zugriffskontrollliste des Hauptzweiges *HKEY_LOCAL_MACHINE* aufnehmen und volle Zugriffsrechte vergeben möchten, so führen Sie mit der `Run`-Methode den `Subinacl`-Befehl mit dem Parameter `/Keyreg HKEY_LOCAL_MACHINE /Grant=AdmInstall=F` aus.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Reg Subinacl /Keyreg HKEY_LOCAL_MACHINE _
    /Grant=FRA-MD-MS-001\AdmInstall=F"

Set oShell = Nothing
```

Listing 453: 2_05_024a.vbs

Ist der Benutzer dagegen schon in der Zugriffskontrollliste enthalten und möchten Sie ihm die vollen Zugriffsberechtigungen explizit verweigern, so wenden Sie stattdessen den `Subinacl`-Befehl mit dem Parameter `/Deny an`.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Reg Subinacl /Keyreg HKEY_LOCAL_MACHINE _
    /Deny=FRA-MD-MS-001\AdmInstall=F"

Set oShell = Nothing
```

Listing 454: 2_05_024b.vbs

Und um den Benutzer wieder aus der Zugriffskontrollliste herauszunehmen, setzen Sie den Subinac1-Befehl mit dem Parameter /Revoke ein.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Reg Subinac1 /Keyreg HKEY_LOCAL_MACHINE _
/Deny=FRA-MD-MS-001\AdmInstall=F"

Set oShell = Nothing
```

Listing 455: 2_05_024c.vbs

Systemmanagement

Die zuverlässige Verwaltung und auch die regelmäßige Überwachung des Windows Servers 2003 zählt zu den wichtigsten Grundvoraussetzung für einen stabilen und sicheren Serverbetrieb. Der Windows Server 2003 ist daher mit einer ganzen Reihe administrativer Tools ausgestattet, womit Sie beispielsweise auf recht einfache Art und Weise Computerkonten anlegen, Logfiles einsehen, Prozesse überwachen und Dienste konfigurieren können.

Die Erfahrung zeigt allerdings, dass viele dieser administrativen Aufgaben häufig noch auf manuellem Wege durchgeführt werden, was sich in vielen Fällen als sehr zeitaufwändig erweisen kann. Die nachfolgenden Rezepte sollen daher nicht nur einen Lösungsansatz unter grafischer Oberfläche bieten, sondern insbesondere auch zeigen, wie sich die wichtigsten Verwaltungs- und Überwachungsaufgaben von der Eingabeaufforderung aus oder per VBScript automatisieren lassen.

127 Computerkonto erstellen

Zum Anlegen eines neuen Computerkontos verwenden Sie unter grafischer Oberfläche das Verwaltungswerkzeug ACTIVE DIRECTORY-BENUTZER UND -COMPUTER, das Sie im STARTMENÜ, genauer gesagt unter VERWALTUNG, per Klick öffnen oder mit `Dsa.msc` auch direkt aufrufen können.

Um ein neues Computerkonto anzulegen, markieren Sie zunächst den entsprechenden Container (z.B. COMPUTERS) oder die jeweilige Organisationseinheit und wählen im Kontextmenü NEU und dann COMPUTER aus. Sie gelangen daraufhin in das Dialogfenster NEUES OBJEKT – COMPUTER, worüber Sie den Computernamen und wahlweise auch den Benutzer oder die Gruppe, die das betreffende Computersystem in die Domäne integrieren dürfen, auswählen.

Im nächsten Schritt haben Sie die Möglichkeit, die GUID (Globally Unique Identifier) des neuen Computersystems festzulegen, sofern es sich bei dem neuen Computersystem um ein zu verwaltendes Computersystem handelt. Anschließend klicken Sie auf FERTIG STELLEN, um das Computerkonto in dem zuvor markierten Container oder innerhalb einer Organisationseinheit zu erstellen.

Nach dem Erstellen des neuen Computerkontos können Sie weitere Eigenschaften (z.B. *Beschreibung*, *Gruppenmitgliedschaft*, *Standortangabe*, *Verwaltungsinformationen* ...) für das Computerkonto konfigurieren, indem Sie aus dem Kontextmenü des Computerkontos heraus die EIGENSCHAFTEN öffnen und auf den zur Verfügung stehenden Registerkarten weitere Optionen setzen.

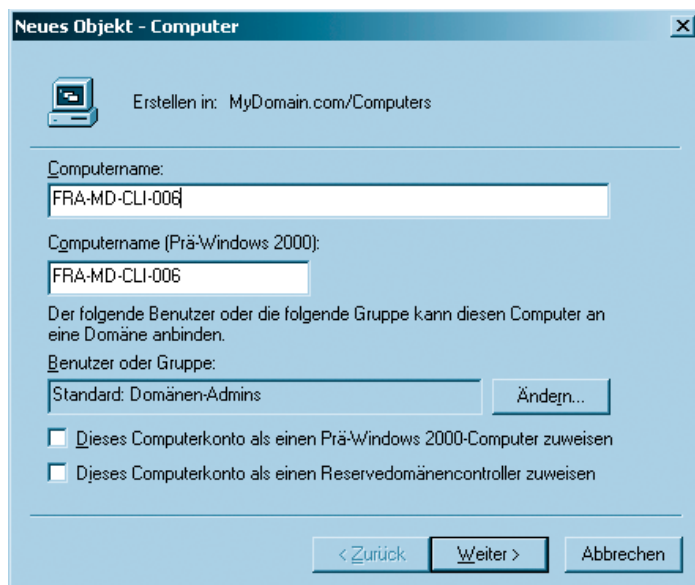


Abbildung 178: Ein Computerkonto erstellen

Die Lösung per Eingabeaufforderung

Mit dem `Dsadd`-Befehl unter Verwendung der Option `Computer` lassen sich neue Computerkonten auch per Eingabeaufforderung erstellen. Der `Dsadd`-Befehl erwartet dabei zumindest die Angabe des definierten Namens für das neue Computerkonto, den Sie mit dem Befehlsaufruf als Parameter übergeben müssen.

Um beispielsweise im Container *Computers* der Domäne *MyDomain.com* ein neues Konto für den Computer *FRA-MD-CLI-006* anzulegen, können Sie folgende Befehlszeile auf dem Domänen-Controller anwenden:

```
@Echo off
Dsadd Computer CN=FRA-MD-CLI-006,CN=Computers,DC=MyDomain,DC=Com
```

Listing 456: 2_06_001a.bat

Der `Dsadd`-Befehl erlaubt dabei die Angabe weiterer Eigenschaften, die Sie mit dem Befehlsaufruf übergeben können. Wenn Sie nicht nur das eigentliche Computerkonto anlegen möchten, sondern zeitgleich auch den SAM-Namen (`-samid`), die Beschreibung (`-desc`) und auch den geografischen Standort des Computersystems (`-loc`) bestimmen wollen, so verwenden Sie stattdessen die folgende Befehlszeile:


```
@Echo off
Dsadd Computer CN=FRA-MD-CLI-006, _
    CN=Computers,DC=MyDomain,DC=Com -samid FRA-MD-CLI-006 _
    -desc "Testclient 6 von MyDomain.com" -loc Frankfurt
```

Listing 457: 2_06_001b.bat

Die Lösung mit VBScript

Auch mit VBScript ist das Erstellen neuer Computerkonten kein Problem, denn Sie benötigen im einfachsten Fall nur eine LDAP-Verbindung zur Domäne (oder wahlweise zu einem Container oder einer Organisationseinheit), um dann mit der `Create`-Methode ein neues Computerobjekt anlegen zu können.

Wenn Sie z.B. im Container *Computers* der Domäne *MyDomain.com* ein neues Konto für *FRA-MD-CLI-006* erstellen möchten, so binden Sie den Active Directory-Pfad des Containers *Computers* (CN=Computers,DC=MyDomain,DC=Com) per `GetObject`-Methode an eine Objektvariable (`oDomain`), wenden die `Create`-Methode gegen die Objektvariable an und übergeben den Objekttyp (`Computer`) wie auch den Namen des Objektes (CN=FRA-MD-CLI-006) als Argument.

Um mit dem Anlegen des Computerkontos zusätzliche Eigenschaften zu konfigurieren, weisen Sie der betreffenden Eigenschaft mit der `Put`-Methode den gewünschten Wert zu. Im nachfolgenden Code-Beispiel wird z.B. der `SamAccountName` definiert und die Kennwort-Option `ADS_UF_PASSWD_NOTREQD` gesetzt. Letzteres ist erforderlich, damit das Computerkonto nicht automatisch deaktiviert wird. Anschließend schreiben Sie die Information mit der `SetInfo`-Methode in die Active Directory-Datenbank, womit das Computerkonto angelegt und auch gespeichert wird.

Option Explicit

```
Dim oDomain
Dim oComputer
Dim strComputerName
Dim strDomain

Const ADS_UF_PASSWD_NOTREQD = &h0020
strComputerName = "FRA-MD-CLI-006"
strDomain = "CN=Computers,DC=MyDomain,DC=Com"

Set oDomain = GetObject("LDAP://" & strDomain)
Set oComputer = oDomain.Create("Computer", _
    "CN=" & strComputerName)
oComputer.Put "sAMAccountName", strComputerName & "$"
oComputer.Put "userAccountControl", ADS_UF_PASSWD_NOTREQD
```

Listing 458: 2_06_001.vbs

```
oComputer.SetInfo
```

```
Set oDomain = Nothing
```

Listing 458: 2_06_001.vbs (Forts.)

128 Computerkonto löschen

Mit der Voraussetzung, dass Sie als *Administrator* angemeldet oder zumindest Mitglied der Gruppe *Konten-Operatoren* oder *Organisations-Admins* sind, können Sie mit Hilfe des Verwaltungswerkzeuges *ACTIVE DIRECTORY-BENUTZER UND -COMPUTER* auf recht einfache Art und Weise ein Computerkonto löschen.

Hierzu starten Sie *ACTIVE DIRECTORY-BENUTZER UND -COMPUTER* aus dem Verwaltungsmenü heraus oder rufen das zuvor genannte Verwaltungswerkzeug mit *Dsa.msc* direkt auf. Anschließend markieren Sie das betreffende Computerkonto und wählen im Kontextmenü den Eintrag *Löschen* aus oder drücken die Taste **[Entf]**. Sie werden daraufhin noch einmal gefragt, ob Sie das ausgewählte Computerkonto auch wirklich löschen möchten. Klicken Sie auf **JA**, um das Löschen zu bestätigen.

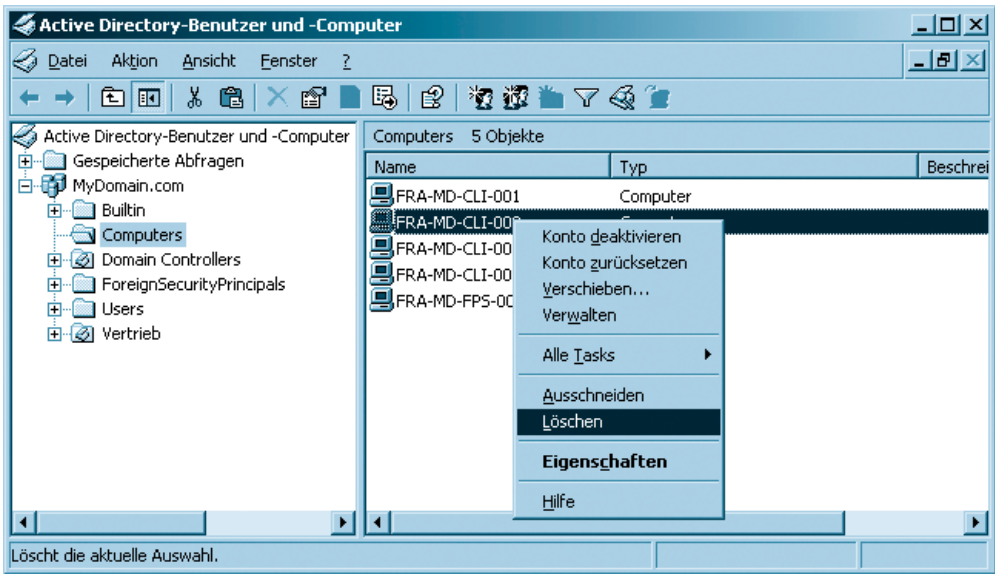


Abbildung 179: Ein Computerkonto löschen

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung lässt sich ein vorhandenes Computerkonto genauso schnell löschen wie unter grafischer Oberfläche. Hierfür stellt der Windows Server 2003 den `Dsrm`-Befehl bereit, dem Sie lediglich den definierten Namen des zu löschenden Objektes übergeben müssen.

Ohne Angabe weiterer Parameter versucht der `Dsrm`-Befehl daraufhin, das angegebene Objekt zu löschen, wobei Sie diesen Vorgang aber von Hand bestätigen müssen. Um diese Rückfrage zu unterdrücken und das Löschen eines Objektes zu forcieren, fügen Sie der Befehlszeile den Parameter `-noprompt` an.

```
@Echo off
Dsrm CN=FRA-MD-CLI-001,CN=Computers,DC=MyDomain,DC=Com _
-noprompt
```

Listing 459: 2_06_002.bat

Die Lösung mit VBScript

Um ein vorhandenes Computerkonto per VBScript zu löschen, verwenden Sie im einfachsten Fall die `Delete`-Methode des ADSI-Providers `LDAP`, der Sie den Objekttyp wie auch den Namen des zu löschenden Objektes als Argument übergeben.

Das nachfolgende Code-Beispiel zeigt eine Möglichkeit, wie Sie das Computerkonto *FRA-MD-CLI-006*, das im Container *Computers* der Domäne *MyDomain.com* enthalten ist, löschen können: Mit der `GetObject`-Methode wird zunächst der Active Directory-Pfad des Containers *Computers* an die Objektvariable `strDomain` gebunden. Anschließend wird die `Delete`-Methode gegen die Objektvariable angewandt, und der zu löschende Objekttyp (`Computer`) wie auch der Name des zu löschenden Objektes (`strComputerName`) werden übergeben.

```
Option Explicit

Dim oDomain
Dim strComputerName
Dim strDomain

strComputerName = "FRA-MD-CLI-006"
strDomain = "CN=Computers,DC=MyDomain,DC=Com"

Set oDomain = GetObject("LDAP://" & strDomain)
oDomain.Delete "Computer", strComputerName

Set oComputer = Nothing
```

Listing 460: 2_06_002.vbs

129 Computer einer Domäne hinzufügen

Einzelne Computersysteme (dies können Clients wie auch Standalone-Server sein), die Mitglied einer Arbeitsgruppe sind, lassen sich unter grafischer Oberfläche ohne großen Aufwand einer Domäne hinzufügen. Denn Sie müssen im Grunde genommen nur die Mitgliedschaft der Arbeitsgruppe lösen und die der Domäne herstellen. Um diese Aufgabe ausführen zu können, müssen allerdings einige Grundvoraussetzungen gegeben sein:

- ▶ **Netzwerkverbindung:** Zwischen dem Client (oder dem Standalone-Server) und dem Domänen-Controller muss eine TCP/IP-Verbindung vorliegen und der Name eines Domänen-Controllers muss per DNS auflösbar sein.
- ▶ **Computerkonto:** Auf dem Domänen-Controller muss ein entsprechendes Computerkonto für den Client (oder den Standalone-Server) eingerichtet werden. Das Computerkonto kann dabei im Vorfeld eingerichtet werden, es lässt sich aber auch im Zuge der Konfiguration der Domänen-Mitgliedschaft anlegen.
- ▶ **Berechtigung:** Auf dem Client oder Standalone-Server, der zu einer Domäne hinzugefügt werden soll, sind administrative Berechtigungen erforderlich, da die Netzwerkidentifikation geändert werden muss. Des Weiteren müssen Sie auf dem Domänen-Controller zumindest die Berechtigung haben, um neue Active Directory-Objekte anlegen zu können.

Um den Standalone-Server mit der Bezeichnung *FRA-MD-FPS-001* in der Domäne *MyDomain.com* aufzunehmen, melden Sie sich zunächst als lokaler Administrator auf dem Standalone-Server an und wechseln in die Eigenschaften der Netzwerkumgebung. Im Menü **ERWEITERT** wählen Sie den Eintrag **NETZWERKIDENTIFIKATION...**, Sie gelangen daraufhin in das Dialogfenster **SYSTEMEIGENSCHAFTEN**. Klicken Sie auf der Registerkarte **COMPUTERNAME** auf **ÄNDERN**, um das Dialogfenster **COMPUTERNAMEN ÄNDERN** aufzurufen. Dort aktivieren Sie die Option **DOMÄNE**, geben den Namen der Domäne (in diesem Fall *MyDomain*) ein und drücken die -Taste.

Sie müssen sich nun dem Domänen-Controller gegenüber durch Eingabe eines Benutzernamens und des dazugehörigen Kennwortes authentifizieren. Verwenden Sie in diesem Fall einen Account, der auf dem Domänen-Controller die ausreichenden Berechtigungen zum Anlegen eines neuen Active Directory-Objektes (z.B. *Administrator*, Mitglied der Gruppe *Konten-Operatoren* oder *Organisations-Admins*) und auch zum Hinzufügen eines Computer-Systems zur Domäne hat.

Nach der Authentifizierung wird – sofern noch nicht geschehen – das Computerkonto angelegt und der Client oder Standalone-Server der Domäne hinzugefügt. Dies wird mit der Meldung **WILLKOMMEN IN DER DOMÄNE MYDOMAIN** bestätigt, woraufhin Sie die Dialogfenster wieder schließen können und das Computersystem neu starten müssen, um die Änderungen wirksam werden zu lassen.

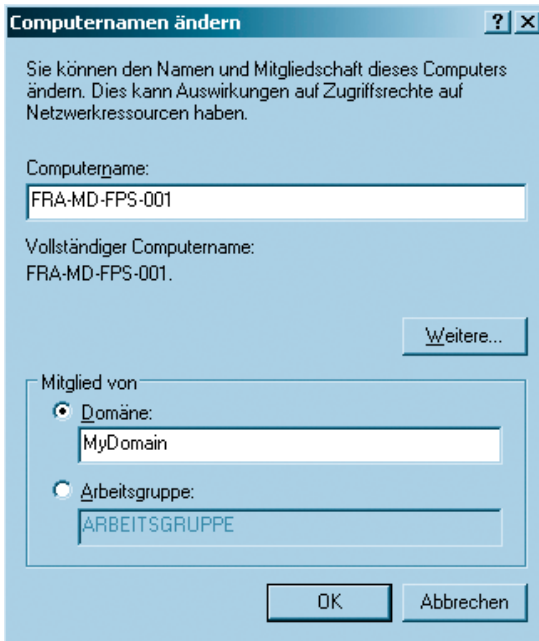


Abbildung 180: Einen Server zur Domäne hinzufügen

Die Lösung per Eingabeaufforderung

Die gleiche Aufgabe lässt sich auch in der Eingabeaufforderung lösen, wofür Sie allerdings eines der Support-Tools für den Windows Server 2003 benötigen. Denn in den Support-Tools ist unter anderem der `Netdom`-Befehl enthalten, womit Sie einen Client oder auch einen Standalone-Server in eine Domäne aufnehmen können.

Um beispielsweise den Standalone-Server *FRA-MD-FPS-001* in der Domäne *MyDomain.com* aufzunehmen, wenden Sie auf dem betreffenden Standalone-Server den `Netdom`-Befehl mit der Option `Join` an und übergeben den Namen des aufzunehmenden Computersystems und auch den Namen der Domäne als Parameter. Und für die Authentifizierung des Domänen-Controllers gegenüber übergeben Sie mit den Parametern `/UserD` und `/PasswordD` den Benutzernamen und auch das dazugehörige Kennwort, wobei Sie das Kennwort nicht unbedingt eingeben müssen. Denn mit dem Parameter `/PasswordD:*` können Sie den `Netdom`-Befehl dazu veranlassen, dass das Kennwort während der Befehlsausführung abgefragt wird.

```
@Echo off
NetDom Join FRA-MD-FPS-001 /Domain:MyDomain _
        /UserD:Administrator /PasswordD:12345678
```

Listing 461: 2_06_003a.bat

Für den Fall, dass das Computerkonto vor dem obigen Befehlsaufruf noch nicht erstellt wurde, so wird dies automatisch im Container COMPUTERS im Root der angegebenen Domäne angelegt. Wenn Sie das Computerkonto dagegen in einer bestimmten Organisationseinheit ablegen möchten, so übergeben Sie den definierten Namen des Zielobjektes mit dem Parameter /OU.

```
@Echo off
NetDom Join FRA-MD-FPS-001 /Domain:MyDomain _
/OU:OU=Vertrieb,DC=MyDomain,DC=Com _
/UserD:Administrator /PasswordD:*
```

Listing 462: 2_06_003b.bat

Die Lösung mit VBScript

Mit VBScript lässt sich ein Computersystem auf recht einfache Art und Weise einer Domäne hinzufügen. Die WMI-Klasse Win32_Computersystem stellt hierfür die JoinDomainOrWorkgroup-Methode bereit, womit Sie ein Computersystem wahlweise einer Arbeitsgruppe oder auch einer Domäne hinzufügen können. Die JoinDomainOrWorkgroup-Methode erwartet die folgenden Argumente, die Sie mit dem Methoden-Aufruf übergeben müssen:

- ▶ **Name:** Der Name der betreffenden Domäne oder der Arbeitsgruppe.
- ▶ **Password:** Das Kennwort des Benutzers, der über ausreichende Berechtigungen zur Aufnahme neuer Computerobjekte im Active Directory verfügt. Diese Anmeldeinformation wird für die Verbindung zum Domänen-Controller benötigt.
- ▶ **Username:** Der Name des Benutzers, der über ausreichende Berechtigungen zur Aufnahme neuer Computerobjekte im Active Directory verfügt.
- ▶ **Accent:** NULL-String oder aber optionale Angabe einer Organisationseinheit, falls das Computerobjekt in einer bestimmten Organisationseinheit und nicht im Default-Container für Computerobjekte (COMPUTERS) angelegt werden soll. Wenn eine Organisationseinheit angegeben wird, so muss der definierte Name der entsprechenden OU übergeben werden.
- ▶ **JoinOptions:** Legt die Optionen (siehe nachfolgende Tabelle) fest, die beim Anwenden der JoinDomainOrWorkgroup-Methode verwendet werden sollen. Die Optionen werden zu Beginn des Skriptes als Konstanten deklariert und mit dem Methodenaufruf übergeben.

Wert	Beschreibung
1	Computer wird zu einer Domäne (und nicht zu einer Arbeitsgruppe) hinzugefügt. Dies ist der Standardwert.
2	Computerkonto wird erstellt.
4	Computerkonto wird gelöscht.
16	Die Aufnahme der Domänenmitgliedschaft ist Teil eines Upgrade-Prozesses von Windows 9.x zu Windows NT/2000/2003.
32	Erlaubt die Aufnahme einer neuen Domänenmitgliedschaft, wenn das Computerobjekt bereits Mitglied einer anderen Domäne ist.
64	Forciert eine ungesicherte Domänen-Mitgliedschaft.
128	Computerkennwort (nicht Benutzerkennwort) wird übergeben.
256	SPN- und DNS-Hostname-Attribute werden erst nach dem Domänenbeitritt geändert.
262144	API-Installation

Tabelle 59: Die Optionen der JoinDomainOrWorkgroup-Methode

Das nachfolgende Code-Beispiel zeigt, wie Sie mit Hilfe der `JoinDomainOrWorkgroup`-Methode den Standalone-Server *FRA-MD-FPS-001* der Domäne *MyDomain.com* hinzufügen und zeitgleich ein entsprechendes Computerkonto anlegen können: Zunächst werden die erforderlichen Variablen und Konstanten deklariert, damit das Skript übersichtlich bleibt und leichter angepasst werden kann. Dann wird mit der `GetObject`-Methode eine Verbindung zum WMI-Namespace `root\cimv2` des betreffenden Computersystems hergestellt und der `Impersonation-Level` auf `Impersonate` gesetzt.

Anschließend wird eine `ExecQuery`-Abfrage mit dem Kriterium `Name` durchgeführt, um das Computerobjekt zu selektieren. Das Ergebnis wird daraufhin in Form einer Kollektion zurückgeliefert und kann innerhalb einer `For Each`-Schleife verarbeitet werden, wobei in diesem Fall die `JoinDomainOrWorkgroup`-Methode gegen das Computerobjekt angewendet wird.

Option Explicit

```
Dim oWMI
Dim colComputer
Dim Computer
Dim strDomain
Dim strComputer
Dim strUserName
Dim strPassword
```

Listing 463: 2_06_003.vbs

```

strDomain = "MyDomain"
strComputer = "FRA-MD-FPS-001"
strUserName = "Administrator"
strPassword = "12345678"
Const NETSETUP_JOIN_DOMAIN = 1
Const NETSETUP_ACCT_CREATE = 2

Set oWMI = GetObject("winmgmts:{impersonationLevel=_
    Impersonate}!\" & strComputer & "\root\cimv2")
Set colComputer = objWMIComputer.Get("Win32_ComputerSystem.Name='" _
    & strComputer & "'")

For Each Computer in colComputer
    Err = Computer.JoinDomainOrWorkGroup(strDomain, _
        strPassword, strDomain & "\" & strUserName, NULL, _
        NETSETUP_JOIN_DOMAIN + NETSETUP_ACCT_CREATE)
Next

Set oWMI = Nothing

```

Listing 463: 2_06_003.vbs (Forts.)

130 Computer aus Domäne herausnehmen

Die Domänenmitgliedschaft eines Computersystems lässt sich unter grafischer Oberfläche recht einfach beenden, da Sie das betreffende Computersystem nur einer Arbeitsgruppe hinzufügen müssen. Wenn das Computersystem (gleich ob Client oder Member-Server) für immer aus der Domäne verbannt werden soll, so löschen Sie im Active Directory auch das dazugehörige Computerkonto, anderenfalls (wenn die Mitgliedschaft nur temporär beendet werden soll) deaktivieren Sie das Computerkonto nur für den Zeitraum der gewünschten Nichtmitgliedschaft.

Um die Mitgliedschaft der Domäne zu beenden und ein Computersystem wieder in einer Arbeitsgruppe aufzunehmen, melden Sie sich zunächst als lokaler Administrator oder als Domänenadministrator an dem betreffenden Computersystem an und wechseln in die Eigenschaften der Netzwerkumgebung. Im Menü ERWEITERT wählen Sie den Eintrag NETZWERKIDENTIFIKATION... aus und klicken anschließend auf ÄNDERN, um das Dialogfenster COMPUTERNAMEN ÄNDERN aufzurufen. Dort aktivieren Sie die Option ARBEITSGRUPPE und geben den Namen der gewünschten Arbeitsgruppe ein.

Diese Änderung wird nun an den Domänen-Controller übergeben, wofür Sie sich dem Domänen-Controller gegenüber authentifizieren müssen. Verwenden Sie hierfür einen Account, der die Berechtigung zum Entfernen von Computerobjekten aus dem Active Directory besitzt, und führen Sie die Anmeldung durch. Sie erhalten daraufhin die Meldung WILLKOMMEN IN DER ARBEITSGRUPPE und müssen das Computersystem neu starten, damit die Änderung wirksam wird.

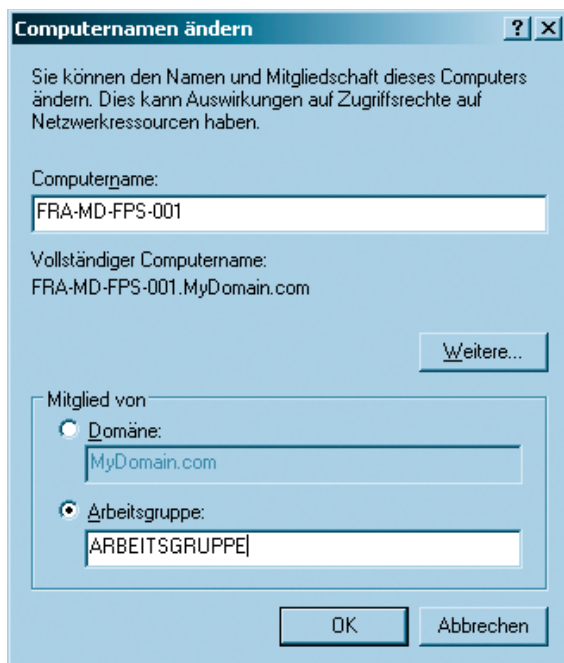


Abbildung 181: Einen Server aus der Domäne herausnehmen

Falls Sie das Computersystem für immer aus der Domäne nehmen möchten, so löschen Sie im Anschluss auch das dazugehörige Computerkonto im Active Directory. Hierzu melden Sie sich am Domänen-Controller an, rufen das Verwaltungswerkzeug ACTIVE DIRECTORY-BENUTZER UND -COMPUTER auf, markieren das betreffende Computerkonto und wählen aus dem Kontextmenü heraus LÖSCHEN aus.

Wenn Sie das Computersystem aber nur für eine gewisse Zeit aus der Domäne entfernen und das Computerkonto daher nur temporär außer Kraft setzen möchten, so deaktivieren Sie das Computerkonto nur. Wählen Sie im Kontextmenü nicht LÖSCHEN, sondern KONTO DEAKTIVIEREN aus. Sie können das Computerkonto dann zu einem späteren Zeitpunkt reaktivieren.

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung können Sie den `Netdom`-Befehl (der Bestandteil der Windows Server 2003 Support-Tools ist) mit der Option `Remove` verwenden, um ein Computersystem aus der Domäne zu entfernen und in einer Arbeitsgruppe aufzunehmen. Der `Netdom`-Befehl erwartet dabei die Angabe des Computernamens und auch den Namen der Domäne, aus der das Computersystem entfernt werden soll, als Parameter. Und da hierfür eine Authentifizierung dem Domänen-Controller gegenüber erforderlich ist, übergeben Sie den Benutzernamen und auch das dazugehörige Kennwort (oder `*` für die Aufforderung zur Kennworteingabe) mit den Parametern `/UserD` und `/PasswordD`, da der Befehl anderenfalls nicht ausgeführt werden kann.

Das nachfolgende Beispiel entfernt den Server *FRA-MD-FPS-001* aus der Domäne *MyDomain.com*, womit das Computersystem automatisch Mitglied der Arbeitsgruppe *MyDomain* wird:

```
@Echo off
NetDom Remove FRA-MD-FPS-001 /Domain:MyDomain _
/UserD:Administrator /PasswordD:12345678
```

Listing 464: 2_06_004a.bat

Damit die Änderung auf dem Computersystem wirksam wird, muss das betreffende Computersystem neu gestartet werden. Dies können Sie mit dem zusätzlichen Parameter */Reboot* unter Angabe der Wartezeit (in Sekunden) veranlassen:

```
@Echo off
NetDom Remove FRA-MD-FPS-001 /Domain:MyDomain _
/UserD:Administrator /PasswordD:12345678 /Reboot:5
```

Listing 465: 2_06_004b.bat

Die Lösung mit VBScript

Ähnlich wie Sie ein Computersystem einer Domäne hinzufügen können, so können Sie den Client oder Server auch wieder aus der Domäne entfernen. Denn die WMI-Klasse *Win32_Computersystem* stellt hierfür die *UnJoinDomainOrWorkgroup*-Methode bereit, der Sie lediglich den Benutzernamen und das Kennwort für den Zugriff auf den Domänen-Controller übergeben müssen und per Argument auch wahlweise festlegen können, ob das Computerkonto gelöscht werden soll oder nicht.

Das nachfolgende Code-Beispiel entfernt den Server *FRA-MD-FPS-001* aus der Domäne *MyDomain.com*, behält dabei aber das Computerkonto im Active Directory bei. Dazu wird per *GetObject*-Methode eine Verbindung zum WMI-Namespace *root\cimv2* des betreffenden Computersystems hergestellt. Mit Hilfe einer *ExecQuery*-Abfrage wird dann das entsprechende Computerobjekt selektiert. Innerhalb einer *For Each*-Schleife wird die *UnJoinDomainOrWorkgroup*-Methode gegen das Computerobjekt angewandt und das Kennwort wie auch der Benutzername für den Zugriff auf den Domänen-Controller als Argument übergeben. Mit dem zusätzlichen Argument *NET_SETUP_ACCT_DELETE* wird der Wert 0 übergeben, womit das Computerkonto im Active Directory nicht gelöscht, sondern beibehalten wird.

```
Option Explicit

Dim oWMI
Dim colComputer
```

Listing 466: 2_06_004.vbs

```
Dim Computer
Dim strComputer
Dim strUserName
Dim strPassword

strComputer = "FRA-MD-FPS-001"
strUserName = "Administrator"
strPassword = "12345678"
Const NETSETUP_ACCT_DELETE = 0

Set oWMI = GetObject("winmgmts:{impersonationLevel=_
    Impersonate}!\" & strComputer & "\root\cimv2")
Set colComputer = oWMI.ExecQuery("SELECT * FROM _
    Win32_ComputerSystem WHERE Name='" & strComputer & "'")

For Each Computer in colComputer
    Err = oComputer.UnJoinDomainOrWorkGroup(strPassword, _
        strUserName, NETSETUP_ACCT_DELETE)
Next

Set oWMI = Nothing
```

Listing 466: 2_06_004.vbs (Forts.)

131 Computer- und Systeminformationen auslesen

Wenn Sie eine Aufstellung der aktuellen Betriebssystemumgebung benötigen, weil Sie vielleicht Teile des Betriebssystems aktualisieren oder die Mindestvoraussetzungen für eine Software-Installation sicherstellen möchten, so rufen Sie im einfachsten Fall die SYSTEMINFORMATIONEN von Windows Server 2003 auf. Denn die SYSTEMINFORMATIONEN enthalten sehr detaillierte Angaben zu den folgenden Betriebssystemkategorien:

- ▶ **Betriebssystem:** In der Betriebssystemübersicht finden Sie die wichtigsten Information zur verwendeten Windows-Version, zum Aktivierungsstatus, zum Systemhersteller und -modell, zu den wichtigsten Systemordnern, zum verwendeten Gebietsschema wie auch zum virtuellen Arbeitsspeicher.
- ▶ **Hardware:** Der Hardware-Bereich gibt Auskunft über die Nutzung der Speicher-, DMA-, E/A- und IRQ-Ressourcen wie auch über eventuell vorhandene Konfliktkonfigurationen.
- ▶ **Komponenten:** In der Übersicht der Komponenten werden wichtige Geräteinformationen zu den internen Hardware-Komponenten und den installierten Ein- und Ausgabegeräten angezeigt.
- ▶ **Software:** Die Software-Übersicht liefert Informationen zu den installierten Treibern, den Diensten, den Prozessen, den Programmgruppen, den Umgebungsvariablen, den Systemtreibern und weiteren mehr.

- ▶ Internet: In der Internet-Übersicht wird eine Zusammenfassung der Internet Explorer-Version, der einzelnen Dateiversionen wie auch der Cache- und Sicherheitseinstellungen angezeigt.

Um die SYSTEMINFORMATIONEN aufzurufen, öffnen Sie das STARTMENÜ, klicken auf ALLE PROGRAMME, dann auf ZUBEHÖR, anschließend auf SYSTEMPROGRAMME und zu guter Letzt auf SYSTEMINFORMATIONEN. Sie können die SYSTEMINFORMATIONEN auch direkt aufrufen, indem Sie das STARTMENÜ öffnen, auf AUSFÜHREN klicken und Msinfo32 eingeben.

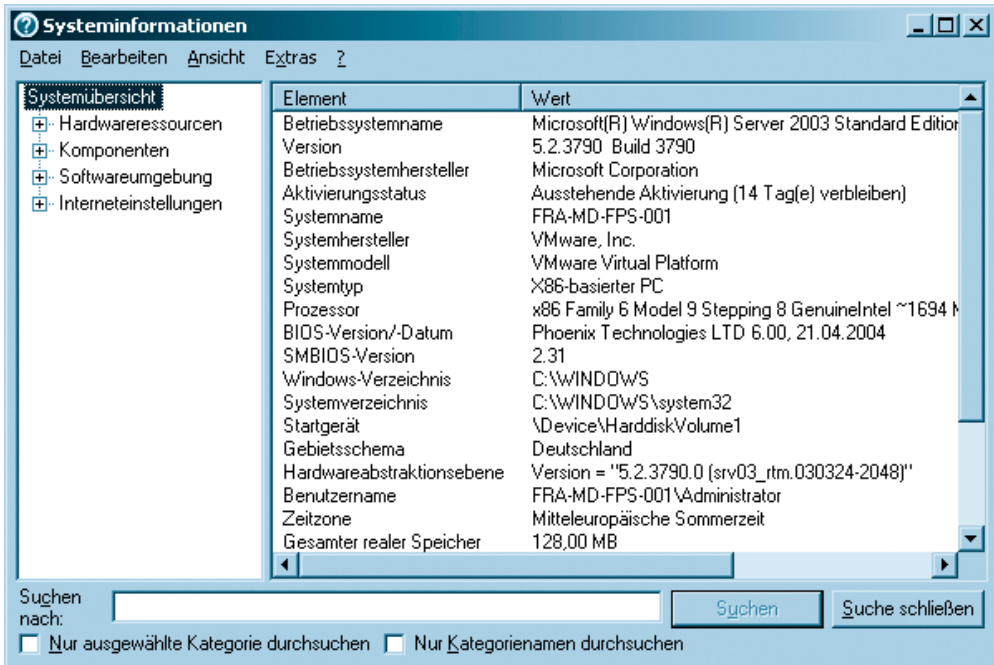


Abbildung 182: Die Systeminformationen von Windows Server 2003

Tipp

Microsoft bietet auf seiner Internetseite (www.microsoft.com) ein Reporting-Tool für Diagnosezwecke (*Microsoft Product Support Reporting Tool – Alliance Edition*) zum Download an, womit Sie einen noch weitaus detaillierteren Report erzeugen können. Dieses Reporting-Tool installiert zunächst einige Diagnosetools, liest dann die Betriebssystemumgebung aus und erzeugt im Ordner `C:\Windows\MPSReports\Alliance\Cab` eine Kabinettdatei mit mehreren Einzelreports, die genaueste Informationen zu den installierten Treibern, der verwendeten Hard- und Software-Umgebung, den Einträgen in den Eventlogs, den aktiven Prozessen, den Sicherheitseinstellungen und vielen weiteren mehr liefern.

Die Lösung per Eingabeaufforderung

Das Tool *Msinfo32.exe* können Sie auch in der Eingabeaufforderung verwenden, um eine Aufstellung der Computer- und Systeminformationen zu erzeugen. Denn mit dem Parameter */Report* lassen sich die Computer- und Systeminformationen, die unter grafischer Oberfläche angezeigt werden, in eine Datei exportieren. Und mit dem Parameter */Categories* können Sie zusätzlich bestimmen, welche Kategorien ausgegeben werden sollen. Folgende Kategorien stehen zur Auswahl:

Kategorie	Beschreibung
SystemSummary	Enthält allgemeine Informationen zum Computersystem und zur aktuellen Betriebssystemversion.
Resources	Enthält Informationen zu möglichen Ressourcenkonflikten, zu den DMA-, IRQ- und Speichereinstellungen.
Components	Enthält Informationen zu den installierten Hardware-Komponenten.
SWEnv	Enthält Informationen zu den verwendeten Systemtreibern, zu aktiven Prozessen, zu den Diensten und zum Autostart.
Internet-Einstellungen	Enthält Informationen zur Konfiguration des Internet Explorers.

Tabelle 60: Die Kategorien, die mit »Msinfo32« abgefragt werden können

Um alle Systeminformationen des aktuellen Computersystems von der Eingabeaufforderung aus abzufragen und das Ergebnis per Ausgabeumleitung in die Datei *C:\Systeminformationen.txt* zu schreiben, verwenden Sie folgende Befehlszeile:

```
@Echo off
"C:\Programme\Gemeinsame Dateien\Microsoft _
  Shared\Msinfo\Msinfo32" /Report C:\Systeminformationen.txt
```

Listing 467: 2_06_005a.bat

Möchten Sie dagegen nur die Betriebssysteminformationen ausgeben, so schränken Sie die Ausgabe der Systeminformationen auf die Kategorie *SystemSummary* ein, indem Sie den Parameter */Categories* anfügen und die gewünschte Kategorie angeben.

```
@Echo off
"C:\Programme\Gemeinsame Dateien\Microsoft _
  Shared\Msinfo\Msinfo32" /Report _
  C:\Systeminformationen.txt /Categories SystemSummary
```

Listing 468: 2_06_005b.bat

Eine weitere Möglichkeit, die wichtigsten Betriebssysteminformationen per Eingabeaufforderung auszugeben, bietet das Tool *Srvinfo.exe*, das Bestandteil des Windows

Server 2003 Resource Kits ist. Ohne Angabe weiterer Parameter gibt der `Srvinfo`-Befehl Informationen zur Betriebssystemversion, zur Installation, zum Produkt, zu Benutzer- und Computernamen wie auch zu den installierten Diensten, zur Netzwerkarte und zu den verwendeten Netzwerkprotokollen aus.

```
@Echo off
Srvinfo
```

Listing 469: 2_06_005c.bat

Die Lösung mit VBScript

Um aus einem VBScript heraus die wichtigsten Computer- und Systeminformationen auszugeben, lesen Sie idealerweise die Eigenschaften der WMI-Klasse `Win32_Computersystem` (siehe *Teil III* dieses Buches) aus, da diese Klasse die wichtigsten Informationen zur Systemumgebung enthält.

Das folgende Code-Beispiel zeigt, wie Sie mit Hilfe der `Win32_Computersystem`-Klasse eine ganze Reihe wichtiger Computer- und Systeminformationen auslesen und am Bildschirm ausgeben können.

Das Skript stellt im ersten Schritt eine Verbindung zum WMI-Namespace `root\cimv2` her, setzt den `Impersonation-Level` auf `Impersonate`, führt eine `ExecQuery`-Abfrage gegen die Klasse `Win32_Computersystem` aus und liest dann die Werte einiger Eigenschaften aus, die mit der `WScript.Echo`-Anweisung am Bildschirm ausgegeben werden.

```
Option Explicit

Dim oWMI
Dim colComputer
Dim Computer
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colComputer = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Computersystem")

For Each Computer in colComputer
    WScript.Echo "Caption: " & Computer.Caption
    WScript.Echo "CurrentTimeZone: " & _
        Computer.CurrentTimeZone
    WScript.Echo "DaylightInEffect: " & _
        Computer.DaylightInEffect
```

Listing 470: 2_06_005.vbs

```
WScript.Echo "Description: " & Computer.Description
WScript.Echo "Domain: " & Computer.Domain
WScript.Echo "DomainRole: " & Computer.DomainRole
WScript.Echo "EnableDaylightSavingsTime: " & _
    Computer.EnableDaylightSavingsTime
WScript.Echo "InstallDate: " & _
    WMIDateStringToDate(Computer.InstallDate)
WScript.Echo "KeyboardPasswordStatus: " & _
    Computer.KeyboardPasswordStatus
WScript.Echo "Manufacturer: " & Computer.Manufacturer
WScript.Echo "Model: " & Computer.Model
WScript.Echo "Name: " & Computer.Name
WScript.Echo "NumberOfProcessors: " & _
    Computer.NumberOfProcessors
WScript.Echo "OEMStringArray: " & strOEMStringArray
WScript.Echo "PartOfDomain: " & Computer.PartOfDomain
WScript.Echo "PowerManagementSupported: " & _
    Computer.PowerManagementSupported
WScript.Echo "PrimaryOwnerContact: " & _
    Computer.PrimaryOwnerContact
WScript.Echo "PrimaryOwnerName: " & _
    Computer.PrimaryOwnerName
WScript.Echo "SystemStartupDelay: " & _
    Computer.SystemStartupDelay
WScript.Echo "SystemStartupOptions: " & _
    strSystemStartupOptions
WScript.Echo "SystemStartupSetting: " & _
    Computer.SystemStartupSetting
WScript.Echo "SystemType: " & Computer.SystemType
WScript.Echo "TotalPhysicalMemory: " & _
    Computer.TotalPhysicalMemory
WScript.Echo "UserName: " & Computer.UserName
```

Next

Set oWMI = Nothing

Listing 470: 2_06_005.vbs (Forts.)

132 Wie lange läuft Windows bereits?

Der Windows Server 2003 stellt unter grafischer Oberfläche leider keine Möglichkeit bereit, womit Sie auf Anhieb die Uptime (= Betriebszeit) des Betriebssystems ermitteln können. Mit Hilfe bestimmter Ereignisse, die im Systemprotokoll mitgeschrieben werden, lässt sich in vielen Fällen zwar der Zeitpunkt des letzten Computerneustarts ermitteln, allerdings kann ein vermehrtes Aufkommen an Systemereignissen dazu führen, dass diese Informationen überschrieben werden, weshalb diese Quelle nicht gerade die zuverlässigste ist.

Um ein Durchsuchen der Logdateien zu vermeiden und die Betriebszeit auf Anhieb ermitteln zu können, lässt sich beispielsweise das Freeware-Tool *MinUpTime* verwenden, das Sie kostenlos von der Internetseite <http://www.tawbaware.com> herunterladen können. Dieses Tool erfordert keine Installation, sondern muss lediglich auf dem Zielsystem in einen Ordner extrahiert werden. Nach dem Start von *MinUpTime* wird ein kleines Fenster mit der aktuellen Uptime eingeblendet, das Sie auf dem Desktop frei platzieren können.

Im Kontextmenü des *MinUpTime*-Fensters lassen sich die Einstellungen aufrufen, worüber Sie zwischen einer kleinen, mittleren und großen Schrift auswählen und auch das Protokollieren der Uptime in eine gesonderte Logdatei veranlassen können.

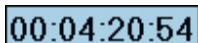


Abbildung 183: Die Windows-Uptime einblenden

Die Lösung per Eingabeaufforderung

Mit den Windows Server 2003-eigenen Bordmitteln lässt sich in der Eingabeaufforderung leider nur der Zeitpunkt des letzten Computerneustarts ermitteln, nicht aber die eigentliche Betriebszeit selbst.

Wenn Sie z.B. aus einer Batch-Datei heraus ermitteln möchten, wann der letzte Boot-Vorgang durchgeführt wurde, so führen Sie eine WMIC-Abfrage unter Verwendung des Alias `OS` aus und lassen sich mit Hilfe der `Get`-Anweisung nur den Wert der Eigenschaft `LastBootUpTime` anzeigen, womit nur das Datum und die Uhrzeit des letzten Computerneustarts im UTC-Format ausgegeben wird.

```
@Echo off
Wmic OS Get LastBootUpTime
```

Listing 471: 2_06_006a.bat

Das Windows Server 2003 Resource Kit enthält das Tool *Srvinfo.exe*, womit Sie eine ganze Reihe wichtiger Systeminformationen abfragen können; unter anderem auch die Uptime des Servers, die am Ende der Systeminformationen angezeigt wird:

```
@Echo off
Srvinfo
```

Listing 472: 2_06_006b.bat

Falls Sie das Windows 2000 Resource Kit zur Hand haben, so können Sie auch per Eingabeaufforderung die exakte Uptime ermitteln. Denn im Windows 2000 Resource Kit ist ein Tool namens *Uptime.exe* enthalten, das Sie nur auf dem betreffenden Com-

putersystem ausführen müssen, um die aktuelle Betriebszeit in Tagen, Stunden, Minuten und Sekunden zu erfahren.

```
@Echo off
Uptime
```

Listing 473: 2_06_006c.bat

Die Lösung mit VBScript

Die aktuelle Uptime lässt sich auch per VBScript berechnen. Hierfür fragen Sie zunächst den Wert der Eigenschaft `LastBootUpTime` der WMI-Klasse `Win32_OperatingSystem` ab, womit Sie die UTC-Time des letzten Boot-Vorgangs ermitteln können. Anschließend konvertieren Sie diese UTC-Time in ein gewöhnliches Datums- und Uhrzeitformat (*dd.mm.yyyy hh:mm*) und errechnen mit Hilfe der `DateDiff`-Funktion die Differenz zwischen der aktuellen Zeit und dem Zeitpunkt des letzten Boot-Vorgangs.

Um diese Aufgabe zu lösen, wenden Sie eine `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_OperatingSystem` an, um das Computerobjekt zu selektieren. Da das Ergebnis der `ExecQuery`-Abfrage in Form einer Kollektion zurückgeliefert wird, setzen Sie eine `For Each`-Schleife für den Zugriff auf das zurückgelieferte Element ein und weisen der Variablen `strUptime` den Wert der Eigenschaft `LastBootUpTime` zu, womit die Variable `strUptime` das Datum und auch die exakte Uhrzeit des letzten Boot-Vorgangs erhält.

Damit Sie das Datum und auch die Uhrzeit des letzten Boot-Vorgangs mit dem aktuellen Datum und der aktuellen Uhrzeit vergleichen können, müssen Sie den Wert der Variablen `strUptime` zunächst in das von VBScript verwendete Datums- und Zeitformat konvertieren, da der letzte Boot-Vorgang im UTC-Format zurückgeliefert wird. Für die Konvertierung lesen Sie mit der `Left`- und `Mid`-Funktion die entsprechenden Datums- und Uhrzeitinformationen aus, speichern sie in entsprechenden Variablen zwischen und setzen sie dann unter neuer Anordnung wieder zusammen.

Und zu guter Letzt wenden Sie die `DateDiff`-Funktion an, um die Differenz zwischen `strUptime` und dem aktuellen Zeitpunkt (`Now`) zu ermitteln, wobei die Angabe in diesem Fall in Minuten (*n*) erfolgen soll. Das Ergebnis wird dann mit der `WScript.Echo`-Anweisung am Bildschirm ausgegeben.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colItems
Dim strItem
Dim strUptime
```

Listing 474: 2_06_006.vbs

```

Dim strJahr
Dim strTag
Dim strMonat
Dim strStunde
Dim strMinute
Dim strSekunde
Dim strDiff

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colItems = oWMI.ExecQuery ("SELECT * FROM _
    Win32_OperatingSystem")

For Each strItem in colItems
    strUptime = strItem.LastBootUpTime
Next

strJahr = Left(strUptime,4)
strMonat = Mid(strUptime,5,2)
strTag = Mid(strUptime,7,2)
strStunde = Mid(strUptime,9,2)
strMinute = Mid(strUptime,11,2)
strSekunde = Mid(strUptime,13,2)

strUptime = strTag & "." & strMonat & "." & strJahr & " " _
    & strStunde & ":" & strMinute & ":" & strSekunde
strDiff = DateDiff("n", strUptime, Now)

WScript.Echo "Windows läuft bereits seit " & strDiff _
    & " Minuten..."

Set oWMI = Nothing

```

Listing 474: 2_06_006.vbs (Forts.)

133 Datum, Uhrzeit und Zeitzone abfragen und bearbeiten

Um unter grafischer Oberfläche das Datum und auch die Uhrzeit abzufragen, führen Sie einen Doppelklick auf die Anzeige der Uhrzeit im Gerätefeld der Task-Leiste aus oder öffnen das Systemsteuerungsobjekt DATUM UND UHRZEIT. Sie gelangen daraufhin in das Dialogfenster EIGENSCHAFTEN VON DATUM UND UHRZEIT, worüber das aktuelle Systemdatum und auch die Systemzeit, die vom Betriebssystem verwendet wird, angezeigt wird.

Für den Fall, dass in den EIGENSCHAFTEN VON DATUM UND UHRZEIT falsche Datums- und Zeitwerte angezeigt werden, so können Sie durch Auswahl eines neuen Datums und durch Festlegen einer neuen Uhrzeit eine Korrektur vornehmen und die Änderungen mit einem Klick auf ÜBERNEHMEN speichern. Falsche Datums- und Zeitwerte können allerdings auch aus der Verwendung einer falschen Zeitzone resultieren, weshalb Sie vor dem Korrigieren von DATUM und UHRZEIT erst die verwendete Zeitzone prüfen und auswählen sollten. Welche Zeitzone verwendet wird, können Sie ebenfalls dem Dialogfenster DATUM UND UHRZEIT entnehmen.



Abbildung 184: Die Zeitzone anzeigen lassen

Der sichere und zuverlässige Netzbetrieb setzt voraus, dass auf allen Computersystemen innerhalb des Netzes gleiche Datums- und Uhrzeiteinstellungen verwendet werden. Um dies zu gewährleisten, können Sie auf Computersystemen, die nicht Mitglied einer Domäne sind, auf der Registerkarte INTERNETZEIT einen Internet-/Intranet-Zeitserver auswählen (und auch neu festlegen) und so die Datums- und Uhrzeiteinstellungen des lokalen Computersystems mit den Einstellungen des Zeitserver synchronisieren. Der Windows Server 2003 nutzt per Default den Microsoft-eigenen Zeitserver mit der URL *Time.Windows.com*, dessen Verwendung allerdings eine funktionstüchtige Internetverbindung voraussetzt. Computersysteme, die einer Domäne angehören, synchronisieren die Datums- und Uhrzeiteinstellungen automatisch mit dem Domänen-Controller.

Um nun die Zeit-Synchronisierungsfunktion auf allen Computersystemen des internen Netzwerkes nutzen zu können, ohne dass alle Clients und Server mit dem Internet verbunden sind, empfiehlt sich der Einsatz eines Intranet-Zeitserver, der Synchronisierungsanfragen innerhalb des Netzwerkes annehmen und auch bedienen kann. Um einen Zeitserver zu definieren, aktivieren Sie auf dem Computersystem, das als Zeitserver fungieren soll, die Richtlinie WINDOWS-NTP-SERVER AKTIVIEREN, die Sie unter COMPUTERKON-

FIGURATION\ADMINISTRATIVE VORLAGEN\SYSTEM\WINDOWS-ZEITDIENST\ZEITANBIETER finden. Für alle anderen Computersysteme aktivieren Sie dagegen die Richtlinie WINDOWS-NTP-CLIENT KONFIGURIEREN (die sich im gleichen Pfad befindet) und legen im Feld NTP-SERVER den Hostnamen (oder die IP-Adresse) des internen Zeitserverns fest.

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus das aktuelle Datum und auch die aktuelle Uhrzeit ermitteln möchten, so können Sie hierfür den `Date`- und `Time`-Befehl verwenden, die ohne Angabe weiterer Parameter zur Eingabe eines neuen Datums- oder Zeitwertes auffordern. Diese Aufforderung können Sie aber mit dem Parameter `/T` unterdrücken und so nur die Ausgabe des Datums bzw. der Uhrzeit erzwingen.

```
@Echo off
Date /T
Time /T
```

Listing 475: 2_06_007a.bat

Die aktuelle Zeitzone lässt sich dagegen mit dem `WMIC`-Befehl ermitteln, womit Sie auf die WMI-Klasse `Win32_TimeZone` (`WMIC`-Alias `TIMEZONE`) zugreifen und die Eigenschaften der Zeitzone abfragen können. Um beispielsweise die Bezeichnung der Zeitzone auszugeben, wenden Sie den `WMIC`-Befehl unter Angabe des Alias `TimeZone` an und fragen mit `Get` nur den Inhalt der Eigenschaft `Caption` ab:

```
@Echo off
WMIC TimeZone Get Caption
```

Listing 476: 2_06_007b.bat

Selbst das Aktivieren der NTP-Server- und NTP-Client-Funktion ist von der Eingabeaufforderung aus kein Problem, da Sie nur einen entsprechenden Wert in die Registrierung schreiben müssen. Um beispielsweise die NTP-Server-Funktion zu aktivieren, ändern Sie den `DWORD`-Wert `Enabled`, der sich im Schlüssel `HKLM\SYSTEM\CURRENT-CONTROLSET\SERVICES\W32TIME\TIMEPROVIDERS\NTPSERVER` befindet, von 0 auf 1.

Hierfür wenden Sie den `Reg Add`-Befehl an, dem Sie zunächst den Schlüssel, dann mit `/v` den Wert (`Enabled`), mit `/t` den Datentyp (`REG_DWORD`) und mit `/d` den Wert (1) übergeben:

```
@Echo off
Reg Add HKLM\SYSTEM\CurrentControlSet\Services\W32Time\
TimeProviders\NTPServer /v Enabled /t REG_DWORD /d 1
```

Listing 477: 2_06_007c.bat

Die NTP-Client-Funktion lässt sich auf die gleiche Art und Weise aktivieren, nur geben Sie in diesem Fall den Schlüssel `NTPClient` (und nicht `NTPServer`) an und legen im Schlüssel `HKLM\SYSTEM\CURRENTCONTROLSET\SERVICES\W32TIME\PARAMETERS`, genauer gesagt in der Zeichenfolge `NTPServer`, den Hostnamen oder die IP-Adresse des entsprechenden NTP-Servers an.

```
@Echo off
Reg Add HKLM\SYSTEM\CurrentControlSet\Services\W32Time\
TimeProviders\NTPClient /v Enabled /t REG_DWORD /d 1
Reg Add HKLM\SYSTEM\CurrentControlSet\Services\W32Time\
Parameters/v NTPServer /t REG_SZ /d 192.168.0.1
```

Listing 478: 2_06_007d.bat

Um von der Eingabeaufforderung aus zu überprüfen, welcher NTP-Server verwendet wird, können Sie auch den `Net Time`-Befehl mit dem Parameter `/QuerySNTP` einsetzen, da dieser Befehl den Hostnamen oder die IP-Adresse des aktuellen NTP-Servers ausgibt.

Die Lösung mit VBScript

Wenn Sie das aktuelle Datum und auch die aktuelle Uhrzeit per VBScript ausgeben möchten, so können Sie im einfachsten Fall die `Date`- und `Time`-Funktion verwenden, Sie können aber auch die Eigenschaften der WMI-Klasse `Win32_LocalTime` auslesen, womit Sie erweiterte Datums- und Uhrzeitinformationen abfragen können (siehe nachfolgende Tabelle).

Eigenschaft	Beschreibung
Day	Tag
Month	Monat
Year	Jahr
Hour	Stunde
Minute	Minute
Second	Sekunden
DayofWeek	Tag der Woche
WeekInMonth	Woche im Monat
Quarter	Quartal im Jahr

Tabelle 61: Die Eigenschaften der Win32_LocalTime-Klasse

Das nachfolgende Listing gibt beispielsweise das aktuelle Datum, die aktuelle Uhrzeit wie auch erweiterte Informationen zum Tag der Woche, der Woche im Monat und zum Quartal aus. Hierzu wird per `GetObject` eine Verbindung zum WMI-Namespace `root\cimv2` des aktuellen Computersystems hergestellt und eine `ExecQuery`-Abfrage

gegen die WMI-Klasse `Win32_LocalTime` durchgeführt, woraufhin die aktuellen Datums- und Zeitwerte als Kollektion zurückgeliefert werden.

Innerhalb einer `For Each`-Schleife werden dann die Werte der Datums- und Zeiteigenschaften ausgelesen und per `WScript.Echo`-Anweisung am Bildschirm ausgegeben.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colDateTime
Dim DateTime

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colDateTime = oWMI.ExecQuery("SELECT * FROM _
    Win32_LocalTime")

For Each DateTime in colDateTime
    WScript.Echo DateTime.Day & "." & DateTime.Month & _
        "." & DateTime.Year
    WScript.Echo DateTime.Hour & ":" & DateTime.Minute & _
        ":" & DateTime.Second & ":" & DateTime.Milliseconds & _
        vbCrLf

    WScript.Echo "Day of week: " & DateTime.DayOfWeek
    WScript.Echo "Week in month: " & DateTime.WeekInMonth
    WScript.Echo "Quarter: " & DateTime.Quarter
Next

Set oWMI = Nothing
```

Listing 479: 2_06_007a.vbs

Die aktuelle Zeitzone lässt sich dagegen durch eine Abfrage der WMI-Klasse `Win32_TimeZone` ermitteln, die Informationen zu den Zeitzoneneinstellungen besitzt.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colTimeZone
Dim TimeZone
```

Listing 480: 2_06_007b.vbs

```

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colTimeZone = oWMI.ExecQuery("SELECT * FROM _
    Win32_TimeZone")

For Each TimeZone in colTimeZone
    WScript.Echo "TimeZone: " & TimeZone.Caption
    WScript.Echo "Name: " & TimeZone.DayLightName
Next

Set oWMI = Nothing

```

Listing 480: 2_06_007b.vbs (Forts.)

Und nicht zuletzt lassen sich auch die NTP-Einstellungen per VBScript aktivieren, da Sie diese Einstellungen direkt in die Registrierung schreiben können, wofür Sie die SetDWORDValue- und SetStringValue-Methode der WMI-Klasse StdRegProv einsetzen können.

Das nachfolgende Beispiel aktiviert die NTP-Server-Funktion auf dem aktuellen Computersystem. Hierzu wird eine Verbindung zum WMI-Namespace `root\default` benötigt, um daraufhin die SetDWORDValue-Methode unter Angabe des hexadezimalen Wertes des Hauptzweiges (HKLM), des Schlüssels (strKey), des Wertes (strValueName) und des eigentlichen DWORD-Wertes (strValue) ausführen zu können. Mit dem Setzen des DWORD-Wertes `Enabled` von 0 auf 1 wird der NTP-Server aktiviert.

```

Option Explicit

Dim oReg
Dim strComputer
Dim strKey
Dim strValueName
Dim strValue

Const HKLM = &H80000002
strComputer = "."
strKey = "SYSTEM\CurrentControlSet\Services\W32Time\
    TimeProviders\NTPServer"
strValueName = "Enabled"
strValue = "1"

Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")

```

Listing 481: 2_06_007c.vbs

```
oReg.SetDWORDValue HKLM, strKey, strValueName, strValue
```

```
Set oReg = Nothing
```

Listing 481: 2_06_007c.vbs (Forts.)

Der NTP-Client lässt sich auf die gleiche Art und Weise aktivieren, allerdings benötigen Sie hier einmal die `SetDWORDValue`-Methode, um den Wert `Enabled` von 0 auf 1 zu setzen, und die `SetStringValue`-Methode, um den eigentlichen NTP-Server als Zeichenfolge zu hinterlegen.

```
Option Explicit
```

```
Dim oReg
Dim strComputer
Dim strKey1
Dim strKey2
Dim strValueName1
Dim strValue1
Dim strValueName2
Dim strValue2
```

```
Const HKLM = &H80000002
strComputer = "."
strKey1 = "SYSTEM\CurrentControlSet\Services\W32Time\
TimeProviders\NTPServer"
strKey2 = "SYSTEM\CurrentControlSet\Services\W32Time\
Parameters"
strValueName1 = "Enabled"
strValue1 = "1"
strValueName2 = "NTPServer"
strValue2 = "192.168.0.1"
```

```
Set oReg=GetObject("winmgmts:{impersonationLevel=_
impersonate}!\\" & strComputer & _
"\root\default:StdRegProv")
oReg.SetDWORDValue HKLM, strKey1, strValueName1, strValue1
oReg.SetStringValue HKLM, strKey2, strValueName2, strValue2
```

```
Set oReg = Nothing
```

Listing 482: 2_06_007d.vbs

134 Systeminformationen als Hintergrundbild anzeigen

Mit dem Tool *BGInfo*, das Sie von der Internetseite www.sysinternals.com kostenlos herunterladen und ohne großen Installationsaufwand direkt einsetzen können, lassen sich die wichtigsten Computer- und Systeminformationen (z.B. *Hostname*, *IP-Adresse*, *MAC-Adresse*, *Server-Typ* ...) als Hintergrundbild darstellen. Dies hat den Vorteil, dass Sie – gerade in einer größeren Server-Landschaft – immer wissen, welchen Server Sie gerade administrieren oder mit welchem Server Sie gerade per Remotedesktop verbunden sind.

Um ein solches Hintergrundbild einzurichten, laden Sie *BGInfo* von der zuvor genannten Internetseite herunter und extrahieren die Datei *BGInfo.exe* in einen beliebigen Ordner. Anschließend wechseln Sie in den WINDOWS EXPLORER und starten das Tool mit einem Doppelklick, woraufhin Sie in das Dialogfenster BGINFO – DEFAULT CONFIGURATION gelangen und die gewünschten Computer- und Systeminformationen im Feld FIELDS auswählen und mit Hilfe der Textgestaltungselemente (Schriftart, Schriftfarbe ...) auch umgestalten können.

Mit Hilfe der Schaltfläche CUSTOM können Sie weitere Felder erzeugen, indem Sie eine Systemvariable, eine Registrierungseinstellung, ein Skript, eine WMI-Abfrage oder Dateiinformationen auswählen und der Liste der Felder hinzufügen.



Abbildung 185: Die Systeminformationen als Hintergrundbild

Wenn Sie das Hintergrundbild entsprechend Ihren Vorstellungen gestaltet haben, so speichern Sie die Konfiguration im Menü FILE mit SAVE AS in eine Datei und übernehmen die Einstellungen mit einem Klick auf APPLY, woraufhin das Hintergrundbild sofort aktiviert wird.

Damit die Informationen, die durch *BGInfo* dargestellt werden, auch aktualisiert werden können, binden Sie das Tool *BGInfo* und auch die dazugehörige Konfigurationsdatei in die allgemein gültige Autostart-Gruppe mit ein, damit die Informationen mit jedem Anmeldevorgang neu eingelesen und angezeigt werden. Hierfür erstellen Sie im Ordner *C:\Dokumente und Einstellungen\All Users\Startmenü\Programme\Autostart* eine Batch-Datei (*.bat) mit einem beliebigen Namen und nehmen folgende Befehlszeile darin auf:

```
C:\BGInfo\BGInfo.exe C:\BGInfo\Konfig.bgi /Timer:0
```

Wenn Sie das Tool *BGInfo*, die Konfigurationsdatei (*.bgi) und wahlweise auch die Batch-Datei auf ein anderes Computersystem kopieren, so können Sie das Hintergrundbild mit den Computer- und Systeminformationen auch auf andere Server übertragen, ohne das Hintergrundbild neu definieren zu müssen.

Die Lösung per Eingabeaufforderung

Wenn Sie unter grafischer Oberfläche bereits eine *BGInfo*-Konfigurationsdatei erstellt haben, so können Sie das Tool *BGInfo* wie auch die Konfigurationsdatei mit Hilfe des Copy-Befehls auf ein anderes Computersystem übertragen und das Hintergrundbild mit dem Befehl *BGInfo Konfig.bgi /Timer:0* aktivieren. Der Parameter */Timer:0* gibt hierbei an, dass *BGInfo* nicht im Konfigurationsmodus gestartet werden soll.

Die nachfolgende Batch-Datei kopiert die erforderlichen Dateien (*BGInfo.exe* und *Desktop.bgi*) in den Windows-Ordners des aktuellen Computersystems und aktiviert das Hintergrundbild:

```
@Echo off
Copy BGInfo.exe C:\Windows
Copy Desktop.bgi C:\Windows
Cd C:\Windows
Bginfo Desktop.bgi /Timer:0
```

Listing 483: 2_06_008.bat

Die Lösung mit VBScript

Auf die gleiche Art und Weise können Sie das Tool *BGInfo* auch per VBScript einrichten und aktivieren. Das nachfolgende Code-Beispiel nutzt hierfür die *Run*-Methode des *Shell*-Objektes und führt die gleichen Befehlszeilen aus, die auch im Lösungsbeispiel für die Eingabeaufforderung zum Einsatz kommen.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Copy BGInfo.exe C:\Windows"
oShell.Run "%comspec% /c Copy Desktop.bgi C:\Windows"
oShell.Run "%comspec% /c CD C:\Windows"
oShell.Run "%comspec% /c BGInfo Desktop.bgi /Timer:0"

Set oShell = Nothing
```

Listing 484: 2_06_008.vbs

135 Windows Update-Service konfigurieren

Der Windows Server 2003 ist mit einer automatischen Update-Funktion ausgestattet, womit Sie das Betriebssystem (und auch einige Microsoft-Applikationen) auf dem neuesten Stand und damit up to date halten können. Sie haben dabei die Möglichkeit, die Update-Pakete entweder über eine Internetverbindung direkt vom Microsoft Update-Server herunterzuladen oder innerhalb des Netzwerkes auch einen eigenen Software Update Service-Server (siehe *Betriebssystem-Update* in *Teil I* dieses Buches) einzusetzen, der gewissermaßen eine Schnittstelle zwischen den im Netzwerk befindlichen Computersystemen und dem Microsoft Update-Server bildet. Letzteres hat den Vorteil, dass nur der SUS-Server (Software Update Service-Server) eine Verbindung zum Microsoft Update-Server benötigt, da die anderen Computersysteme ihre Updates vom internen SUS-Server beziehen können, denn dieser speichert die Update-Pakete zwischen und reicht sie auf Anforderung an die internen Computersysteme weiter.

Die Windows Update-Funktion ist per Default aktiviert und lässt sich zum einen über das Systemsteuerungsobjekt *System* (Registerkarte AUTOMATISCHE UPDATES) konfigurieren, zum anderen aber auch per Gruppenrichtlinie und auch direkt in der Registrierung. Um die Einstellungen nicht auf jedem einzelnen Computersystem konfigurieren zu müssen, sollten Sie die Einstellungen mit Hilfe der Gruppenrichtlinien vornehmen.

Öffnen Sie hierzu die Gruppenrichtlinien (je nach Konfiguration die lokalen oder die der Domäne), und erweitern Sie zunächst COMPUTER-KONFIGURATION, dann ADMINISTRATIVE VORLAGEN und anschließend WINDOWS UPDATE, und führen Sie einen Doppelklick auf die Richtlinie AUTOMATISCHE UPDATES KONFIGURIEREN aus, um die Eigenschaften aufzurufen. Wählen Sie die Option AKTIVIERT aus, und legen Sie im Feld AUTOMATISCHE UPDATES KONFIGURIEREN fest, wie mit dem Download und der Installation der Update-Pakete verfahren werden soll:

- ▶ Vor Download und Installation benachrichtigen
- ▶ Autom. downloaden, aber vor Installation benachrichtigen
- ▶ Autom. downloaden und laut Zeitplan installieren
- ▶ Lokalem Administrator ermöglichen, Einstellungen auszuwählen

Im Feld GEPLANTER INSTALLATIONSTAG wählen Sie den gewünschten Wochentag und im Feld GEPLANTE INSTALLATIONSZEIT die Uhrzeit für die automatische Installation der Update-Pakete aus und speichern die geänderten Einstellungen mit einem Klick auf ÜBERNEHMEN.

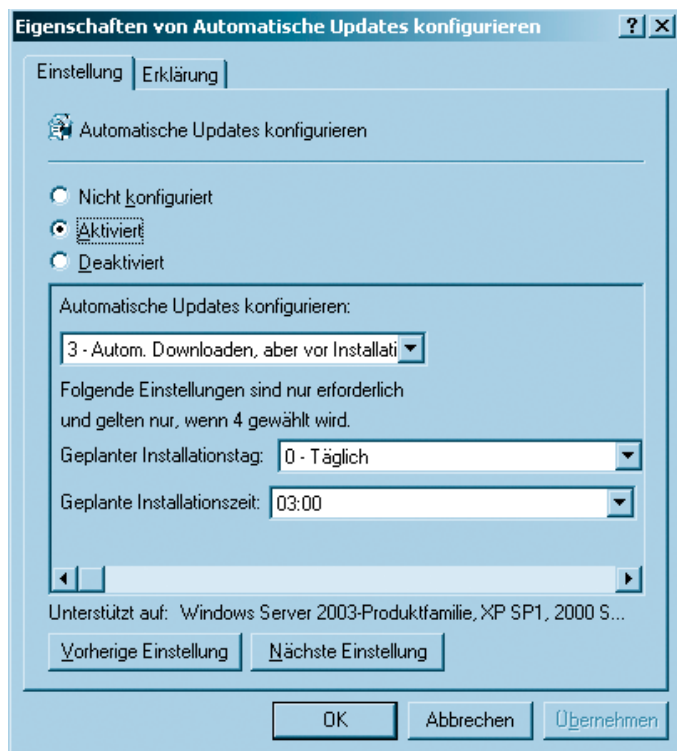


Abbildung 186: Die Windows Update-Richtlinie konfigurieren

Wenn Sie einen internen SUS-Server betreiben, so müssen Sie auf den SUS-Clients auch die Richtlinie INTERNEN PFAD FÜR DEN MICROSOFT UPDATEDIENST ANGEBEN konfigurieren, um den internen SUS-Server bekannt zu machen, denn standardmäßig wird immer der Microsoft Update-Server verwendet.

Hierzu öffnen Sie unter COMPUTER-KONFIGURATION\ADMINISTRATIVE VORLAGEN\WINDOWS-KOMPONENTEN\WINDOWS UPDATE die zuvor genannte Richtlinie und aktivieren zunächst die Option AKTIVIEREN. Anschließend geben Sie im Feld INTERNER UPDATE-DIENST ZUM ERMITTELN VON UPDATES die URL des SUS-Servers ein und legen im FELD INTRANETSERVER FÜR DIE STATISTIKEN die URL des Statistikservers (dies ist in den meisten Fällen ebenfalls der SUS-Server) fest. Speichern Sie die Änderungen mit einem Klick auf ÜBERNEHMEN ab.

Die Lösung per Eingabeaufforderung

Die AUTOMATISCHE UPDATES-Funktion lässt sich auch per Eingabeaufforderung konfigurieren, indem Sie mit dem `Reg Add`-Befehl die entsprechenden Policy-Einstellungen direkt in die Registrierung schreiben. Dazu legen Sie im Schlüssel `HKLM\SOFTWARE\POLICIES\MICROSOFT\WINDOWS` einen neuen Schlüssel mit der Bezeichnung `WINDOWSUPDATE` an und darunter einen weiteren Schlüssel mit dem Namen `AU`.

Anschließend erstellen Sie im Schlüssel `AU` die `DWORD`-Werte `AUOptions`, `NoAutoUpdate`, `ScheduledInstallDay`, `ScheduledInstallTime` und `UseWUServer` und legen jeweils die Einstellungen fest. Die möglichen Einstellungen und die dazugehörigen Werte können Sie der nachfolgenden Tabelle entnehmen.

Einstellung	Wert	Beschreibung
<code>AUOptions</code>	2	Vor Download und Installation benachrichtigen
<code>AUOptions</code>	3	Autom. downloaden, aber vor Installation benachrichtigen
<code>AUOptions</code>	4	Autom. downloaden und laut Zeitplan installieren
<code>AUOptions</code>	5	Lokalem Administrator ermöglichen, Einstellungen auszuwählen
<code>NoAutoUpdate</code>	0	Aktiviert die automatische Updates-Funktion
<code>NoAutoUpdate</code>	1	Deaktiviert die automatische Updates-Funktion
<code>ScheduledInstallDay</code>	0	Täglich
<code>ScheduledInstallDay</code>	1 – 7	Sonntag–Samstag
<code>ScheduledInstallTime</code>	0 – 23	00:00–23:00
<code>UseWUServer</code>	0	Deaktiviert die Verwendung eines SUS-Servers
<code>UseWUServer</code>	1	Aktiviert die Verwendung eines SUS-Servers

Tabelle 62: Die `DWORD`-Werte für die automatische Updates-Funktion

Wenn Sie einen internen SUS-Betreiber betreiben und die Update-Pakete nicht vom Microsoft Update-Server, sondern vom SUS-Server beziehen möchten, so legen Sie im Schlüssel `HKLM\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate` die beiden Zeichenfolgen `WUServer` und `WUStatusServer` an und hinterlegen die Adresse der beiden Server als eigentliche Zeichenfolge.

Wert	Beschreibung
<code>WUServer</code>	URL des SUS-Servers
<code>WUStatusServer</code>	URL des Statistikservers

Tabelle 63: Die `REG_SZ`-Werte für den SUS-Server

Mit den nachfolgenden Befehlszeilen wird die AUTOMATISCHE UPDATES-Funktion des lokalen Computersystems konfiguriert. Der Download der Update-Pakete erfolgt täglich um 03:00 Uhr, wobei die Update-Pakete automatisch heruntergeladen werden, vor der Installation aber eine Benachrichtigung erfolgt. Des Weiteren wird ein SUS-Server verwendet, dessen Adresse *HTTP://SUS* lautet, ebenso die des Statistikservers.

```
@Echo off
Reg Add HKLM\SOFTWARE\Policies\Microsoft\Windows\
  WindowsUpdate\AU /v AUOptions /t REG_DWORD /d 3 /f
Reg Add HKLM\SOFTWARE\Policies\Microsoft\Windows\
  WindowsUpdate\AU /v NoAutoUpdate /t REG_DWORD /d 0 /f
Reg Add HKLM\SOFTWARE\Policies\Microsoft\Windows\
  WindowsUpdate\AU /v ScheduledInstallDay _
  /t REG_DWORD /d 0 /f
Reg Add HKLM\SOFTWARE\Policies\Microsoft\Windows\
  WindowsUpdate\AU /v ScheduledInstallTime _
  /t REG_DWORD /d 3 /f
Reg Add HKLM\SOFTWARE\Policies\Microsoft\Windows\
  WindowsUpdate\AU /v UseWUServer /t REG_DWORD /d 1 /f
Reg Add HKLM\SOFTWARE\Policies\Microsoft\Windows\
  WindowsUpdate /v WUServer /t REG_SZ /d http://sus /f
Reg Add HKLM\SOFTWARE\Policies\Microsoft\Windows\
  WindowsUpdate /v WUStatusServer _
  /t REG_SZ /d http://sus /f
```

Listing 485: 2_06_009.bat

Die Lösung mit VBScript

Um die AUTOMATISCHE UPDATES-Funktion per VBScript zu konfigurieren, gehen Sie genauso vor wie in der Eingabeaufforderung und schreiben die entsprechenden Policy-Einstellungen wieder direkt in die Registrierung.

Um die DWORD-Werte für die AUTOMATISCHE UPDATES-Funktion in den Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\WINDOWSUPDATE\AU schreiben zu können, wenden Sie die SetDWORDValue-Methode der WMI-Klasse StdRegProv an und übergeben den hexadezimalen Wert des Hauptschlüssels, den Namen des Schlüssels (aber ohne vorangestellten Hauptzweig), den Namen des DWORD-Wertes wie auch den Wert selbst als Argument.

Und wenn Sie zeitgleich den SUS-Server konfigurieren möchten, so schreiben Sie die beiden Zeichenfolgen WUServer und WUStatusServer mit Hilfe der SetStringValue-Methode, die ebenfalls durch die WMI-Klasse StdRegProv bereitgestellt wird, in den Schlüssel HKLM\SOFTWARE\MICROSOFT\WINDOWS\WINDOWSUPDATE und hinterlegen die Adressen der beiden Server als Zeichenfolge.

Option Explicit

```
Dim oReg
Dim strComputer
Dim strKey1
Dim strKey2
Dim strValueName1, strValue1
Dim strValueName2, strValue2
Dim strValueName3, strValue3
Dim strValueName4, strValue4
Dim strValueName5, strValue5
Dim strValueName6, strValue6
Dim strValueName7, strValue7

Const HKLM = &H80000002
strComputer = "."
strKey1 = "SOFTWARE\Policies\Microsoft\Windows\_\
    WindowsUpdate\AU"
strKey2 = "SOFTWARE\Policies\Microsoft\Windows\_\
    WindowsUpdate"

strValueName1 = "AUOptions"
strValue1 = "3"
strValueName2 = "NoAutoUpdate"
strValue2 = "0"
strValueName3 = "ScheduledInstallDay"
strValue3 = "0"
strValueName4 = "ScheduledInstallTime"
strValue4 = "3"
strValueName5 = "UseWUServer"
strValue5 = "1"
strValueName6 = "WUServer"
strValue6 = "http://sus"
strValueName7 = "WUStatusServer"
strValue7 = "http://sus"

Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")
oReg.SetDWORDValue HKLM, strKey1, strValueName1, strValue1
oReg.SetDWORDValue HKLM, strKey1, strValueName2, strValue2
oReg.SetDWORDValue HKLM, strKey1, strValueName3, strValue3
oReg.SetDWORDValue HKLM, strKey1, strValueName4, strValue4
oReg.SetDWORDValue HKLM, strKey1, strValueName5, strValue5

oReg.SetStringValue HKLM, strKey2, strValueName6, strValue6
```

Listing 486: 2_06_009.vbs

```
oReg.SetStringValue HKLM, strKey2, strvalueName7, strValue7
```

```
Set oReg = Nothing
```

Listing 486: 2_06_009.vbs (Forts.)

136 Welche Applikationen sind installiert?

Um unter grafischer Oberfläche herauszufinden, welche Applikationen bereits installiert wurden, rufen Sie lediglich das Systemsteuerungsobjekt SOFTWARE auf. Sie gelangen daraufhin automatisch in die Übersicht der installierten Software-Produkte und können anhand der Einträge ersehen, welche Produkte im Einzelnen schon eingerichtet wurden.

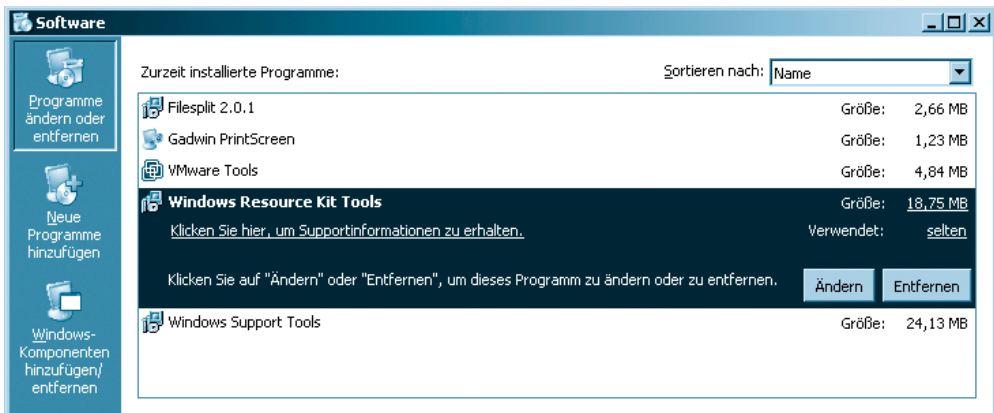


Abbildung 187: Die Anzeige der installierten Applikationen

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung lassen sich die installierten Applikationen ebenfalls abfragen, da Sie im einfachsten Fall nur den Inhalt des Schlüssels HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\UNINSTALL auslesen müssen, um eine Übersicht der installierten Software-Produkte zu erhalten. Denn mit jeder (oder zumindest mit fast jeder) Software-Installation wird im UNINSTALL-Schlüssel ein untergeordneter Schlüssel für die betreffende Applikation erstellt, worin die applikationsspezifischen Werte (z.B. *Display-Name*, *Display-Version*, *Publisher* ...) hinterlegt werden.

Da die Bezeichnungen der untergeordneten Schlüssel sowohl die Objekt-ID der Applikation (falls es sich um eine MSI-Anwendung handelt) als auch den eigentlichen Applikationsnamen (bei Nicht-MSI-Anwendungen) enthalten können, ist das Abfragen der Schlüsselbezeichnungen irrelevant. Mit Hilfe des `Reg Query`-Befehls und der Parameter `/v` und `/s` können Sie aber eine rekursive Abfrage gegen die untergeordneten Schlüssel eines bestimmten Schlüssels durchführen und jeweils den Inhalt eines

speziellen Wertes (z.B. *Displayname*) auslesen. Die Zeichenfolge *Displayname* enthält dabei die Bezeichnung der Applikation, wie sie auch in der Übersicht der installierten Software-Produkte unter grafischer Oberfläche angezeigt wird, und eignet sich daher sehr gut für eine solche Abfrage.

Die nachfolgende `Reg Query`-Befehlszeile führt eine rekursive Abfrage (`/s`) gegen den Schlüssel `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\UNINSTALL` aus und schreibt den Wert der Zeichenfolge *Displayname* (der Name der Applikation) in die Datei *Software.txt*:

```
@Echo off
Reg Query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\_
Uninstall /v Displayname /s > Software.txt
```

Listing 487: 2_06_010a.bat

Eine weitere Möglichkeit, die installierten Applikationen auszulesen, bietet der `WMIC`-Befehl mit dem Alias `Product`, womit Sie von der Eingabeaufforderung aus auf die WMI-Klasse `Win32_Product` zugreifen und die Eigenschaften der `Product`-Objekte abfragen können.

Allerdings hat diese Methode zwei Nachteile: Der Einsatz des `WMIC Product`-Befehls setzt voraus, dass auf dem betreffenden Windows Server 2003 die Windows-Komponente `WMI-ANBIETER FÜR WINDOWS INSTALLER` (dies ist eine Komponente der `VERWALTUNGS- UND ÜBERWACHUNGSPROGRAMME`) installiert ist. Des Weiteren lassen sich damit nur `MSI`-Installationen abfragen, womit Sie in den meisten Fällen nur ein Teilergebnis erhalten.

```
@Echo off
WMIC Product > Product.txt
```

Listing 488: 2_06_010b.bat

Die Lösung mit VBScript

Die einfachste und auch komfortabelste Möglichkeit, die installierten Applikationen per VBScript auszulesen, bietet in diesem Fall die `Run`-Methode des `Shell`-Objektes, da Sie hiermit den `Reg Query`-Befehl aus einem VBScript ausführen und die Zeichenfolge *Display-Name* der untergeordneten Schlüssel von `HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\UNINSTALL` rekursiv auslesen können.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
```

Listing 489: 2_06_010.vbs

```
oShell.Run "%comspec% /c Reg Query _
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\_
Uninstall /v Displayname /s > Software.txt "
```

```
Set oShell = Nothing
```

Listing 489: 2_06_010.vbs (Forts.)

137 Ereignisprotokollierung für das Herunterfahren deaktivieren

Der Windows Server 2003 ist mit einer automatischen Ereignisprotokollierung für das Herunterfahren des Computersystems ausgestattet und zeichnet per Default alle Shutdown- und Restart-Ereignisse im Systemprotokoll der EREIGNISANZEIGE auf. Und für den Fall, dass ein ungeplantes Shutdown- oder Restart-Ereignis eintritt, so wird im Ordner C:\Windows\System32\LogFiles\ Shutdown eine XML-Datei mit dem aktuellen Systemstatus generiert, die zu Troubleshooting-Zwecken an Microsoft übermittelt werden kann.

Aus Gründen der Systemüberwachung ist diese spezielle Ereignisprotokollierung mit Sicherheit von Vorteil. Wenn Sie den Server aber des Öfteren starten müssen, weil Sie gerade Hard- oder Software-Installationen vornehmen oder diverse Konfigurationsarbeiten durchführen, so kann sich das Systemstatusdatenfeature als recht nervig erweisen, da Sie mit jedem Herunterfahren oder Neustarten des Computersystems einen Grund (und wahlweise auch einen Kommentar) für das bevorstehende Shutdown- oder Restart-Ereignis eingeben müssen.

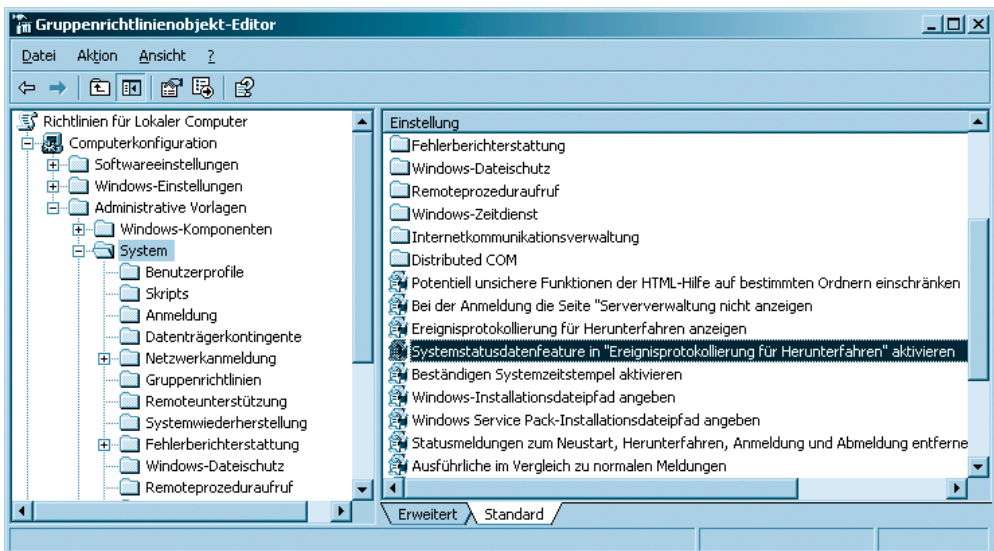


Abbildung 188: Die Systemstatusdatenfeature-Richtlinie

Sie können die EREIGNISPROTOKOLLIERUNG FÜR DAS HERUNTERFAHREN des Computersystems jedoch deaktivieren, indem Sie die Richtlinie SYSTEMSTATUSDATENFEATURE IN »EREIGNISPROTOKOLLIERUNG FÜR HERUNTERFAHREN« AKTIVIEREN deaktivieren. Hierzu öffnen Sie die Gruppenrichtlinien (je nach Server-Konfiguration die lokalen Gruppenrichtlinien oder die der Domäne) und erweitern zunächst COMPUTERKONFIGURATION, dann ADMINISTRATIVE VORLAGEN, anschließend SYSTEM und markieren dann die zuvor genannte Richtlinie. Rufen Sie im Kontextmenü der Richtlinie die EIGENSCHAFTEN auf, und aktivieren Sie dann die Option DEAKTIVIEREN, um die Richtlinie zu deaktivieren und das Systemstatusdatenfeature auszuschalten.

Die Lösung per Eingabeaufforderung

Die Ereignisprotokollierung für das Herunterfahren des Servers lässt sich auch von der Eingabeaufforderung aus deaktivieren, da Sie eine lokale Richtlinie auch direkt in der Registrierung setzen bzw. verändern können.

Um die lokale Richtlinie zu definieren, legen Sie mit Hilfe des Reg Add-Befehls den Schlüssel HKLM\SOFTWARE\Policies\Microsoft\Windows NT\Reliability an und erzeugen in dem Schlüssel einen neuen DWORD-Wert mit der Bezeichnung ShutDownReasonOn, dem Sie den Wert 1 zuweisen (der Wert 0 würde die Ereignisprotokollierung für das Herunterfahren wieder aktivieren).

```
@Echo off
Reg Add "HKLM\SOFTWARE\Policies\Microsoft\Windows _
NT\Reliability" /v ShutDownReasonOn /t REG_DWORD /d 1
```

Listing 490: 2_06_011.bat

Die Lösung mit VBScript

Auf die gleiche Art und Weise lässt sich die Ereignisprotokollierung für das Herunterfahren des Servers auch per VBScript deaktivieren.

Das Anlegen des DWORD-Wertes nehmen Sie dabei mit der SetDWORDValue-Methode der WMI-Klasse StdRegProv vor, der Sie den hexadezimalen Wert des Hauptzweiges (HKLM = &H80000002), den Schlüssel (SOFTWARE\Policies\Microsoft\Windows NT\Reliability), die Bezeichnung des Wertes (ShutDownReasonOn) und auch den Wert (1) selbst als Argument übergeben.

```
Option Explicit

Dim oReg
Dim strComputer
Dim strKey
Dim strValueName
```

Listing 491: 2_06_011.vbs

```

Dim strValue

Const HKLM = &H80000002
strComputer = "."
strKey = "SOFTWARE\Policies\Microsoft\Windows NT\Reliability"
strValueName = "ShutDownReasonOn"
strValue = "1"

Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")
oReg.SetDWORDValue HKLM, strKey, strValueName, strValue

Set oReg = Nothing

```

Listing 491: 2_06_011.vbs (Forts.)

138 Systemereignisse abfragen

In den Ereignisprotokollen der Ereignisanzeige wird eine ganze Reihe von Informationen, Warnungen und Fehler zum Betriebssystem wie auch zur verwendeten Hard- und Software aufgezeichnet, womit sich der Windows Server 2003 sehr gut überwachen lässt. Diese Systemereignisse lassen sich mit Hilfe der Ereignisanzeige jederzeit einsehen und des Troubleshootings wegen oder zu statischen Zwecken auswerten. Die Ereignisanzeige untergliedert sich dabei in die folgenden Protokolle, wobei je nach Server-Rolle und installierten Applikationen auch noch weitere Protokolle vorhanden sein können:

- ▶ **Anwendungsprotokoll:** Im Anwendungsprotokoll werden Ereignisse zu den Anwendungen festgehalten. Wenn beispielsweise eine Applikation abgestürzt ist, nicht gestartet werden konnte oder sonstige Warnungen und Fehler meldet, so sind diese Ereignisse im Anwendungsprotokoll zu finden.
- ▶ **Systemprotokoll:** Das Systemprotokoll enthält alle Informationen, Warnungen und Fehler zu den Systemkomponenten. Wenn Treiber nicht geladen, Dienste nicht gestartet oder die Festplatten- und Speicherressourcen zur Neige gehen, so finden Sie diese Einträge im Systemprotokoll.
- ▶ **Sicherheitsprotokoll:** Das Sicherheitsprotokoll enthält Ereignisse zu erfolgreichen und fehlerhaften An- und Abmeldevorgängen, zu Änderungen der Sicherheitseinstellungen und zu den einzelnen Zugriffen.

Um die Systemereignisse unter grafischer Oberfläche auslesen zu können, öffnen Sie das STARTMENÜ, wechseln in das Untermenü VERWALTUNG und klicken auf EREIGNISANZEIGE oder wählen im STARTMENÜ den Eintrag AUSFÜHREN und geben `Eventvwr.msc` ein. Anschließend markieren Sie das entsprechende Ereignisprotokoll (*Anwendung, Sicher-*

heit, System), woraufhin die einzelnen Ereignisse in der rechten Fensterhälfte angezeigt und durch einen Klick auf die jeweiligen Spaltentitel auch nach *Typ*, *Datum*, *Uhrzeit*, *Quelle* und dergleichen mehr sortiert werden können. Um weitere Informationen zu einem Ereignis zu erhalten, öffnen Sie das jeweilige Ereignis durch einen Doppelklick, woraufhin alle Angaben (z.B. *Ereignis-ID*, *Ereignisquelle*, *Beschreibung* ...) angezeigt werden.

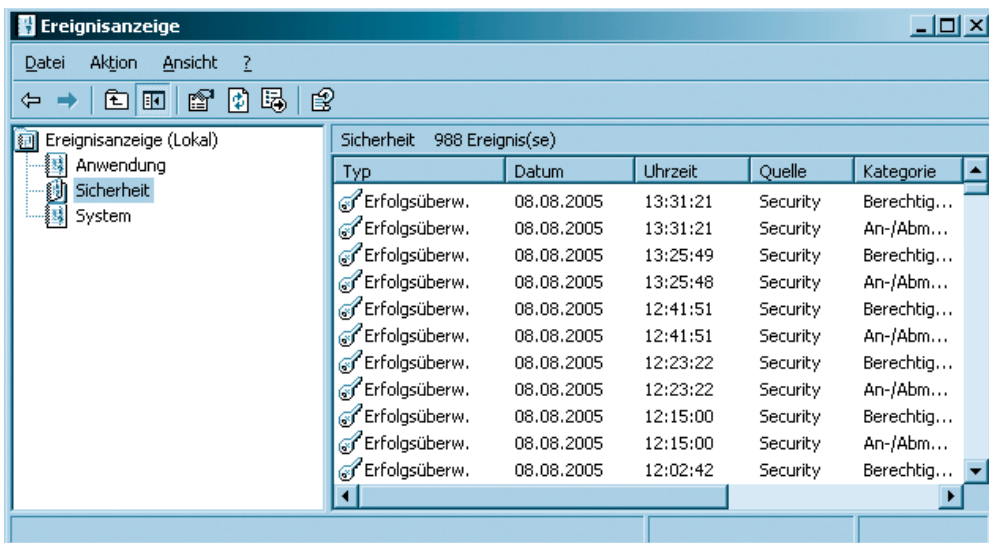


Abbildung 189: Die Einträge des Sicherheitsprotokolls

Je nach Größe der einzelnen Protokolle und nach Anzahl der Ereignisse kann sich die Suche nach bestimmten Ereignissen als recht mühsam erweisen. Die EREIGNISANZEIGE ist daher mit einer Filter-Funktion ausgestattet, womit Sie bestimmte Ereignisse herausfiltern können. Wenn Sie z.B. im Systemprotokoll nur die Ereignisse des Typs FEHLER anzeigen lassen möchten, so wählen Sie im Kontextmenü von SYSTEM erst ANSICHT und dann FILTER aus, um in die Einstellungen der Filter-Funktion zu gelangen. In den Auswahlfeldern EREIGNISQUELLE und KATEGORIE wählen Sie den Eintrag (ALLE) aus und aktivieren nur den Ereignistyp FEHLER. Mit einem Klick auf ÜBERNEHMEN wird die Filter-Funktion angewandt, woraufhin nur die Ereignisse angezeigt werden, die vom Typ FEHLER sind.

Um wieder alle Ereignisse einzublenden, wählen Sie im Kontextmenü des Protokolls ANSICHT und dann ALLE EINTRÄGE aus, woraufhin der Filter wieder deaktiviert wird.

Tipp

Wenn Sie Warnungen und Fehler in den Ereignisprotokollen vorfinden, so sollten Sie der Systemstabilität wegen nach den Ursachen sehen und die Probleme umgehend beheben. Die Beschreibung des Ereignisses gibt in der Regel entsprechende Hinweise darauf, womit Sie den Fehler lokalisieren und anschließend auch bereinigen können. Falls Sie Schwierigkeiten bei der Interpretation der Ereignisbeschreibung haben, so können Sie mit Hilfe der Ereignis-ID auf der Internetseite www.microsoft.com nach möglichen Lösungsansätzen suchen. Sehr empfehlenswert ist auch die Internetseite www.eventid.net, da viele Administratoren über dieses Forum Erfahrungen in Sachen Ereignisprotokollierung austauschen und Sie in den meisten Fällen gleich mehrere Lösungsansätze vorfinden.

Die Lösung per Eingabeaufforderung

Wenn Sie die Systemereignisse von der Eingabeaufforderung aus abfragen möchten, so verwenden Sie hierfür den WMIC-Befehl unter Verwendung des Alias NTEvent, da Sie mit diesem Befehl eine WMI-Abfrage gegen die Klasse Win32_NTEventLogFile durchführen und die Eigenschaften der Logfile-Einträge auslesen können.

Ohne Angabe weiterer Parameter gibt der WMIC NTEvent-Befehl alle Ereignisse aus allen Protokollen unter Angabe aller Eigenschaften am Bildschirm aus. Um die Ausgabe der WMIC-Abfrage auf ein bestimmtes Ereignisprotokoll und auf die wichtigsten Eigenschaften zu beschränken, selektieren Sie mit der Where-Klausel das gewünschte Protokoll (*System*, *Application*, *Security*) und lesen mit der Get-Anweisung nur die Eigenschaften Type (Ereignistyp), TimeGenerated (Datum und Uhrzeit), SourceName (Quelle), Category (Kategorie), EventCode (Ereignis-ID), User (Benutzername), ComputerName und Message (Beschreibung) aus. Damit erhalten Sie die gleichen Informationen zurück, wie sie auch in der Ereignisanzeige unter grafischer Oberfläche angezeigt werden.

```
@Echo off
Wmic NTEvent Where "LogFile='System'" Get Type, _
    TimeGenerated, SourceName, Category, EventCode, User, _
    ComputerName, Message
```

Listing 492: 2_06_012.bat

Die Lösung mit VBScript

Auf ähnlichem Wege lassen sich die Systemereignisse auch per VBScript abfragen. Will heißen: Sie nehmen auch in diesem Fall wieder WMI zu Hilfe, führen eine Abfrage gegen die Klasse Win32_NTLogEvent durch und fragen die Eigenschaften der Event-Objekte ab.

Das nachfolgende Listing gibt beispielsweise alle Ereignisse des Systemprotokolls unter Angabe von *Typ*, *Datum*, *Uhrzeit*, *Quelle*, *Kategorie*, *Ereignis-ID*, *Benutzername*, *Computername* und *Beschreibung* am Bildschirm aus.

Option Explicit

Dim oWMI

Dim colEvents

Dim strEvent

Dim strComputer

strComputer = "."

Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_impersonate}!\" & strComputer & "\root\cimv2")

Set colEvents = oWMI.ExecQuery ("SELECT * FROM _Win32_NTLogEvent WHERE Logfile = 'System'")

For Each strEvent in colEvents

Wscript.Echo "Type: " & strEvent.Type

Wscript.Echo "Time Generated: " & strEvent.TimeGenerated

Wscript.Echo "Source: " & strEvent.SourceName

Wscript.Echo "Category: " & strEvent.Category

Wscript.Echo "Event-ID: " & strEvent.EventCode

Wscript.Echo "User: " & strEvent.User

Wscript.Echo "Computer Name: " & strEvent.ComputerName

Wscript.Echo "Message: " & strEvent.Message & VBCRLF

Next

Set oWMI = Nothing

Listing 493: 2_06_012.vbs

139 Eigenschaften der Ereignisprotokolle ändern

Jedes Ereignisprotokoll verfügt über gewisse Eigenschaften, die Sie – administrative Berechtigungen vorausgesetzt – jederzeit ändern können. So lässt sich beispielsweise der Name des Protokolls ändern, die Größe der Protokolldatei festlegen wie auch das Verhalten bei Erreichen der maximalen Protokollgröße bestimmen. Um diese Eigenschaften ändern zu können, rufen Sie im STARTMENÜ, genauer gesagt unter VERWALTUNG, die EREIGNISANZEIGE auf und markieren in der linken Fensterhälfte das Protokoll, dessen Eigenschaften Sie ändern möchten. Anschließend öffnen Sie das Kontextmenü des Protokolls und wählen EIGENSCHAFTEN aus, um in die Protokolleinstellungen zu gelangen.

Im Feld ANZEIGENAME können Sie nun eine neue Bezeichnung für das Protokoll eingeben, indem Sie den aktuellen Anzeigenamen einfach durch einen neuen Wert überschreiben. Wenn Sie dagegen die Größe des Protokolls erhöhen möchten, so wählen Sie im Feld MAXIMALE PROTOKOLLGRÖSSE die gewünschte Größe (Angabe in KByte) aus, wobei der eingegebene Wert durch 64 teilbar sein muss. Größen, die diese Anforderung nicht erfüllen, werden nicht übernommen und müssen korrigiert werden.

Um das Verhalten bei Erreichen der maximalen Protokollgröße zu bestimmen, wählen Sie eine der folgenden Optionen aus:

- ▶ Ereignisse bei Bedarf überschreiben: Mit dieser Option legen Sie fest, dass bei Erreichen der maximalen Protokollgröße die ältesten Ereignisse durch neue Ereignisse überschrieben werden. Dies hat den Vorteil, dass Ereignisse immer protokolliert werden können, da die Protokolldatei vor dem Überlaufen geschützt ist.
- ▶ Ereignisse überschreiben, die älter als x Tage sind: Diese Option bewirkt, dass bei Erreichen der maximalen Protokollgröße nur solche Ereignisse überschrieben werden können, die älter als x Tage sind. Bei einem vermehrten Aufkommen an Ereignissen innerhalb kurzer Zeit kann diese Option zur Folge haben, dass neue Ereignisse nicht protokolliert werden können.
- ▶ Ereignisse nie überschreiben: Diese Option sollten Sie nur in Verbindung mit einer ausreichenden Protokollgröße verwenden, da bei Erreichen der maximalen Protokollgröße keine Ereignisse überschrieben und damit auch nicht protokolliert werden können. Je nach Sicherheitseinstellung kann diese Option das automatische Herunterfahren des Servers bedeuten.

Wenn Sie die Eigenschaften überprüft und bei Bedarf auch entsprechend geändert haben, so übernehmen Sie die Einstellungen mit einem Klick auf ÜBERNEHMEN, woraufhin die Eigenschaften gespeichert werden.

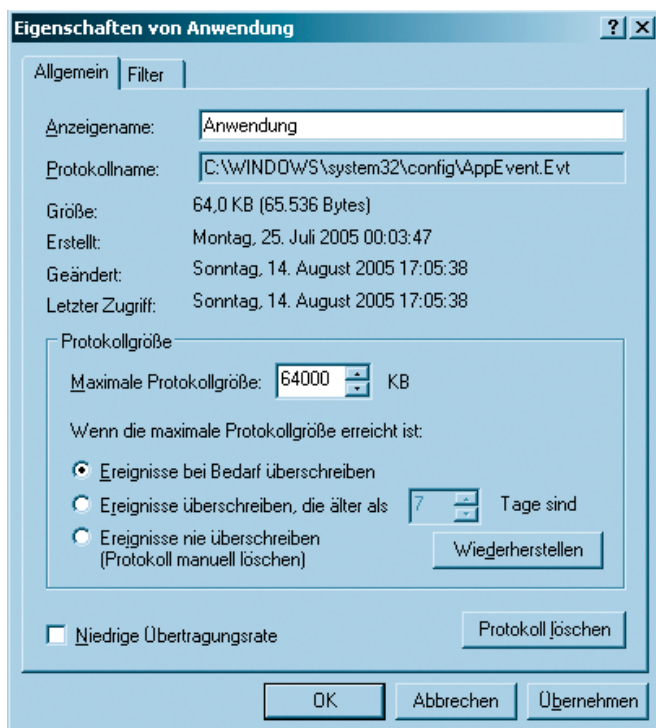


Abbildung 190: Die Eigenschaften des Anwendungsprotokolls

Die Lösung per Eingabeaufforderung

Die Eigenschaften der Ereignisprotokolle lassen sich mit dem WMIC-Befehl auch von der Eingabeaufforderung aus ändern. Hierfür wenden Sie den Alias `NTEventLog` an, womit Sie auf die WMI-Klasse `Win32_NTEventLogFile` zugreifen und mit der `Set-Anweisung` die Eigenschaften `MaxFileSize` (Größe der Protokolldatei) und `OverwriteOutDated` (Verhalten bei Erreichen der maximalen Protokollgröße) ändern können.

Um beispielsweise für alle Ereignisprotokolle eine Größe von 64.000 KByte festzulegen, wenden Sie den WMIC `NTEventLog`-Befehl an und weisen der Eigenschaft `MaxFileSize` mit `Set` den neuen Wert zu, wobei Sie die Größe in diesem Fall in Bytes ($64000 * 1024 = 65536000$) angeben müssen. Wenn Sie zeitgleich festlegen möchten, dass Ereignisse, die älter als 90 Tage sind, überschrieben werden können, so weisen Sie der Eigenschaft `OverwriteOutDated` den Wert 90 zu. Wenn die Ereignisse bei Bedarf überschrieben werden können, erhält `OverwriteOutDated` den Wert 0, und wenn das Überschreiben der Ereignisse gänzlich vermieden werden soll, so weisen Sie der Eigenschaft den Wert 4294967295 zu.

```
@Echo off
Wmic NTEventLog Set MaxFileSize=65536000, _
    OverwriteOutDated=90
```

Listing 494: 2_06_013.bat

Die Lösung mit VBScript

Auf die gleiche Art und Weise können Sie die Eigenschaften der Ereignisprotokolle auch per VBScript ändern. Hierzu selektieren Sie die Ereignisprotokolle mit der `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_NTEventLogFile` und weisen der Eigenschaft `MaxFileSize` die Größe der Protokolldatei (Angabe in Bytes) zu.

Der Eigenschaft `OverwriteOutDated` dagegen übergeben Sie den Wert 0, wenn die Ereignisse bei Bedarf überschrieben werden können, die Anzahl der Tage (1-365), wenn Ereignisse mit einem bestimmten Alter überschrieben werden oder aber durch Angabe des Wertes 4294967295 nie überschrieben werden dürfen. Mit der `Put-Methode`, der Sie den hexadezimalen Wert 20000 übergeben, speichern Sie die neuen Einstellungen schließlich ab.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colLogFile
Dim LogFile
```

Listing 495: 2_06_013.vbs

```

Const wbemFlagUseAmendedQualifiers = &h20000
strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate,(Security)}!\\" & strComputer & "\root\cimv2")
Set colLogFile = oWMI.ExecQuery ("SELECT * FROM _
    Win32_NTEventLogFile")

For Each LogFile in colLogFile
    LogFile.MaxFileSize = 65536000
    LogFile.OverwriteOutDated = 90
    LogFile.Put_ wbemFlagUseAmendedQualifiers
Next

Set oWMI = Nothing

```

Listing 495: 2_06_013.vbs (Forts.)

140 Ereignisprotokolle exportieren

Die Standardeinstellungen der Ereignisprotokolle sehen eine maximale Protokollgröße von 16.384 KByte vor. Wenn diese Größe erreicht wird, werden per Default ältere Ereignisse durch neue Ereignisse überschrieben, womit länger zurückliegende Ereignisse kaum noch eingesehen, geschweige denn: analysiert, werden können.

Mit Hilfe der SPEICHERN UNTER-Funktion der EREIGNISANZEIGE können Sie die einzelnen Protokolle exportieren und damit sichern. Dies hat den Vorteil, dass Sie im Notfall auch noch auf ältere Protokolle zurückgreifen und die Ereignisse (insbesondere die Warnungen und Fehler) zur Analyse heranziehen können.

Um ein Protokoll zu exportieren, rufen Sie zunächst die EREIGNISANZEIGE auf. Öffnen Sie hierzu das STARTMENÜ, erweitern Sie VERWALTUNG, und klicken Sie auf EREIGNISANZEIGE, oder wählen Sie im STARTMENÜ den Eintrag AUSFÜHREN, und geben Sie Eventvwr.msc ein. Anschließend markieren Sie das Protokoll, das Sie exportieren möchten, und wählen im Kontextmenü des Protokolls PROTOKOLLDATTEI SPEICHERN UNTER... aus. Sie gelangen daraufhin in einen SPEICHERN UNTER-Dialog, worüber Sie das Protokoll als Protokolldatei (*.evt) oder als Textdatei (*.csv, *.txt) speichern können.

Je nachdem, wie viele Ereignisse im Durchschnitt protokolliert werden, führen Sie die zuvor genannten Schritte in regelmäßigen Abständen durch, um auch zu einem späteren Zeitpunkt noch bestimmte Ereignisse nachverfolgen zu können.

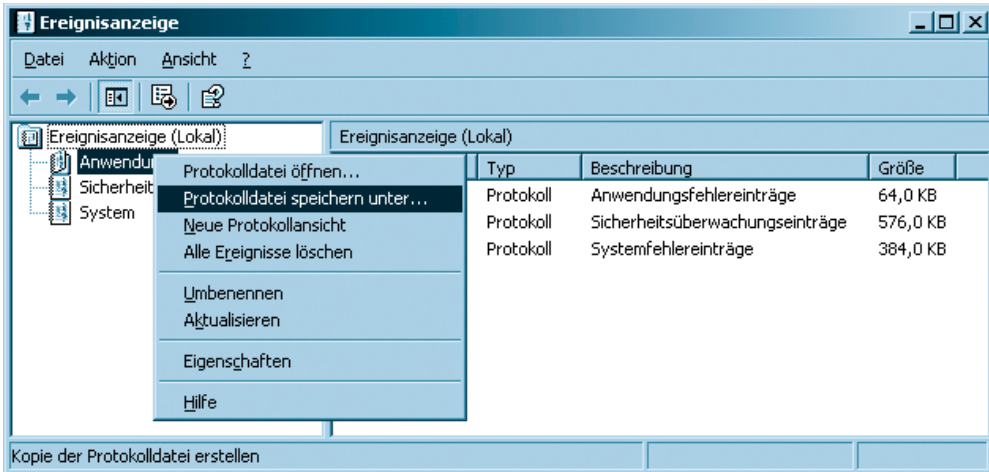


Abbildung 191: Das Anwendungsprotokoll exportieren

Die Lösung per Eingabeaufforderung

Gleiches lässt sich natürlich auch von der Eingabeaufforderung aus realisieren, indem Sie mit dem WMIC-Befehl eine Verbindung zur WMI-Klasse Win32_NTEventLogFile, die durch den Alias NTEventLog repräsentiert wird, herstellen und die BackupEventLog-Methode aufrufen. Denn die BackupEventLog-Methode ermöglicht den Export der angegebenen Protokolldatei (*appevent*, *secevent*, *sysevent*) in eine beliebige Datei, wobei das Format der Protokolldatei verwendet wird, damit die Datei zu einem späteren Zeitpunkt wieder mit der Ereignisanzeige geöffnet und eingesehen werden kann.

Das nachfolgende Beispiel sichert den Inhalt des Anwendungsprotokolls (*appevent*) in die Datei *C:\Application.evt*:

```
@Echo off
Wmic NTEventLog Where "FileName='appevent'" _
    Call BackupEventLog C:\Application.evt
```

Listing 496: 2_06_014.bat

Die Lösung mit VBScript

Um ein Ereignisprotokoll per VBScript zu sichern, verwenden Sie ebenfalls die BackupEventLog-Methode der WMI-Klasse Win32_NTEventLogFile. Hierzu stellen Sie im ersten Schritt eine Verbindung zum WMI-Namespace root\cimv2 des betreffenden Computersystems her, setzen den Impersonation-Level auf Impersonate und übergeben das Backup-Privileg. Anschließend selektieren Sie mit einer ExecQuery-Abfrage das Ereignisprotokoll (*Application*, *Security* oder *System*) und wenden dann die BackupEventLog-Methode gegen das Protokollobjekt an, der Sie den Pfad und Dateinamen der Zielfeile als Parameter übergeben.

```

Option Explicit

Dim oWMI
Dim strComputer
Dim colLogFile
Dim LogFile

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate,(Backup)}!\\" & strComputer & "\root\cimv2")
Set colLogFile = oWMI.ExecQuery ("SELECT * FROM _
    Win32_NTEventLogFile WHERE LogFileName='Application'")

For Each logfile in colLogFile
    Err = LogFile.BackupEventLog_
        ("C:\Backup\Logs\Application.evt")
Next

Set oWMI = Nothing

```

Listing 497: 2_06_014.vbs

141 Eigene Systemereignisse erstellen

Der Windows Server 2003 stellt leider keine entsprechenden Möglichkeiten bereit, womit Sie unter grafischer Oberfläche eigene Systemereignisse erstellen können. Das Betriebssystem stellt aber entsprechende Hilfsmittel zur Verfügung, um Ereignisse von der Eingabeaufforderung aus und auch per VBScript zu erzeugen.

Die Lösung per Eingabeaufforderung

Mit dem `EventCreate`-Befehl, der fester Bestandteil des Windows Server 2003-Befehlssatzes ist, lassen sich problemlos eigene Ereignisse erstellen. Der `EventCreate`-Befehl erwartet dabei die folgenden Angaben, die Sie mit dem Befehlsaufruf als Parameter übergeben müssen:

- ▶ Ereignis-ID: Mit dem Parameter `/ID` legen Sie die ID des Ereignisses fest, wobei Sie für eigene Ereignisse die IDs von 1 bis 1000 verwenden können.
- ▶ Ereignisprotokoll: Mit dem Parameter `/L` legen Sie das Ereignisprotokoll (*Application, Security, System*) fest.
- ▶ Typ: Mit dem Parameter `/T` übergeben Sie den Typ des Ereignisses. Das Anwendungs- und Systemprotokoll unterstützt dabei die Typen *Information, Warning* und *Error*, das Sicherheitsprotokoll dagegen *Success* und *Failure*.
- ▶ Beschreibung: Mit dem Parameter `/D` übergeben Sie die Beschreibung zum Ereignis (z.B. *Das Skript xyz wurde erfolgreich ausgeführt*).

Die nachfolgende `EventCreate`-Befehlszeile erzeugt im Systemprotokoll des aktuellen Computersystems ein Ereignis mit der ID 1000 vom Typ `Information` und legt die Beschreibung `Das Systemprotokoll wurde erfolgreich exportiert` fest.

```
@Echo off
EventCreate /ID 1000 /L System /T Information _
/D "Das Systemprotokoll wurde erfolgreich exportiert"
```

Listing 498: 2_06_015.bat

Die Lösung mit VBScript

Mit der `LogEvent`-Methode des `Shell`-Objektes lassen sich neue Ereignisse auch per VBScript mit nur wenigen Code-Zeilen erstellen. Die `LogEvent`-Methode erwartet dabei den dezimalen Wert des Ereignistyps (siehe nachfolgende Tabelle) wie auch die Beschreibung des Ereignisses als Argument.

Während Sie mit dem `CreateEvent`-Befehl auch das Protokoll bestimmen und die Ereignis-ID festlegen können, werden Ereignisse, die mit der `LogEvent`-Methode erstellt werden, grundsätzlich mit der ID 4 in das Anwendungsprotokoll geschrieben.

Ereignis-Typ	Wert
Erfolgreich	0
Fehler	1
Warnung	2
Information	4
Erfolgsüberwachung	8
Fehlerüberwachung	16

Tabelle 64: Die Ereignistypen der LogEvent-Methode und ihre Werte

Das nachfolgende Code-Beispiel erzeugt im Anwendungsprotokoll ein neues Ereignis vom Typ `Information`, das die ID 4 und die Beschreibung `Das Systemprotokoll wurde erfolgreich exportiert` enthält:

```
Option Explicit

Dim oShell
Dim strMessage

strMessage = "Das Systemprotokoll wurde erfolgreich _
exportiert"
Const EVENT_INFORMATION = 4
```

Listing 499: 2_06_015.vbs

```
Set oShell = Wscript.CreateObject("Wscript.Shell")
oShell.LogEvent EVENT_INFORMATION, strMessage
```

```
Set oShell = Nothing
```

Listing 499: 2_06_015.vbs (Forts.)

142 Aktive Prozesse auslesen

Wenn Sie unter grafischer Oberfläche herausfinden möchten, welche Applikationen und Hintergrundprozesse gerade ausgeführt werden, so brauchen Sie nur den TASK-MANAGER aus dem Kontextmenü der TASK-LEISTE aufzurufen und die Registerkarte PROZESSE in den Vordergrund zu holen. Denn auf dieser Registerkarte werden alle aktiven Prozesse unter Angabe ihres Namens, des Benutzernamens, der CPU- und Speichernutzung aufgeführt.

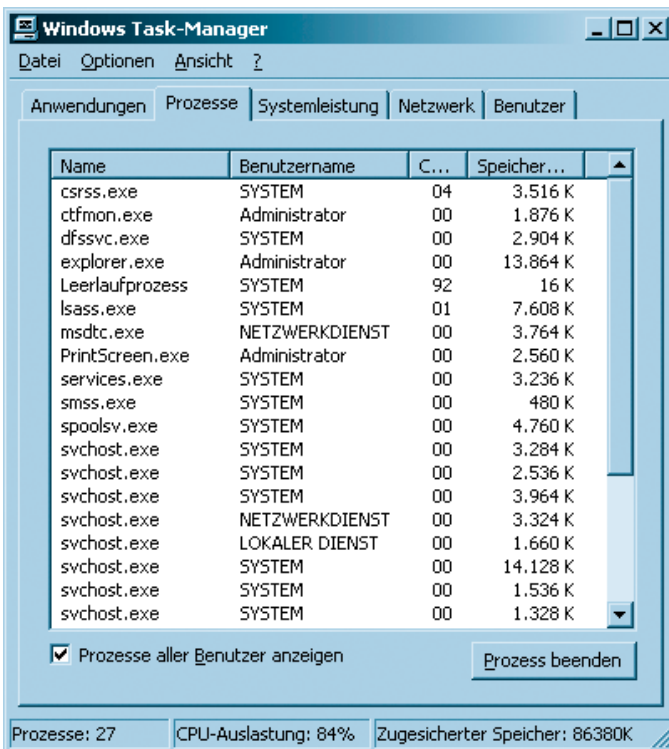


Abbildung 192: Die aktiven Prozesse

Die Lösung per Eingabeaufforderung

Mit dem `Tasklist`-Befehl, der zum festen Befehlsumfang von Windows Server 2003 gehört, lassen sich die Prozesse auch von der Eingabeaufforderung aus anzeigen. Ohne Angabe weiterer Parameter gibt der `Tasklist`-Befehl dabei die Prozesse unter Angabe des Namens, der Prozess-ID, des Sitzungsnamens, der Sitzungsnummer wie auch der Speichernutzung am Bildschirm aus.

Mit Hilfe des Parameters `/v` können Sie die Prozessauflistung um weitere Informationen ergänzen, da mit `/v` zusätzlich der Status des Prozesses, der Benutzername, die CPU-Zeit und auch der Fenstertitel mit ausgegeben wird.

```
@Echo off
Tasklist /v
```

Listing 500: 2_06_016a.bat

Eine weitere Möglichkeit der Prozessauflistung bietet der `Wmic`-Befehl unter Angabe des Alias `Process`, womit Sie von der Eingabeaufforderung aus eine WMI-Abfrage gegen die Klasse `Win32_Process` durchführen und die Eigenschaften der Prozesse auslesen können.

Um beispielsweise alle Prozesse unter Angabe des Namens, der Prozess-ID und des Dateinamens auszugeben, wenden Sie den `Wmic`-Befehl unter Angabe des Alias `Process` an und lesen mit Hilfe der `Get`-Anweisung nur die Eigenschaften `Caption`, `ExecutablePath` und `ProcessID` aus.

```
@Echo off
Wmic Process Get Caption,ExecutablePath,ProcessID
```

Listing 501: 2_06_016b.bat

Die Lösung mit VBScript

Mit Hilfe einer WMI-Abfrage lassen sich die aktiven Prozesse des Computersystems auch per VBScript abfragen. Denn die WMI-Klasse `Win32_Process` stellt eine ganze Reihe von Prozesseigenschaften (siehe *Teil III* dieses Buches) zur Verfügung, die Sie jederzeit auslesen können:

Das nachfolgende Code-Beispiel zeigt, wie Sie die Prozesseigenschaften *Name*, *Dateiname* und *Prozess-ID* per WMI abfragen können. Zunächst wird eine `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_Process` durchgeführt, um die einzelnen Prozessobjekte zu selektieren. Da das Ergebnis als Kollektion zurückgeliefert wird, werden die Eigenschaften `Name`, `ExecutablePath` und `ProcessID` in einer `For Each`-Schleife ausgelesen und per `WScript.Echo`-Anweisung am Bildschirm ausgegeben.

```

Option Explicit

Dim oWMI
Dim colProcess
Dim Process
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "impersonationLevel=_
    impersonate)!\" & strComputer & "\root\cimv2")
Set colProcess = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Process")

For Each Process in colProcess
    Wscript.Echo "Name: " & Process.Name
    Wscript.Echo "Dateiname: " & Process.ExecutablePath
    Wscript.Echo "Prozess-ID: " & Process.ProcessID & VBCRLF
Next

Set oWMI = Nothing

```

Listing 502: 2_06_016.vbs

143 Speicherauslastung aktiver Prozesse abfragen

Wenn Sie unter grafischer Oberfläche herausfinden möchten, wie die Speichernutzung der einzelnen Prozesse aussieht, so brauchen Sie nur den TASK-MANAGER aus dem Kontextmenü der TASK-LEISTE zu öffnen und die Registerkarte PROZESSE in den Vordergrund zu holen. Denn in der Übersicht der aktiven Prozesse wird unter anderem auch die Speicherauslastung angezeigt, worüber Sie die Speichernutzung des jeweiligen Prozesses erkennen können.

Tipp

Im *Windows Server 2003 Resource Kit*, das Sie kostenlos von der Internetseite www.microsoft.com herunterladen können, ist ein Tool mit der Bezeichnung *Empty.exe* enthalten, womit Sie nicht genutzten Speicher aktiver Prozesse freigeben können. Dadurch können Sie die Speicherauslastung senken und die Performance des Servers wieder ein Stück weit erhöhen, ohne die Prozesse beenden und neu starten zu müssen. Denn *Empty.exe* gibt den reservierten Speicherbereich (*Working Set*) der Prozesse automatisch frei.

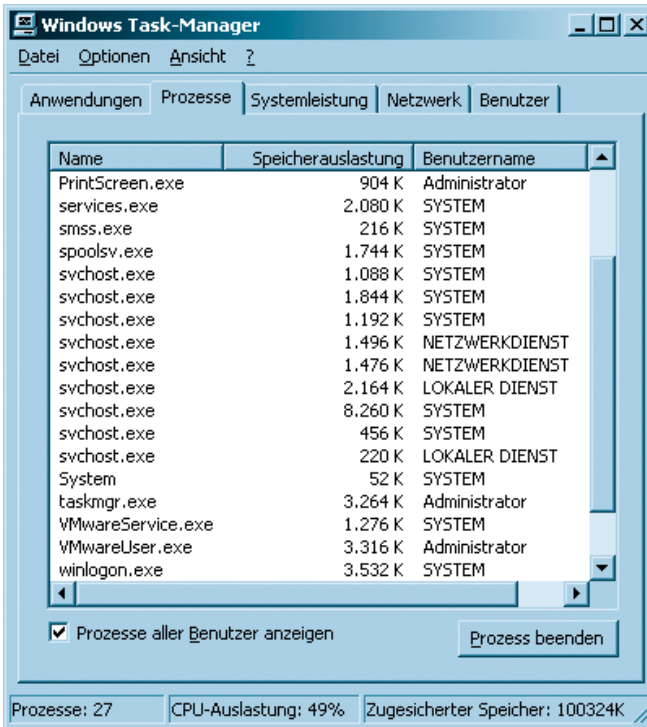


Abbildung 193: Die Speicherauslastung der Prozesse anzeigen lassen

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus ermitteln möchten, wie viel Speicher ein bestimmter Prozess verwendet, so können Sie den `Tasklist`-Befehl mit der Filter-Option (`/Fi`) verwenden und sich den betreffenden Prozess unter Angabe des Namens, der Prozess-ID, der Sitzungsinformation und auch der Speicherauslastung am Bildschirm anzeigen lassen.

Um beispielsweise die Prozessinformationen (und damit auch die Speicherauslastung) des Prozesses `cmd.exe` ausgeben zu lassen, wenden Sie den `Tasklist`-Befehl mit der Filter-Option an und selektieren alle Prozesse, deren `ImageName` gleich `cmd.exe` ist:

```
@Echo off
Tasklist /Fi "ImageName eq cmd.exe"
```

Listing 503: 2_06_017a.bat

Eine weitere Möglichkeit, die Speicherauslastung eines Prozesses zu ermitteln, bietet der `WMIC`-Befehl in Verbindung mit dem Alias `Process`, da Sie mit dem `WMIC Process`-Befehl eine Abfrage gegen die WMI-Klasse `Win32_Process` durchführen und so den

Wert der Eigenschaft `WorkingSetSize` auslesen können. Diese Eigenschaft enthält die aktuelle Speichernutzung, wobei die Angabe in Bytes erfolgt.

```
@Echo off
WMIC Process Get Caption,WorkingSetSize
```

Listing 504: 2_06_017b.bat

Die Lösung mit VBScript

Um die aktuelle Speichernutzung eines Prozesses per VBScript zu ermitteln, nutzen Sie ebenfalls eine WMI-Abfrage und fragen den Wert der Eigenschaft `WorkingSetSize` ab, die durch die WMI-Klasse `Win32_Process` bereitgestellt wird.

```
Option Explicit

Dim oWMI
Dim colProcess
Dim Process
Dim strComputer

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colProcess = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Process")

For Each Process in colProcess
    Wscript.Echo "Caption: " & Process.Caption
    Wscript.Echo "WorkingSetSize: " & Process.WorkingSetSize _
        & VBCRLF
Next

Set oWMI = Nothing
```

Listing 505: 2_06_017.vbs

144 Einzelne Prozesse beenden

Hin und wieder kommt es vor, dass Applikationen oder einzelne Hintergrundprozesse nicht mehr richtig funktionieren oder überhaupt nicht mehr auf eine Anforderung reagieren und sich daher auch nicht mehr richtig beenden bzw. neu starten lassen. In solchen Fällen bleibt meist nur, den Prozess (oder die Applikation) mit Hilfe des TASK-MANAGERS zu beenden und die ausführbare Datei auf eine etwas unsanfte Methode zu beenden, womit ungespeicherte Daten allerdings verloren gehen.

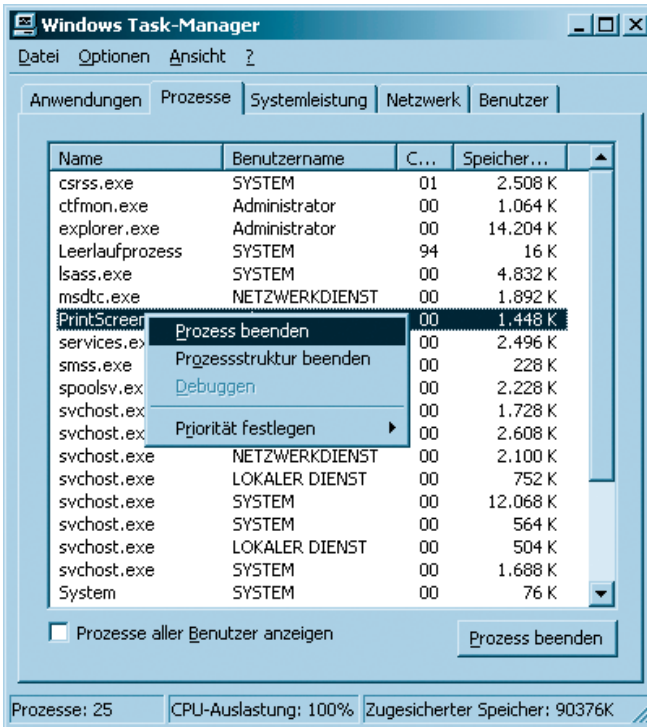


Abbildung 194: Einen Prozess beenden

Um einen Prozess zu beenden, rufen Sie aus dem Kontextmenü der TASK-LEISTE den TASK-MANAGER auf und wechseln auf die Registerkarte PROZESSE. Klicken Sie auf die Spalte NAME, um die Ansicht der Prozesse nach Namen zu sortieren, und markieren Sie den Prozess, den Sie beenden möchten (z.B. *Cmd.exe*, wenn das Eingabeaufforderungsfenster nicht mehr reagiert). Anschließend öffnen Sie das Kontextmenü des Prozesses und wählen den Eintrag *Prozess beenden* aus, woraufhin der Prozess sofort gestoppt und beendet wird.

Falls eine Applikation oder ein Dienst, der mehrere untergeordnete Prozesse zur gleichen Zeit ausführt, nicht mehr reagiert, so können Sie über das Kontextmenü des Prozesses auch die komplette PROZESSSTRUKTUR beenden und anschließend durch Neustarten der Applikation oder des Dienstes wieder neu aufrufen.

Die Lösung per Eingabeaufforderung

Der Windows Server 2003 stellt mehrere Möglichkeiten bereit, womit Sie von der Eingabeaufforderung aus Prozesse beenden können. Im einfachsten Fall wenden Sie den *Taskkill*-Befehl an, dem Sie mit dem Parameter */IM* den Namen des Prozesses übergeben, der beendet werden soll.

```
@Echo off
Taskkill /IM cmd.exe
```

Listing 506: 2_06_018a.bat

Der Taskkill-Befehl erlaubt das Beenden der Prozesse auch durch Angabe der jeweiligen Prozess-ID. Wenn Sie beispielsweise den Prozess mit der Bezeichnung *Cmd.exe*, der die ID 1876 besitzt, beenden möchten, so übergeben Sie dem Taskkill-Befehl mit dem Parameter /PID die entsprechende Prozess-ID.

```
@Echo off
Taskkill /PID 1876
```

Listing 507: 2_06_018b.bat

Die aktiven Prozesse lassen sich auch per WMI beenden, indem Sie zunächst mit der Where-Klausel das Prozessobjekt selektieren und dann mit Call die Terminate-Methode aufrufen.

```
@Echo off
Wmic Process WHERE "Caption='cmd.exe'" Call Terminate
```

Listing 508: 2_06_018c.bat

Die Lösung mit VBScript

Die Terminate-Methode der WMI-Klasse Win32_Process können Sie auch in einem VBScript verwenden, um einen Prozess zu beenden. Hierzu stellen Sie mit der GetObject-Methode eine Verbindung zum WMI-Namespaces `root\cimv2` des betreffenden Computersystems her und selektieren mit Hilfe der ExecQuery-Abfrage das betreffende Prozessobjekt, indem Sie mit der Where-Klausel den Namen (Caption) oder die Prozess-ID (ProcessID) abfragen. Die ExecQuery-Abfrage gibt daraufhin das Prozessobjekt (sofern gefunden) als Array-Element zurück. In einer For Each-Schleife können Sie dann die Terminate-Methode gegen das Prozessobjekt anwenden, um den Prozess zu beenden.

```
Option Explicit

Dim oWMI
Dim colProcess
Dim Process
Dim strComputer
```

Listing 509: 2_06_018.vbs

```

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colProcess = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Process WHERE Caption='cmd.exe'")

For Each Process in colProcess
    Process.Terminate()
Next

Set oWMI = Nothing

```

Listing 509: 2_06_018.vbs (Forts.)

145 Priorität eines Prozesses festlegen

Für den Fall, dass Sie einen komplexen Prozess gestartet haben und dem Prozess etwas mehr CPU-Leistung zuweisen möchten, so können Sie dem Prozess mit Hilfe des TASK-MANAGERS eine höhere Priorität zuteilen und in vielen Fällen die Ausführungsgeschwindigkeit des Prozesses etwas erhöhen. Allerdings geht dies natürlich zu Lasten der anderen Prozesse, da die CPU-Leistung in diesem Fall nur neu aufgeteilt wird.

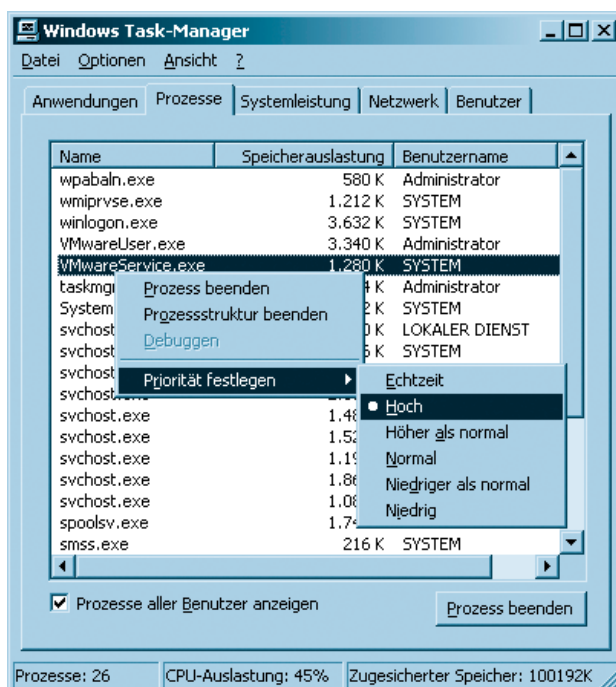


Abbildung 195: Die Priorität festlegen

Um die Priorität eines Prozesses zu erhöhen (oder wahlweise auch zu verringern), rufen Sie zunächst den TASK-MANAGER aus dem Kontextmenü der TASK-LEISTE auf und wechseln auf die Registerkarte PROZESSE. Anschließend markieren Sie den betreffenden Prozess und wählen im Kontextmenü erst PRIORITÄT FESTLEGEN und dann die gewünschte Priorität (*Niedrig*, *Niedriger als normal*, *Normal*, *Höher als normal*, *Hoch* oder *Echtzeit*) aus (siehe Abbildung 195).

Mit der Auswahl einer anderen Priorität wird dem Prozess (je nach Auswahl) mehr oder weniger CPU-Leistung zugesprochen, wobei diese Änderung aber nur für die derzeitige Ausführungsphase gilt. Wenn Sie den Prozess beenden und neu starten, wird dem Prozess wieder die Standardpriorität (*Normal* oder *Hoch*) zugewiesen.

Die Lösung per Eingabeaufforderung

Um von der Eingabeaufforderung aus die Priorität eines Prozesses zu verändern, nutzen Sie der Einfachheit halber den Wmic-Befehl, da Sie mit Hilfe des Alias `Process` auf die WMI-Klasse `Win32_Process` zugreifen und dadurch die `SetPriority`-Methode anwenden können, der Sie nur den Wert der gewünschten Priorität (siehe nachfolgende Tabelle) übergeben müssen.

Priorität	Wert
Niedrig	64
Niedriger als normal	16384
Normal	32
Höher als normal	32768
Hoch	128
Echtzeit	256

Tabelle 65: Die Prozessprioritäten und ihre Werte

Um beispielsweise die Priorität des Prozesses `VMWareService.exe` von *Hoch* auf *Normal* zu ändern, wenden Sie den Wmic-Befehl unter Angabe des Alias `Process` an, selektieren den Prozess aufgrund seines Namens (`Name`) oder seiner Prozess-ID (`ProcessID`) und wenden dann die `SetPriority`-Methode mit dem Wert 32 (= *Normal*) an.

```
@Echo off
Wmic Process Where "Name='VMWareService.exe'" _
    Call SetPriority 32
```

Listing 510: 2_06_019.bat

Die Lösung mit VBScript

Auf dem gleichen Weg, wie Sie die Priorität eines Prozesses von der Eingabeaufforderung aus ändern, legen Sie die gewünschte Priorität auch per VBScript fest. Sie nutzen WMI und wenden die `SetPriority`-Methode gegen das Prozessobjekt an, der Sie den dezimalen Wert der gewünschten Priorität (siehe Tabelle 65: Die Prozessprioritäten und ihre Werte auf Seite 618) übergeben.

Um beispielsweise die Priorität des Prozesses `VMWareService.exe` von *Hoch* auf *Normal* herunterzustufen, stellen Sie mit der `GetObject`-Methode erst eine Verbindung zum WMI-Namespacespace `root\cimv2` des betreffenden Computersystems her, selektieren dann mit `ExecQuery` das Prozessobjekt und wenden anschließend die `SetPriority`-Methode gegen das Prozessobjekt an, wobei Sie der `SetPriority`-Methode den dezimalen Wert der entsprechenden Priorität übergeben.

Option Explicit

```
Dim oWMI
Dim colProcess
Dim Process
Dim strComputer
Dim intPriority

intPriority = 32
strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colProcess = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Process WHERE Caption='cmd.exe'")

For Each Process in colProcess
    Process.SetPriority(intPriority)
Next

Set oWMI = Nothing
```

Listing 511: 2_06_019.vbs

146 Welche Dienste sind installiert?

Um unter grafischer Oberfläche herauszufinden, welche Dienste auf einem bestimmten Computersystem installiert sind, wechseln Sie lediglich in die Übersicht der Dienste des betreffenden Computersystems und lassen sich die einzelnen Dienste anzeigen.

Öffnen Sie hierzu das STARTMENÜ, erweitern Sie VERWALTUNG, und klicken Sie anschließend auf DIENSTE, oder klicken Sie im STARTMENÜ auf AUSFÜHREN, und geben Sie `Services.msc` ein, um die Übersicht der Dienste direkt aufzurufen. Sie gelangen daraufhin in die Dienste-Übersicht und können sich die einzelnen Dienste, die gerade installiert sind, anzeigen lassen.

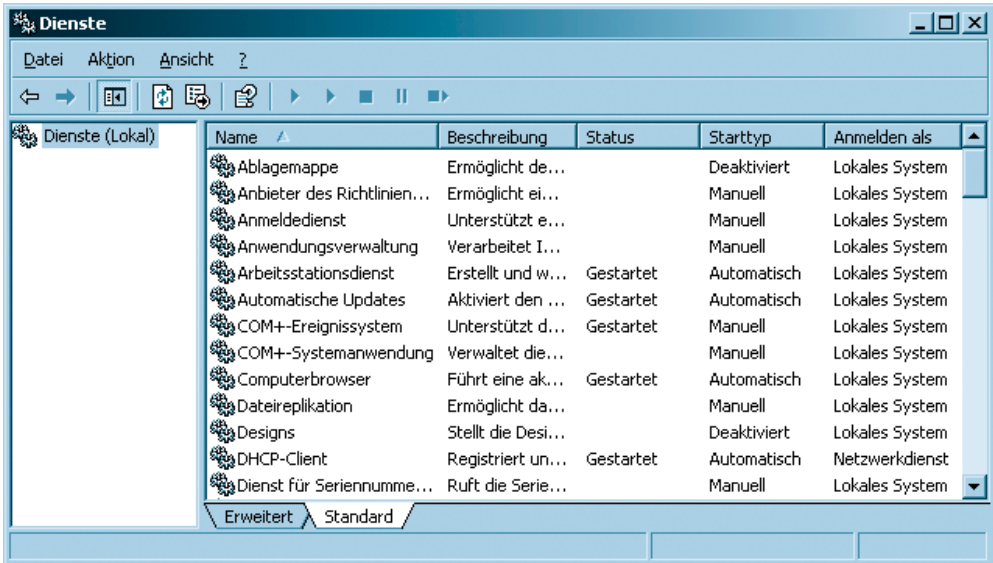


Abbildung 196: Die Übersicht der Dienste

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung haben Sie mehrere Möglichkeiten, sich die installierten Dienste eines bestimmten Computersystems anzeigen zu lassen. Wenn Sie beispielsweise nur eine Auflistung der Dienste erzeugen wollen, die gerade gestartet sind, so geben Sie den Befehl `Net Start` ein, woraufhin die Namen der gestarteten Dienste ausgegeben werden. Das nachfolgende Beispiel liest z.B. die gestarteten Dienste des aktuellen Computersystems aus und schreibt das Ergebnis per Ausgabeumleitung in die Datei `Services.txt`:

```
@Echo off
Net Start > Services.txt
```

Listing 512: 2_06_020a.bat

Während der `Net Start`-Befehl lediglich die Bezeichnungen der gestarteten Dienste ausgibt, können Sie sich mit dem `Sc Query`-Befehl die Bezeichnung, den vom System verwendeten Namen wie auch Typ, Status und Exit-Codes der gestarteten Dienste anzeigen lassen.


```
@Echo off  
Sc query > Services.txt
```

Listing 513: 2_06_020b.bat

Möchten Sie dagegen eine Liste aller Dienste und damit unabhängig von ihrem Status (*Gestartet* oder *Beendet*) erzeugen, so setzen Sie den WMIC-Befehl ein und führen eine Abfrage gegen die WMI-Klasse `Win32_Service` durch, die Informationen zu den einzelnen Diensten besitzt.

Der Befehl `WMIC Service` liest ohne Angabe weiterer Parameter alle Dienstinformationen (siehe *Teil III* dieses Buch) der installierten Dienste aus und gibt das Ergebnis am Bildschirm aus. Wenn Sie nur bestimmte Eigenschaften der Dienste auslesen möchten, so können Sie die Ausgabe mit Hilfe der `Get`-Anweisung auf bestimmte Eigenschaften beschränken. Um beispielsweise nur die Bezeichnung und den internen Namen aller Dienste auszugeben und das Ergebnis in die Datei *Services.txt* umzuleiten, verwenden Sie folgende Befehlszeile:

```
@Echo off  
WMIC Service Get Caption, Name > Services.txt
```

Listing 514: 2_06_020c.bat

Mit Hilfe einer Abfrage lassen sich auch nur die Informationen bestimmter Dienste auslesen. Wenn Sie z.B. alle Eigenschaften des Dienstes `ABLAGEMAPPE` auflisten möchten, so führen Sie folgende Abfrage aus:

```
@Echo off  
WMIC Service Where Caption="Ablagemappe" > Services.txt
```

Listing 515: 2_06_020d.bat

Die Lösung mit VBScript

Mit VBScript lassen sich die installierten Dienste eines Computersystems ebenfalls auslesen, wobei Sie hier ähnlich vorgehen wie in der Eingabeaufforderung. Sie verwenden auch hier eine WMI-Abfrage, um Informationen zu den installierten Diensten zu erhalten.

Das nachfolgende Code-Beispiel zeigt eine solche Lösung: Nach der Deklaration der benötigten Variablen wird mit der `GetObject`-Methode eine Verbindung zum WMI-Namespaces `root\cimv2` des betreffenden Computersystems hergestellt und eine `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_Service` ohne Angabe weiterer Parameter durchgeführt, woraufhin eine Kollektion aller Dienste zurückgeliefert wird. Innerhalb

einer For Each-Schleife werden dann die einzelnen Eigenschaften (siehe *Teil III* dieses Buches) des jeweiligen Elements ausgelesen und per WScript.Echo-Anweisung am Bildschirm ausgegeben.

```
Option Explicit
```

```
Dim oWMI
Dim colServices
Dim Service
Dim strComputer
```

```
strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colServices = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Service")

For Each Service in colServices
    Wscript.Echo "AcceptPause: " & Service.AcceptPause
    Wscript.Echo "AcceptStop: " & Service.AcceptStop
    Wscript.Echo "Caption: " & Service.Caption
    Wscript.Echo "CheckPoint: " & Service.CheckPoint
    Wscript.Echo "CreationClassName: " & _
        Service.CreationClassName
    Wscript.Echo "Description: " & Service.Description
    Wscript.Echo "DesktopInteract: " & Service.DesktopInteract
    Wscript.Echo "DisplayName: " & Service.DisplayName
    Wscript.Echo "ErrorControl: " & Service.ErrorControl
    Wscript.Echo "ExitCode: " & Service.ExitCode
    Wscript.Echo "InstallDate: " & Service.InstallDate
    Wscript.Echo "Name: " & Service.Name
    Wscript.Echo "PathName: " & Service.PathName
    Wscript.Echo "ProcessID: " & Service.ProcessID
    Wscript.Echo "ServiceSpecificExitCode: " & _
        Service.ServiceSpecificExitCode
    Wscript.Echo "ServiceType: " & Service.ServiceType
    Wscript.Echo "Started: " & Service.Started
    Wscript.Echo "StartMode: " & Service.StartMode
    Wscript.Echo "StartName: " & Service.StartName
    Wscript.Echo "State: " & Service.State
    Wscript.Echo "Status: " & Service.Status
    Wscript.Echo "SystemCreationClassName: " & _
        Service.SystemCreationClassName
    Wscript.Echo "SystemName: " & Service.SystemName
    Wscript.Echo "TagID: " & Service.TagID
    Wscript.Echo "WaitHint: " & Service.WaitHint
```

Next

Set oWMI = Nothing

Listing 516: 2_06_020.vbs (Forts.)

147 Neue Dienste registrieren

Der Windows Server 2003 ermöglicht zwar das Erstellen neuer Dienste von der Eingabeaufforderung aus und auch per VBScript, aber leider nicht unter grafischer Oberfläche. Falls Sie aber im Besitz des *Windows 2000 Resource Kits* sind, so können Sie hierfür das Tool *Srvinstw.exe* verwenden, womit Sie auf recht einfache Art und Weise auch per GUI einen neuen Dienst registrieren können.

Um beispielsweise das Tool *Srvany.exe*, das Bestandteil des *Windows Server 2003 Resource Kits* ist, als Dienst zu registrieren und damit auch gewöhnliche Applikationen als Hintergrunddienst ausführen lassen zu können, starten Sie zunächst das Tool *Srvinstw.exe* mit einem Doppelklick und wählen im ersten Schritt die Option **INSTALL A SERVICE** aus. Anschließend legen Sie fest, ob Sie den Dienst auf dem lokalen Computersystem (LOCAL MACHINE) oder aber auf einem Remote-System (REMOTE MACHINE) registrieren möchten, und geben im nächsten Step einen aussagekräftigen Namen für den neuen Dienst (z.B. *SrvAny*) ein.

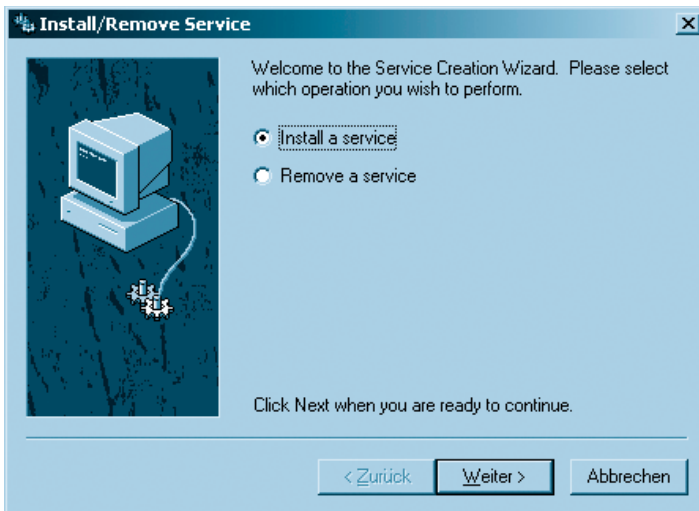


Abbildung 197: Einen neuen Dienst registrieren

Im nächsten Schritt wählen Sie die Datei *Srvany.exe* aus, übernehmen daraufhin die Standardeinstellung **SERVICE IS ITS OWN PROCESS** und legen den Benutzerkontext fest, unter dessen Verwendung der Dienst ausgeführt werden soll. Wenn der neue Dienst mit dem Dienst-Account von Windows Server 2003 gestartet werden soll, so über-

nehmen Sie die Standardeinstellung `SYSTEM ACCOUNT` und bestimmen zu guter Letzt nur noch den Starttyp (*Automatic, Manual, Disabled*). Da neue Dienste per Default nicht gestartet werden, rufen Sie nach der Einrichtung des neuen Dienstes die Übersicht der Dienste auf und starten den soeben eingerichteten Dienst von Hand, um die Funktionen des Dienstes nutzen zu können.

Die Lösung per Eingabeaufforderung

Um einen neuen Dienst per Eingabeaufforderung zu registrieren, können Sie den `Sc`-Befehl mit der Option `Create` verwenden, der zumindest den Namen des Dienstes wie auch den Pfad zur ausführbaren Datei als Parameter erwartet.

Mit dem zusätzlichen Parameter `Type=` können Sie zeitgleich auch den Diensttyp (*Own, Share, Interact, Filesys* oder *Rec*) bestimmen. Den Startmodus (*Boot, System, Auto, Demand, Disabled*) legen Sie dagegen mit dem Parameter `Start=` fest. Um einen Anzeigenamen festzulegen, verwenden Sie den Parameter `Display=` und übergeben die gewünschte Bezeichnung.

Das nachfolgende Beispiel legt auf dem aktuellen Computersystem einen neuen Dienst mit dem Anzeige- und Dienstnamen `MyService` an, bestimmt den Diensttyp `Own` und den Startmodus `Auto` und verwendet die Datei `C:\Windows\System32\MyService.exe` als Executable für den Dienst.

```
@Echo off
Sc Create MyService Type= Own Start= Auto _
    BinPath= C:\Windows\System32\MyService.exe _
    DisplayName= MyService
```

Listing 517: 2_06_021.bat

Die Lösung mit VBScript

Die WMI-Klasse `Win32_BaseService` stellt mit der `Create`-Methode eine Möglichkeit bereit, womit Sie einen neuen Dienst auch per VBScript einrichten können. Die `Create`-Methode erwartet dabei die Angabe der dienstspezifischen Argumente, die Sie mit dem Methodenaufruf in der angegebenen Reihenfolge übergeben müssen:

Position	Argument	Beschreibung
1	Name	Name des Dienstes
2	DisplayName	Anzeigenname des Dienstes
3	PathName	Pfad zur ausführbaren Datei des Dienstes

Tabelle 66: Die Argumente der Create-Methode

Position	Argument	Beschreibung
4	ServiceType	Typ des Dienstes (1 = Kernaltreiber, 2 = Dateisystemtreiber, 4 = Adaptertreiber, 8 = vom Dateisystem erkannter Treiber, 16 = eigener Prozess, 32 = freigegebener Prozess, 256 = interaktiver Prozess)
5	ErrorControl	Fehlerkontrolle des Dienstes (0 = Ignorieren, 1 = Normal, 2 = Schwerwiegend, 3 = Kritisch)
6	StartMode	Starttyp des Dienstes (Boot, System, Automatic, Manual, Disabled)
7	DesktopInteract	Interaktion mit dem Desktop (True, False)
8	StartName	Benutzername, unter dessen Verwendung der Dienst ausgeführt werden soll
9	StartPassword	Kennwort des Benutzers, unter dessen Verwendung der Dienst ausgeführt werden soll
10	LoadOrderGroup	Ladegruppe des Dienstes
11	LoadOrderGroupDependencies	Vom Dienst abhängige Ladegruppe(n)
12	ServiceDependencies	Abhängige Dienste

Tabelle 66: Die Argumente der Create-Methode (Forts.)

Um beispielsweise einen neuen Dienst mit der Bezeichnung *MyService* einzurichten, stellen Sie mit der `GetObject`-Methode eine Verbindung zum WMI-Namespace `root\cimv2` des betreffenden Computersystems her und setzen den `ImpersonationLevel` auf `Impersonate`. Anschließend wenden Sie die `Get`-Methode unter Angabe der `Win32_BaseService`-Klasse an, um eine Verbindung zu der angegebenen Klasse herzustellen.

Danach setzen Sie die `Create`-Methode ein, um das neue Dienstobjekt anzulegen, wobei Sie mit dem Methodenaufruf die dienstspezifischen Argumente `Name`, `DisplayName`, `PathName`, `ServiceType`, `ErrorControl`, `StartMode`, `DesktopInteract` und `StartName` übergeben.

```
Option Explicit

Dim oWMI
Dim oService
Dim strComputer

Const OWN_PROCESS = 16
Const NOT_INTERACTIVE = True
strComputer = "."
```

Listing 518: 2_06_021.vbs

```

Set oWMI = GetObject("winmgmts:" & " impersonationLevel=_
    impersonate)!\" & strComputer & "\root\cimv2")
Set oService = oWMI.Get("Win32_BaseService")

Err = oService.Create ("MyService", "MyService", _
    "C:\Windows\System32\MyService.exe", OWN_PROCESS, _
    2,"Automatic", NOT_INTERACTIVE, ".\LocalSystem" ,")

Set oWMI = Nothing
Set oService = Nothing

```

Listing 518: 2_06_021.vbs (Forts.)

148 Vorhandene Dienste löschen

Mit der Voraussetzung, dass ein bestimmter Dienst nicht mehr benötigt wird und auch nicht in Abhängigkeiten zu anderen Diensten steht, so können Sie den nicht benötigten Dienst aus der Systemkonfiguration löschen.

Der Windows Server 2003 stellt unter grafischer Oberfläche keine Möglichkeit bereit, womit Sie einen Dienst so ohne weiteres löschen können. Falls Sie aber im Besitz des Windows 2000 Resource Kits sind, so können Sie das Tool *Srvinstw.exe* verwenden, um den Dienst auch unter grafischer Oberfläche zu löschen. Anderenfalls nutzen Sie die Möglichkeiten der Eingabeaufforderung.

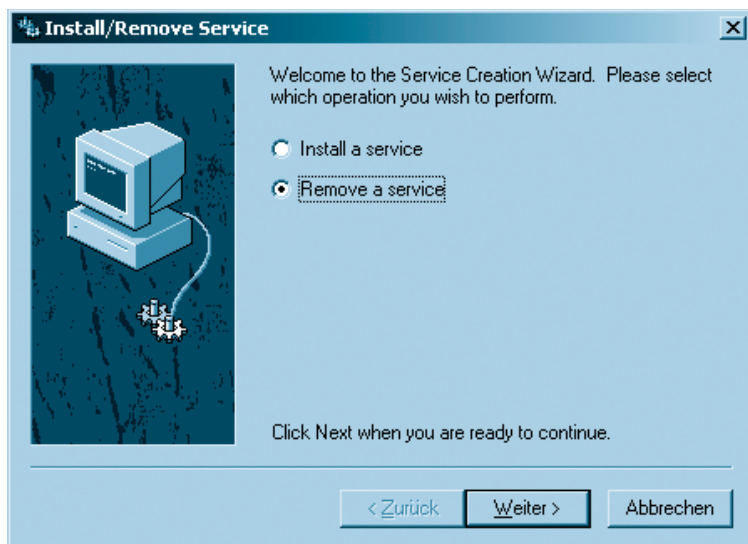


Abbildung 198: Einen Dienst entfernen

Starten Sie das Tool *Srvinstw.exe*, und wählen Sie im ersten Dialogfenster die Option REMOVE A SERVICE aus. Im nächsten Schritt geben Sie an, ob der Dienst auf dem lokalen Computersystem (LOCAL MACHINE) oder auf einem Remote-System (REMOTE MACHINE) gelöscht werden soll, und geben bei Auswahl der Option REMOTE MACHINE den betreffenden Computernamen ein. Und zu guter Letzt wählen Sie den Dienst aus, den Sie aus der Systemkonfiguration löschen möchten, und klicken auf FERTIG STELLEN, woraufhin der Dienst endgültig gelöscht wird.

Die Lösung per Eingabeaufforderung

Um einen vorhandenen Dienst von der Eingabeaufforderung aus zu löschen, wenden Sie den *Sc*-Befehl mit der Option *Delete* an und übergeben den internen Dienstnamen als Parameter. Der angegebene Dienst wird daraufhin ohne jegliche Rückfrage vom Computersystem gelöscht; und zwar unabhängig vom Status (*Gestartet* oder *Beendet*) des Dienstes.

```
@Echo off
Sc Delete MyService
```

Listing 519: 2_06_022.bat

Die Lösung mit VBScript

Ein vorhandener Dienst lässt sich natürlich auch per VBScript löschen. Die WMI-Klasse *Win32_Service* stellt hierfür die *Delete*-Methode bereit, die Sie ohne Angabe weiterer Argumente gegen das entsprechende Service-Objekt anwenden können.

Das nachfolgende Code-Beispiel zeigt eine solche Lösung. Im ersten Schritt wird per *ExecQuery*-Abfrage das Service-Objekt mit der Bezeichnung *MyService* selektiert. Da die *ExecQuery*-Abfrage das Ergebnis als Array-Element zurückliefert, erfolgt der Zugriff auf das Service-Objekt mit Hilfe einer *For Each*-Schleife. Innerhalb dieser *For Each*-Schleife wird der Service mit der *StopService*-Methode zunächst beendet und anschließend mit der *Delete*-Methode gelöscht.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colService
Dim Service

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
```

Listing 520: 2_06_022.vbs

```
Set colService = oWMI.ExecQuery ("SELECT * FROM _
Win32_Service WHERE Name = 'MyService'")

For Each Service in colService
    Service.StopService()
    Service.Delete()
Next

Set oWMI = Nothing
```

Listing 520: 2_06_022.vbs (Forts.)

149 Dienste anhalten und fortsetzen

Einige Dienste sind in der Lage, die Steuerungsanfrage *Pause* entgegenzunehmen, womit Sie den betreffenden Dienst kurzfristig anhalten und zu einem späteren Zeitpunkt wieder fortsetzen können. Dies hat gegenüber dem Beenden eines Dienstes den Vorteil, dass die aktiven Dienstanforderungen weiterhin ausgeführt werden können, da nur neue Dienstanforderungen abgelehnt werden.

System-
management

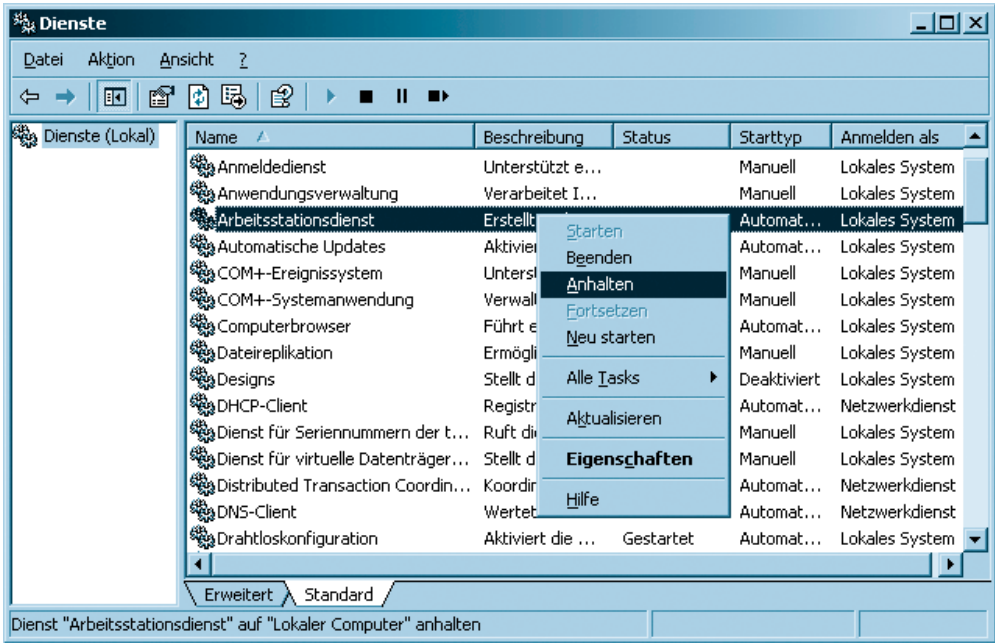


Abbildung 199: Einen Dienst anhalten

Um einen Dienst kurzfristig anzuhalten, wechseln Sie zunächst in die Übersicht der Dienste, die Sie im STARTMENÜ, genauer gesagt unter VERWALTUNG, öffnen oder mit Services.msc auch direkt aufrufen können. Markieren Sie den betreffenden Dienst,

und öffnen Sie dessen Kontextmenü. Wenn der Dienst die Steuerungsanfrage *Pause* akzeptiert, so finden Sie im Kontextmenü den Eintrag *ANHALTEN* vor. Wählen Sie den Eintrag aus, um den Dienst kurzfristig anzuhalten. Die aktiven Dienstanforderungen (z.B. Client-Requests) werden nach wie vor noch ausgeführt, neue Dienstanforderungen werden dagegen abgelehnt.

Wenn Sie den Dienst zu einem späteren Zeitpunkt fortsetzen möchten, so rufen Sie wieder die Übersicht der Dienste auf, öffnen das Kontextmenü des betreffenden Dienstes und wählen *FORTSETZEN* aus, woraufhin der Dienst wieder neue Dienstanfragen entgegennimmt.

Die Lösung per Eingabeaufforderung

Der Status eines Dienstes lässt sich auch per Eingabeaufforderung verändern. Hierfür stellt der Windows Server 2003 die Befehle *Net.exe* und *Sc.exe* bereit, die Sie unter anderem auch zum Anhalten eines Dienstes verwenden können. Um beispielsweise den Taskplaner-Dienst aus einer Batch-Datei heraus anzuhalten, können Sie den *Net*-Befehl mit der Option *Pause* verwenden, dem Sie lediglich den Namen des Dienstes als Parameter übergeben.

```
@Echo off
Net Pause Taskplaner
```

Listing 521: 2_06_023a.bat

Die gleiche Möglichkeit bietet auch der *Sc*-Befehl unter Verwendung der Option *Pause*, der im Gegensatz zum *Net*-Befehl allerdings den internen Dienstnamen als Parameter erwartet:

```
@Echo off
Sc Pause Schedule
```

Listing 522: 2_06_023b.bat

Um die Ausführung des Dienstes fortzusetzen, wenden Sie den *Net*-Befehl mit der Option *Continue* an und übergeben den Namen des Dienstes:

```
@Echo off
Net Continue Taskplaner
```

Listing 523: 2_06_023c.bat

Oder aber Sie setzen den *Sc*-Befehl mit der Option *Continue* ein und übergeben den internen Dienstnamen.

```
@Echo off
Sc Continue Schedule
```

Listing 524: 2_06_023d.bat

Die Lösung mit VBScript

Wenn Sie per VBScript einen Dienst anhalten möchten, so können Sie die `PauseService`-Methode der WMI-Klasse `Win32_Service` verwenden. Hierfür selektieren Sie das Dienstobjekt, wofür im nachfolgenden Beispiel die `ExecQuery`-Abfrage unter Angabe des internen Dienstnamens verwendet wird. Anschließend wenden Sie die `PauseService`-Methode gegen das Dienstobjekt an, woraufhin der Dienst sofort angehalten wird.

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colService
Dim Service

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colService = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Service WHERE Name = 'Lanmanworkstation'")

For Each Service in colService
    Service.PauseService()
Next

Set oWMI = Nothing
```

Listing 525: 2_06_023a.vbs

Wenn Sie den Dienst zu einem späteren Zeitpunkt wieder fortsetzen möchten, so können Sie hierfür die gleichen Code-Zeilen verwenden. Nur mit dem Unterschied, dass Sie zum Fortsetzen des Dienstes die `ResumeService`-Methode gegen das Dienstobjekt anwenden und nicht die `PauseService`-Methode.

```
Option Explicit

Dim oWMI
```

Listing 526: 2_06_023b.vbs

```
Dim strComputer
Dim colService
Dim Service

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=impersonate}!\" _
    & strComputer & "\root\cimv2")
Set colService = oWMI.ExecQuery ("SELECT * FROM Win32_Service WHERE _
    Name = 'Lanmanworkstation'")

For Each Service in colService
    Service.ResumeService()
Next

Set OWMi = Nothing
```

Listing 526: 2_06_023b.vbs (Forts.)

150 Dienste starten und beenden

Mit der Installation von Windows Server 2003 wird zwar eine ganze Reihe nützlicher Dienste eingerichtet, allerdings werden aus Sicherheits- und Performance-Gründen nicht alle Dienste automatisch gestartet. Um die Funktionen eines beendeten Dienstes nutzen zu können, müssen Sie den entsprechenden Dienst erst starten.

Hierzu rufen Sie die Übersicht der Dienste auf, indem Sie das STARTMENÜ öffnen, das Untermenü VERWALTUNG erweitern und DIENSTE auswählen oder aber im STARTMENÜ auf AUSFÜHREN klicken und *Services.msc* eingeben. In der Übersicht der Dienste markieren Sie den Dienst, den Sie starten möchten, und rufen aus dem Kontextmenü heraus die EIGENSCHAFTEN auf. Auf der Registerkarte ALLGEMEIN prüfen Sie den Starttyp (*Deaktiviert*, *Manuell*, *Automatisch*) und wählen *Manuell* oder *Automatisch* aus, falls der Dienst deaktiviert sein sollte. Anschließend klicken Sie auf STARTEN, um den Dienst zu starten (siehe Abbildung 200).

Wenn Sie den Dienst zu einem späteren Zeitpunkt wieder beenden möchten, so wechseln Sie wieder in die Übersicht der Dienste, markieren den entsprechenden Dienst und wählen im Kontextmenü BEENDEN aus.

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung können Sie den *Net-* und *Sc-*Befehl verwenden, um einzelne Dienste zu starten und wahlweise auch zu beenden. Wenn Sie beispielsweise den Dienst *Arbeitsstationsdienst* starten möchten, so können Sie den *Net-*Befehl mit der Option *Start* verwenden, dem Sie lediglich den Anzeigenamen des Dienstes als Parameter übergeben müssen.

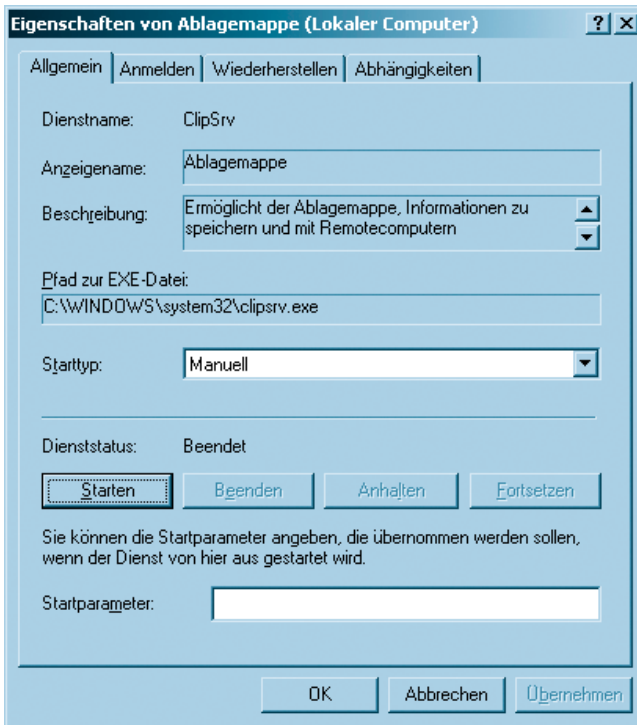


Abbildung 200: Einen Dienst starten

```
@Echo off
Net Start Arbeitsstationsdienst
```

Listing 527: 2_06_024a.bat

Gleiches lässt sich auch mit dem `Sc`-Befehl realisieren, der ebenfalls die Option `Start` erfordert, um einen Dienst zu starten. Im Gegensatz zum `Net`-Befehl benötigt der `Sc`-Befehl allerdings den internen Dienstnamen, den Sie mit dem Befehlsaufruf als Parameter übergeben müssen.

```
@Echo off
Sc Start Lanmanworkstation
```

Listing 528: 2_06_024b.bat

Um einen Dienst wieder zu beenden, verwenden Sie den `Net`-Befehl mit der Option `Stop` und übergeben den Anzeigennamen des entsprechenden Dienstes als Parameter.

```
@Echo off
Net Stop Arbeitsstationsdienst
```

Listing 529: 2_06_024c.bat

Oder Sie setzen den Sc-Befehl mit der Option Stop ein und übergeben den internen Dienstnamen, womit Sie einen Dienst ebenfalls beenden können.

```
@Echo off
Sc Stop Lanmanworkstation
```

Listing 530: 2_06_024d.bat

Die Lösung mit VBScript

Die WMI-Klasse Win32_Service stellt die Methoden StartService und StopService bereit, womit Sie einzelne Dienste starten und auch wieder beenden können. Um diese Methoden anwenden zu können, stellen Sie per GetObject eine Verbindung zum WMI-Namespace root\cimv2 des betreffenden Computersystems her und selektieren das entsprechende Dienstobjekt. Anschließend wenden Sie die StartService- bzw. StopService-Methode gegen das Dienstobjekt an, woraufhin der Dienst gestartet bzw. beendet wird.

Das nachfolgende Code-Beispiel zeigt, wie Sie mit Hilfe der StartService-Methode der Win32_Service-Klasse den Arbeitsstationsdienst (interner Dienstname: Lanmanworkstation) starten können:

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colService
Dim Service

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\\" & strComputer & "\root\cimv2")
Set colService = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Service WHERE Name = 'Lanmanworkstation'")

For Each Service in colService
    Service.StartService()
Next

Set oWMI = Nothing
```

Listing 531: 2_06_024a.vbs

Auf ähnliche Art und Weise lässt sich der Arbeitsstationsdienst auch wieder beenden. Nur eben mit dem Unterschied, dass Sie zum Beenden eines Dienstes die `StopService`-Methode gegen das entsprechende Dienstobjekt anwenden:

```
Option Explicit

Dim oWMI
Dim strComputer
Dim colService
Dim Service

strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\cimv2")
Set colService = oWMI.ExecQuery ("SELECT * FROM _
    Win32_Service WHERE Name = 'Lanmanworkstation'")

For Each Service in colService
    Service.StopService()
Next

Set oWMI = Nothing
```

Listing 532: 2_06_024b.vbs

151 Eigenschaften der Dienste ändern

Mit Hilfe der Diensteigenschaften lassen sich bestimmte Verhaltensweisen eines Dienstes beeinflussen. So können Sie in den Eigenschaften eines Dienstes beispielsweise den `STARTTYP` festlegen und damit bestimmen, ob ein Dienst deaktiviert, manuell oder aber mit dem Systemstart automatisch gestartet werden soll. Des Weiteren können Sie in den Eigenschaften eines Dienstes den Kontext (System- oder bestimmtes Benutzerkonto) bestimmen, unter dessen Verwendung der Dienst ausgeführt wird. Und nicht zuletzt können Sie in den Diensteigenschaften auch die Wiederherstellungsparameter definieren und damit festlegen, wie das Betriebssystem auf einen auftretenden Dienstfehler reagieren soll (z.B. *Dienst neu starten*, *Computer starten ...*).

Um beispielsweise den Starttyp des Dienstes `AUTOMATISCHE UPDATES` von *Deaktiviert* auf *Automatisch* zu ändern und den Wiederherstellungsparameter *Dienst neu starten* zu setzen, rufen Sie zunächst die Übersicht der Dienste auf. Hierzu öffnen Sie das `STARTMENÜ`, erweitern `VERWALTUNG` und klicken auf `DIENTE` oder wählen im `STARTMENÜ` den Eintrag `AUSFÜHREN` aus und geben `Services.msc` ein. Anschließend markieren Sie den Dienst `AUTOMATISCHE UPDATES` und wählen im Kontextmenü des Dienstes den Eintrag `EIGENSCHAFTEN` aus.

Auf der Registerkarte `ALLGEMEIN` im Feld `STARTTYP` wählen Sie den Eintrag *Automatisch* aus, womit der Dienst aktiviert und beim Systemstart automatisch gestartet

wird. Damit der Dienst sofort (und nicht erst mit dem nächsten Systemstart) gestartet wird, betätigen Sie zusätzlich die Schaltfläche **STARTEN**. Anschließend wechseln Sie auf die Registerkarte **WIEDERHERSTELLEN** und wählen im Feld **ERSTER FEHLER** den Eintrag *Dienst neu starten* aus, womit der Dienst automatisch neu gestartet wird, falls es zu einem Fehler bei der Dienstausführung kommt. Mit einem Klick auf **ÜBERNEHMEN** werden die neuen Eigenschaften in die Registrierung geschrieben und damit dauerhaft gespeichert.

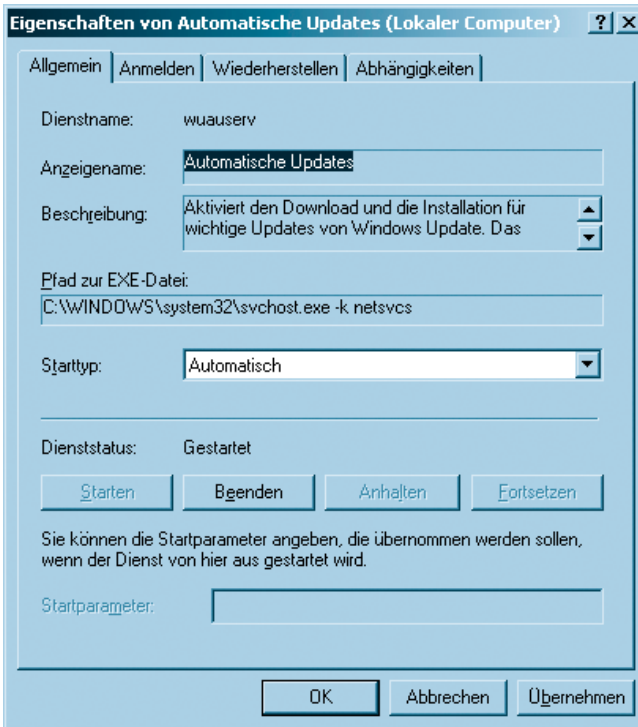


Abbildung 201: Die Eigenschaften eines Dienstes ändern

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung verwenden Sie den `Sc`-Befehl mit der Option `Config`, um die Eigenschaften eines Dienstes zu ändern. Um beispielsweise den Starttyp des Dienstes *Automatische Updates* von *Deaktiviert* auf *Automatisch* zu setzen, wenden Sie den `Sc`-Befehl mit der Option `Config` an, übergeben den internen Dienstnamen und mit dem Parameter `Start=` den Wert `Auto`.

```
@Echo off
Sc Config Wuauserv Start= Auto
```

Listing 533: 2_06_025a.bat

Um dagegen den Anzeigenamen des Dienstes von *Automatische Updates* auf *Update-Service* zu ändern, verwenden Sie folgende Befehlszeile:

```
@Echo off
Sc Config Wuauserv DisplayName= UpdateService
```

Listing 534: 2_06_025b.bat

Die Lösung mit VBScript

Auch mit VBScript ist das Bearbeiten der Diensteigenschaften kein Problem, da die WMI-Klasse Win32_Service, die Informationen zu den einzelnen Diensten besitzt, die Change-Methode zum Ändern der Eigenschaften bereitstellt. Die Change-Methode akzeptiert dabei die folgenden Argumente, die beim Methodenaufruf allerdings auch in der gleichen Reihenfolge bzw. in der gleichen Position angegeben werden müssen:

Position	Argument	Beschreibung
1	DisplayName	Anzeigename des Dienstes
2	PathName	Pfad zur ausführbaren Datei des Dienstes
3	ServiceType	Typ des Dienstes (1 = Kernaltreiber, 2 = Dateisystemtreiber, 4 = Adaptertreiber, 8 = vom Dateisystem erkannter Treiber, 16 = eigener Prozess, 32 = freigegebener Prozess, 256 = interaktiver Prozess)
4	ErrorControl	Fehlerkontrolle des Dienstes (0 = Ignorieren, 1 = Normal, 2 = Schwerwiegend, 3 = Kritisch)
5	StartMode	Starttyp des Dienstes (Boot, System, Automatic, Manual, Disabled)
6	DesktopInteract	Interaktion mit dem Desktop (True, False)
7	StartName	Benutzername, unter dessen Verwendung der Dienst ausgeführt werden soll
8	StartPassword	Kennwort des Benutzers, unter dessen Verwendung der Dienst ausgeführt werden soll
9	LoadOrderGroup	Ladegruppe des Dienstes
10	LoadOrderGroupDependencies	Vom Dienst abhängige Ladegruppe(n)
11	ServiceDependencies	Abhängige Dienste

Tabelle 67: Die Argumente der Change-Methode

Um beispielsweise den Starttyp des Dienstes *Automatische Updates* von *Deaktiviert* auf *Aktiviert* zu verändern, führen Sie eine ExecQuery-Abfrage durch und verwenden die Eigenschaft Name als Kriterium, um das entsprechende Dienstobjekt zu selektieren.

Da das Ergebnis der `ExecQuery`-Abfrage als Kollektion zurückgeliefert wird, wenden Sie die `Change`-Methode innerhalb einer `For Each`-Schleife gegen das entsprechende Dienstobjekt an und übergeben (an Position 5) der Eigenschaft `StartMode` den Wert `Automatic`.

Option Explicit

Dim oWMI

Dim strComputer

Dim colService

Dim Service

strComputer = "."

Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_impersonate}!\" & strComputer & "\root\cimv2")

Set colService = oWMI.ExecQuery ("SELECT * FROM _Win32_Service WHERE Name = 'Wuauserv'")

For Each Service in colService

Err = Service.Change(,,, "Automatic")

Next

Set oWMI = Nothing

Listing 535: 2_06_025.vbs

152 Größe der Auslagerungsdatei verändern

Windows Server 2003 verwendet per Default eine Auslagerungsdatei mit einer dynamischen Größe, wobei die Anfangsgröße das 1,5-fache und die Endgröße das 3,0-fache des installierten Hauptspeichers (in MByte) beträgt. Wenn diese Größe nicht ausreicht (was sich mit Hilfe des Systemmonitors jederzeit herausfinden lässt) und eine maximale Belegung der Auslagerungsdatei vorherrscht, so können Sie die Größe neu definieren und bei Bedarf auch eine statische Größe festlegen, um ein ständiges Re-Sizing der Auslagerungsdatei zu vermeiden.

Hierzu rufen Sie das Systemsteuerungsobjekt `SYSTEM` auf und wechseln auf die Registerkarte `ERWEITERT`. In der Sektion `SYSTEMLEITUNG` betätigen Sie die Schaltfläche `EINSTELLUNGEN`, wechseln dann auf die Registerkarte `ERWEITERT` und klicken anschließend auf `ÄNDERN`, um das Dialogfenster `VIRTUELLER ARBEITSSPEICHER` aufzurufen. Legen Sie in den Feldern `ANFANGSGRÖSSE` und `MAXIMALE GRÖSSE` die gewünschte Größe der Auslagerungsdatei pro vorhandener Festplatte fest (verwenden Sie in beiden Feldern die gleiche Größe, um eine statische Größe zu definieren), und klicken Sie anschließend auf `OK`, um die Änderung zu übernehmen (siehe Abbildung 202). Starten Sie das Computersystem neu, um die geänderte Einstellung wirksam werden zu lassen.

Die Lösung per Eingabeaufforderung

Die Konfigurationseinstellungen der Auslagerungsdatei sind in der Registrierung (Schlüssel HKLM\SYSTEM\CURRENTCONTROLSET\CONTROL\SESSION MANAGER\MEMORY MANAGEMENT) als mehrteilige Zeichenfolge hinterlegt und lassen sich dadurch auch per Eingabeaufforderung verändern, da Sie mit dem `Reg Add`-Befehl auf recht einfache Art und Weise neue Werte in die Registrierung schreiben und so auch vorhandene Werte ersetzen können.

Wenn Sie z.B. die Größeneinstellungen für die Auslagerungsdatei (`C:\Pagefile.sys`) verändern und künftig eine statische Größe von **256 MByte** verwenden möchten, so wenden Sie den `Reg Add`-Befehl unter Angabe des Schlüssels an und übergeben mit dem Parameter `/v` die Bezeichnung des Wertes (`PagingFiles`), mit dem Parameter `/t` den Typ des Wertes (`REG_MULTI_SZ`) und mit dem Parameter `/d` den eigentlichen Wert (`"C:\Pagefile.sys 256 256"`). Und da diese mehrteilige Zeichenfolge bereits existiert, erzwingen Sie das Überschreiben der Zeichenfolge mit dem Parameter `/f`.

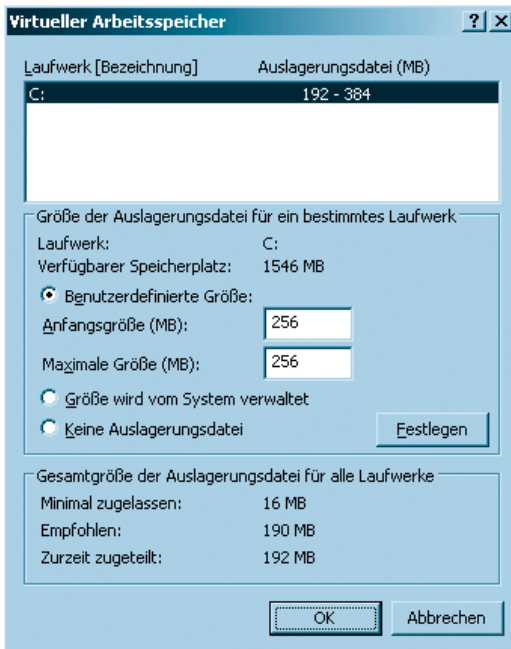


Abbildung 202: Die Größe der Auslagerungsdatei ändern

```
@Echo off
Reg Add "HKLM\SYSTEM\CurrentControlSet\Control\Session _
    Manager\Memory Management" /v PagingFiles _
    /t REG_MULTI_SZ /d "C:\PageFile.sys 256 256" /f
```

Listing 536: 2_06_026.bat

Die Lösung mit VBScript

Die Größe der Auslagerungsdatei lässt sich auch mit VBScript bearbeiten, da Sie nur die Registrierungseinstellungen für die Auslagerungsdatei mit neuen Werten überschreiben müssen. Und hierfür stellt die WMI-Klasse StdRegProv die SetMultiStringValue-Methode bereit, womit Sie eine neue mehrteilige Zeichenfolge in die Registrierung schreiben bzw. eine mehrteilige Zeichenfolge ersetzen können.

Das nachfolgende Code-Beispiel zeigt eine solche Lösung. Dabei wird die SetMultiStringValue-Methode angewendet, um die mehrteilige Zeichenfolge in die Registrierung zu schreiben. Der Methode wird der hexadezimale Wert des Hauptzweiges (HKLM), der Schlüssel (strKey), die Bezeichnung des Wertes (strValueName) und die eigentliche Zeichenfolge (strValue) übergeben, die zu Beginn des Skriptes als Variablen deklariert wurden.

Option Explicit

```
Dim oReg
Dim strComputer
Dim strKey
Dim strValueName
Dim strValue
```

```
Const HKLM = &H80000002
strComputer = "."
strKey = "SYSTEM\CurrentControlSet\Control\Session _
    Manager\Memory Management"
strValueName = "PagingFiles"
strValue = Array("C:\Pagefile.sys 256 256")
```

```
Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\" & strComputer & _
    "\root\default:StdRegProv")
oReg.SetMultiStringValue HKLM, strKey, strValueName, strValue
```

```
Set oReg = Nothing
```

Listing 537: 2_06_026.vbs

153 Umgebungsvariablen auslesen

Mit der Installation des Betriebssystems (und auch durch die Einrichtung einiger Applikationen) werden so genannte Umgebungsvariablen eingerichtet, die bestimmte Systeminformationen und auch Systempfade enthalten. Die Umgebungsvariable NUMBER_OF_PROCESSORS enthält beispielsweise die Anzahl der installierten Prozessoren, die Umgebungsvariable WINDIR den Pfad zum Windows-Ordner und die Umgebungsvariable SYSTEMDRIVE den Root-Pfad des Systemlaufwerks.

Um unter grafischer Oberfläche herauszufinden, welche Umgebungsvariablen eingerichtet sind und welche Werte sie enthalten, öffnen Sie das Systemsteuerungsobjekt SYSTEM, wechseln auf die Registerkarte ERWEITERT und klicken auf die Schaltfläche UMGEBUNGSVARIABLEN. Oder aber Sie lassen sich die SYSTEMINFORMATIONEN anzeigen, indem Sie das STARTMENÜ öffnen, auf AUSFÜHREN klicken und Msinfo32 eingeben. Anschließend erweitern Sie die Sektion SOFTWAREUMGEBUNG und klicken auf UMGEBUNGSVARIABLEN, um eine etwas übersichtlichere Aufstellung der Umgebungsvariablen zu erhalten (siehe Abbildung 203).

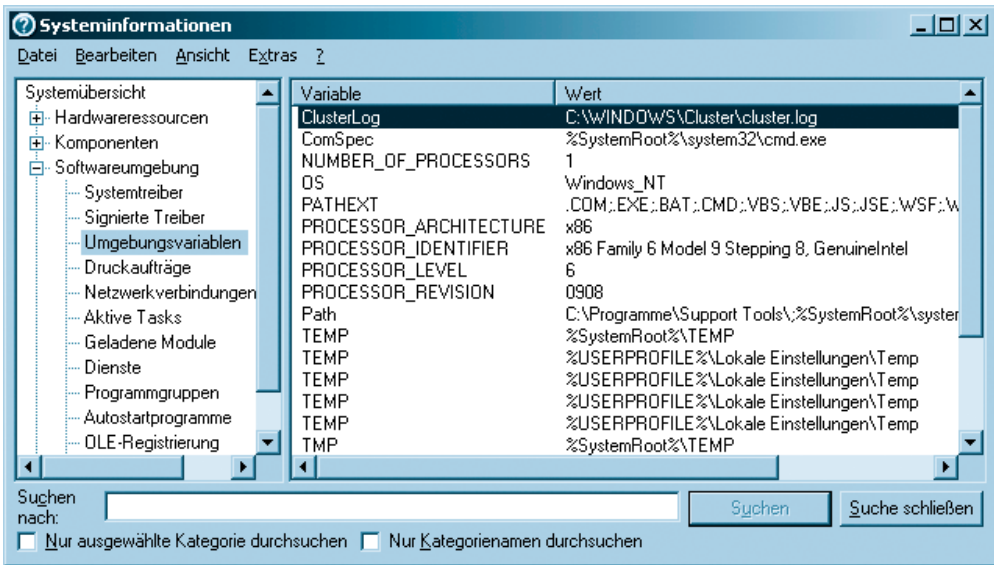


Abbildung 203: Die Anzeige der Umgebungsvariablen

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung können Sie sich die Umgebungsvariablen ebenfalls anzeigen lassen. Im einfachsten Fall verwenden Sie hierfür den Set-Befehl (ohne Angabe weiterer Parameter), der alle system- und benutzerspezifischen Umgebungsvariablen unter Angabe des Namens und des Wertes am Bildschirm ausgibt.

```
@Echo off
Set
```

Listing 538: 2_06_027a.bat

Alternativ dazu können Sie auch eine WMI-Abfrage durchführen, um eine Aufstellung der Umgebungsvariablen zu erhalten. Hierfür wenden Sie den Wmic-Befehl mit dem Alias Environment an, woraufhin die system- und benutzerspezifischen Umgebungsvariablen unter Angabe von Namen, Beschreibung, Status, Variablentyp, Benutzername und Wert am Bildschirm ausgegeben werden.

Das nachfolgende Beispiel liest die system- und benutzerspezifischen Umgebungsvariablen des aktuellen Computersystems aus und schreibt alle Eigenschaften der Umgebungsvariablen in die Datei *Var.txt*:

```
@Echo off
WMIC Environment > Var.txt
```

Listing 539: 2_06_027b.bat

Mit Hilfe einer Abfrage lassen sich die system- und benutzerspezifischen Umgebungsvariablen auch getrennt voneinander ausgeben. Denn die WMI-Klasse *Win32_Environment*, die Informationen zu den einzelnen Umgebungsvariablen besitzt, stellt unter anderem die Eigenschaft *Systemvariable* zur Verfügung, deren Wert gleich *True* ist, wenn es sich um eine systemspezifische Umgebungsvariable handelt. Bei einer benutzerspezifischen Umgebungsvariablen dagegen ist der Wert gleich *False*.

Die nachfolgende *Wmic*-Befehlszeile gibt z.B. alle Umgebungsvariablen unter Angabe des Namens und des Wertes am Bildschirm aus, wenn der Wert der Eigenschaft *Systemvariable* gleich *True* ist:

```
@Echo off
WMIC Environment WHERE "Systemvariable='True'" _
    Get Name, VariableValue
```

Listing 540: 2_06_027c.bat

Die Lösung mit VBScript

Auch mit VBScript ist das Auslesen der system- und benutzerspezifischen Umgebungsvariablen kein Problem, indem Sie eine Verbindung zur WMI-Klasse *Win32_Environment* (WMI-Namespace *root\cimv2*) herstellen, die gewünschten Umgebungsvariablen selektieren und deren Eigenschaften (*Caption*, *Description*, *InstallDate*, *Name*, *Status*, *Systemvariable*, *Username* und *VariableValue*) abfragen.

Das nachfolgende Code-Beispiel zeigt, wie Sie die system- und die benutzerspezifischen Umgebungsvariablen des aktuellen Computersystems (*strComputer = "."*) per WMI abfragen können: Zunächst wird mit der *GetObject*-Methode eine Verbindung zum WMI-Namespace *root\cimv2* hergestellt und der *Impersonation-Level* auf *Impersonate* gesetzt. Anschließend wird eine Abfrage gegen die WMI-Klasse *Win32_Environment* (ohne Angabe eines Suchkriteriums) durchgeführt, woraufhin die Umgebungsvariablen in Form einer Kollektion nach *colVar* übertragen werden. Innerhalb einer *For Each*-Schleife werden die Eigenschaften *Name* und *VariableValue* abgefragt und per *WScript-Echo*-Anweisung am Bildschirm ausgegeben.

```

Option Explicit

Dim oWMI
Dim strcomputer
Dim colVar
Dim var

strComputer = "."

Set oWMI = GetObject("winmgmts:{impersonationLevel=_
    Impersonate}\\\" & strComputer & "\root\cimv2")
Set colVar = oWMI.ExecQuery("SELECT * FROM _
    Win32_Environment")

For Each Var in colVar
    WScript.Echo Var.Name & " = " & Var.VariableValue
Next

Set oWMI = Nothing

```

Listing 541: 2_06_027.vbs

Möchten Sie nicht alle Umgebungsvariablen ausgeben, sondern nur die system- oder benutzerspezifischen Variablen, so können Sie die `ExecQuery`-Abfrage um eine `WHERE`-Klausel erweitern und damit nur die Umgebungsvariablen selektieren, deren Wert der Eigenschaft `Systemvariable` gleich `True` (systemspezifische Umgebungsvariable) oder `False` (benutzerspezifische Umgebungsvariable) ist.

```

Set colVar = oWMI.ExecQuery("SELECT * FROM _
    Win32_Environment WHERE Systemvariable='True'")

```

154 Umgebungsvariablen bearbeiten

Die benutzer- und systemspezifischen Umgebungsvariablen lassen sich jederzeit bearbeiten, wobei Sie die Änderungen aber mit Bedacht vornehmen sollten, da fehlerhafte Umgebungsvariablen die Stabilität des Betriebssystems oder die von Applikationen beeinträchtigen können.

Wenn Sie beispielsweise die systemspezifische Umgebungsvariable `TEMP` bearbeiten und die Variable künftig nicht mehr nach `C:\Windows\Temp`, sondern nach `C:\Temp` verweisen lassen möchten, so rufen Sie zunächst das Systemsteuerungsobjekt `SYSTEM` auf, wechseln auf die Registerkarte `ERWEITERT` und betätigen die Schaltfläche `UMGEBUNGSVARIABLEN`. Sie gelangen daraufhin in das Dialogfenster `UMGEBUNGSVARIABLEN`, worüber Sie im oberen Teil des Dialogfensters die benutzerspezifischen Umgebungsvariablen einsehen und auch bearbeiten können und im unteren Teil die systemspezifischen Variablen.

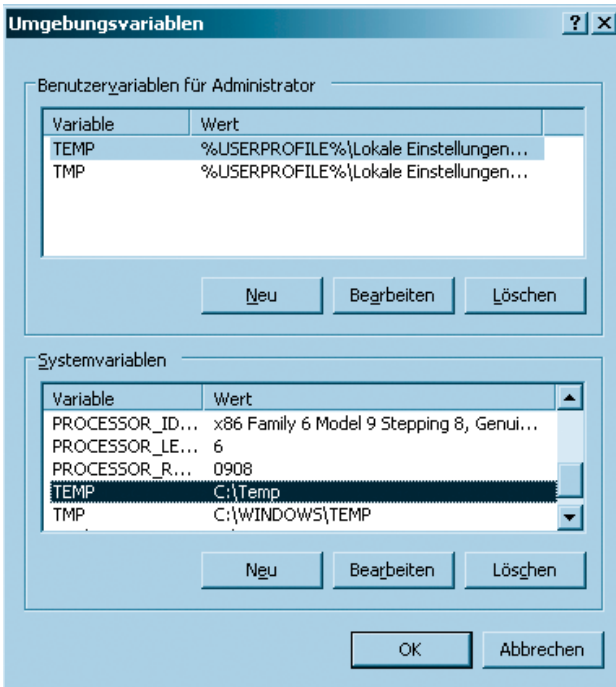


Abbildung 204: Die Systemvariable Temp bearbeiten

Um die systemspezifische Variable TEMP zu modifizieren, markieren Sie im unteren Teil des Dialogfensters den Eintrag TEMP und klicken auf BEARBEITEN. Daraufhin wird das Dialogfenster SYSTEMVARIABLE BEARBEITEN geöffnet, und Sie können im Feld WERT DER VARIABLEN einen neuen Pfad für TEMP eingeben, z.B. *C:\Temp* oder *%Systemdrive%\Temp*. Nach der Eingabe des neuen Pfades schließen Sie die beiden Dialogfenster wieder. Der neue Pfad wird nun in der Übersicht der Systemvariablen angezeigt und ist auch mit sofortiger Wirkung gültig.

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung stehen Ihnen mehrere Möglichkeiten zur Verfügung, womit Sie die system- und benutzerspezifischen Umgebungsvariablen bearbeiten können. Im einfachsten Fall wenden Sie den *Set*-Befehl an und weisen der betreffenden Umgebungsvariablen per Gleichheitszeichen einen neuen Wert zu. Hierbei müssen Sie allerdings berücksichtigen, dass sich der *Set*-Befehl primär auf die benutzerspezifischen Umgebungsvariablen auswirkt, was sich speziell bei den Variablen TEMP und TMP bemerkbar macht. Denn wenn Sie der Variablen TEMP oder TMP per *Set*-Befehl einen neuen Pfad zuweisen, so wird in diesem Fall die benutzerspezifische Umgebungsvariable des Benutzers geändert, der gerade angemeldet ist bzw. den *Set*-Befehl ausführt.

```
@Echo off
Set TEMP = C:\Temp
```

Listing 542: 2_06_028a.bat

Eine weitere Möglichkeit, die Umgebungsvariablen zu bearbeiten, bietet der `Setx`-Befehl, der im Gegensatz zum `Set`-Befehl eine Unterscheidung zwischen den system- und benutzerspezifischen Umgebungsvariablen erlaubt. Um beispielsweise die benutzerspezifische Umgebungsvariable `TEMP` zu bearbeiten und den Pfad von `%USER-PROFILE%\Lokale Einstellungen\Temp` nach `C:\Temp` zu verlegen, wenden Sie folgende Befehlszeile an:

```
@Echo off
SetX TEMP C:\Temp
```

Listing 543: 2_06_028b.bat

Möchten Sie dagegen die systemspezifische Umgebungsvariable `TEMP` bearbeiten, so fügen Sie der `Setx`-Befehlszeile den Parameter `/M` an und legen damit fest, dass es sich nicht um eine benutzerspezifische Umgebungsvariable (dies ist die Standardeinstellung) handelt, sondern um eine systemspezifische Variable.

```
@Echo off
SetX TEMP C:\Temp /M
```

Listing 544: 2_06_028c.bat

Und zu guter Letzt können Sie die benutzer- und systemspezifischen Umgebungsvariablen auch mit dem `Reg Add`-Befehl bearbeiten, da die Umgebungsvariablen in der Registrierung hinterlegt sind und per `Reg Add` mit neuen Werten überschrieben werden können. Um beispielsweise der benutzerspezifischen Umgebungsvariablen `TEMP` den Wert `C:\Temp` zuweisen zu können, wenden Sie den `Reg Add`-Befehl unter Angabe des Schlüssels `HKCU\Environment` an und übergeben mit dem Parameter `/v` den Namen der Umgebungsvariable (in diesem Fall `TEMP`), mit dem Parameter `/t` den Typ des neuen Wertes (`REG_SZ` = Zeichenfolge) und mit `/d` die eigentliche Zeichenfolge (`C:\Temp`).

```
@Echo off
Reg Add HKCU\Environment /v TEMP /t REG_SZ /d C:\Temp
```

Listing 545: 2_06_028d.bat

Wenn Sie dagegen die systemspezifische Umgebungsvariable TEMP von C:\Windows\Temp auf C:\Temp verlegen möchten, so geben Sie stattdessen den Schlüssel HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment an und übergeben den Namen der Zeichenfolge (TEMP), den Typ (REG_SZ) wie auch die eigentliche Zeichenfolge (C:\Temp) als Parameter.

```
@Echo off
Reg Add "HKLM\SYSTEM\CurrentControlSet\Session _
    Manager\Environment" /v TEMP /t REG_SZ /d C:\Temp
```

Listing 546: 2_06_028e.bat

Die Lösung mit VBScript

Wenn Sie per VBScript eine benutzer- oder systemspezifische Umgebungsvariable ändern möchten, so können Sie hierfür die `VariableValue`-Methode der WMI-Klasse `Win32_Environment` nutzen, der Sie lediglich den neuen Wert als Argument übergeben müssen.

Das nachfolgende Listing zeigt ein Beispiel für den Einsatz der `VariableValue`-Methode. Zunächst wird die betreffende Umgebungsvariable mit der `ExecQuery`-Abfrage anhand des Namens der Umgebungsvariablen und auch des Namens des Accounts (Benutzername oder System) selektiert. Dann wird die `VariableValue`-Methode gegen das Variablenobjekt angewendet, wobei mit dem Aufruf der Methode gleichzeitig auch der neue Wert der Variablen übergeben wird.

Um zwischen benutzer- und systemspezifischen Umgebungsvariablen zu unterscheiden, fragen Sie lediglich den Wert der Eigenschaft `UserName` ab. Dieser enthält den Namen des Benutzers (Format *Computer- oder Domänenname\Benutzername*), wenn es sich um eine benutzerspezifische Umgebungsvariable handelt, anderenfalls ist der Wert gleich `System` (im Falle einer systemspezifischen Umgebungsvariablen).

```
Option Explicit

Dim oWMI
Dim colVariables
Dim objItem

strComputer = "."
Set oWMI = GetObject("winmgmts:\\." & strComputer & "\root\cimv2")
Set colVariables = oWMI.ExecQuery ("SELECT * FROM Win32_Environment _
    WHERE Name = 'Temp' AND UserName='System'")

For Each objItem in colVariables
    objItem.VariableValue = "C:\Temp"
```

Listing 547: 2_06_028.vbs

```
objItem.Put_  
Next
```

```
Set oWMI = Nothing
```

Listing 547: 2_06_028.vbs (Forts.)

155 Computer herunterfahren und neu starten

Wenn Sie das lokale Computersystem, an dem Sie gerade angemeldet sind, herunterfahren oder neu starten möchten, so wählen Sie im STARTMENÜ den Eintrag HERUNTERFAHREN aus. Sie gelangen daraufhin automatisch in das Dialogfenster WINDOWS HERUNTERFAHREN und können im Feld WIE MÖCHTEN SIE VORGEHEN? auswählen, ob Sie das Computersystem herunterfahren und damit ausschalten oder aber neu starten möchten.

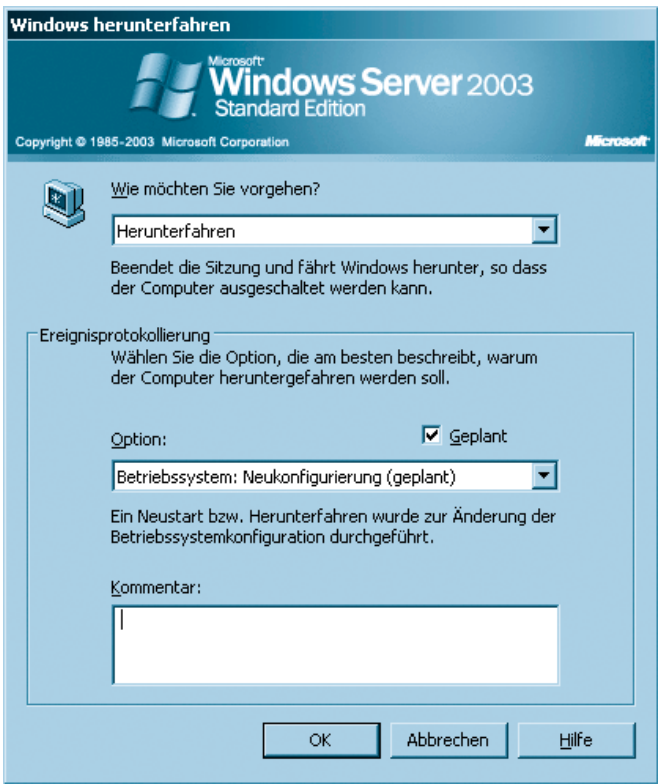


Abbildung 205: Die Ereignisprotokollierung zum Herunterfahren des Computersystems

Und sofern die Ereignisprotokollierung für das Herunterfahren des Computersystems aktiviert ist (siehe auch Ereignisprotokollierung für das Herunterfahren deaktivieren auf Seite 598), wählen Sie im Feld OPTIONEN einen Grund für das Herunterfahren bzw. für den Neustart aus und geben einen Kommentar dazu ein (dies ist ein Pflichtfeld, es muss also etwas eingetragen werden). Diese Informationen können Sie später mit Hilfe der Ereignisanzeige von Windows Server 2003 zu statistischen Zwecken oder auch für das Troubleshooting auswerten.

Wenn Sie dagegen ein Remote-System herunterfahren oder neu starten möchten, so können Sie mit Hilfe des REMOTEDESKTOPS eine RDP-Verbindung zu dem betreffenden Remote-System herstellen und die zuvor genannten Schritte auf dem Zielsystem ausführen.

Hierzu öffnen Sie das STARTMENÜ, erweitern den Eintrag VERWALTUNG und klicken anschließend auf REMOTEDESKTOPS, um in die Übersicht der Remotedesktop-Verbindungen zu gelangen. Sofern die entsprechende Verbindung bereits eingerichtet wurde, markieren Sie den Verbindungseintrag und wählen im Kontextmenü VERBINDUNG HERSTELLEN aus oder fügen eine neue Verbindung unter Angabe des Computernamens oder der IP-Adresse des Remote-Systems und der Anmeldeinformationen (Benutzername und Kennwort) hinzu und stellen dann die Verbindung zum Zielsystem her.

Die Lösung per Eingabeaufforderung

Um das lokale Computersystem von der Eingabeaufforderung aus herunterzufahren oder neu zu starten, verwenden Sie den `Shutdown`-Befehl, der zum festen Befehlsumfang von Windows Server 2003 gehört.

Der `Shutdown`-Befehl erwartet dabei den Parameter `/s`, wenn Sie das Computersystem vollständig herunterfahren und ausschalten möchten, wobei standardmäßig ein Zeitlimit von 30 Sekunden verwendet wird, bis der `Shutdown` auch wirklich durchgeführt wird. Mit dem Parameter `/t` und der Anzahl der Sekunden (0-600) können Sie diese Wartezeit jedoch verkürzen oder wahlweise auch erhöhen.

```
@Echo off
Shutdown /s /t 10
```

Listing 548: 2_06_029a.bat

Wenn Sie das Computersystem dagegen nicht herunterfahren, sondern neu starten möchten, so übergeben Sie dem `Shutdown`-Befehl den Parameter `/r`, woraufhin das Computersystem nach einer Wartezeit von 30 Sekunden neu gestartet wird. Auch hier lässt sich die Wartezeit wieder verändern, indem Sie mit dem Parameter `/t` eine neue Wartezeit übergeben.

```
@Echo off
Shutdown /r /t 60
```

Listing 549: 2_06_029b.bat

Der Shutdown-Befehl lässt sich natürlich auch zum Herunterfahren und Neustarten eines Remote-Systems verwenden, sofern der angemeldete Benutzer des lokalen Computersystems über entsprechende Berechtigungen auf dem Remote-System verfügt. Hierfür fügen Sie der Befehlszeile den Parameter /m an und übergeben den Computernamen oder die IP-Adresse des Remote-Systems.

```
@Echo off
Shutdown /r /t 60 /m \\192.168.0.100
```

Listing 550: 2_06_029c.bat

Und für den Fall, dass Sie die Ausführung des Shutdown-Befehls abbrechen möchten, so können Sie das Herunterfahren bzw. Neustarten des Computersystems verhindern, indem Sie den Shutdown-Befehl mit dem Parameter /a anwenden.

Die Lösung mit VBScript

Mit Hilfe der Win32Shutdown-Methode der WMI-Klasse Win32_OperatingSystem lässt sich ein Computersystem auch per VBScript herunterfahren oder neu starten. Die Win32Shutdown-Methode erwartet dabei nur die Angabe der gewünschten Shutdown-Option, die Sie mit dem Methodenaufruf als Argument übergeben müssen. Folgende Optionen sind möglich:

Option	Wert
Abmelden	0
Abmelden erzwingen	4
Herunterfahren	1
Herunterfahren erzwingen	5
Neu starten	2
Neu starten erzwingen	6
Ausschalten	8
Ausschalten erzwingen	12

Tabelle 68: Die Win32Shutdown-Optionen

Das nachfolgende Skript zeigt ein Anwendungsbeispiel für die `Win32ShutDown`-Methode und fährt das aktuelle Computersystem herunter. Dabei wird im ersten Schritt eine Verbindung zum WMI-Namespace `root\cimv2` des betreffenden Computersystems hergestellt, der `Impersonation-Level` auf `Impersonate` gesetzt und explizit das `ShutDown-Privileg` übergeben.

Anschließend wird mit der `ExecQuery`-Abfrage das `OperatingSystem`-Objekt selektiert und die `Win32ShutDown`-Methode unter Angabe der entsprechenden `Shutdown-Option` (in diesem Fall 0 für das Herunterfahren) gegen das Objekt angewendet, woraufhin das Computersystem dann heruntergefahren wird.

Option Explicit

```
Dim oWMI
Dim strComputer
Dim colOS
Dim OS
```

```
strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate,(ShutDown)}!\\" & strComputer & "\root\cimv2")
Set colOS = oWMI.ExecQuery ("Select * from _
    Win32_OperatingSystem")
```

```
For Each OS in colOS
    OS.Win32Shutdown(0)
Next
```

```
Set oWMI = Nothing
```

Listing 551: 2_06_029.vbs

Server-Rollen

Mit der reinen Installation von Windows Server 2003 ist es in der Regel nicht getan. Denn meist müssen Sie nach der Installation noch eine Reihe von Konfigurationsarbeiten vornehmen, um beispielsweise das Server-System zu einem Domänen-Controller heraufzustufen, die Domänenfunktionsebene und auch die Gesamtstrukturfunktionsebene anzuheben, den DHCP-Server zu autorisieren und entsprechende Bereiche zu konfigurieren und vieles mehr.

Windows Server 2003 hält nun eine ganze Reihe von Möglichkeiten bereit, womit Sie die gängigsten Serverkonfigurationen von der Eingabeaufforderung aus oder wahlweise auch per VBScript vereinfachen und gegebenenfalls auch vollautomatisieren können. Die nachfolgenden Rezepte zeigen daher, wie Sie die wichtigsten Konfigurationsaufgaben, die oftmals nach der Installation eines Server-Systems anstehen, unter grafischer Oberfläche durchführen und wie Sie die Aufgaben mit entsprechenden Befehlen und VBScript-Befehlszeilen automatisieren können.

156 Server zum Domänen-Controller heraufstufen

Um einen Standalone- oder Member-Server zu einem Domänen-Controller heraufzustufen, melden Sie sich zunächst als lokaler Administrator an dem betreffenden Server an und rufen im STARTMENÜ, genauer gesagt unter VERWALTUNG, den SERVER-KONFIGURATIONSSASSISTENT auf. Im Dialogfenster SERVER-KONFIGURATIONSSASSISTENT klicken Sie zweimal auf WEITER, um zur Auswahl der KONFIGURATIONSOPTION zu gelangen. Aktivieren Sie die Option BENUTZERDEFINIERTE KONFIGURATION, und klicken Sie anschließend auf WEITER. In der Auswahl der SERVERFUNKTION markieren Sie den Eintrag DOMÄNENCONTROLLER (ACTIVE DIRECTORY).

Sofern noch kein DNS-Server eingerichtet und auch kein DNS-Client konfiguriert wurde, werden Sie nun auf die fehlende DNS-Konfiguration aufmerksam gemacht. Sie können an dieser Stelle entscheiden, ob Sie auf diesem Server auch einen DNS-Server einrichten oder – falls Sie bereits einen DNS-Server haben – nur den DNS-Client konfigurieren möchten. Um mit dem Active Directory-Server auch den DNS-Server zu installieren, wählen Sie die Option NEIN, DNS AUF DIESEM COMPUTER INSTALLIEREN UND KONFIGURIEREN aus und geben daraufhin den DNS-Namen der zu konfigurierenden Domäne (*MyDomain.com*) ein. Daraufhin wird der NetBIOS-Name (*MyDomain*) ermittelt und angezeigt, übernehmen Sie diese Einstellung mit einem Klick auf WEITER.

Als Nächstes werden die Standardpfade für den Datenbank- und Protokollordner angezeigt. Klicken Sie auf WEITER, um jeweils die Standardeinstellung *C:\Windows\NTDS* und daraufhin auch den Standardpfad für die Freigabe *SYSVOL (C:\Windows\Sysvol)* zu übernehmen. Im nächsten Schritt gilt es, das Berechtigungsmodell auszuwählen. Legen Sie die Option NUR MIT WINDOWS 2000- ODER WINDOWS SERVER 2003-BETRIEBSSYSTEMEN KOMPATIBLE BERECHTIGUNGEN fest, wenn Sie im Netzwerk nur Server

mit den Betriebssystemen *Windows 2000 Server* und *Windows Server 2003* betreiben und damit nur authentifizierten Benutzern Zugriff gewähren möchten.

Anschließend legen Sie das Kennwort für die Verzeichnisdienst-Wiederherstellung fest und beenden den ASSISTENTEN ZUM INSTALLIEREN VON ACTIVE DIRECTORY mit einem Klick auf FERTIG STELLEN. Der Active Directory-Dienst wird nun mit den zuvor ausgewählten Optionen eingerichtet. Nach Abschluss der Konfiguration müssen Sie noch einen Computerneustart durchführen, um die geänderten Einstellungen wirksam werden zu lassen, dann ist die Einrichtung von Active Directory und damit auch das Heraufstufen des Servers zum Domänen-Controller abgeschlossen.

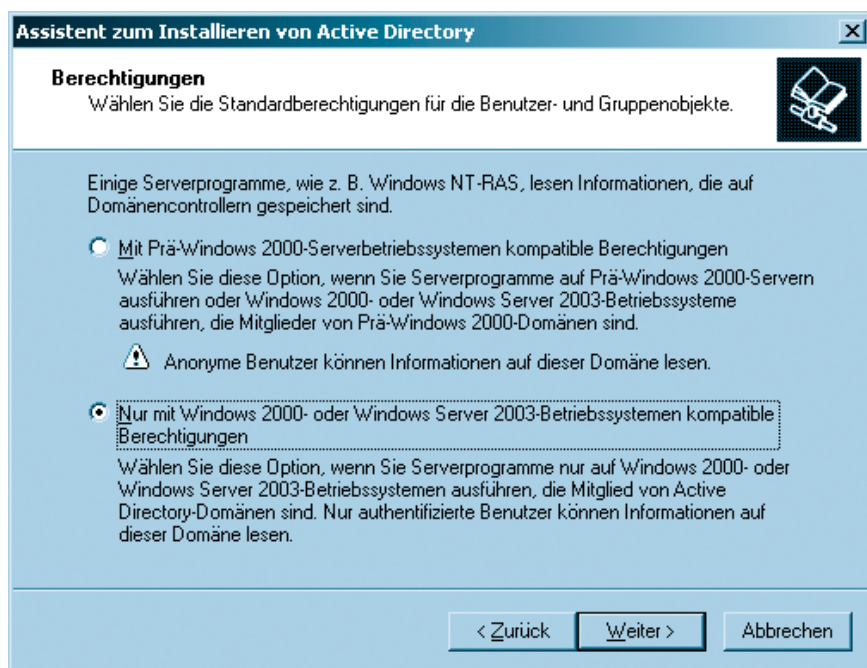


Abbildung 206: Das Berechtigungsmodell festlegen

Die Lösung per Eingabeaufforderung

Einen Standalone- oder Member-Server können Sie auch von der Eingabeaufforderung aus zu einem Domänen-Controller heraufstufen. Denn der ASSISTENT ZUM INSTALLIEREN VON ACTIVE DIRECTORY (*Dcpromo.exe*) erlaubt die Verwendung einer Antwortdatei, worin alle Konfigurationseinstellungen bereits hinterlegt sind, und ermöglicht so eine unbeaufsichtigte Installation von Active Directory. Mit unbeaufsichtigter Installation ist an dieser Stelle die automatisierte Einrichtung von Active Directory gemeint, da der ASSISTENT ZUM INSTALLIEREN VON ACTIVE DIRECTORY alle Einstellungen, die Sie normalerweise unter grafischer Oberfläche auswählen oder eingeben müssen, aus der Antwortdatei liest und automatisch konfiguriert.

Bevor Sie den Server von der Eingabeaufforderung aus zum Domänen-Controller heraufstufen und Active Directory installieren können, müssen Sie zunächst eine entsprechende Antwortdatei erstellen und die erforderlichen Konfigurationseinstellungen hinterlegen. Der `Dcpromo`-Befehl stellt hierzu eine ganze Reihe von Parametern zur Verfügung, die Sie in einer Textdatei aufnehmen und mit den entsprechenden Werten versehen müssen.

Die nachfolgende Tabelle enthält die Parameter, die zum Heraufstufen des Domänen-Controllers, zur automatischen Einrichtung des DNS-Servers und auch zur Erstellung einer neuen Domäne in einer neuen Gesamtstruktur erforderlich sind. Eine Übersicht aller Parameter finden Sie in *Teil III* des Buches.

Parameter	Beschreibung
<code>AllowAnonymousAccess=Yes No</code>	Legt das Berechtigungsmodell fest. Der Wert <i>Yes</i> bestimmt, dass das Berechtigungsmodell <i>Mit Prä-Windows 2000-Serverbetriebssystemen kompatible Berechtigungen</i> verwendet wird, der Wert <i>No</i> dagegen benutzt <i>Nur mit Windows 2000- oder Windows Server 2003-Betriebssystemen kompatible Berechtigungen</i> .
<code>AutoConfigDNS=Yes No</code>	Bestimmt die automatische DNS-Konfiguration. Der Wert <i>Yes</i> konfiguriert DNS automatisch, der Wert <i>No</i> dagegen nicht.
<code>DatabasePath=xxx</code>	Legt den Pfad für die Active Directory-Datenbanken fest (z.B. <code>C:\Windows\NTDS</code>).
<code>DisableCancelForDnsInstall=Yes No</code>	Blendet die Abbrechen-Schaltfläche während der DNS-Installation aus (<i>Yes</i>) oder ein (<i>No</i>).
<code>DNSOnNetwork=Yes No</code>	Bestimmt, ob die DNS-Konfiguration bereits durchgeführt wurde (<i>Yes</i>) oder während der Active Directory-Installation automatisch durchgeführt werden soll (<i>No</i>).
<code>DomainNetBiosName=xxx</code>	Bestimmt den NetBIOS-Namen der Domäne (z.B. <code>MyDomain</code>).
<code>LogPath=xxx</code>	Legt den Pfad für die Protokolldateien fest (z.B. <code>C:\Windows\NTDS</code>).
<code>NewDomain=Tree Child Forest</code>	Legt den Domänentyp fest. Der Wert <i>Forest</i> bestimmt, dass eine neue Root-Domäne in einer neuen Gesamtstruktur erstellt werden soll. Der Wert <i>Tree</i> legt eine Domänenstruktur in einer bestehenden Gesamtstruktur fest, und der Wert <i>Child</i> bestimmt eine untergeordnete Domäne in einer bestehenden Domänenstruktur.
<code>NewDomainDNSName=xxx</code>	Bestimmt den DNS-Namen für die neue Domäne (z.B. <code>MyDomain.com</code>).

Tabelle 69: Die wichtigsten Parameter für das Heraufstufen eines Servers

Parameter	Beschreibung
Password=xxx	Legt das Kennwort des Benutzers fest, der ausreichende Berechtigungen zum Heraufstufen des Servers zu einem Domänen-Controller hat.
RebootOnSuccess=Yes No	Legt den automatischen Computerneustart nach der Active Directory-Installation fest. Der Wert <i>Yes</i> führt den Computerneustart durch, der Wert <i>No</i> dagegen nicht.
ReplicaOrNewDomain=Replica Domain	Bestimmt den Typ des Domänen-Controllers und legt dabei fest, ob der Domänen-Controller der erste Domänen-Controller der Domäne (<i>Domain</i>) wird oder ein zusätzlicher Domänen-Controller (<i>Replica</i>).
SafeModeAdminPassword=xxx	Legt das Kennwort für die Verzeichnis-Wiederherstellung fest.
SysVolPath=xxx	Legt den Pfad für das freigegebene Systemvolume fest (z.B. C:\Windows\SysVol).
TreeOrChild=Tree Child	Legt fest, ob die neue Domäne die Root-Domäne einer neuen Gesamtstruktur (<i>Tree</i>) wird oder Teil (<i>Child</i>) einer bestehenden Gesamtstruktur.
UserDomain=xxx	Legt die Domäne des Benutzers fest.
UserName=xxx	Legt den Namen des Benutzers fest, der ausreichende Berechtigungen zum Heraufstufen des Servers zu einem Domänen-Controller hat (z.B. Administrator).

Tabelle 69: Die wichtigsten Parameter für das Heraufstufen eines Servers (Forts.)

Das nachfolgende Listing enthält den Inhalt der Antwortdatei *DC_Create.txt*, worin alle Konfigurationseinstellungen hinterlegt sind, um einen Server zum Domänen-Controller heraufzustufen, eine neue Root-Domäne (*MyDomain.com*) in einer neuen Gesamtstruktur anzulegen, die DNS-Konfiguration automatisch durchzuführen und nach Abschluss aller Konfigurationsarbeiten einen automatischen Computerneustart herbeizuführen. Dabei wurden exakt die gleichen Einstellungen hinterlegt, die auch unter grafischer Oberfläche (siehe Seite 651) verwendet wurden.

```
[DCInstall]
AllowAnonymousAccess = No
AutoConfigDNS = Yes
DatabasePath = C:\Windows\NTDS
DisableCancelForDnsInstall = No
DNSOnNetwork = Yes
DomainNetBiosName = MyDomain
LogPath = C:\Windows\NTDS
NewDomain = Forest
```

Listing 552: DC_Create.txt

```
NewDomainDNSName = MyDomain.com
Password = 12345678
RebootOnSuccess = Yes
ReplicaOrNewDomain = Domain
SafeModeAdminPassword = 12345678
SysVolPath = C:\Windows\Sysvol
TreeOrChild = Tree
UserDomain = MyDomain.com
UserName = Administrator
```

Listing 552: DC_Create.txt (Forts.)

Jede Dcpromo-Antwortdatei muss dabei in der ersten Zeile die Sektion [DCInstall] enthalten und im Anschluss alle Parameter aufweisen, die zur Einrichtung des Active Directory erforderlich sind. Wenn Parameter fehlen oder ungültige Werte enthalten, so kann die Active Directory-Installation nicht fehlerfrei durchgeführt werden und erfordert damit einen manuellen Eingriff in den Einrichtungsvorgang bzw. eine manuelle Korrektur.

Wenn Sie die Antwortdatei erstellt haben, so können Sie den Server nun zum Domänen-Controller heraufstufen und die Active Directory-Installation vornehmen, indem Sie von der Eingabeaufforderung aus den Dcpromo-Befehl mit dem Parameter /Answer: ausführen und die Antwortdatei übergeben.

```
@Echo off
Dcpromo /Answer:DC_Create.txt
```

Listing 553: 2_07_001.bat

Achtung

Führen Sie diese Batch-Datei bitte nicht auf einem Server aus, der bereits zu einem Domänen-Controller heraufgestuft wurde. Denn der erneute Aufruf von *Dcpromo.exe* bewirkt, dass der Domänen-Controller herabgestuft und – je nach Domänen-Modell – Active Directory entfernt wird.

Die Lösung mit VBScript

Wenn Sie das Heraufstufen des Servers zum Domänen-Controller und die Einrichtung von Active Directory mit Hilfe von VBScript durchführen möchten, so nutzen Sie die Run-Methode des Shell-Objektes und führen – wie auch in der Eingabeaufforderung – den Dcpromo-Befehl mit dem Parameter /Answer: aus und übergeben die entsprechende Antwortdatei. Die Einrichtung des Domänen-Controllers und der Domäne erfolgt mit den Parametern, die in der angegebenen Antwortdatei (siehe vorheriger Lösungsabschnitt) hinterlegt sind.

```
Option Explicit
Dim oShell
```

```
Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Dcpromo /Answer:DC_Create.txt"
```

```
Set oShell = Nothing
```

Listing 554: 2_07_001.vbs

157 Domänen-Controller zum Server herabstufen

Wenn Sie einen Domänen-Controller zum Standalone- oder Member-Server herabstufen und gleichzeitig auch Active Directory entfernen möchten, so rufen Sie aus dem Verwaltungsmenü den SERVERKONFIGURATIONS-ASSISTENT auf, übergehen die ersten beiden Dialogseiten mit WEITER und markieren in der Auswahl der Serverfunktionen den Eintrag DOMÄNENCONTROLLER (ACTIVE DIRECTORY). Mit einem Klick auf WEITER gelangen Sie zur Dialogseite BESTÄTIGUNG FÜR ENTFERNEN EINER FUNKTION, auf der Sie die Option DOMÄNENCONTROLLERFUNKTION ENTFERNEN aktivieren. Daraufhin wird wieder der ASSISTENT ZUM INSTALLIEREN VON ACTIVE DIRECTORY gestartet. Wenn der Domänen-Controller über den globalen Katalog verfügt, so erhalten Sie nun eine Warnung, dass Sie den Zugriff auf andere globale Kataloge sicherstellen müssen. Bestätigen Sie diese Warnung mit OK, und geben Sie im nächsten Schritt an, ob dies der letzte Domänen-Controller im Netzwerk ist. Aktivieren Sie die Option DIESER SERVER IST DER LETZTE DOMÄNENCONTROLLER IN DER DOMÄNE, um Active Directory zu entfernen und den Server herabzustufen (siehe Abbildung 207).

Anschließend bestätigen Sie das Entfernen der Anwendungsverzeichnispartitionen und aktivieren im nächsten Schritt die Option ALLE ANWENDUNGSVERZEICHNISPARTITIONEN AUF DIESEM DOMÄNENCONTROLLER LÖSCHEN.

Sie werden nun aufgefordert, ein neues Kennwort für den lokalen Administrator festzulegen. Geben Sie das entsprechende Kennwort ein, bestätigen Sie es im darauf folgenden Eingabefeld, und klicken Sie auf der nächsten Dialogseite auf WEITER, woraufhin Active Directory entfernt und der Domänen-Controller zum Standalone- oder Member-Server herabgestuft wird. Dieser Vorgang kann einige wenige Minuten in Anspruch nehmen. Nach Abschluss aller Konfigurationsarbeiten klicken Sie auf FERTIG STELLEN, um den Assistenten zu beenden, und führen dann einen Computerneustart durch, um die neue Konfiguration wirksam werden zu lassen.

Die Lösung per Eingabeaufforderung

Wenn Sie den Domänen-Controller von der Eingabeaufforderung aus zu einem Standalone- oder Member-Server herabstufen möchten, so wenden Sie wieder den Befehl `Dcpromo /Answer: an` und übergeben die Antwortdatei mit den Parametern, worüber Sie die Konfigurationsoptionen festlegen.

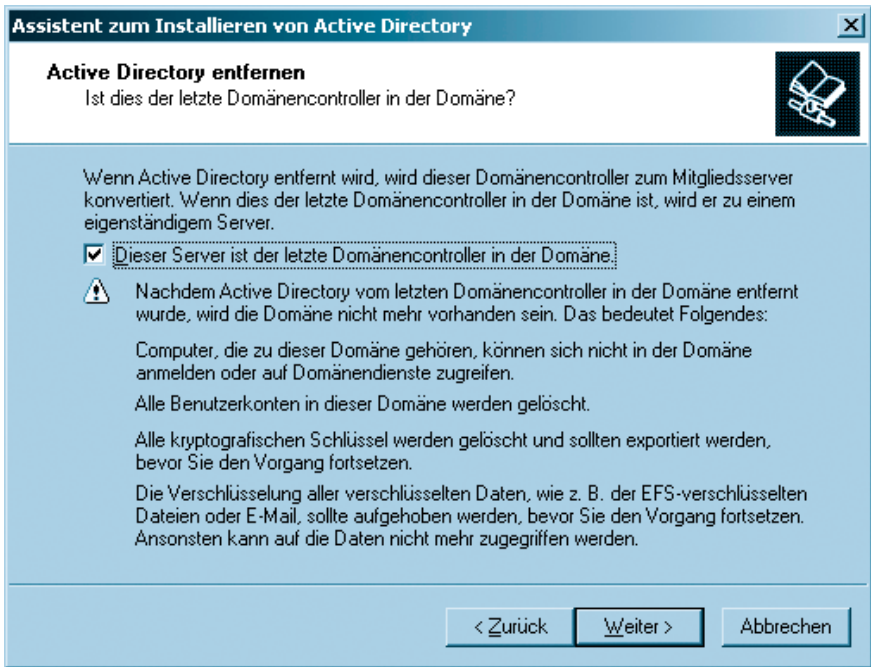


Abbildung 207: Festlegen, ob dies der letzte Domänen-Controller der Domäne ist

Die nachfolgende Tabelle enthält eine kurze Übersicht der wichtigsten Parameter, die Sie zum Herabstufen eines Domänen-Controllers zu einem Standalone-Server angeben müssen. Eine vollständige Übersicht der Dcpromo-Parameter finden Sie in *Teil III* des Buches.

Parameter	Beschreibung
AdministratorPassword = xxx	Legt das Kennwort des Administrators fest.
IsLastDCInDomain = Yes No	Gibt an, ob dies der letzte Domänen-Controller in der Domäne (Yes) ist oder nicht (No). Der Wert Yes bewirkt dabei auch das Entfernen von Active Directory.
RebootOnSuccess = Yes No	Legt den automatischen Computerneustart nach der Konfiguration fest. Der Wert Yes führt den Computerneustart automatisch durch, der Wert No dagegen nicht.
RemoveApplicationPartitions = Yes No	Gibt an, ob die Anwendungsverzeichnispartitionen entfernt werden sollen (Yes) oder nicht (No).

Tabelle 70: Die wichtigsten Parameter für das Herabstufen eines Domänen-Controllers

Um beispielsweise den einzigen Domänen-Controller innerhalb einer kleinen Domäne zu einem Standalone-Server herabzustufen und die Domäne damit aufzulösen, erstellen Sie eine Antwortdatei mit folgendem Inhalt:

```
[DCInstall]
AdministratorPassword = 12345678
ISLastDCInDomain = Yes
RebootOnSuccess = Yes
RemoveApplicationPartitions = Yes
```

Listing 555: DC_Remove.txt

Anschließend öffnen Sie auf dem Domänen-Controller die Eingabeaufforderung, geben den Befehl `Dcpromo /Answer: ein` und übergeben die soeben erstellte Antwortdatei. Daraufhin wird der Assistent zum Installieren von Active Directory gestartet, und das automatisierte Entfernen der Serverfunktion *Domänen-Controller* und das Löschen des Active Directory beginnt.

```
@Echo off
Dcpromo /Answer:DC_Remove.txt
```

Listing 556: 2_07_002.bat

Die Lösung mit VBScript

Um den Domänen-Controller aus einem VBScript heraus zu einem Standalone- oder Member-Server herabzustufen, setzen Sie wieder die `Run`-Methode des `Shell`-Objektes ein und führen den `Dcpromo /Answer:-Befehl` unter Angabe der Antwortdatei aus.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Dcpromo /Answer:DC_Remove.txt"

Set oShell = Nothing
```

Listing 557: 2_07_002.vbs

158 Domänenfunktionsebene heraufstufen

Eine Domänenfunktionsebene legt fest, welche zusätzlichen Funktionalitäten im Active Directory zur Verfügung stehen. Wenn Sie beispielsweise ein Server-System mit Windows Server 2003 einrichten, diesen zu einem Domänen-Controller heraufstufen und anschließend eine neue Domäne einrichten, so wird per Default die Domänenfunktionsebene *Windows 2000 gemischt* verwendet. Damit stehen die grundlegenden Active Directory-Funktionalitäten zur Verfügung, und es können Domänen-Controller mit den Betriebssystemen Windows NT 4.0, Windows 2000 und auch Windows Server 2003 in der Domäne betrieben werden.

Wenn Sie nun zusätzliche Active Directory-Funktionen wie beispielsweise das Tool zum Umbenennen von Domänen-Controllern, die Möglichkeit der unterschiedlichen Speicherorte für Benutzer- und Computerkonten oder die Möglichkeit der vollständigen Gruppenverschachtelung nutzen möchten, so müssen Sie hierfür die Domänenfunktionsebene von *Windows 2000 gemischt* auf *Windows Server 2003* anheben. Denn viele Zusatzfunktionen werden erst mit dem Aktivieren der Domänenfunktionsebene *Windows Server 2003* aktiviert und stehen daher nicht von Anfang an zur Verfügung.

Nachfolgend finden Sie eine kleine Aufstellung der zusätzlichen Active Directory-Funktionen, die mit dem Aktivieren der Domänenfunktionsebene *Windows Server 2003* zur Verfügung gestellt werden (weitere Informationen zu den einzelnen Domänenfunktionsebenen finden Sie auf der Internetseite www.microsoft.com).

- ▶ Tool für die Umbenennung eines Domänen-Controllers
- ▶ Unterschiedliche Speicherorte für Benutzer- und Computerkonten
- ▶ Aktualisieren des Anmeldezeitstempels
- ▶ Kennwortschutz für InetOrgPerson-Objekte
- ▶ Universelle Gruppen für Sicherheits- und Verteilergruppen
- ▶ Vollständige Gruppenverschachtelung
- ▶ Konvertierung von Sicherheits- und Verteilergruppen
- ▶ SID-Verlauf

So nützlich die zusätzlichen Active Directory-Funktionen der Domänenfunktionsebene *Windows Server 2003* auch sind: Beim Heraufstufen der Domänenfunktionsebene müssen Sie berücksichtigen, dass:

- ▶ das Aktivieren der Domänenfunktionsebene voraussetzt, dass in der Domäne ausschließlich Domänen-Controller mit Windows Server 2003 betrieben werden. Domänen-Controller, auf denen Windows NT 4.0 oder Windows 2000 eingerichtet ist, werden mit Verwendung der Domänenfunktionsebene *Windows Server 2003* nicht mehr unterstützt und müssen vorher zu Windows Server 2003 migriert werden.

- das Heraufstufen und Aktivieren der Domänenfunktionsebene Windows Server 2003 nicht mehr rückgängig gemacht werden kann. Sie können die Domänenfunktionsebene zwar jederzeit anheben, im Nachhinein aber nicht mehr herunterstufen.

Die nachfolgende Tabelle zeigt, welche Domänenfunktionsebenen es gibt und welche Windows-basierten Server-Betriebssysteme jeweils unterstützt werden.

Domänenfunktionsebene	Server-Betriebssystemunterstützung
Windows 2000 gemischt	Windows NT 4.0, Windows 2000, Windows Server 2003
Windows 2000 pur	Windows 2000, Windows Server 2003
Windows Server 2003-interim	Windows NT 4.0, Windows Server 2003
Windows Server 2003	Windows Server 2003

Tabelle 71: Die Domänenfunktionsebenen

Um nun die Domänenfunktionsebene auf *Windows Server 2003* heraufzustufen, öffnen Sie zunächst die Management-Konsole ACTIVE DIRECTORY-BENUTZER UND -COMPUTER. Klicken Sie hierfür auf START, wechseln Sie in das Menü VERWALTUNG, und rufen Sie dort die zuvor genannte Management-Konsole auf, oder klicken Sie stattdessen auf START, dann auf AUSFÜHREN, und geben Sie *Dsa.msc* ein, womit Sie die Management-Konsole auch direkt aufrufen können.

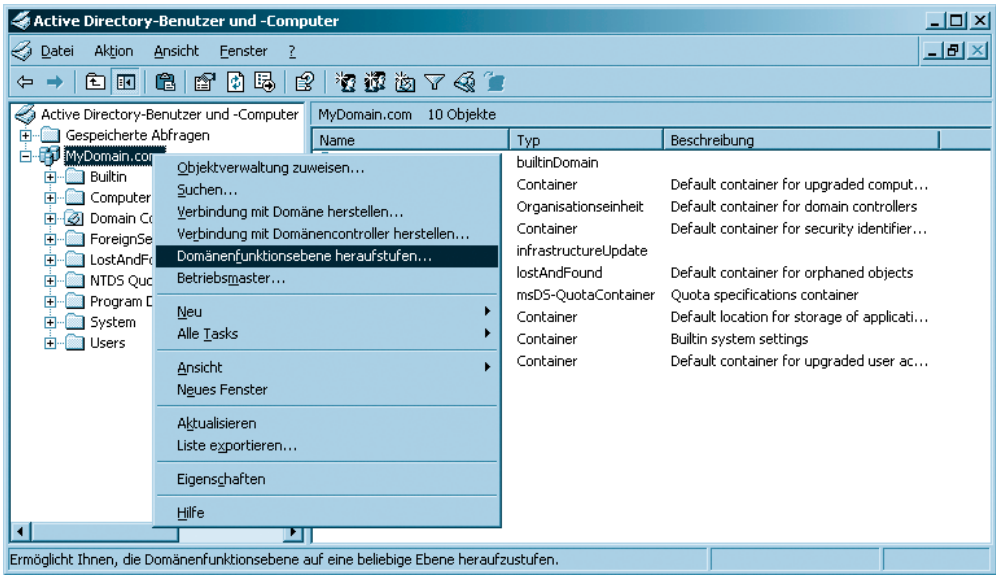


Abbildung 208: Das Dialogfenster »Domänenfunktionsebene heraufstufen« aufrufen

Markieren Sie in der Konsolenstruktur die entsprechende Domäne, und wählen Sie im Kontextmenü den Eintrag **DOMÄNENFUNKTIONSEBENE HERAUFSTUFEN...** aus. Sie gelangen daraufhin in das Dialogfenster **DOMÄNENFUNKTIONSEBENE HERAUFSTUFEN**. Wählen Sie die Funktionsebene *Windows Server 2003* aus, und klicken Sie auf **HERAUFSTUFEN**. Die Domänenfunktionsebene wird daraufhin geändert.

Die Lösung per Eingabeaufforderung

Leider gibt es keinen entsprechenden Befehl, womit Sie die Domänenfunktionsebene so ohne weiteres von der Eingabeaufforderung aus heraufstufen können. Da die Domänenfunktionsebene aber lediglich über die Werte der Active Directory-Attribute `msDS-Behavior-Version` und `NTMixedDomain` festgelegt wird und Sie diese Werte mit Hilfe des `Ldifde`-Subbefehls `Replace` modifizieren können, haben Sie dennoch die Möglichkeit, das Heraufstufen der Domänenfunktionsebene von der Eingabeaufforderung aus vorzunehmen.

Die nachfolgende Tabelle zeigt, welche Werte die beiden Attributen in den jeweiligen Domänenfunktionsebenen besitzen:

Domänenfunktionsebene	msDS-Behavior-Version	NTMixedDomain
Windows 2000 gemischt	0	0
Windows 2000 pur	0	1
Windows Server 2003	2	0

Tabelle 72: Die Werte der Attribute »msDS-Behavior-Version« und »NTMixedDomain«

Um die Werte der beiden Attribute zu verändern, setzen Sie den `Ldifde`-Befehl ein und importieren den Inhalt einer kleinen `Ldifde`-Skriptdatei, worüber Sie den `Ldifde`-Subbefehl `Modify` zum Ändern eines Wertes ausführen können.

Die nachfolgende `Ldifde`-Befehlszeile zeigt ein solches Beispiel. Mit dem Parameter `-i` wird der Importmodus bestimmt, und mit dem Parameter `-f` wird die Datei übergeben, deren Inhalt eingelesen werden soll.

```
@Echo off
Ldifde -i -f Change_DomainFL1.ldf
Ldifde -i -f Change_DomainFL2.ldf
```

Listing 558: 2_07_003.bat

Das nachfolgende Listing zeigt den Inhalt der Datei *MyScript1.ldf*, worüber der Wert des Attributes `ntmixeddomain` auf 0 gesetzt wird. Die erste Zeile enthält dabei den definierten Namen der Domäne, deren Domänenfunktionsebene heraufgestuft werden soll. In der zweiten Zeile wird die auszuführende Aktion (*modify* = ändern) festgelegt, in der dritten Zeile wird das Attribut, dessen Wert ersetzt (*replace*) soll, bestimmt, und mit der vierten Zeile wird der neue Wert zugewiesen. Mit dem Zeichen

– und der anschließenden Leerzeile wird das Ende des Skriptes definiert (diese Abschlusskennzeichnung muss unbedingt vorhanden sein, da die Befehlsausführung sonst scheitert).

```
dn: DC=MyDomain,DC=Com
changetype: modify
replace: ntmixeddomain
ntmixeddomain: 0
-
(Leerzeile)
```

Listing 559: Change_DomainFL1.ldf

Und mit dem folgenden Skript lässt sich der Wert des Attributes `msDS-Behavior` auf 2 setzen, womit die Domänenfunktionsebene *Windows Server 2003* aktiviert wird.

```
dn: DC=MyDomain,DC=Com
changetype: modify
replace: msDS-Behavior-Version
msDS-Behavior-Version: 2
-
(Leerzeile)
```

Listing 560: Change_DomainFL2.ldf

Die Lösung mit VBScript

Um die Domänenfunktionsebene mit VBScript heraufzustufen, stellen Sie zunächst per LDAP eine Verbindung zum Root der Domäne her und fragen den Wert des Attributes `msDS-Behavior-Version` ab. Wenn dieser ungleich 2 ist, so weisen Sie mit der `Put`-Methode dem Attribut `ntMixedDomain` den Wert 0 und dem Attribut `msDS-Behavior-Version` den Wert 2 zu. Um die Änderungen zu übernehmen, wenden Sie anschließend die `SetInfo`-Methode an, woraufhin die neuen Werte zurückgeschrieben werden.

Option Explicit

```
Dim oRootDSE
Dim oDomain
```

```
Set oRootDSE = GetObject("LDAP://RootDSE")
Set oDomain = GetObject("LDAP://" & oRootDSE.Get("DefaultNamingContext"))
```

```
If oDomain.Get("msDS-Behavior-Version") <> 2 Then
    oDomain.Put "ntMixedDomain", 0
```

Listing 561: 2_07_003.vbs

```
oDomain.Put "msDS-Behavior-Version", 2
oDomain.SetInfo
End If
```

```
Set oRootDSE = Nothing
Set oDomain = Nothing
```

Listing 561: 2_07_003.vbs (Forts.)

159 Gesamtstrukturfunktionsebene heraufstufen

In Ergänzung zur Domänenfunktionsebene gibt es noch die Gesamtstrukturfunktionsebene, die sich im Gegensatz zur Domänenfunktionsebene auf die Gesamtstruktur auswirkt und sich nicht nur auf die aktuelle Domäne beschränkt.

Die nachfolgende Tabelle zeigt, welche Gesamtstrukturfunktionsebenen es gibt und welche Server-Betriebssysteme von Microsoft jeweils unterstützt werden.

Gesamtstrukturfunktionsebene	Server-Betriebssystemunterstützung
Windows 2000	Windows NT 4.0, Windows 2000, Windows Server 2003
Windows Server 2003-interim	Windows NT 4.0, Windows Server 2003
Windows Server 2003	Windows Server 2003

Tabelle 73: Die Gesamtstrukturfunktionsebenen

Wenn Sie eine neue Gesamtstruktur einrichten, so wird per Default die Gesamtstrukturfunktionsebene *Windows 2000* verwendet. Mit der Voraussetzung, dass alle Domänen-Controller innerhalb der Gesamtstruktur mit Windows Server 2003 ausgestattet sind und (auch künftig!) keine Windows NT 4.0- und Windows 2000-Systeme betrieben werden, so können Sie die Gesamtstrukturfunktionsebene auf *Windows Server 2003* heraufstufen. Dadurch können Sie die folgenden zusätzlichen Funktionen aktivieren (detaillierte Informationen zu den einzelnen Funktionen finden Sie auf der Internetseite www.microsoft.com):

- ▶ Verbesserte Replikation des globalen Katalogs
- ▶ Funktionsschema-Objekte
- ▶ Gesamtstrukturvertrauensstellungen
- ▶ Verknüpfte Wert-Replikation
- ▶ Tool zum Umbenennen einer Domäne
- ▶ Dynamische Erweiterungsklassen
- ▶ ObjectClass-Änderungen für das InetOrtPerson-Attribut

Das Heraufstufen der Gesamtstrukturfunktionsebene setzt voraus, dass auf allen Domänen-Controllern Windows Server 2003 eingerichtet ist. Und vor dem Aktivieren der zusätzlichen Funktionen müssen Sie wieder berücksichtigen, dass Sie die Gesamtstrukturfunktionsebene zwar problemlos heraufstufen, im Nachhinein aber nicht mehr heruntersetzen können.

Um nun die Funktionsebene für die Gesamtstruktur heraufzusetzen, nehmen Sie die Management-Konsole ACTIVE DIRECTORY-DOMÄNEN UND -VERTRAUENSSTELLUNGEN zu Hilfe. Klicken Sie zum Öffnen der Management-Konsole auf START, anschließend auf VERWALTUNG und dann auf die zuvor genannte Management-Konsole, oder geben Sie unter START/AUSFÜHREN den Befehl DOMAIN.MSC ein, womit Sie die Management-Konsole auch direkt aufrufen können.

Markieren Sie in der linken Konsolenstruktur den Eintrag ACTIVE DIRECTORY-DOMÄNEN UND -VERTRAUENSSTELLUNGEN, und wählen Sie im Kontextmenü den Eintrag GESAMTSTRUKTURFUNKTIONSEBENE HERAUFSTUFEN... aus. Sie gelangen daraufhin in das gleichnamige Dialogfenster. Wählen Sie dort die Funktionsebene *Windows Server 2003* für die Gesamtstruktur aus, und klicken Sie anschließend auf HERAUFSTUFEN.

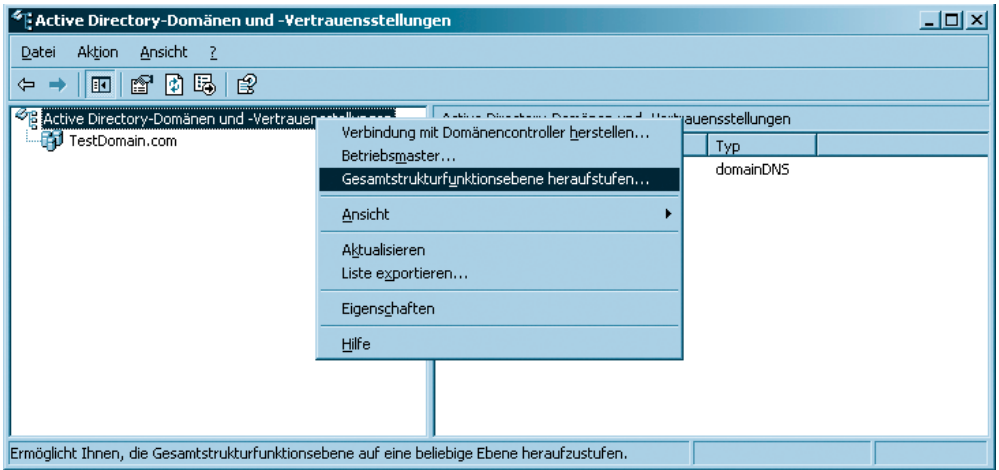


Abbildung 209: Die Gesamtstrukturfunktionsebene heraufstufen

Die Lösung per Eingabeaufforderung

Die Gesamtstrukturfunktionsebene lässt sich in der Eingabeaufforderung auf die gleiche Art und Weise ändern wie auch die Domänenfunktionsebene. Sie wenden hierfür also auch den `ldifde`-Befehl an und importieren ein kleines `ldifde`-Skript, worüber der Sub-Befehl `Modify` ausgeführt und der Wert des Attributes `msDS-Behavior-Version` entsprechend geändert wird.

Die nachfolgende Tabelle zeigt die möglichen Gesamtstrukturfunktionsebenen unter Angabe der entsprechenden Werte des Attributes `msDS-Behavior-Version`.

Gesamtstrukturfunktionsebene	msDS-Behavior-Version
Windows 2000	0
Windows Server 2003-interim	1
Windows Server 2003	2

Tabelle 74: Die Werte des Attributes »msDS-Behavior-Version«

Um die Gesamtstrukturfunktionsebene *Windows Server 2003* zu aktivieren, setzen Sie den `Ldifde`-Befehl mit dem Parameter `-i` ein, um den Import-Modus einzuschalten, und fügen der Befehlszeile dann mit dem Parameter `-f Change_ForestFL.ldf` den Namen der zu importierenden Datei (in diesem Fall *Change_ForestFL.ldf*) an.

```
@Echo off
Ldifde -i -f Change_ForestFL.ldf
```

Listing 562: 2_07_004.bat

Mit der obigen `Ldifde`-Befehlszeile werden die folgenden Befehlszeilen importiert. Zu Beginn des kleinen `Ldifde`-Skriptes wird der definierte Name des Containers `Partitions`, der sich unterhalb von `Configuration` befindet, übergeben. Dann wird der Sub-Befehl `Modify` angewendet und der Name des zu modifizierenden Attributes (`msDS-Behavior-Version`) wie auch der neue Wert (2) übergeben. Als Abschlusskennzeichnung folgt das `-` und darauf eine Leerzeile, womit das Ende des Skriptes gekennzeichnet wird.

```
dn: CN=Partitions,CN=Configuration,DC=MyDomain,DC=Com
changetype: modify
replace: msDS-Behavior-Version
msDS-Behavior-Version: 2
-
(Leerzeile)
```

Listing 563: Change_ForestFL.ldf

Die Lösung mit VBScript

Wenn Sie die Funktionsebene der Gesamtstruktur per VBScript auf Windows Server 2003 heraufstufen möchten, so stellen Sie per LDAP erst eine Verbindung zum Root der Domäne und anschließend zum Container `Partitions` her. Anschließend prüfen Sie den Wert des Attributes `msDS-Behavior-Version`. Wenn das Attribut einen Wert ungleich 2 besitzt, so setzen Sie den Wert 2 mit Hilfe der `Put`-Methode und speichern die Änderung dann mit der `SetInfo`-Methode ab.

```
Option Explicit
```

```
Dim oRootDSE
```

```
Dim oForest
```

```
Set oRootDSE = GetObject("LDAP://RootDSE")
```

```
Set oForest = GetObject("LDAP://CN=Partitions," & _  
oRootDSE.Get("ConfigurationNamingContext"))
```

```
If oForest.Get("msDS-Behavior-Version") <> 2 then
```

```
oForest.Put "msDS-Behavior-Version", 2
```

```
oForest.SetInfo
```

```
End if
```

```
Set oRootDSE = Nothing
```

```
Set oForest = Nothing
```

Listing 564: 2_07_004.vbs

160 Auf welchem Domänen-Controller ist der globale Katalog aktiviert?

Mit der Einrichtung einer neuen Gesamtstruktur wird auf dem ersten Domänen-Controller der Gesamtstruktur automatisch ein globaler Katalog erzeugt. Dieser globale Katalog stellt eine Verzeichnisdatenbank dar, in der eine vollständige Kopie aller Active Directory-Objekte der eigenen Domäne und ein Teil der Active Directory-Objekte der anderen Domänen, die der Gesamtstruktur angehören, enthalten sind.

Der Inhalt des globalen Katalogs wird per Replikation zwischen den einzelnen Domänen-Controllern aufgebaut und damit auch up to date gehalten. Welche Active Directory-Objekte der anderen Domänen dabei im globalen Katalog aufgenommen werden, ist im Schema genauestens definiert und lässt sich mit Hilfe der Management-Konsole ACTIVE DIRECTORY-SCHEMA (vor der ersten Verwendung müssen Sie die Konsole mit dem Befehl `Regsvr32 Schmmgmt.dll` erst registrieren) bei Bedarf auch verändern.

Sinn und Zweck des globalen Katalogs liegt beispielsweise im schnellen Auffinden von Active Directory-Objekten innerhalb der Gesamtstruktur, in der domänenübergreifenden Authentifizierung unter Verwendung von Benutzerprinzipalnamen (engl. = User Principal Name) und im Auflösen der universellen Gruppenmitgliedschaften in einer Multi-Domänen-Umgebung. Der globale Katalog muss dabei nicht unbedingt auf einem bestimmten Domänen-Controller abgelegt sein, sondern kann jederzeit auch auf andere Domänen-Controller übertragen werden.

Um nun unter grafischer Oberfläche und mit Windows Server 2003-eigenen Bordmitteln herauszufinden, auf welchem Domänen-Controller der globale Katalog aktiviert ist, nutzen Sie die Management-Konsole ACTIVE DIRECTORY-STANDORTE UND -DIENSTE (klicken Sie hierzu auf START, wechseln Sie dann in das Menü VERWALTUNG, und öff-

nen Sie anschließend ACTIVE DIRECTORY-STANDORTE UND -DIENSTE). Erweitern Sie in der linken Fensterhälfte den Standort, dann den ersten Domänen-Controller, und markieren Sie unterhalb davon NTDS SETTINGS. Rufen Sie anschließend aus dem Kontextmenü heraus die EIGENSCHAFTEN auf. Auf der Registerkarte ALLGEMEIN finden Sie die Option GLOBALER KATALOG. Wenn diese Option aktiviert ist, dann verfügt der zuvor ausgewählte Domänen-Controller über den globalen Katalog. Anderenfalls (wenn die Option nicht aktiviert ist) schließen Sie die Eigenschaften der NTDS-Einstellungen wieder und öffnen die des nächsten Domänen-Controllers.

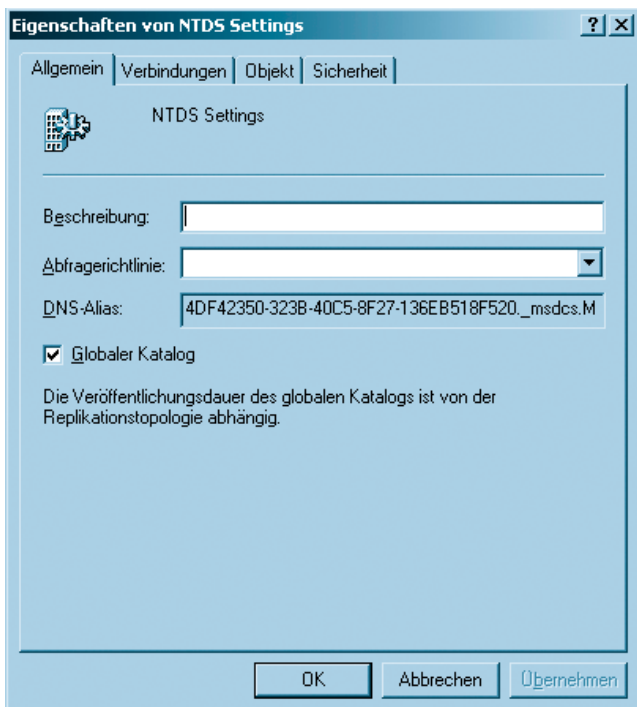


Abbildung 210: Auf diesem Domänen-Controller ist der globale Katalog aktiviert.

Beim Einsatz mehrerer Domänen-Controller pro Standort kann sich diese manuelle Suche als recht aufwändig erweisen, da Sie jeden einzelnen Domänen-Controller zunächst markieren, dann dessen NTDS-Einstellungen aufrufen und anschließend den Status der Option GLOBALER KATALOG überprüfen müssen. Weitaus einfacher (und auch wesentlich schneller) finden Sie den globalen Katalog jedoch mit Hilfe des Replikationsmonitors (*Replmon.exe*), der Bestandteil der *Windows Server 2003 Resource Kit-Tools* ist. Denn mit dem Replikationsmonitor brauchen Sie nur einen beliebigen Domänen-Controller des Standorts zur Liste der zu überwachenden Server hinzuzufügen und im Kontextmenü des Domänen-Controllers den Eintrag SHOW GLOBAL CATALOG SERVER IN ENTERPRISE auszuwählen. Sie gelangen daraufhin in eine Übersicht, worüber der Domänen-Controller, auf dem der globale Katalog aktiviert ist, angezeigt wird (siehe Abbildung 211).

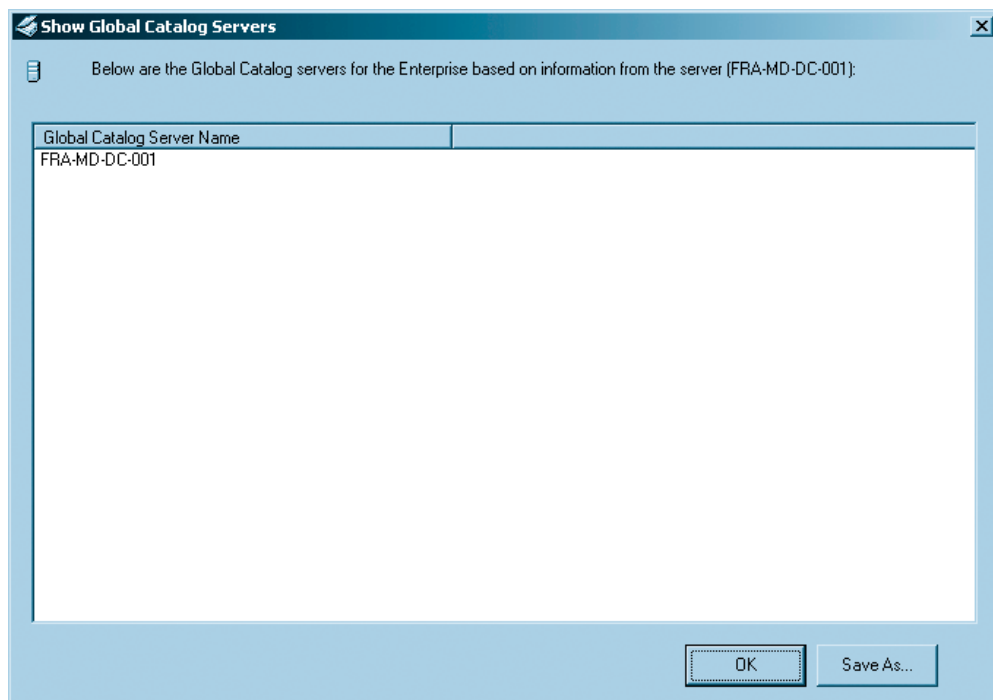


Abbildung 211: Der Replikationsmonitor hilft beim schnellen Finden des »Globalen Katalogs«.

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung ist es ein Leichtes, den globalen Katalog ausfindig zu machen. Denn mit dem Befehl `Dsquery Server -isgc` können Sie die aktuelle Domäne nach dem globalen Katalog durchsuchen und so den Hostnamen des entsprechenden Domänen-Controllers ausgeben lassen.

Wenn Sie nicht nur innerhalb der aktuellen Domäne nach dem globalen Katalog suchen möchten, sondern beispielsweise in der Gesamtstruktur oder innerhalb des aktuellen Standortes, so können Sie mit dem zusätzlichen Parameter `-Forest` und `-Site` den Gültigkeitsbereich der Suchanfrage verändern. Ohne Angabe von `-Forest` oder `-Site` wird per Default nur innerhalb der aktuellen Domäne gesucht.

Die nachfolgende `Dsquery`-Befehlszeile durchsucht beispielsweise die Gesamtstruktur nach dem Domänen-Controller, auf dem der globale Katalog aktiviert ist, und gibt dessen Hostnamen am Bildschirm aus:

```
@Echo off
Dsquery Server -Forest -isgc
```

Listing 565: 2_07_005a.bat

Wenn Sie stattdessen innerhalb des aktuellen Standortes nach dem globalen Katalog suchen möchten, so ersetzen Sie den Parameter `-Forest` durch `-Site`.

```
@Echo off
Dsquery Server -Site -isgc
```

Listing 566: 2_07_005b.bat

Die Lösung mit VBScript

Um per VBScript zu ermitteln, auf welchem Domänen-Controller innerhalb der aktuellen Domäne der globale Katalog aktiviert ist, können Sie das ADO Connection-Objekt für eine Abfrage der Active Directory-Datenbank nutzen. Hierzu erzeugen Sie im ersten Schritt eine neue Instanz des ADO Connection-Objektes und weisen der Eigenschaft `Provider` den Wert `ADsDSOObject` und der Eigenschaft `Open` den Wert `Active Directory Provider` zu.

Anschließend führen Sie mit der `Execute`-Methode eine Suche durch und übergeben mit dem ersten Parameter den Pfad zum Container `Configuration` (`strADsPath`), mit dem zweiten Parameter den anzuwendenden Filter (`strFilter`), mit dem dritten Parameter das auszulesende Attribut (`strAttributes`) und mit dem letzten Parameter den Suchbereich (`strScope`).

Zu guter Letzt positionieren Sie den `RecordSet`-Datensatzzeiger auf den ersten Datensatz, fragen den Wert des Feldes mit dem Index 0 ab und geben den DNS-Namen des Domänen-Controllers aus, auf dem der globale Katalog aktiviert ist.

```
Option Explicit
```

```
Dim oRootDSE
Dim oConn
Dim oRS
Dim oNTDS
Dim oServer
Dim strDomain
Dim strADsPath
Dim strFilter
Dim strAttributes
Dim strScope
```

```
strDomain = "MyDomain.com"
Set oRootDSE = GetObject("LDAP://" & strDomain & "/" & "RootDSE")
strADsPath = "<LDAP://" & oRootDSE.Get("ConfigurationNamingContext") & ">";
strFilter = "(&(ObjectCategory=NTDSDSA)(Options=1));"
strAttributes = "DistinguishedName;"
```

Listing 567: 2_07_005a.vbs

```

strScope = "SubTree"

Set oConn = CreateObject("ADODB.Connection")
oConn.Provider = "AdsDSOObject"
oConn.Open "Active Directory Provider"
Set oRS = oConn.Execute(strADsPath & strFilter & strAttributes & strScope)
oRS.MoveFirst

While Not oRS.EOF
    Set oNTDS = GetObject("LDAP://" & oRS.Fields(0).Value)
    Set oServer = GetObject(oNTDS.Parent)
    Wscript.Echo oServer.Get("DNSHostName")
    oRS.MoveNext
Wend

Set oRootDSE = Nothing
Set oConn = Nothing
Set oRS = Nothing
Set oNTDS = Nothing
Set oServer = Nothing

```

Listing 567: 2_07_005a.vbs (Forts.)

Auf die gleiche Art und Weise können Sie auch den Domänen-Controller mit dem globalen Katalog innerhalb eines Standortes ermitteln, wobei Sie die Suche aber nicht direkt im Container Configuration beginnen, sondern im Container Servers des entsprechenden Standorts.

```

Option Explicit

Dim oRootDSE
Dim oConn
Dim oRS
Dim oNTDS
Dim oServer
Dim strSite
Dim strForest
Dim strADsPath
Dim strFilter
Dim strAttributes
Dim strScope

strSite = "Standardname-des-ersten-Standorts"
strForest = "MyDomain.com"

```

Listing 568: 2_07_005b.vbs

```

Set oRootDSE = GetObject("LDAP://" & strForest & "/RootDSE")
strADsPath = "<LDAP://CN=servers,CN=" & strSite & ",CN=sites," & _
    oRootDSE.Get("ConfigurationNamingContext") & ">";"
strFilter = "(ObjectCategory=NTDSDSA);"
strAttributes = "DistinguishedName;"
strScope = "SubTree"

Set oConn = CreateObject("ADODB.Connection")
oConn.Provider = "ADsDSOObject"
oConn.Open "Active Directory Provider"
Set oRS = oConn.Execute(strADsPath & strFilter & strAttributes & strScope)
oRS.MoveFirst

strFilter = "(&(ObjectCategory=NTDSDSA)(Options=1));"
Set oRS = oConn.Execute(strADsPath & strFilter & strAttributes & strScope)
oRS.MoveFirst

While Not oRS.EOF
    Set oNTDS = GetObject("LDAP://" & oRS.Fields(0).Value)
    Set oServer = GetObject(oNTDS.Parent)
    Wscript.Echo oServer.Get("DNSHostName")
    oRS.MoveNext
Wend

Set oRootDSE = Nothing
Set oConn = Nothing
Set oRS = Nothing
Set oNTDS = Nothing
Set oServer = Nothing

```

Listing 568: 2_07_005b.vbs (Forts.)

161 Globalen Katalog aktivieren und deaktivieren

Um auf einem bestimmten Domänen-Controller den globalen Katalog zu aktivieren oder zu deaktivieren, nehmen Sie unter grafischer Oberfläche wieder die Management-Konsole ACTIVE DIRECTORY-STANDORTE UND -DIENSTE zu Hilfe. Klicken Sie hierzu auf START, dann auf VERWALTUNG und anschließend auf ACTIVE DIRECTORY-STANDORTE UND -DIENSTE, und erweitern Sie unterhalb des entsprechenden Standortes den Domänen-Controller, auf dem Sie den globalen Katalog aktivieren oder deaktivieren möchten. Wählen Sie im Kontextmenü von NTDS SETTINGS (unterhalb des Domänen-Controllers) die Eigenschaften aus, und aktivieren (oder deaktivieren) Sie auf der Registerkarte ALLGEMEIN die Option GLOBALER KATALOG.

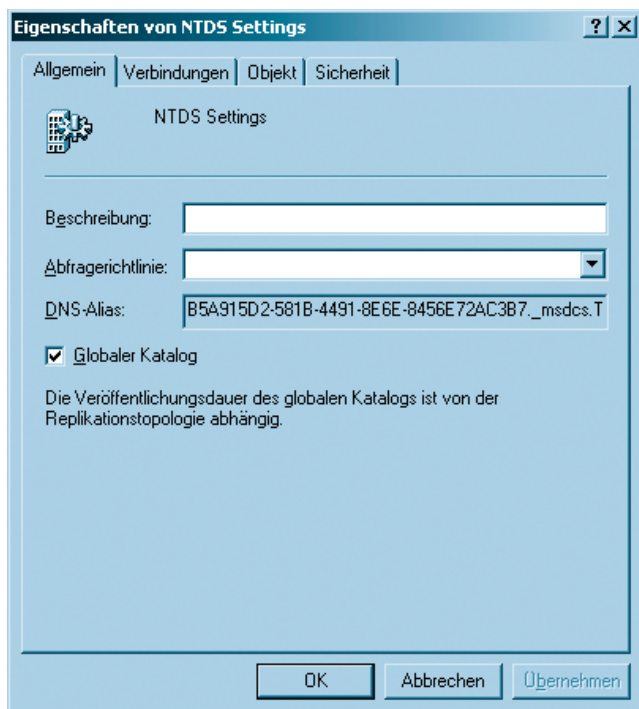


Abbildung 212: Den globalen Katalog aktivieren

Die Lösung per Eingabeaufforderung

Wenn Sie den globalen Katalog von der Eingabeaufforderung aktivieren oder deaktivieren möchten, so können Sie hierfür den `Dsmod Server`-Befehl mit dem Parameter `-isgc` verwenden. Der `Dsmod Server`-Befehl erwartet dabei den definierten Namen des Servers als Parameter wie auch den Parameter `-isgc`, dem Sie den Wert `Yes` (globalen Katalog aktivieren) oder `No` (globalen Katalog deaktivieren) übergeben.

Die nachfolgende `Dsmod Server`-Befehlszeile aktiviert den globalen Katalog auf dem Domänen-Controller *FRA-MD-DC-003*.

```
@Echo off
Dsmod Server "CN=FRA-MD-DC-003,CN=Servers,CN=Standardname-des-ersten-
Standorts, CN=Sites, CN=Configuration, DC=MyDomain, DC=Com" -isgc Yes
```

Listing 569: 2_07_006.bat

Um den globalen Katalog zu deaktivieren, wenden Sie die gleiche `Dsmod Server`-Befehlszeile an, übergeben dem Parameter `-isgc` jedoch den Wert `No`.

Die Lösung mit VBScript

Um den globalen Katalog per VBScript zu aktivieren, stellen Sie im ersten Schritt eine LDAP-Verbindung zum Root des Domänen-Controllers, dann zum Attribut `DSServiceName` (dieses Attribut enthält den definierten Namen der NTDS-Einstellungen) und daraufhin zu den eigentlichen NTDS-Einstellungen her. Anschließend wenden Sie die `Put`-Methode an und weisen dem Attribut `Options` den Wert 1 zu. Die geänderte Einstellung schreiben Sie mit der `SetInfo`-Methode zurück, um sie dauerhaft zu speichern.

Option Explicit

```
Dim oRootDSE
Dim oDSRoot
Dim oDSName
Dim blnOptions
Dim strComputer
```

```
strComputer = "FRA-MD-DC-003"
Set oRootDSE = GetObject("LDAP://" & strComputer & "/RootDSE")
oDSName = oRootDSE.Get("DSServiceName")
Set oDSRoot = GetObject("LDAP://" & strComputer & "/" & oDSName)
oDSRoot.Put "Options", 1
oDSRoot.SetInfo
```

```
Set oRootDSE = Nothing
Set oDSRoot = Nothing
Set oDSName = Nothing
```

Listing 570: 2_07_006.vbs

Um den globalen Katalog zu deaktivieren, können Sie das gleiche Listing verwenden, wobei Sie dem Attribut `Options` jedoch den Wert 0 zuweisen müssen.

162 Inhaber der FSMO-Rollen suchen und finden

Innerhalb einer Gesamtstruktur gibt es fünf verschiedene Betriebsmaster-Funktionen (engl. = *Flexible Single Master Operations*), die einem (oder auch mehreren) Domänen-Controller(n) zugewiesen werden. Folgende Betriebsmaster-Funktionen (von nun an *FSMO-Rollen* genannt) werden verwendet:

- **RID-Master:** Die FSMO-Rolle *RID-Master* ist auf dem Domänen-Controller zu finden, der für die Verarbeitung von RID-Poolanforderungen aller Domänen-Controller innerhalb einer Domäne verantwortlich ist. Es gibt pro Domäne einen RID-Master.

- ▶ **PDC-Emulator:** Die FSMO-Rolle *PDC-Emulator* wird auf dem Domänen-Controller ausgeführt, der sich gegenüber Clients, Member-Servern und Domänen-Controllern älterer Betriebssysteme als PDC (*Primary Domain Controller*) ausgibt. Es gibt einen PDC-Emulator pro Domäne.
- ▶ **Infrastruktur-Master:** Die FSMO-Rolle des *Infrastruktur-Masters* wird auf dem Domänen-Controller ausgeführt, der für die Aktualisierung der SID und des definierten Namen eines Objektes in einem domänenübergreifenden Objektverweis verantwortlich ist. Es gibt einen Infrastruktur-Master pro Domäne.
- ▶ **Domänennamen-Master:** Die FSMO-Rolle *Domänennamen-Master* wird auf dem Domänen-Controller ausgeführt, der für die Änderungen an dem Domänenname-space, der für die Gesamtstruktur gilt, verantwortlich ist. Es gibt einen Domänennamen-Master pro Gesamtstruktur.
- ▶ **Schemamaster:** Die FSMO-Rolle *Schemamaster* wird auf dem Domänen-Controller ausgeführt, der für die Aktualisierung des Verzeichnisschemas verantwortlich ist. Es gibt einen Schemamaster pro Gesamtstruktur.

Um nun den RID-, PDC- und Infrastruktur-Master einer Domäne zu ermitteln, rufen Sie die Management-Konsole ACTIVE DIRECTORY-BENUTZER UND -COMPUTER auf. Klicken Sie hierzu auf START, erweitern Sie VERWALTUNG, und führen Sie einen Klick auf den Eintrag ACTIVE DIRECTORY-BENUTZER UND -COMPUTER aus, oder wählen Sie START, dann AUSFÜHREN, und geben Sie `Dsa.msc` ein, womit Sie die Management-Konsole auch direkt aufrufen können.



Abbildung 213: Den RID-Master ermitteln

Markieren Sie im Konsolenstamm (linke Fensterhälfte) die betreffende Domäne, und wählen Sie im Kontextmenü der Domäne BETRIEBSMASTER... aus. Sie gelangen daraufhin in das Dialogfenster BETRIEBSMASTER und können der Registerkarte RID den DNS-Namen des RID-Masters, der Registerkarte PDC den DNS-Namen des PDC-Emulators und der Registerkarte INFRASTRUKTUR den DNS-Namen des Infrastruktur-Masters entnehmen (siehe Abbildung 213).

Der Schemamaster einer Gesamtstruktur lässt sich dagegen mit der Management-Konsole ACTIVE DIRECTORY-SCHEMA ermitteln, wobei Sie diese Management-Konsole vor der ersten Verwendung zunächst registrieren müssen, da sie per Default nicht aktiviert wird. Klicken Sie daher auf START, dann auf AUSFÜHREN, und geben Sie `Regsvr32 Schmmgmt.dll` ein, woraufhin die zuvor genannte Management-Konsole registriert und damit auch aktiviert wird.

Rufen Sie anschließend mit MMC eine leere Management-Konsole auf, klicken Sie dann auf DATEI, und wählen Sie SNAP-IN HINZUFÜGEN/ENTFERNEN... aus. Klicken Sie auf HINZUFÜGEN, und wählen Sie ACTIVE DIRECTORY-SCHEMA aus, um die soeben registrierte Management-Konsole dem noch leeren Konsolenstamm hinzuzufügen, und kehren Sie dann wieder zur eigentlichen Management-Konsole zurück.

Im Konsolenstamm markieren Sie ACTIVE DIRECTORY-SCHEMA und öffnen dessen Kontextmenü. Dort wählen Sie BETRIEBSMASTER... aus, woraufhin Sie in das Dialogfenster SCHEMAMASTER ÄNDERN gelangen. Dem Feld AKTUELLER SCHEMAMASTER (ONLINE) können Sie nun den DNS-Namen des Domänen-Controllers entnehmen, der die Rolle des Schemamasters innerhalb der Gesamtstruktur hat.

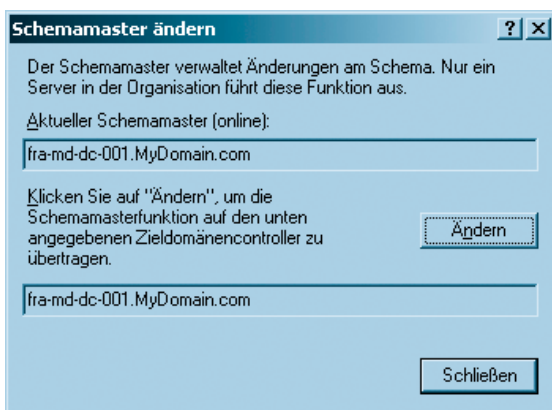


Abbildung 214: Den Schemamaster ermitteln

Um den Domänennamen-Master der Gesamtstruktur herauszufinden, nehmen Sie die Management-Konsole ACTIVE DIRECTORY-DOMÄNEN UND -VERTRAUENSSTELLUNGEN zu Hilfe, die Sie mit einem Klick auf START und dann auf VERWALTUNG per Klick öffnen oder mit `Domain.msc` auch direkt aufrufen können.

Markieren Sie im Konsolenstamm den obersten Eintrag mit der Bezeichnung ACTIVE DIRECTORY-DOMÄNEN UND -VERTRAUENSSTELLUNGEN, und wählen Sie im Kontextmenü den Eintrag BETRIEBSMASTER... aus. Daraufhin wird das Dialogfenster BETRIEBSMASTER ÄNDERN geöffnet, und Sie können dem Feld DOMÄNENNAMEN-BETRIEBSMASTER den DNS-Namen des Domänen-Controllers entnehmen, der innerhalb der Gesamtstruktur als Domänennamen-Master fungiert.

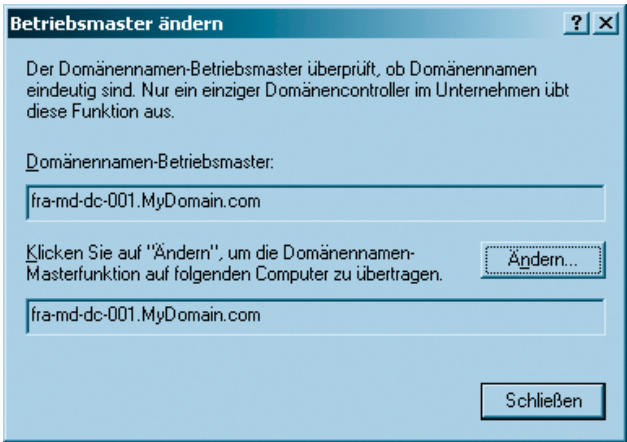


Abbildung 215: Den Domänennamen-Master anzeigen lassen

Die Lösung per Eingabeaufforderung

Um die Inhaber der FSMO-Rollen zu finden, können Sie in der Eingabeaufforderung den `Dsquery Server`-Befehl verwenden, dem Sie den Parameter `-hasfsmo` unter Angabe der entsprechenden Rolle anfügen. Folgende Parameter sind möglich:

Rolle	Parameter
Schemamaster	Schema
Domänennamen-Master	Name
Infrastrukturmaster	Infr
PDC-Emulator	PDC
RID-Master	RID

Tabelle 75: Die Parameter der einzelnen FSMO-Rollen

Wenn Sie beispielsweise den Schemamaster der Gesamtstruktur suchen, so können Sie hierfür die folgende `Dsquery Server`-Befehlszeile verwenden, die den definierten Namen des aktuellen Schemamasters und damit auch den Hostnamen des Domänen-Controllers am Bildschirm ausgibt:


```
@Echo off
Dsquery Server -hasfsmo Schema
```

Listing 571: 2_07_007a.bat

Möchten Sie dagegen die Inhaber aller FSMO-Rollen ermitteln und das Ergebnis nicht am Bildschirm ausgeben, sondern in eine Datei schreiben, so können Sie die nachfolgenden Dsquery Server-Befehlszeilen verwenden:

```
@Echo off
Dsquery Server -hasfsmo Schema > Result.txt
Dsquery Server -hasfsmo Name >> Result.txt
Dsquery Server -hasfsmo Infr >> Result.txt
Dsquery Server -hasfsmo PDC >> Result.txt
Dsquery Server -hasfsmo RID >> Result.txt
```

Listing 572: 2_07_007b.bat

Die Lösung mit VBScript

Um die Inhaber der FSMO-Rollen per VBScript zu ermitteln, stellen Sie lediglich eine LDAP-Verbindung zur jeweiligen Objekt-Klasse her und fragen dann den Wert des Attributes `FSMORoleOwner` ab, woraufhin Sie den definierten Namen des entsprechenden Domänen-Controllers zurückerhalten.

Das nachfolgende Listing zeigt ein Beispiel dafür, wie Sie die Inhaber der fünf FSMO-Rollen ausfinden machen können. Zunächst wird der PDC-Emulator der aktuellen Domäne ermittelt, wofür eine LDAP-Verbindung zur Domäne hergestellt und der Wert des Attributes `FSMORoleOwner` ermittelt wird.

Daraufhin folgt das Abfragen des aktuellen RID-Masters. Diese Abfrage setzt eine Verbindung zur Objekt-Klasse `RIDManager` voraus, die im Container `System` der Domäne abgelegt ist. Mit dem Abfragen des Attributes `FSMORoleOwner` wird der definierte Name des aktuellen RID-Masters zurückgeliefert.

Der Infrastruktur-Master lässt sich ermitteln, indem Sie eine LDAP-Verbindung zu `Infrastructure` (dieses Objekt liegt im Root der Domäne) herstellen und den Wert des Attributes `FSMORoleOwner` ermitteln, woraufhin Sie den definierten Namen des aktuellen Infrastruktur-Masters zurückerhalten.

Um den Schemamaster der Gesamtstruktur herauszufinden, stellen Sie eine Verbindung zum Schema her und fragen wieder den Wert des Attributes `FSMORoleOwner` ab. Und last but not least, ermitteln Sie noch den Domänennamen-Master, wofür Sie eine LDAP-Verbindung zum Container `Partitions` (dieses Objekt liegt unterhalb von `Configuration`) herstellen und wieder den Wert des Attributes `FSMORoleOwner` ermitteln.

Option Explicit

```
Dim oRootDSE
Dim oDomainDN
Dim oSchemaDN
Dim oConfigDN
Dim oPDC
Dim oRID
Dim oSchema
Dim oInfra
Dim oName
Dim strDomain
```

```
strDomain = "MyDomain.com"
Set oRootDSE = GetObject("LDAP://" & strDomain & "/RootDSE")
oDomainDN = oRootDSE.Get("DefaultNamingContext")
oSchemaDN = oRootDSE.Get("SchemaNamingContext")
oConfigDN = oRootDSE.Get("ConfigurationNamingContext")
```

```
Set oPDC = GetObject("LDAP://" & oDomainDN)
Wscript.Echo "PDC Emulator: " & oPDC.fsmoroleowner & VBCRLF
```

```
Set oRID = GetObject("LDAP://CN=RID Manager$,CN=System," & oDomainDN)
Wscript.Echo "RID Master: " & oRID.fsmoroleowner & VBCRLF
```

```
Set oInfra = GetObject("LDAP://CN=Infrastructure," & oDomainDN)
Wscript.Echo "Infrastructure Master: " & oInfra.fsmoroleowner & VBCRLF
```

```
Set oSchema = GetObject("LDAP://" & oSchemaDN)
Wscript.Echo "Schema Master: " & oSchema.fsmoroleowner & VBCRLF
```

```
Set oName = GetObject("LDAP://CN=Partitions," & oConfigDN)
Wscript.Echo "Domain Naming Master: " & oName.fsmoroleowner & VBCRLF
```

```
Set oRootDSE = Nothing
Set oDomainDN = Nothing
Set oSchemaDN = Nothing
Set oConfigDN = Nothing
Set oPDC = Nothing
Set oRID = Nothing
Set oSchema = Nothing
Set oInfra = Nothing
Set oName = Nothing
```

163 FSMO-Rollen auf andere Server übertragen

Bedingt durch eine Veränderung innerhalb der IT-Infrastruktur kann es erforderlich sein, die FSMO-Rollen auf andere Domänen-Controller zu übertragen. Dies kann beispielsweise der Fall sein, wenn der Domänen-Controller, der zurzeit Inhaber einer FSMO-Rolle ist, durch ein anderes Computersystem ersetzt werden soll.

Um die FSMO-Rollen *RID-Master*, *PDC-Emulator* und *Infrastruktur-Master* auf einen anderen Domänen-Controller zu übertragen, öffnen Sie zunächst die Management-Konsole ACTIVE DIRECTORY-BENUTZER UND -COMPUTER. Klicken Sie hierzu auf START, wechseln Sie dann in das Menü VERWALTUNG, und klicken Sie dort den gleichnamigen Menüeintrag an, oder klicken Sie im Startmenü auf AUSFÜHREN, und geben Sie `Dsa.msc` ein, womit Sie die zuvor genannte Management-Konsole auch direkt aufrufen können.

Markieren Sie im Konsolenstamm die Domäne, und wählen Sie im Kontextmenü den Eintrag VERBINDUNG MIT DOMÄNENCONTROLLER HERSTELLEN... aus. Markieren Sie den Domänen-Controller, der die zuvor genannte(n) Rolle(n) übernehmen soll, und klicken Sie auf OK. Markieren Sie daraufhin die Domäne im Konsolenstamm erneut, und wählen Sie im Kontextmenü den Eintrag BETRIEBSMASTER... aus, um das Dialogfenster Betriebsmaster zu öffnen.

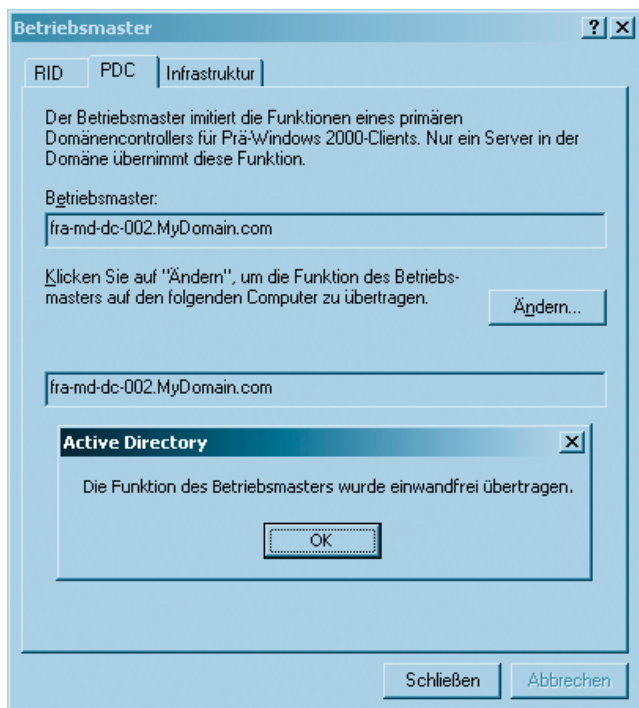


Abbildung 216: Die FSMO-Rolle »PDC-Emulator« übertragen

Je nachdem, welche FSMO-Rolle Sie auf den zuvor verbundenen Domänen-Controller übertragen möchten, aktivieren Sie nun die entsprechende Registerkarte (RID, PDC oder INFRASTRUKTUR) und klicken dann auf ÄNDERN. Sie werden daraufhin gefragt, ob Sie die Funktion des Betriebsmasters auch wirklich ändern wollen. Klicken Sie auf JA, um die Rolle auf den neuen Domänen-Controller zu übertragen, und schließen Sie das Dialogfenster wieder.

Auf ähnliche Art und Weise lässt sich auch die FSMO-Rolle *Domänennamen-Master* auf einen anderen Domänen-Controller übertragen. Rufen Sie hierzu die Management-Konsole ACTIVE DIRECTORY-DOMÄNEN UND -VERTRAUENSSTELLUNGEN auf, die Sie im Startmenü (Menü VERWALTUNG) anklicken oder mit `Domain.msc` auch direkt aufrufen können.

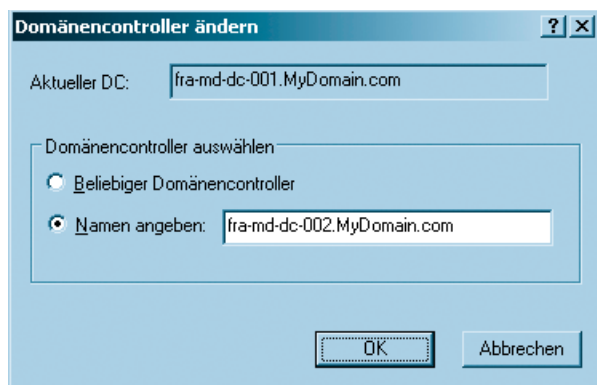


Abbildung 217: Den DNS-Hostnamen des neuen Domänen-Controllers angeben

Markieren Sie in der linken Fensterhälfte den Namen der Management-Konsole (ACTIVE DIRECTORY-DOMÄNEN UND -VERTRAUENSSTELLUNGEN), und wählen Sie im Kontextmenü den Eintrag VERBINDUNG MIT DOMÄNENCONTROLLER HERSTELLEN... aus. Sie gelangen daraufhin in ein Dialogfenster, worüber Sie den Domänen-Controller, auf den Sie die FSMO-Rolle Domänennamen-Master übertragen möchten, auswählen müssen.

Nehmen Sie die Auswahl vor, und klicken Sie anschließend auf OK. Markieren Sie im Konsolenstamm der linken Fensterhälfte erneut den Namen der Management-Konsole, und wählen Sie diesmal im Kontextmenü den Eintrag BETRIEBSMASTER... aus. Klicken Sie auf ÄNDERN, um die FSMO-Rolle Domänennamen-Master auf den zuvor ausgewählten Domänen-Controller zu übertragen, bestätigen Sie die Sicherheitsmeldung mit einem Klick auf JA, und schließen Sie das Dialogfenster wieder.

Auch die Rolle des *Schemamasters* ist schnell übertragen. Hierzu benötigen Sie jedoch die Management-Konsole ACTIVE DIRECTORY-SCHEMA, die Sie vor dem ersten Aufruf erst von Hand registrieren müssen. Öffnen Sie also das Startmenü, klicken Sie auf AUSFÜHREN, und geben Sie den Befehl `Regsvr32 Schmmgmt.dll` ein. Öffnen Sie anschließend mit MMC eine leere Management-Konsole, und fügen Sie dem Konsolenstamm das Snap-In ACTIVE DIRECTORY-SCHEMA hinzu, das Sie nach der Registrierung nun in der Übersicht der verfügbaren Snap-Ins finden.

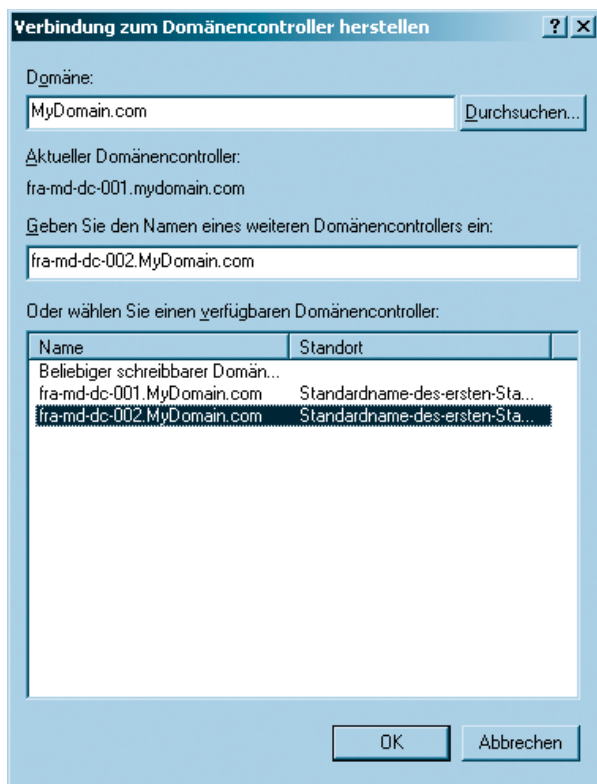


Abbildung 218: Den künftigen Domännennamen-Master auswählen

Markieren Sie in der linken Fensterhälfte den Eintrag **ACTIVE DIRECTORY-SCHEMA**, und wählen Sie im Kontextmenü den Eintrag **DOMÄNENCONTROLLER ÄNDERN...** aus. Daraufhin wird ein Dialogfenster geöffnet, worüber Sie auswählen können, ob die FSMO-Rolle auf einen beliebigen oder (durch Angabe des DNS-Hostnamens) auf einen bestimmten Domänen-Controller übertragen werden soll. Wenn Sie den DNS-Hostnamen des künftigen *Schemamasters* selbst bestimmen möchten, so aktivieren Sie die Option **NAMEN ANGEBEN** und hinterlegen den DNS-Hostnamen des entsprechenden Domänen-Controllers. Klicken Sie anschließend auf **OK**, um die Rolle zu übertragen.

Die Lösung per Eingabeaufforderung

Mit Hilfe des `Ntdsutil`-Befehls lassen sich die fünf FSMO-Rollen auch von der Eingabeaufforderung aus auf einen anderen Domänen-Controller transferieren. Wenn Sie `Ntdsutil` dabei interaktiv nutzen möchten, so wechseln Sie durch Eingabe von `Ntfdutil` zunächst in die `Ntdsutil`-spezifische Eingabeaufforderung, wechseln dann mit `Roles` in den *Roles*-Kontext, geben erst `Connections` und anschließend `Connect to Server` `Hostname` (`Hostname` ist hierbei durch den Namen oder durch die IP-Adresse des künftigen Rolleninhabers zu ersetzen) ein, um eine Verbindung zu dem Domänen-Controller herzustellen, auf den Sie eine FSMO-Rolle übertragen möchten.

Anschließend wechseln Sie mit Q eine Ebene höher und geben Transfer Rolle (Rolle ist durch den entsprechenden Ntdsutil-Parameter zu ersetzen, siehe nachfolgende Tabelle) ein, um eine bestimmte FSMO-Rolle auf den zuvor verbundenen Domänen-Controller zu übertragen. Folgende Rollen können transferiert werden:

FSMO-Rolle	Ntdsutil-Parameter
Schemamaster	Schema Master
Domänennamen-Master	Domain Naming Master
Infrastrukturmaster	Infrastructure Master
PDC-Emulator	PDC
RID-Master	RID

Tabelle 76: Die FSMO-Rollen und die Ntdsutil-spezifischen Bezeichnungen

Wenn Sie beispielsweise die FSMO-Rolle RID Master übertragen möchten, so geben Sie Transfer RID ein. Sie erhalten daraufhin eine Meldung und müssen den Transfer der Rolle durch einen Klick auf JA bestätigen. Diese Interaktion können Sie umgehen, indem Sie den Ntdsutil-eigenen Subbefehl Popups Off eingeben, womit Sie keine Aktionen mehr bestätigen müssen.

Der Einsatz des Ntdsutil-Befehls setzt nicht unbedingt die Nutzung des interaktiven Modus voraus. Denn Sie können alle Ntdsutil-spezifischen Subbefehle, die Sie zur Übertragung einer FSMO-Rolle eingeben müssen, auch in einer Textdatei hinterlegen und den Ntdsutil-Befehl dazu veranlassen, die Befehle aus einer externen Datei einzulesen.

Die nachfolgende Ntdsutil-Befehlszeile verwendet die Eingabeumleitung, woraufhin alle Ntdsutil-spezifischen Befehle nicht von der Tastatur erwartet, sondern aus der angegebenen Datei eingelesen und anschließend Zeile für Zeile verarbeitet werden.

```
@Echo off
Ntdsutil < Transfer_RID.txt
```

Listing 574: 2_07_008.bat

Die Datei Transfer_RID.txt enthält den folgenden Inhalt, den Sie nur geringfügig anpassen müssen, um auch andere Rollen übertragen zu können(lediglich der Host-name des Servers und die FSMO-Rolle müssen ersetzt werden):

```
Popups Off
Roles
Connections
Connect to Server FRA-MD-DC-002
```

Listing 575: Transfer_RID.txt

Q
Transfer RID Master

Listing 575: Transfer_RID.txt (Forts.)

Die Lösung mit VBScript

Da der Ntdsutil-Befehl der Eingabeaufforderung eine sehr komfortable Möglichkeit des FSMO-Rollentransfers darstellt, ist es nahe liegend, diesen Befehl auch für eine VBScript-basierte Lösung einzusetzen. Denn mit der Run-Methode des Shell-Objektes lässt sich dieser Befehl auch aus einem VBScript heraus aufrufen.

Das nachfolgende Listing führt beispielsweise den Ntdsutil-Befehl aus und verwendet die Eingabeumleitung, um die Ntdsutil-spezifischen Subbefehle aus der Textdatei Transfer_RID.txt zu lesen (die Textdatei hat dabei den gleichen Inhalt wie die gleichnamige Datei aus dem vorherigen Abschnitt).

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Ntdsutil < Transfer_RID.txt"

Set oShell = Nothing
```

Listing 576: 2_07_008.vbs

164 Welche Vertrauensstellungen sind eingerichtet?

Mit Hilfe von Vertrauensstellungen können Sie eine Verbindung zwischen zwei (oder wahlweise auch mehreren) Domänen herstellen, womit den Benutzern aus der einen Domäne die Ressourcen der anderen Domäne zur Verfügung gestellt werden können. Vertrauensstellungen können dabei unidirektional (ein- oder ausgehend) wie auch bidirektional sein. Wenn Sie beispielsweise eine ausgehende Vertrauensstellung von der Domäne A zur Domäne B einrichten, dann können sich die Benutzer der Domäne A gegenüber der Domäne B authentifizieren und (je nach Berechtigung) auf die Ressourcen der Domäne B zugreifen. Die Benutzer der Domäne B dagegen können sich nicht der Domäne A gegenüber authentifizieren, da die Vertrauensstellung nur unidirektional ist. Etwas anders sieht es aus, wenn zwischen der Domäne A und der Domäne B eine bidirektionale Vertrauensstellung eingerichtet wird. Denn dann können sich sowohl die Benutzer aus der Domäne A der Domäne B gegenüber und die Benutzer der Domäne B der Domäne A gegenüber authentifizieren und jeweils auf die Ressourcen der anderen Domäne zugreifen. Sofern die erteilten Berechtigungen dies natürlich zulassen.

Wenn Sie nun unter grafischer Oberfläche herausfinden möchten, welche Vertrauensstellungen bereits eingerichtet wurden, so öffnen Sie die Management-Konsole ACTIVE DIRECTORY-DOMÄNEN UND -VERTRAUENSSTELLUNGEN (klicken Sie hierzu auf START, und erweitern Sie VERWALTUNG), wählen im Kontextmenü der Domäne EIGENSCHAFTEN aus und wechseln auf die Registerkarte VERTRAUENSSTELLUNGEN.

Sie gelangen daraufhin in das Dialogfenster EIGENSCHAFTEN VON DOMÄNENNAME und können dem oberen Feld alle eingehenden und dem unteren Feld alle ausgehenden Vertrauensstellungen entnehmen.

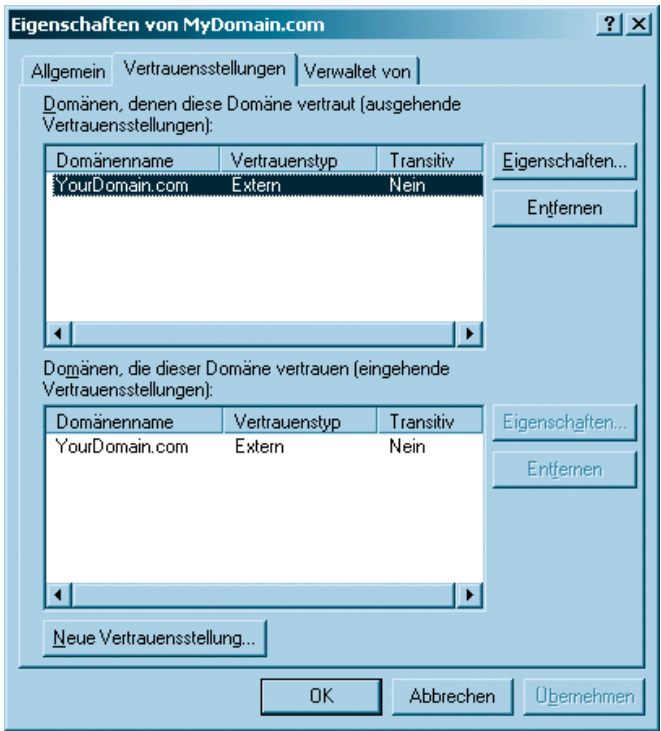


Abbildung 219: Die Vertrauensstellungen anzeigen lassen

Die Lösung per Eingabeaufforderung

Mit dem NetDom-Befehl, der Bestandteil der Windows Server 2003 Support-Tools ist, lassen sich die bereits eingerichteten Vertrauensstellungen auch von der Eingabeaufforderung aus abfragen. Hierzu wenden Sie lediglich den NetDom-Befehl mit dem Parameter Query Trust an, woraufhin alle bidirektionalen, eingehenden und auch ausgehenden Vertrauensstellungen unter Angabe der Richtung, der Zieldomäne und auch des Vertrauensstellungstyps am Bildschirm angezeigt werden.


```
@Echo off
NetDom Query Trust
```

Listing 577: 2_07_009.bat

Die Lösung mit VBScript

Um die bereits eingerichteten Vertrauensstellungen per VBScript abzufragen, stellen Sie zunächst eine LDAP-Verbindung zum Root der Domäne her, selektieren dann mit Hilfe der `Filter`-Funktion die Vertrauensstellungen und fragen dann für jedes zurückgelieferte Objekt die Werte der Eigenschaften `TrustPartner` (Name der Zieldomäne), `TrustDirection` (Richtung der Vertrauensstellung), `TrustType` (Art der Vertrauensstellung) und `TrustAttributes` (Eigenschaft der Vertrauensstellung) ab, die Sie entweder mit der `WScript.Echo`-Anweisung am Bildschirm ausgeben oder per `FileSystemObject` auch in eine Datei schreiben können.

Der Wert der Eigenschaft `TrustPartner` wird dabei als Zeichenfolge zurückgegeben und gibt Auskunft über den Namen der Domäne. Die Werte der Eigenschaften `TrustDirection`, `TrustType` und `TrustAttributes` werden dagegen als dezimale Werte zurückgeliefert. Die nachfolgende Tabelle zeigt, welche Werte die Eigenschaft `TrustDirection` besitzen kann:

Richtung	Wert
TRUST_DIRECTION_DISABLED	0
TRUST_DIRECTION_INBOUND	1
TRUST_DIRECTION_OUTBOUND	2
TRUST_DIRECTION_BIDIRECTIONAL	3

Tabelle 77: Die gültigen Werte für die Eigenschaft »TrustDirection«

Der nachfolgenden Tabelle können Sie die gültigen Werte und auch die jeweiligen Bedeutungen der Eigenschaft `TrustType` entnehmen.

Typ	Wert
TRUST_TYPE_DOWNLEVEL	1
TRUST_TYPE_UPLEVEL	2
TRUST_TYPE_MIT	3
TRUST_TYPE_DCE	4

Tabelle 78: Die gültigen Werte für die Eigenschaft »TrustType«

Auch die Eigenschaft `TrustAttributes` liefert einen dezimalen Wert zurück. Welche Werte gültig sind, zeigt die nachfolgende Tabelle.

Eigenschaft	Wert
TRUST_ATTRIBUTE_NON_TRANSITIVE	1
TRUST_ATTRIBUTE_UPLEVEL_ONLY	2
TRUST_ATTRIBUTE_QUARANTINED_DOMAIN	4
TRUST_ATTRIBUTE_FOREST_TRANSITIVE	8
TRUST_ATTRIBUTE_CROSS_ORGANIZATION	16
TRUST_ATTRIBUTE_WITHIN_FOREST	32
TRUST_ATTRIBUTE_TREAT_AS_EXTERNAL	64

Tabelle 79: Die gültigen Werte für die Eigenschaft »TrustAttributes«

Das nachfolgende Listing zeigt ein Beispiel dafür, wie Sie per VBScript die Vertrauensstellung der Domäne (strDomain) abfragen und die zuvor genannten Eigenschaften mit der WScript-Echo-Anweisung am Bildschirm ausgeben können.

```
Option Explicit

Dim oRootDSE
Dim oTrusts
Dim strDomain
Dim objItem

strDomain = "MyDomain.com"
Set oRootDSE = GetObject("LDAP://" & strDomain & "/RootDSE")
Set oTrusts = GetObject("LDAP://CN=System," & _
    oRootDSE.Get("DefaultNamingContext"))
oTrusts.Filter = Array("TrustedDomain")

For Each objItem in oTrusts
    WScript.Echo "Trusted Domain(s): " & objItem.TrustPartner
    WScript.Echo "Direction: " & objItem.TrustDirection
    WScript.Echo "Type: " & objItem.TrustType
    WScript.Echo "Attributes: " & objItem.TrustAttributes & VBCRLF
Next

Set oRootDSE = Nothing
Set oTrusts = Nothing
```

Listing 578: 2_07_009.vbs

165 Neue Vertrauensstellungen einrichten

Um beispielsweise zwischen der lokalen Domäne *MyDomain.com* und der Zieldomäne *YourDomain.com* eine bidirektionale Vertrauensstellung vom Typ *Extern* einzurichten, rufen Sie zunächst die Management-Konsole ACTIVE DIRECTORY-DOMÄNEN UND -VERTRAUENSSTELLUNGEN auf. Sie finden die Management-Konsole im Startmenü (unterhalb von VERWALTUNG).

Markieren Sie im Konsolenstamm die Domäne, und rufen Sie aus dem Kontextmenü heraus die EIGENSCHAFTEN auf. Wechseln Sie anschließend auf die Registerkarte VERTRAUENSSTELLUNGEN, und klicken Sie auf NEUE VERTRAUENSSTELLUNG... Daraufhin wird der ASSISTENT FÜR NEUE VERTRAUENSSTELLUNGEN geöffnet, schließen Sie den Willkommensbildschirm mit einem Klick auf WEITER. Geben Sie im Dialogfenster VERTRAUENSSTELLUNGSNAME den DNS-Namen der Zieldomäne (in diesem Fall *YourDomain.com*) ein, und klicken Sie auf WEITER.

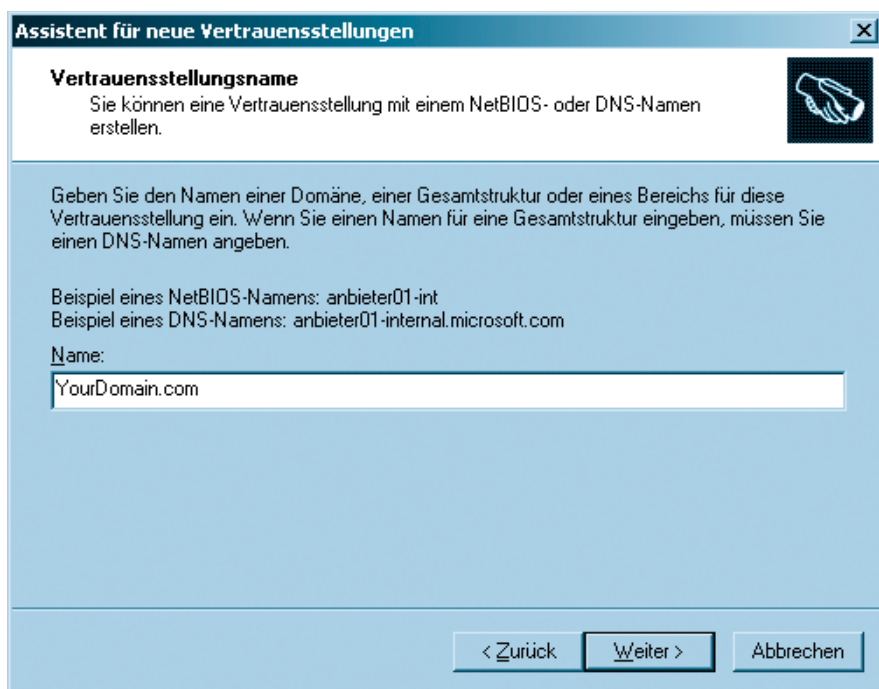


Abbildung 220: Den DNS-Namen der externen Zieldomäne festlegen

Sie gelangen nun in das Dialogfenster RICHTUNG DER VERTRAUENSSTELLUNG und müssen nun festlegen, ob die Vertrauensstellung *bidirektional* ist und damit in beide Richtungen erfolgen soll oder aber *unidirektional* ist und ausschließlich *eingehend* oder aber *ausgehend* ist. Legen Sie die gewünschte Richtung fest, und klicken Sie anschließend auf WEITER, um mit der Auswahl der Vertrauensstellungsseite fortzufahren. Folgende Optionen sind möglich:

- ▶ **NUR FÜR DIESE DOMÄNE:** Aktivieren Sie diese Option, wenn Sie die Vertrauensstellung nur in der lokalen Domäne konfigurieren möchten. Die Vertrauensstellung muss damit auch in der Zieldomäne durch den dortigen Administrator erstellt werden. Mit der Auswahl dieser Option müssen Sie ein Vertrauensstellungs-Kennwort vergeben, das auch beim Einrichten der Vertrauensstellung innerhalb der Zieldomäne verwendet werden muss.
- ▶ **FÜR DIESE DOMÄNE UND DIE ANGEGEBENE DOMÄNE:** Wählen Sie diese Option aus, wenn Sie die erforderlichen Account-Informationen für die Zieldomäne besitzen und sich damit gegenüber der Zieldomäne als Administrator authentifizieren können. Damit können Sie die Vertrauensstellung gleichzeitig in der lokalen und auch in der Zieldomäne erstellen und ersparen sich damit das gesonderte Erstellen der Vertrauensstellung in der Zieldomäne. In diesem Fall müssen Sie kein Vertrauensstellungs-Kennwort vergeben.

Nehmen Sie die entsprechende Auswahl vor, und klicken Sie auf **Weiter**. Wenn Sie die Vertrauensstellung für die lokale Domäne und auch für die Zieldomäne erstellen möchten, so müssen Sie sich nun der Zieldomäne gegenüber authentifizieren (siehe Abbildung 221).

Abbildung 221: Die Authentifizierung der Zieldomäne gegenüber

Anschließend werden Ihnen die Konfigurationsparameter für die neue Vertrauensstellung noch einmal angezeigt. Klicken Sie auf **WEITER**, um mit dem Erstellen der Vertrauensstellung zu beginnen, oder klicken Sie gegebenenfalls auf **ZURÜCK**, womit Sie die gewählten Optionen notfalls noch einmal korrigieren können.

Mit einem Klick auf WEITER wird die Vertrauensstellung nun eingerichtet, und das Ergebnis wird im Dialogfenster VERTRAUENSSTELLUNGERSTELLUNG ABGESCHLOSSEN angezeigt. Wenn die Vertrauensstellung erfolgreich erstellt werden konnte und wenn Sie die Vertrauensstellung für beide Domänen konfiguriert haben, so wählen Sie (nach einem Klick auf WEITER) die Option JA, AUSGEHENDE VERTRAUENSSTELLUNG BESTÄTIGEN und im nächsten Schritt JA, EINGEHENDE VERTRAUENSSTELLUNG BESÄTIGEN aus, woraufhin die Vertrauensstellung jeweils von der Gegenseite bestätigt wird und Sie deren Funktionstüchtigkeit damit überprüfen können.

Die Lösung per Eingabeaufforderung

Mit dem `Netdom`-Befehl, der in den *Windows Server 2003 Support-Tools* enthalten ist, können Sie neue Vertrauensstellungen auch von der Eingabeaufforderung aus erstellen. Wenn Sie beispielsweise eine ausgehende (unidirektionale) externe Vertrauensstellung von der Domäne *MyDomain.com* zur Domäne *YourDomain.com* erzeugen möchten, so wenden Sie den `Netdom`-Befehl an, legen mit `/Domain:Domänenname` den Namen der Zieldomäne fest, übergeben daraufhin den Namen der lokalen Domäne, fügen dann den Parameter `/Add` an und übergeben mit `/UserD:` und `/PasswordD:` die Anmeldeinformationen für den Zugriff auf die Zieldomäne (mit `NetDom` erzeugte Vertrauensstellungen werden immer zeitgleich in beiden Domänen konfiguriert).

```
@Echo off
Netdom Trust /Domain:YourDomain.com MyDomain.com /Add /UserD:Administrator _
/PasswordD:12345678
```

Listing 579: 2_07_010a.bat

Wenn Sie statt einer unidirektionalen Vertrauensstellung eine bidirektionale Vertrauensstellung erzeugen wollen, so fügen Sie dem Parameter `/Add` noch ein `/Twoway` an.

```
@Echo off
Netdom Trust /Domain:YourDomain.com MyDomain.com /Add /Twoway _
/UserD:Administrator /PasswordD:12345678
```

Listing 580: 2_07_010b.bat

Die Lösung mit VBScript

Mit Hilfe der `Run`-Methode des `Shell`-Objektes lassen sich Vertrauensstellungen auch mit VBScript sehr leicht anlegen, da Sie die `NetDom`-Befehlszeile der Eingabeaufforderung auch aus einem VBScript heraus starten können.

Um beispielsweise eine ausgehende Vertrauensstellung von der Domäne *MyDomain.com* zur Domäne *YourDomain.com* anzulegen und zur Authentifizierung der Zieldomäne gegenüber den Benutzernamen *Administrator* und das Kennwort *12345678* zu verwenden, können Sie das folgende Listing verwenden.

```
Option Explicit
Dim oShell
```

```
Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Netdom Trust /Domain:YourDomain.com MyDomain.com _
/Add /UserD:Administrator /PasswordD:12345678"
```

```
Set oShell = Nothing
```

Listing 581: 2_07_010.vbs

166 Vertrauensstellungen überprüfen

Für den Fall, dass Sie gerade eine externe Vertrauensstellung oder eine Gesamtstruktur- oder Verknüpfungsvertrauensstellung eingerichtet haben und deren Funktionstüchtigkeit testen möchten, so können Sie die Vertrauensstellung jederzeit überprüfen. Hierzu rufen Sie unter grafischer Oberfläche die Management-Konsole ACTIVE DIRECTORY-DOMÄNEN UND -VERTRAUENSSTELLUNGEN auf und öffnen aus dem Kontextmenü der betreffenden Domäne die EIGENSCHAFTEN. Anschließend wechseln Sie auf die Registerkarte VERTRAUENSSTELLUNGEN, markieren die entsprechende Vertrauensstellung und klicken auf EIGENSCHAFTEN.

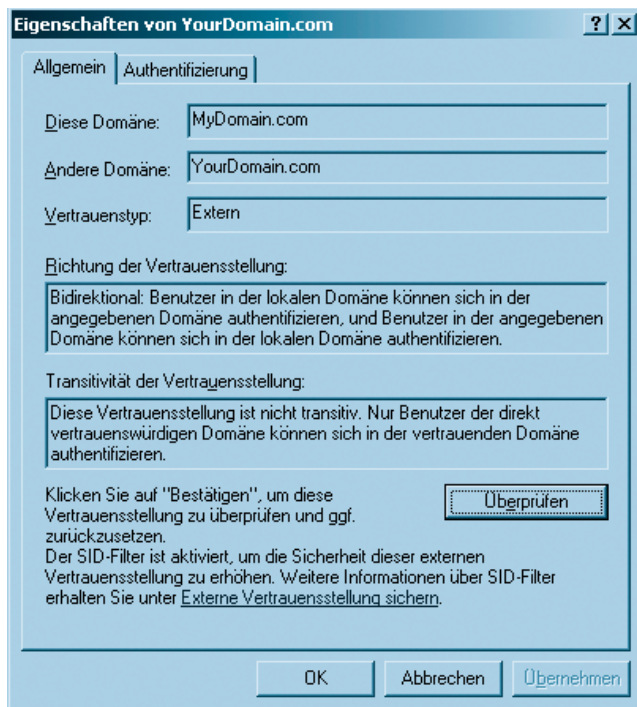


Abbildung 222: Eine Vertrauensstellung überprüfen

Führen Sie anschließend einen Klick auf die Schaltfläche ÜBERPRÜFEN aus, und legen Sie fest, ob nur die Vertrauensstellung der lokalen Domäne oder zeitgleich auch die der Zieldomäne überprüft werden soll. Letzteres setzt voraus, dass Sie sich der Zieldomäne gegenüber als Domänen- oder Organisationsadministrator authentifizieren können. Wählen Sie die entsprechende Option aus, und geben Sie gegebenenfalls den Benutzernamen und auch das Kennwort für die Anmeldung an der Zieldomäne ein. Klicken Sie dann auf OK.

Die Vertrauensstellung zwischen der lokalen Domäne und der Zieldomäne wird daraufhin überprüft, und Sie erhalten eine Meldung darüber, ob die Vertrauensstellung vorhanden und aktiv oder fehlerhaft ist.

Die Lösung per Eingabeaufforderung

Eine Vertrauensstellung lässt sich auch von der Eingabeaufforderung aus überprüfen. Hierfür setzen Sie wieder das Support-Tool *NetDom.exe* mit dem Parameter *Trust* ein und übergeben mit */Domain:* zunächst den Namen der Zieldomäne und anschließend auch den Namen der lokalen Domäne. Mit dem Parameter */Verify* geben Sie an, dass die Vertrauensstellung überprüft werden soll, und mit */UserD:* und */PasswordD:* übergeben Sie den Benutzernamen und auch das Kennwort des Administrators auf der Gegenseite.

Die nachfolgende *NetDom*-Befehlszeile führt eine Überprüfung der Vertrauensstellung von *MyDomain.com* zu *YourDomain.com* unter Verwendung des Benutzers *Administrators* (und des Kennwortes *123QWEasd*) des Administrators der Zieldomäne durch.

```
@Echo off
NetDom Trust /Domain:YourDomain.com MyDomain.com /Verify _
/UserD:Administrator /PasswordD:123QWEasd
```

Listing 582: 2_07_011.bat

Die Lösung mit VBScript

Mit Hilfe einer WMI-Abfrage lässt sich eine Vertrauensstellung auch per VBScript überprüfen. Wenn Sie beispielsweise die ausgehende Vertrauensstellung zur Zieldomäne *YourDomain.com* abfragen und deren Status ermitteln möchten, so stellen Sie im ersten Schritt eine Verbindung zum WMI-Namespace *root\MicrosoftActiveDirectory* des betreffenden Computersystems (*strComputer*) her und setzen dann die *ExecQuery*-Abfrage gegen die WMI-Klasse *Microsoft_DomainTrustStatus* ein, um die entsprechenden Vertrauensstellungen zu selektieren. Das nachfolgende Listing nutzt beispielsweise die Eigenschaft *TrustedDomain*, um die Vertrauensstellungen nach dem Namen der Zieldomäne zu selektieren.

Die *ExecQuery*-Abfrage liefert das Ergebnis daraufhin in Form einer Kollektion zurück, und Sie können innerhalb einer *For Each*-Schleife beispielsweise die Werte der Eigenschaften *TrustedDomain* (Name der Zieldomäne), *TrustDirection* (1 = einge-

hend, 2 = ausgehend, 3 = bidirektional), TrustStatus (Status) und TrustIsOk (True/False) abfragen und per WScript.Echo-Anweisung am Bildschirm ausgeben.

```
Option Explicit
```

```
Dim oWMI
Dim colTrusts
Dim objItem
Dim strComputer
```

```
strComputer = "."
Set oWMI = GetObject("winmgmts:" & "{impersonationLevel=_
    impersonate}!\" & strComputer & "\root\MicrosoftActiveDirectory")
Set colTrusts = oWMI.ExecQuery ("SELECT * FROM Microsoft_DomainTrustStatus _
    Where TrustedDomain='YourDomain.com'")
```

```
For Each objItem in colTrusts
    Wscript.Echo "Trusted Domain: " & objItem.TrustedDomain
    Wscript.Echo "Direction: " & objItem.TrustDirection
    Wscript.Echo "Status: " & objItem.TrustStatus
    Wscript.Echo "Active: " & objItem.TrustIsOK
Next
```

```
Set oWMI = Nothing
```

Listing 583: 2_07_011.vbs

167 Eine Vertrauensstellung entfernen

Für den Fall, dass Sie eine eingehende, ausgehende oder auch bidirektionale Vertrauensstellung eingerichtet haben, die Sie künftig nicht mehr benötigen, so können Sie die jeweilige Vertrauensstellung ohne großen Aufwand wieder entfernen.

Hierzu nehmen Sie die Management-Konsole ACTIVE DIRECTORY-DOMÄNEN UND -VERTRAUENSSTELLUNGEN zu Hilfe, die Sie entweder aus dem Menü VERWALTUNG heraus oder durch Eingabe von `Domain.msc` auch direkt aufrufen können. Markieren Sie im Konsolenstamm in der linken Fensterhälfte die Domäne, und rufen Sie aus dem Kontextmenü heraus die EIGENSCHAFTEN auf. Holen Sie dann die Registerkarte VERTRAUENSSTELLUNGEN in den Vordergrund, und markieren Sie die eingehende oder ausgehende Vertrauensstellung, die Sie entfernen möchten.

Klicken Sie anschließend auf ENTFERNEN, und legen Sie daraufhin fest, ob Sie die Vertrauensstellung nur aus der lokalen Domäne oder zeitgleich in beiden Domänen entfernen möchten. Wenn Sie die Option NEIN, DIE VERTRAUENSSTELLUNG NUR AUS DER LOKALEN DOMÄNE ENTFERNEN auswählen, so bleibt die Vertrauensstellung in der anderen Domäne unberührt und muss dort ebenfalls gelöscht werden. Wenn Sie stattdessen die Option JA, DIE VERTRAUENSSTELLUNG AUS DER LOKALEN UND DER ANDEREN DOMÄNE

ENTFERNEN auswählen, so müssen Sie sich der anderen Domäne gegenüber als Domänen- oder Organisationsadministrator authentifizieren. Die Vertrauensstellung wird daraufhin in der lokalen und – wenn die Authentifizierung erfolgreich durchgeführt wurde – auch in der anderen Domäne entfernt.

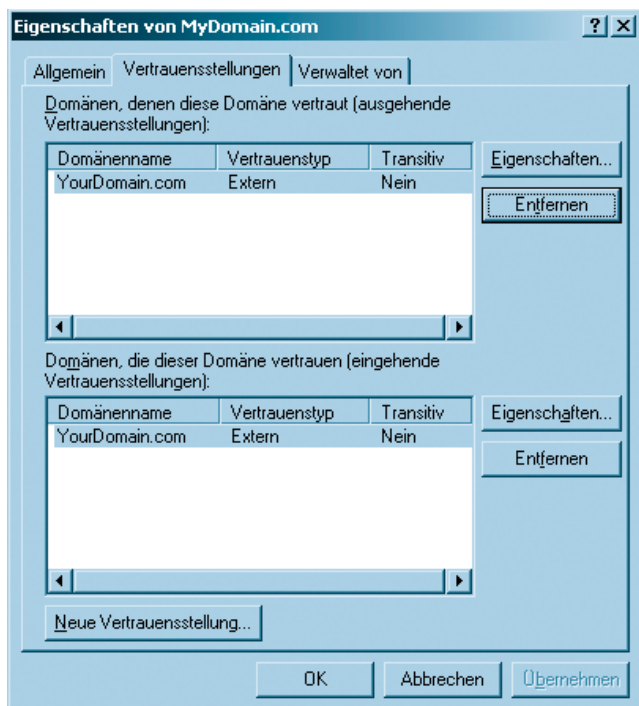


Abbildung 223: Eine Vertrauensstellung entfernen

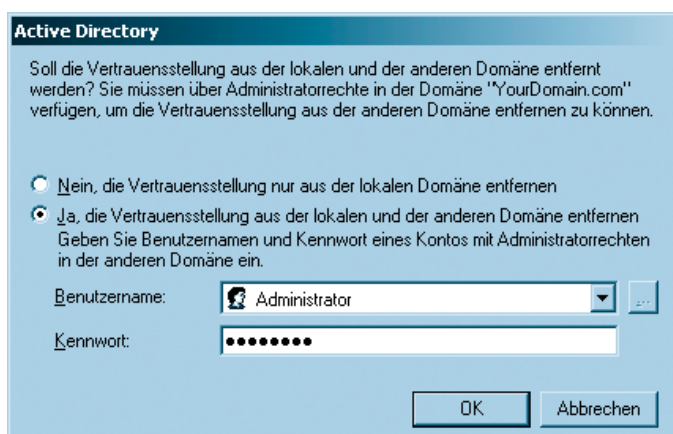


Abbildung 224: Die Vertrauensstellung in beiden Domänen entfernen

Die Lösung per Eingabeaufforderung

Mit dem `NetDom`-Befehl, der Bestandteil der *Windows Server 2003 Support-Tools* ist, lassen sich Vertrauensstellungen auch von der Eingabeaufforderung aus entfernen. Wenn in der lokalen Domäne *MyDomain.com* beispielsweise eine ausgehende Vertrauensstellung mit der Zieldomäne *YourDomain.com* eingerichtet ist und Sie die Vertrauensstellung nun in beiden Domänen löschen möchten, so wenden Sie den `NetDom`-Befehl mit dem Parameter `Trust` an, übergeben mit `/Domain:` den Namen der Zieldomäne und anschließend auch den Namen der lokalen Domäne. Mit `/Remove` legen Sie fest, dass die Vertrauensstellung entfernt werden soll, und fügen anschließend mit `/UserD:` und `/PasswordD:` die Account-Informationen für die Authentifizierung auf der Gegenseite an.

```
@Echo off
NetDom Trust /Domain:YourDomain.com MyDomain.com /Remove _
/UserD:Administrator /PasswordD:123QWEasd
```

Listing 584: 2_07_012a.bat

Wenn Sie eine bidirektionale Vertrauensstellung entfernen und die Vertrauensstellung in beiden Domänen entfernen möchten, so fügen Sie dem Parameter `/Remove` noch ein `/Tway` an.

```
@Echo off
NetDom Trust /Domain:YourDomain.com MyDomain.com /Remove /Tway _
/UserD:Administrator /PasswordD:12345678
```

Listing 585: 2_07_012b.bat

Die Lösung mit VBScript

Wenn Sie eine Vertrauensstellung per VBScript entfernen möchten, so stellen Sie im ersten Schritt eine Verbindung zum Root der Domäne und darauf folgend zum untergeordneten Container-System her. Anschließend wenden Sie die `Delete`-Methode an und übergeben den Namen der Zieldomäne und den Namen der lokalen Domäne als Parameter.

```
Option Explicit

Dim oRootDSE
Dim oTrust
Dim strTrustingDomain
Dim strTrustedDomain

strTrustingDomain = "Mydomain.com"
```

Listing 586: 2_07_012.vbs

```

strTrustedDomain = "Yourdomain.com"

Set oRootDSE = GetObject("LDAP://" & strTrustingDomain & "/RootDSE")
Set oTrust = GetObject("LDAP://CN=System," & _
    oRootDSE.Get("DefaultNamingContext"))
oTrust.Delete "TrustedDomain", "CN=" & strTrustedDomain

Set oRootDSE = Nothing
Set oTrust = Nothing

```

Listing 586: 2_07_012.vbs (Forts.)

168 Einen DHCP-Server autorisieren

Wenn Sie einen Member-Server oder Domänen-Controller eingerichtet haben, der innerhalb des Active Directory als DHCP-Server fungieren soll, so müssen Sie den DHCP-Server zunächst im Active Directory autorisieren. Denn erst nach der erfolgreichen Autorisierung kann der DHCP-Server die Clients im Netzwerk mit freien IP-Adressen versorgen. Damit soll verhindert werden, dass »versehentlich« DHCP-Server im Netzwerk aufgenommen und Clients möglicherweise mit doppelten oder gar falschen IP-Adressen versorgt werden, womit es zu massiven Störungen des Netzwerkbetriebes kommen würde.

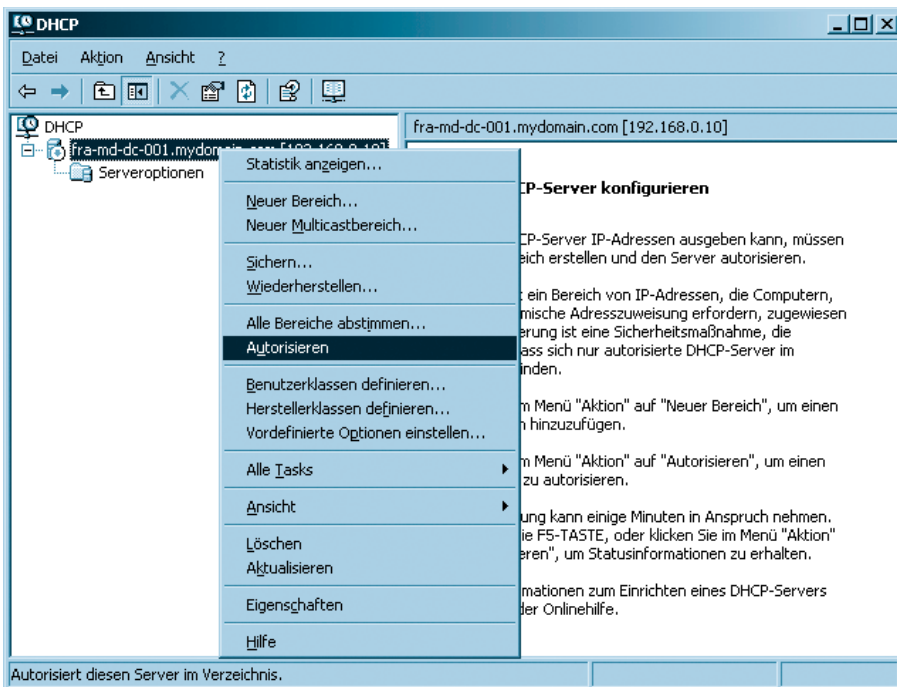


Abbildung 225: Einen DHCP-Server autorisieren

Um einen neuen DHCP-Server unter grafischer Oberfläche zu autorisieren, rufen Sie zunächst die Management-Konsole DHCP auf. Klicken Sie hierzu auf START, erweitern Sie dann das Menü VERWALTUNG, und rufen Sie anschließend DHCP auf, oder wählen Sie stattdessen im Startmenü den Eintrag AUSFÜHREN aus, und geben Sie `Dhcpmgmt.msc` ein, womit Sie die zuvor genannte Management-Konsole auch direkt aufrufen können. In der Management-Konsole DHCP markieren Sie im Konsolenstamm den DHCP-Server, den Sie autorisieren möchten, und wählen im Kontextmenü den Eintrag AUTORISIEREN aus.

Die Lösung per Eingabeaufforderung

Das Autorisieren eines DHCP-Servers lässt sich natürlich auch von der Eingabeaufforderung aus vornehmen. Windows Server 2003 stellt hierfür den `Netsh`-Befehl zur Verfügung, womit Sie zahlreiche Netzwerkeinstellungen und viele netzwerkspezifische Server-Funktionen von der Eingabeaufforderung aus abfragen und auch modifizieren können.

`Netsh` wird für gewöhnlich im interaktiven Modus genutzt. Das bedeutet, dass Sie durch Eingabe von `Netsh` die `Netsh`-spezifische Eingabeaufforderung aufrufen, dann in den entsprechenden Kontext (Funktionsgruppe) wechseln und durch Eingabe der `Netsh`-Subbefehle bestimmte Informationen abfragen oder Änderungen an der Netzwerkkonfiguration vornehmen können. `Netsh` erlaubt die Anwendung der `Netsh`-eigenen Subbefehle aber auch von der »gewöhnlichen« Eingabeaufforderung aus, wofür Sie nicht erst in die `Netsh`-eigene Eingabeaufforderung wechseln und dort die Befehle eingeben müssen. Sie können `Netsh` also auch in einer Batch-Datei verwenden, um Änderungen an der Netzwerkkonfiguration vornehmen zu können, indem Sie dem `Netsh`-Befehl die Subbefehle einfach als Parameter anfügen.

Um nun einen DHCP-Server im Active Directory zu autorisieren, wenden Sie den `Netsh`-Befehl an und wechseln durch Eingabe von `DHCP` in den DHCP-Kontext (dieser Kontext steht erst nach dem Einrichten des DHCP-Servers zur Verfügung). Anschließend setzen Sie den Subbefehl `Add Server` ein und übergeben den DNS-Namen des DHCP-Servers und dessen IP-Adresse als Parameter.

Die nachfolgende `Netsh`-Befehlszeile zeigt ein Beispiel dafür, wie Sie den Server mit dem DNS-Namen `FRA-MD-DC-001.MyDomain.com`, der die IP-Adresse `192.168.0.10` besitzt, im Active Directory autorisieren können, ohne hierfür in die `Netsh`-eigene Eingabeaufforderung wechseln zu müssen.

```
@Echo off
Netsh Dhcp Add Server FRA-MD-DC-001.MyDomain.com 192.168.0.10
```

Listing 587: 2_07_013.bat

Die Lösung mit VBScript

Um einen DHCP-Server per VBScript zu autorisieren, nutzen Sie die gleiche Möglichkeit wie auch in der Eingabeaufforderung. Will heißen: Sie setzen den `Netsh`-Befehl ein, übergeben die erforderlichen Parameter und führen die Befehlszeile mit der `Run`-Methode des `Shell`-Objektes aus.

Das nachfolgende Listing führt beispielsweise den `Netsh`-Befehl aus und autorisiert den DHCP-Server mit dem DNS-Namen `FRA-MD-DC-001.MyDomain.com` und der IP-Adresse `192.168.0.10` im Active Directory:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Netsh Dhcp Add Server _
    FRA-MD-DC-001.MyDomain.com 192.168.0.10"

Set oShell = Nothing
```

Listing 588: 2_07_013.vbs

169 DHCP-Bereiche definieren

Wenn Sie einen DHCP-Server eingerichtet und diesen auch im Active Directory entsprechend autorisiert haben, so müssen Sie im nächsten Schritt einen IP-Adressbereich definieren und die freien IP-Adressen, die den DHCP-Clients später zugewiesen werden sollen, hinterlegen. Unter grafischer Oberfläche nehmen Sie hierfür die Management-Konsole DHCP zu Hilfe, die Sie im Menü `VERWALTUNG` nur anzuklicken brauchen oder mit `Dhcpmgmt.msc` auch direkt aufrufen können.

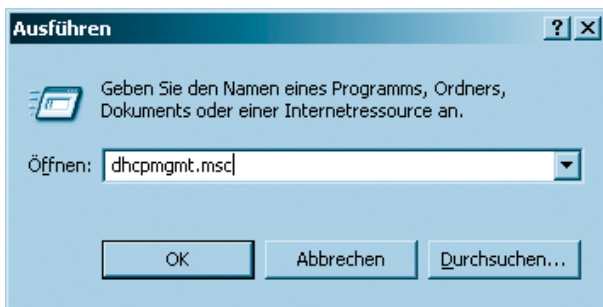


Abbildung 226: Die Management-Konsole »Dhcp« direkt aufrufen

Markieren Sie im Konsolenstamm den DHCP-Server, für den Sie einen neuen Bereich definieren möchten, und wählen Sie im Kontextmenü den Eintrag `NEUER BEREICH...` aus. Daraufhin wird der `BEREICHSERSTELLUNGS-ASSISTENT` geöffnet. Schließen Sie den

Willkommensbildschirm mit einem Klick auf WEITER, und legen Sie im nächsten Schritt einen aussagekräftigen Namen und wahlweise auch eine kurze Beschreibung für den IP-Adressbereich fest. Klicken Sie auf WEITER, und legen Sie die Start-IP-Adresse und auch die End-IP-Adresse fest, und bestimmen Sie die Subnetzmaske nach der Länge oder der IP-Adresse.

Wenn Sie beispielsweise die IP-Adressen von *192.168.0.100* bis *192.168.0.200* für die dynamische IP-Adresszuweisung bereitstellen und die Subnetzmaske *255.255.255.0* verwenden möchten, so legen Sie die IP-Adresse *192.168.0.100* als Start-IP-Adresse und die *192.168.0.200* als End-IP-Adresse fest. Die Subnetzmaske wird daraufhin automatisch berechnet und als Default-Wert angezeigt, den Sie nur zu übernehmen brauchen (siehe Abbildung 227).

Abbildung 227: Die Start- und End-IP-Adresse und die Subnetzmaske festlegen

Klicken Sie auf WEITER, Sie gelangen daraufhin in das Dialogfenster AUSSCHLÜSSE HINZUFÜGEN. Hier können Sie bestimmte IP-Adressen, die in dem zuvor hinterlegten IP-Adressbereich liegen, von der dynamischen Zuweisung durch den DHCP-Server ausschließen. Wenn Sie beispielsweise die IP-Adressen *192.168.0.150* bis *192.168.0.160* für andere Zwecke benötigen und nicht für die dynamische IP-Adresszuweisung bereitstellen möchten, so geben Sie die Start-IP-Adresse *192.168.0.150* und die End-IP-Adresse *192.168.0.160* ein. Klicken Sie dann auf HINZUFÜGEN. Die IP-Adressen werden später nicht an anfragende DHCP-Clients weitergegeben (siehe Abbildung 228).

Abbildung 228: Bestimmte IP-Adressen aus dem IP-Adressbereich ausschließen

Klicken Sie anschließend auf WEITER, um die Lease-Dauer festzulegen. Die Lease-Dauer bestimmt, wie lange ein DHCP-Client eine dynamische IP-Adresse verwenden darf, bevor sie automatisch gelöst wird und daraufhin erneuert werden muss. Standardmäßig wird eine Lease-Dauer von acht Tagen vorgegeben. Sie können die Dauer verringern, aber auch vergrößern. Die Lease-Dauer ist insbesondere dann zu berücksichtigen, wenn Sie nur wenige IP-Adressen zur Verfügung haben und der IP-Adressbereich vielleicht nicht für alle DHCP-Clients ausreicht. Denn wenn Sie die Lease-Dauer auf beispielsweise acht Stunden heruntersetzen, so werden die vergebenen IP-Adressen umso schneller wieder frei und können damit umso schneller an andere DHCP-Clients weitergegeben werden. In der Regel sind nicht alle DHCP-Clients immer online. Es macht also keinen Sinn, die ohnehin schon wenigen IP-Adressen bei längeren Offline-Zeiten der DHCP-Clients in Nutzung zu halten. Besser ist, sie nach nur kurzer Zeit wieder freizugeben und sie damit für andere Clients verfügbar zu machen.

Klicken Sie auf WEITER, woraufhin Sie in das Dialogfenster DHCP-OPTIONEN KONFIGURIEREN gelangen. Dort können Sie festlegen, ob Sie die DHCP-Optionen hinterlegen möchten oder die Einstellungen gar nicht bzw. im Nachhinein vornehmen wollen (siehe Seite 711). Klicken Sie NEIN, DIESE OPTIONEN SPÄTER KONFIGURIEREN, um mit dem Erstellen des neuen Bereiches fortzufahren, ohne die DHCP-Optionen zu konfigurieren. Im nächsten Schritt haben Sie die Möglichkeit, den neuen DHCP-Bereich zu aktivieren, womit der neue Bereich nach dem Fertigstellen des Assistenten sofort verwendet werden kann. Wählen Sie die Option JA, DIESEN BEREICH JETZT AKTIVIEREN aus, um den Bereich zu akti-

vieren, und klicken Sie anschließend auf FERTIG STELLEN, um den Vorgang abzuschließen. Der IP-Adressbereich ist nun eingerichtet und – sofern Sie den Bereich auf der letzten Dialogseite aktiviert haben – auch sofort verfügbar.

Die Lösung per Eingabeaufforderung

Mit dem `Netsh`-Befehl lässt sich nicht nur ein DHCP-Server im Active Directory autorisieren, sondern auch ein neuer DHCP-Bereich definieren. Wenn Sie beispielsweise auf dem DHCP-Server mit der IP-Adresse `192.168.0.10` einen neuen Bereich für das Netzwerk `192.168.0.0 / 255.255.255.0` eröffnen und hierbei den Namen *Standardbereich* vergeben möchten, so wenden Sie den `Netsh`-Befehl an, wechseln mit `Dhcp` in den entsprechenden Kontext, stellen mit `Server 192.168.0.10` eine Verbindung zum DHCP-Server her, weisen mit `Add Scope` an, dass Sie einen neuen Bereich definieren möchten, und übergeben die IP-Adresse und auch die Subnetzmaske für das Netzwerk.

Sobald der Bereich angelegt ist, können Sie den Bereich mit Scope IP-Adresse (z.B. *Scope 192.168.0.0*) selektieren und den IP-Adressbereich durch Angabe der Start- und End-IP-Adresse hinzufügen. Hierzu übergeben Sie mit `Scope IP-Adresse` die IP-Adresse des zuvor erstellten Bereichs, fügen den Subbefehl `Add IPRange` an und übergeben die Start-IP-Adresse und – gefolgt von einem Leerzeichen – auch die End-IP-Adresse.

```

C:\>netsh
netsh>dhcp
netsh dhcp>Server 192.168.0.10
netsh dhcp server>Add Scope 192.168.0.0 255.255.255.0 Standard-Bereich

Der Befehl wurde erfolgreich ausgeführt.
netsh dhcp server>Scope 192.168.0.0

Der aktuelle Bereichskontext wurde in Bereich 192.168.0.0 geändert.
netsh dhcp server scope>Add IPRange 192.168.0.100 192.168.0.200

Der Befehl wurde erfolgreich ausgeführt.
netsh dhcp server scope>

```

Abbildung 229: Mit Netsh einen neuen Bereich erstellen

Der DHCP-Bereich lässt sich natürlich auch einrichten, ohne hierfür die `Netsh`-eigene Eingabeaufforderung aufrufen zu müssen. Wenn Sie den `Netsh`-Befehl beispielsweise in einer Batch-Datei verwenden möchten, so übergeben Sie dem `Netsh`-Befehl einfach die erforderlichen Subbefehle als Parameter.


```
@Echo off
Netsh Dhcp Server 192.168.0.10 Add Scope 192.168.0.0 _
    255.255.255.0 Standardbereich
Netsh Dhcp Server 192.168.0.10 Scope 192.168.0.0 _
    Add IPRange 192.168.0.100 192.168.0.200
```

Listing 589: 2_07_014.bat

Die Lösung mit VBScript

Auf die gleiche Art und Weise lässt sich ein DHCP-Bereich auch per VBScript definieren. Will heißen, Sie setzen auch hier den Netsh-Befehl unter Angabe der erforderlichen Parameter (siehe vorheriger Abschnitt) ein und führen die jeweiligen Netsh-Befehlszeilen mit der Run-Methode des Shell-Objektes aus.

```
Option Explicit
Dim oShell
Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Netsh Dhcp Server 192.168.0.10 Add Scope _
    192.168.0.0 255.255.255.0 Standardbereich"
oShell.Run "%comspec% /c Netsh Dhcp Server 192.168.0.10 Scope 192.168.0.0 _
    Add IPRange 192.168.0.100 192.168.0.200"
Set oShell = Nothing
```

Listing 590: 2_07_014.vbs

170 DHCP-Bereiche aktivieren und deaktivieren

Wenn Sie einen neuen DHCP-Bereich definiert haben, so müssen Sie diesen Bereich erst aktivieren, bevor Sie ihn nutzen können. Denn solange ein DHCP-Bereich nicht aktiviert wurde, werden die IP-Adressen aus dem Bereich auch nicht an anfragende DHCP-Clients weitergegeben.

Unter grafischer Oberfläche erfolgt das Aktivieren eines DHCP-Bereiches meist schon mit dem Anlegen eines neuen DHCP-Bereiches, denn der BEREICHSERSTELLUNGS-ASSISTENT hält hierfür eine entsprechende Option bereit und erlaubt das automatische Aktivieren des Bereiches nach Fertigstellung des Assistenten. Für den Fall, dass Sie die Aktivierung aber auf einen späteren Zeitpunkt verschoben haben, den Bereich aber nun verwenden wollen, so müssen Sie den DHCP-Bereich entsprechend aktivieren.

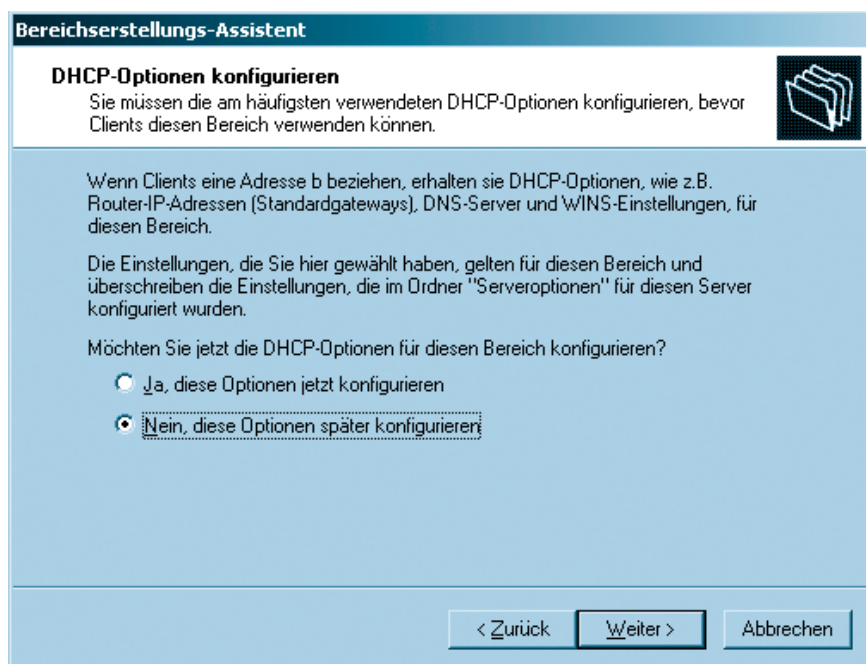


Abbildung 230: Die DHCP-Optionen zu einem späteren Zeitpunkt konfigurieren

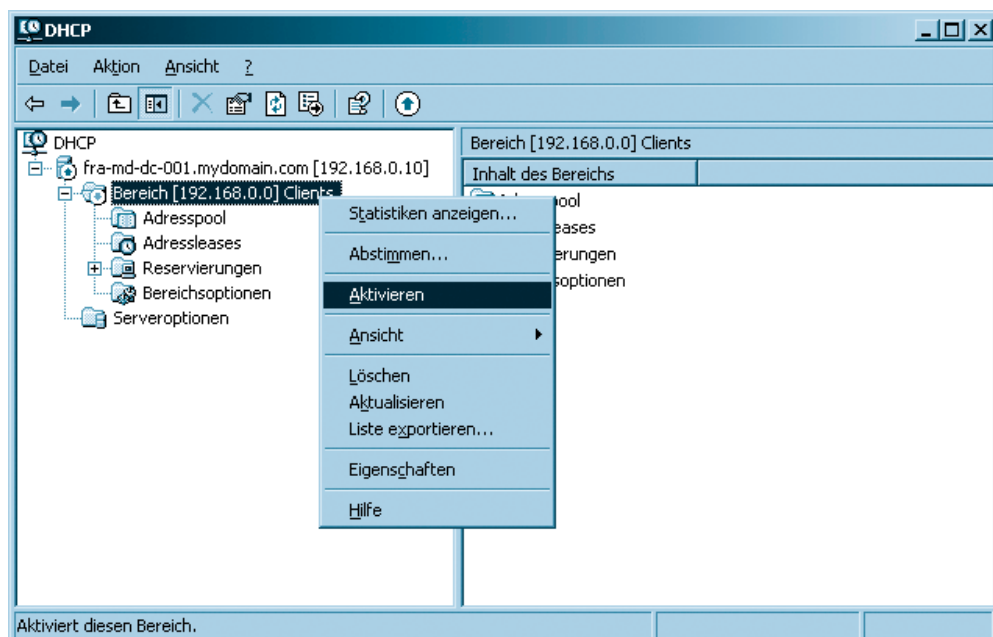


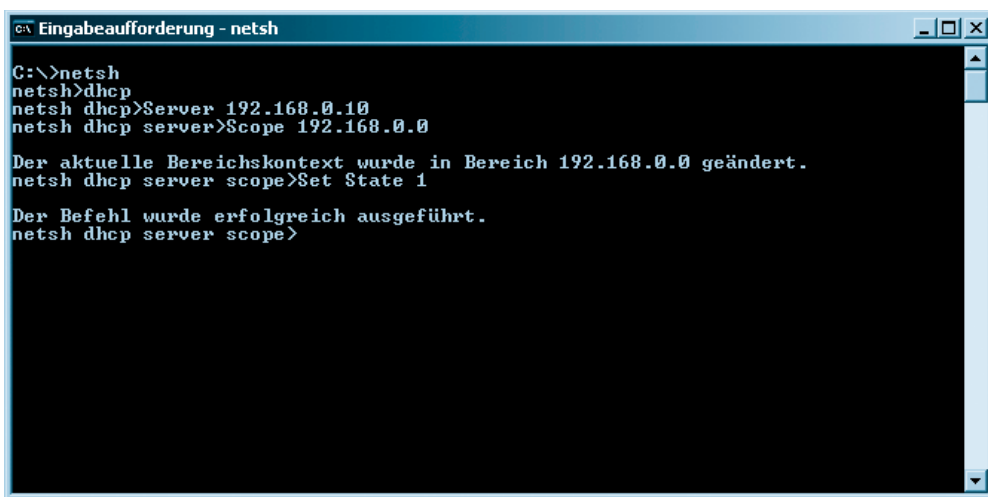
Abbildung 231: Einen DHCP-Bereich aktivieren

Rufen Sie hierzu die Management-Konsole DHCP auf. Öffnen Sie das STARTMENÜ, erweitern Sie VERWALTUNG, und klicken Sie anschließend auf DHCP, oder wählen Sie im STARTMENÜ den Eintrag AUSFÜHREN, und geben Sie `Dhcpmgmt.msc` ein, um die Management-Konsole DHCP direkt aufzurufen. Erweitern Sie in der linken Fensterhälfte der DHCP-Konsolenstruktur den entsprechenden DHCP-Server, und markieren Sie unterhalb davon den Bereich, den Sie aktivieren wollen (ein aktivierter DHCP-Bereich wird mit einem grünen Pfeil, der nach oben zeigt, gekennzeichnet). Wählen Sie im Kontextmenü des Bereichs den Eintrag AKTIVIEREN aus, um den Bereich zu aktivieren.

Wenn Sie den DHCP-Bereich zu einem späteren Zeitpunkt wieder deaktivieren möchten, so markieren Sie den entsprechenden DHCP-Bereich erneut und wählen im Kontextmenü den Eintrag DEAKTIVIEREN aus. Die IP-Adressen dieses Bereiches werden dann nicht mehr den anfragenden DHCP-Clients zugewiesen.

Die Lösung per Eingabeaufforderung

Mit dem Netsh-Befehl lässt sich ein vorhandener DHCP-Bereich auch von der Eingabeaufforderung aus aktivieren (und bei Bedarf natürlich auch wieder deaktivieren). Hierzu wenden Sie den Netsh-Befehl an, wechseln mit DHCP in den DHCP-spezifischen Kontext, stellen mit `Server IP-Adresse` eine Verbindung zu dem entsprechenden DHCP-Server her, selektieren dann mit `Scope IP-Adresse` den DHCP-Bereich, den Sie aktivieren wollen, und legen mit `Set State 1` fest, dass der DHCP-Bereich aktiviert werden soll.



```
C:\>netsh
netsh>dhcp
netsh dhcp>Server 192.168.0.10
netsh dhcp server>Scope 192.168.0.0

Der aktuelle Bereichskontext wurde in Bereich 192.168.0.0 geändert.
netsh dhcp server scope>Set State 1

Der Befehl wurde erfolgreich ausgeführt.
netsh dhcp server scope>
```

Abbildung 232: Mit Netsh einen DHCP-Bereich aktivieren

Die nachfolgende Netsh-Befehlszeile stellt eine Verbindung zum Server mit der IP-Adresse `192.168.0.10` her, selektiert den DHCP-Bereich `192.168.0.0` und setzt den Status auf den Wert `1`, womit der Bereich sofort aktiviert wird.

```
@Echo off
Netsh DhcP Server 192.168.0.10 Scope 192.168.0.0 Set State 1
```

Listing 591: 2_07_015.bat

Wenn Sie den soeben aktivierten DHCP-Bereich zu einem späteren Zeitpunkt wieder deaktivieren wollen, so können Sie die gleiche Netsh-Befehlszeile verwenden. Sie müssen lediglich dem Subbefehl Set State den Wert 0 übergeben.

Die Lösung mit VBScript

Um einen DHCP-Bereich per VBScript zu aktivieren (und gegebenenfalls später wieder zu deaktivieren), nutzen Sie die gleiche Möglichkeit wie auch in der Eingabeaufforderung. Sprich: Sie wenden den Netsh-Befehl an, den Sie mit der Run-Methode des Shell-Objektes auch aus einem VBScript heraus ausführen können und übergeben dem Netsh-Befehl die erforderlichen Parameter (siehe vorheriger Abschnitt).

Um beispielsweise eine Verbindung zum DHCP-Server mit der IP-Adresse 192.168.0.10 herzustellen und den DHCP-Bereich mit der IP-Adresse 192.168.0.0 zu aktivieren, können Sie folgende Codezeilen verwenden:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Netsh DhcP Server 192.168.0.10 _
    Scope 192.168.0.0 Set State 1"
Set oShell = Nothing
```

Listing 592: _07_015.vbs

Wenn Sie den DHCP-Bereich zu einem späteren Zeitpunkt wieder deaktivieren möchten, so setzen Sie das gleiche Skript ein, tauschen jedoch den Parameter Set State 1 durch Set State 0 aus.

171 Adressbereiche ausschließen

Mit dem Ausschließen von Adressbereichen können Sie die Bereitstellung bestimmter IP-Adressen, die sich innerhalb eines definierten DHCP-Bereiches befinden, kurzfristig aufheben. Das hat den Vorteil, dass Sie einzelne IP-Adressen oder wahlweise auch größere IP-Adressbereiche aus einem bereits definierten DHCP-Bereich ausschließen und damit für andere Zwecke nutzen können, ohne den DHCP-Bereich neu konfigurieren zu müssen.

Wenn Sie beispielsweise einen DHCP-Bereich von *192.168.0.100* bis *192.168.0.200* definiert haben und die IP-Adressen von *192.168.0.150* bis *192.168.0.160* ab sofort nicht mehr für DHCP-Clients zur Verfügung stellen möchten, so können Sie den Bereich *192.168.0.150* bis *192.168.0.160* aus dem DHCP-Bereich ausschließen, womit die IP-Adressen dann nicht mehr für die dynamische Zuweisung verwendet werden.

Um nun einen bestimmten IP-Adressbereich innerhalb eines DHCP-Bereiches auszuschließen, rufen Sie zunächst die Management-Konsole DHCP auf. Öffnen Sie hierfür das STARTMENÜ, erweitern Sie VERWALTUNG, und klicken Sie anschließend auf DHCP, oder wählen Sie im STARTMENÜ den Eintrag AUSFÜHREN, und geben Sie *Dhcpmgmt.msc* ein, womit Sie die Management-Konsole auch direkt aufrufen können. Erweitern Sie anschließend den DHCP-Bereich, den Sie bearbeiten wollen, und markieren Sie unterhalb davon den Eintrag ADRESSPOOL. Öffnen Sie das Kontextmenü von ADRESSPOOL, und wählen Sie NEUER AUSGESCHLOSSENER BEREICH... aus, woraufhin das Dialogfenster AUSSCHLUSS HINZUFÜGEN geöffnet wird.

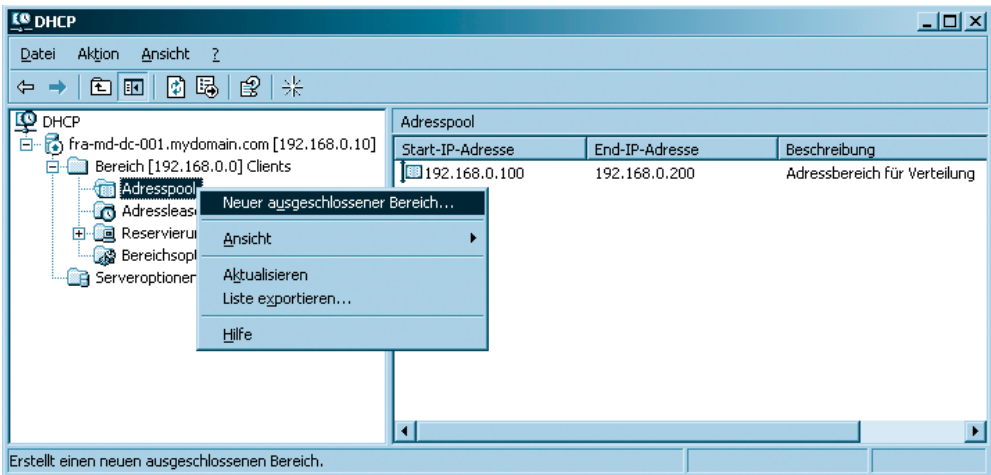


Abbildung 233: Einen neuen DHCP-Bereich ausschließen

Geben Sie die Start- und auch die End-IP-Adresse des auszuschließenden Bereiches ein (in dem hier genannten Beispiel wären dies die *192.168.0.150* und die *192.168.0.160*), und klicken Sie auf HINZUFÜGEN, um den ausgeschlossenen Bereich zu übernehmen. Schließen Sie das Dialogfenster mit einem Klick auf SCHLIESSEN, der ausgeschlossene Bereich ist damit erstellt, und die IP-Adressen, die sich in dem ausgeschlossenen Bereich befinden, werden ab sofort nicht mehr an DHCP-Clients zugewiesen.

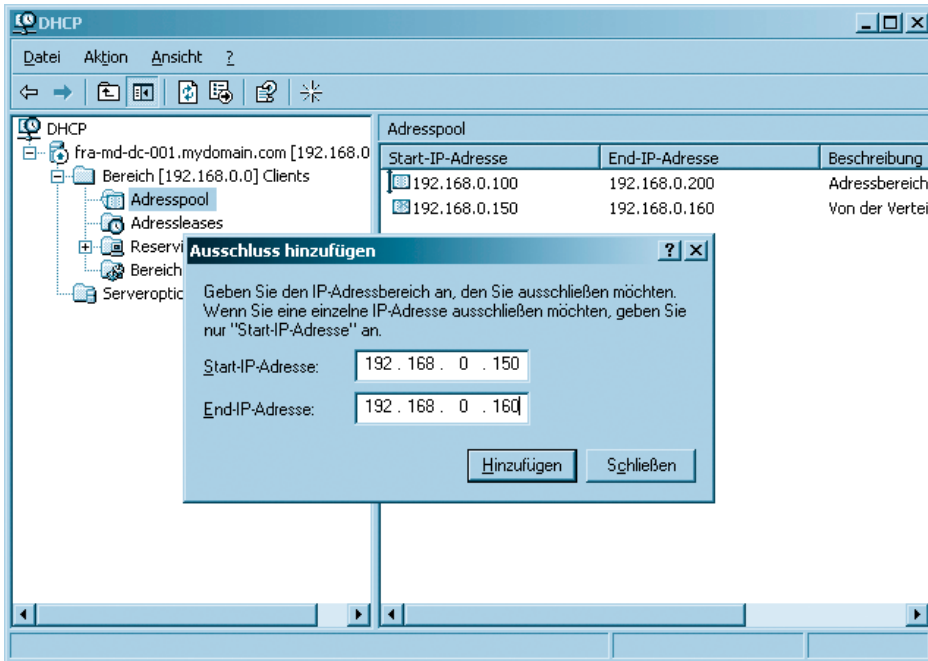


Abbildung 234: Einen IP-Adressbereich ausschließen

Die Lösung per Eingabeaufforderung

Gleiches lässt sich natürlich auch via Eingabeaufforderung realisieren, wofür Sie wieder den `Netsh`-Befehl verwenden können. Um beispielsweise auf dem DHCP-Server mit der IP-Adresse `192.168.0.10` den DHCP-Bereich `192.168.0.0` zu bearbeiten und die IP-Adressen von `192.168.0.150` bis `192.168.0.160` auszuschließen, wenden Sie den `Netsh`-Befehl an, wechseln durch Eingabe von `Dhcp` in den DHCP-spezifischen Kontext, stellen dann mit `Server 192.168.0.10` eine Verbindung zum DHCP-Server her und selektieren anschließend mit `Scope 192.168.0.0` den DHCP-Bereich, für den Sie eine Bereichsausschließung konfigurieren möchten. Mit dem Subbefehl `Add ExcludeRange`, dem Sie die Start- und auch die End-IP-Adresse des auszuschließenden Bereichs als Parameter übergeben, konfigurieren Sie dann den auszuschließenden Bereich. Durch Eingabe von `EXIT` können Sie die `Netsh`-spezifische Eingabeaufforderung daraufhin wieder verlassen (siehe Abbildung 235).

Der auszuschließende Bereich lässt sich natürlich auch ohne Einsatz der `Netsh`-spezifischen Eingabeaufforderung konfigurieren. Hierzu übergeben Sie dem `Netsh`-Befehl einfach die zuvor genannten Subbefehle als Parameter. Die nachfolgende `Netsh`-Befehlszeile zeigt ein Beispiel dafür, wie Sie eine Verbindung zum DHCP-Server mit der IP-Adresse `192.168.0.10` herstellen, den DHCP-Bereich `192.168.0.0` selektieren und anschließend den auszuschließenden Bereich von `192.168.0.150` bis `192.168.0.160` festlegen können.

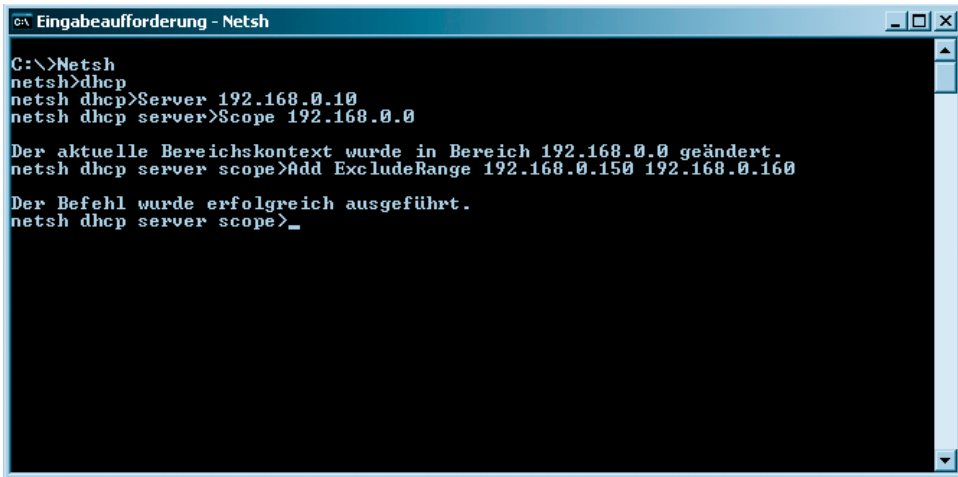


Abbildung 235: Mit Netsh einen auszuschließenden Bereich definieren

```
@Echo off
Netsh Dhcp Server 192.168.0.10 Scope 192.168.0.0 _
    Add ExcludeRange 192.168.0.150 192.168.0.160
```

Listing 593: 2_07_016.bat

Die Lösung mit VBScript

Wenn Sie die Bereichsausschließung per VBScript vornehmen wollen, so setzen Sie hierfür die Run-Methode des Shell-Objektes ein, womit Sie den Netsh-Befehl der Eingabeaufforderung auch per VBScript ausführen können. Dem Netsh-Befehl übergeben Sie dabei die erforderlichen Parameter (siehe vorheriger Abschnitt), womit Sie zunächst die Verbindung zum DHCP-Server herstellen, dann den DHCP-Bereich selektieren und zu guter Letzt durch Angabe der Start- und End-IP-Adresse den Bereich ausschließen.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Netsh Dhcp Server 192.168.0.10 Scope 192.168.0.0 _
    Add ExcludeRange 192.168.0.150 192.168.0.160"
Set oShell = Nothing
```

Listing 594: 2_07_016.vbs

172 DHCP-Adressen reservieren

Mit dem Reservieren einer DHCP-Adresse können Sie den DHCP-Server dazu veranlassen, dass einem bestimmten DHCP-Client immer die gleiche IP-Adresse zugewiesen wird. So ist der DHCP-Client stets unter der gleichen IP-Adresse erreichbar, ohne dass hierfür eine statische IP-Adresse vergeben werden muss.

Wenn die Client-Systeme innerhalb des Netzwerkes z.B. ausschließlich mit dynamisch zugewiesenen IP-Adressen versorgt werden, eines dieser Client-Systeme aber als Testsystem fungiert und daher stets unter der gleichen IP-Adresse erreichbar sein soll, so können Sie innerhalb des DHCP-Bereiches eine IP-Adresse für dieses Client-System reservieren und durch Hinterlegen der entsprechenden MAC-Adresse dafür sorgen, dass die reservierte IP-Adresse nur dem DHCP-Client mit der hinterlegten MAC-Adresse zugewiesen wird.

Um nun eine IP-Adresse für einen bestimmten DHCP-Client zu reservieren, rufen Sie zunächst die Management-Konsole DHCP auf, die Sie im STARTMENÜ, genauer gesagt unter VERWALTUNG, per Klick öffnen oder durch Eingabe von `Dhccpmgmt.msc` auch direkt aufrufen können. Anschließend erweitern Sie in der linken Fensterhälfte den DHCP-Server und darunter den entsprechenden Bereich. Im Kontextmenü von RESERVIERUNGEN wählen Sie den Eintrag NEUE RESERVIERUNG... aus, Sie gelangen daraufhin in das Dialogfenster NEUE RESERVIERUNG.

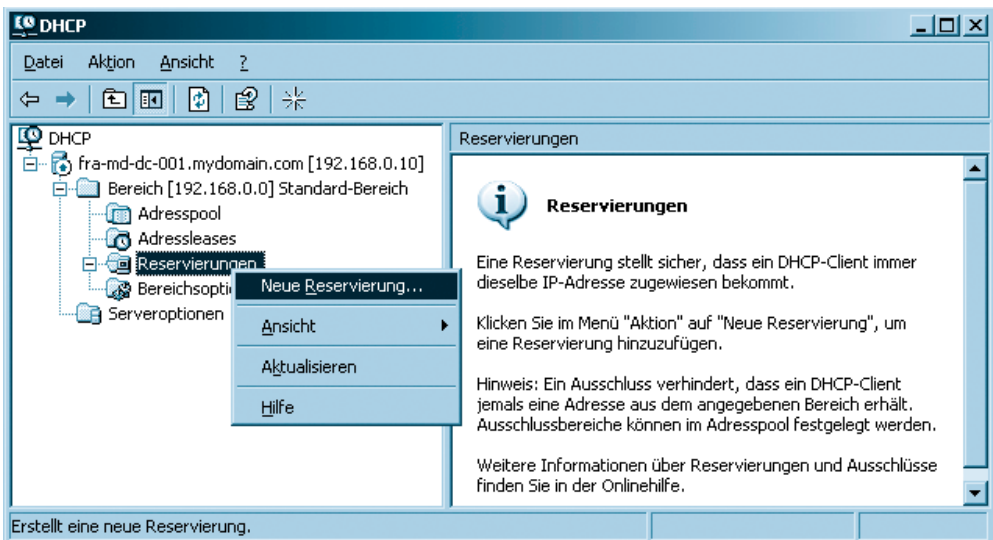


Abbildung 236: Eine neue Reservierung erstellen

Legen Sie im Feld RESERVIERUNGSNAME eine kurze, aber möglichst aussagekräftige Bezeichnung für die Reservierung fest, und hinterlegen Sie im Feld IP-ADRESSE die zu reservierende IP-Adresse. Im Feld MAC-ADRESSE geben Sie die MAC-Adresse des DHCP-Clients ein, für den die IP-Adresse reserviert werden soll. Optional können Sie im Feld BESCHREIBUNG einen kurzen Beschreibungstext für die Reservierung festlegen

und zudem auch auswählen, ob die Reservierung nur für DHCP- oder BOOTP-Anfragen oder für beide Typen gelten soll. Klicken Sie auf HINZUFÜGEN, um die Reservierung zu erzeugen, und schließen Sie das Dialogfenster wieder.

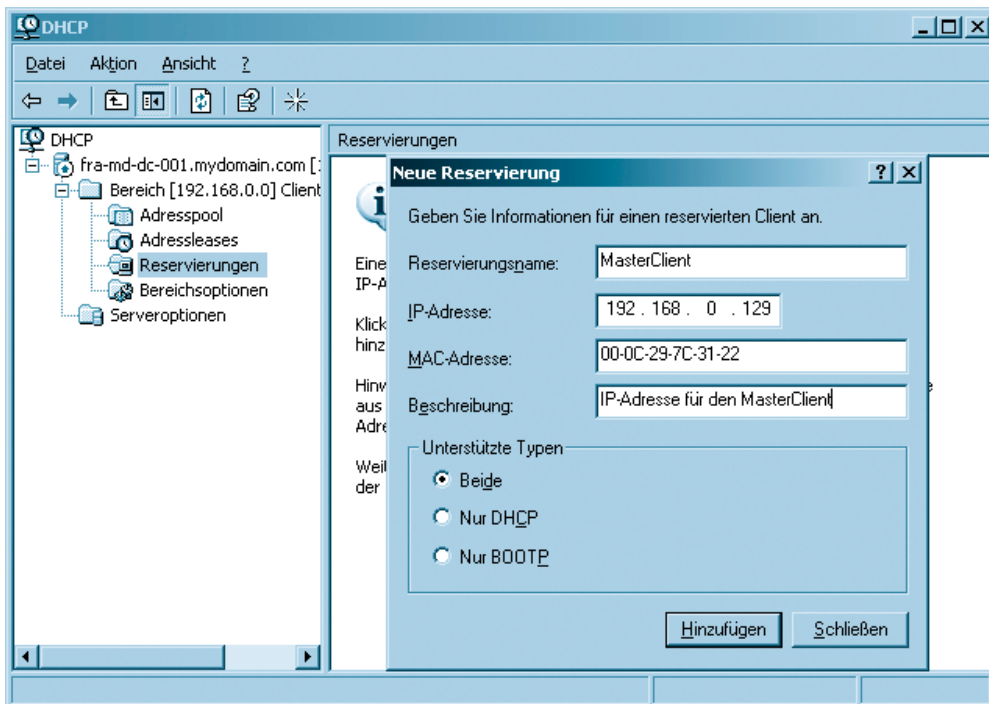


Abbildung 237: Eine IP-Adresse reservieren

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung erfolgt die Reservierung einer IP-Adresse wieder mit dem Netsh-Befehl. Wechseln Sie durch Eingabe von Netsh zunächst in die Netsh-eigene Eingabeaufforderung und anschließend mit Dhcp in den entsprechenden Kontext. Stellen Sie dann mit Server IP-Adresse eine Verbindung zum DHCP-Server her, und selektieren Sie mit Scope IP-Adresse den DHCP-Bereich, für den Sie eine Reservierung konfigurieren möchten. Mit Add ReserveDip IP-Adresse MAC-Adresse erzeugen Sie dann eine neue Reservierung, wobei Sie darauf achten müssen, dass die MAC-Adresse ohne jegliche Trennzeichen eingegeben werden muss, da die MAC-Adresse anderenfalls nicht als solche erkannt wird und die Befehlsausführung damit misslingt. Optional können Sie dann noch eine Bezeichnung und auch eine kurze Beschreibung für die Reservierung festlegen, indem Sie die entsprechende(n) Zeichenfolge(e) dem Netsh-Befehl als zusätzlichen Parameter übergeben. Und zu guter Letzt bestimmen Sie durch die optionale Angabe von Both, DHCP oder BOOTP, für welchen Typ die Reservierung gelten soll.

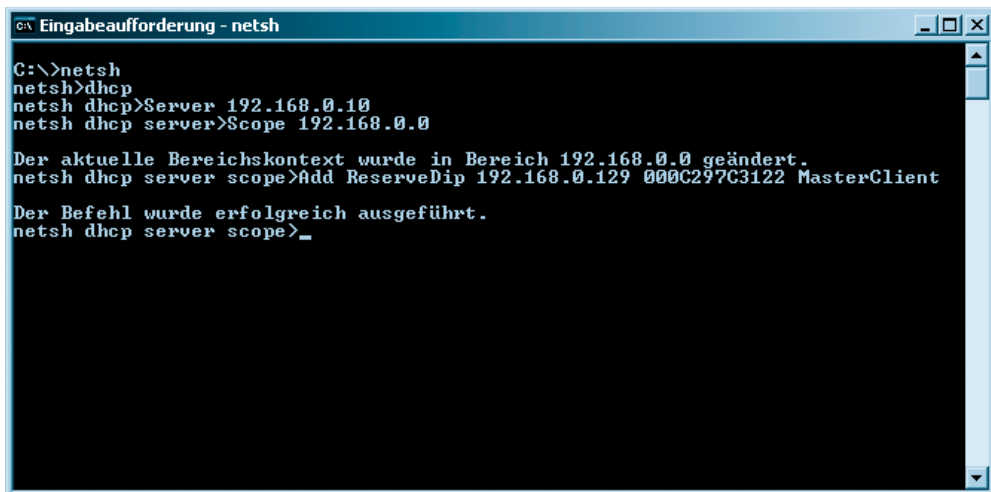


Abbildung 238: Mit Netsh eine neue Reservierung erzeugen

Die nachfolgende Netsh-Befehlszeile zeigt ein Beispiel dafür, wie Sie zunächst eine Verbindung zum DHCP-Server mit der IP-Adresse *192.168.0.10* herstellen, dann den DHCP-Bereich mit der IP-Adresse *192.168.0.0* selektieren und mit *Add ReserveDip* die IP-Adresse *192.168.0.129* für den Client mit der MAC-Adresse *00-0C-29-7C-31-22* reservieren können. Durch Angabe der Zeichenfolge *MasterClient* legen Sie zeitgleich den Reservierungsnamen fest.

```
@Echo off
Netsh Dhcp Server 192.168.0.10 Scope 192.168.0.0 _
Add ReserveDip 192.168.0.129 000C297C3122 MasterClient
```

Listing 595: 2_07_017.bat

Die Lösung mit VBScript

Um die Reservierung einer IP-Adresse mit VBScript vorzunehmen, nutzen Sie wieder die *Run*-Methode des *Shell*-Objektes und führen die Netsh-Befehlszeile der Eingabeaufforderung aus, der Sie nur die erforderlichen Parameter (siehe vorheriger Abschnitt) zu übergeben brauchen.

Wenn Sie beispielsweise auf dem DHCP-Server mit der IP-Adresse *192.168.0.10* einen DHCP-Bereich mit der IP-Adresse *192.168.0.0* eingerichtet haben und diesem Bereich eine Reservierung hinzufügen möchten, so dass der DHCP-Client mit der MAC-Adresse *00-0C-29-7C-31-22* ausschließlich die IP-Adresse *192.168.0.129* zugewiesen bekommt, so können Sie hierfür das nachfolgende Skriptbeispiel nutzen:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Netsh Dhcip Server 192.168.0.10 Scope 192.168.0.0 _
Add ReserveDip 192.168.0.129 000C297C3122 MasterClient"
Set oShell = Nothing
```

Listing 596: 2_07_017.vbs

173 Server- und Bereichsoptionen konfigurieren

Mit dem Konfigurieren von Server- und Bereichsoptionen haben Sie die Möglichkeit, mit der Zuweisung einer dynamischen IP-Adresse nicht nur die eigentliche IP-Adresse an den DHCP-Client zu übergeben, sondern zeitgleich auch noch andere Konfigurationseinstellungen wie beispielsweise die IP-Adresse des DNS-Servers, die des Gateways und die des Zeitserver. Die Konfigurationseinstellungen können Sie dabei als Server-Optionen hinterlegen, womit die Einstellungen für alle DHCP-Bereiche des DHCP-Servers gelten. Sie können die Einstellungen aber auch nur für einen bestimmten Bereich konfigurieren. Hierbei sei angemerkt, dass Bereichsoptionen eine höhere Priorität haben und Server-Optionen automatisch überschrieben werden, falls Einstellungen sowohl als Server-Option als auch als Bereichsoption konfiguriert wurden.

Wenn Sie Server- und Bereichsoptionen definieren und damit bestimmte Konfigurationseinstellungen hinterlegen möchten, so rufen Sie zunächst die Management-Konsole DHCP auf, die Sie im STARTMENÜ unter VERWALTUNG finden. Erweitern Sie in der linken Fensterhälfte den DHCP-Server, für den Sie die Einstellungen vornehmen wollen, und markieren Sie den Eintrag SERVER-OPTIONEN, wenn Sie globale Einstellungen, die für alle Bereiche gelten sollen, vornehmen wollen. Wenn Sie dagegen Bereichsoptionen hinterlegen möchten, so erweitern Sie den entsprechenden Bereich, markieren unterhalb davon den Eintrag BEREICHSOPTIONEN und wählen im Kontextmenü OPTIONEN KONFIGURIEREN... aus.

Sie gelangen daraufhin in ein Dialogfenster, worüber Sie zunächst die zu konfigurierende Option markieren und – je nach ausgewählter Option – die Einstellung hinterlegen müssen. Wenn Sie beispielsweise die Einstellung des DNS-Servers hinterlegen wollen, so markieren Sie zunächst die Option DNS-SERVER und legen dann den Hostnamen oder aber die IP-Adresse des zu verwendenden DNS-Servers fest. Mit einem Klick auf OK wird die Einstellung übernommen und mit der Zuweisung einer dynamischen IP-Adresse an den DHCP-Client übertragen und dort verwendet.

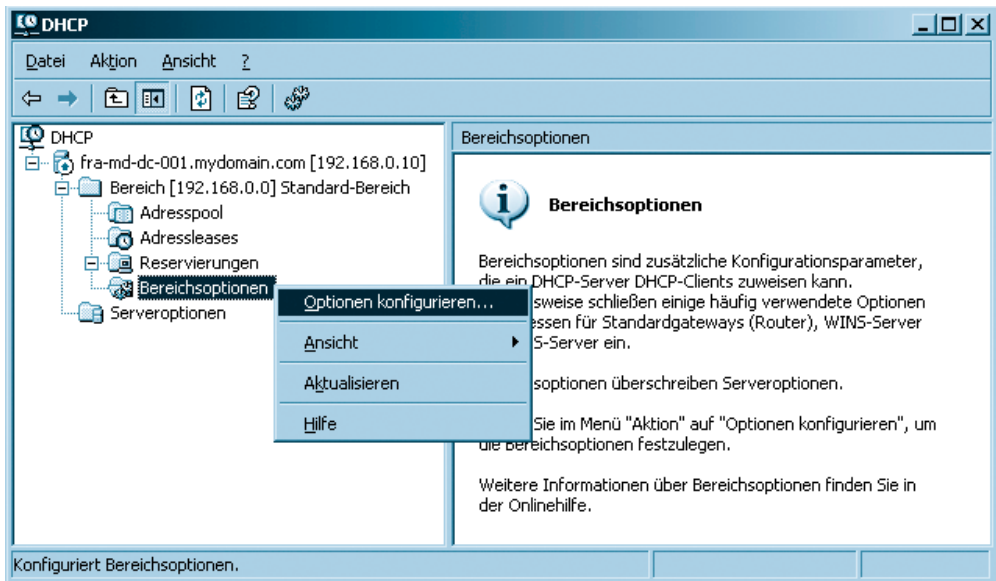


Abbildung 239: Die Bereichsoptionen konfigurieren

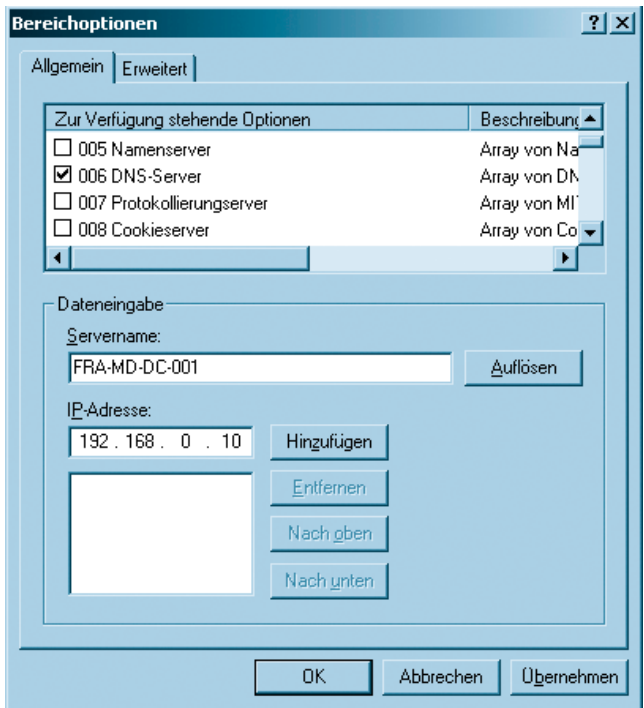


Abbildung 240: Die IP-Adresse des DNS-Servers hinterlegen

Die Lösung per Eingabeaufforderung

Der `Netsh`-Befehl, der standardmäßig im Befehlsumfang von Windows Server 2003 enthalten ist, erlaubt das Konfigurieren der Server- und Bereichsoptionen auch von der Eingabeaufforderung aus. Hierzu stellt der `Netsh`-Befehl den Subbefehl `Set OptionValue` zur Verfügung, dem Sie lediglich die entsprechende Optionskennung und auch den gewünschten Optionswert als Parameter übergeben.

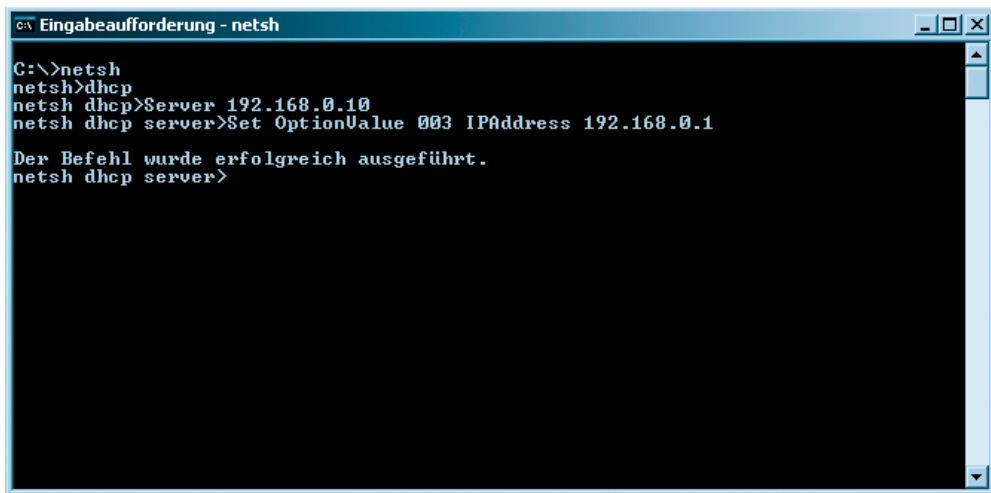
Die nachfolgende Tabelle zeigt eine kleine Übersicht der am häufigsten verwendeten Optionskennungen unter Angabe einer kurzen Beschreibung:

Optionskennung	Beschreibung
003	Router
004	Zeitserver
005	Namensserver
006	DNS-Server
012	Hostname des Clients
015	DNS-Domänenname
018	TFTP-Datei für Optionserweiterungen
028	Broadcast-Adresse
042	NTP-Server
044	WINS-/NBNS-Server
045	NetBIOS über TCP/IP
046	WINS-/NBT-Knotentyp
047	NetBIOS-Bereichskennung
066	Hostname des TFTP-Startservers
067	Name der TFTP-Startdatei
069	SMTP-Server
070	POP3-Server
071	NNTP-Server
072	WWW-Server
073	Finger-Server
074	IRC-Server

Tabelle 80: Die wichtigsten Optionskennungen und ihre Bedeutung

Um beispielsweise auf dem DHCP-Server mit der IP-Adresse `192.168.0.10` eine Server-Option zu konfigurieren und die IP-Adresse `192.168.0.1` des Routers zu hinterlegen, rufen Sie durch Eingabe von `Netsh` zunächst die `Netsh`-eigene Eingabeaufforderung auf, wechseln dann mit `Dhcp` in den DHCP-spezifischen Kontext und stellen mit `Server`

192.168.0.10 eine Verbindung zum DHCP-Server her. Anschließend wenden Sie den Subbefehl `Set OptionValue` an, dem Sie die Optionskennung 003 und den Optionswert `IPAddress 192.168.0.1` als Parameter übergeben.



```
C:\>netsh
netsh>dhcp
netsh dhcp>Server 192.168.0.10
netsh dhcp server>Set OptionValue 003 IPAddress 192.168.0.1

Der Befehl wurde erfolgreich ausgeführt.
netsh dhcp server>
```

Abbildung 241: Mit Netsh eine Server-Option konfigurieren

Gleiches lässt sich natürlich auch in einer einzigen Befehlszeile unterbringen, womit Sie die Server-Option beispielsweise auch per Batch-Datei konfigurieren können. Die nachfolgende Netsh-Befehlszeile zeigt ein solches Beispiel:

```
@Echo off
Netsh Dhcp Server 192.168.0.10 Set OptionValue 003 IPAddress 192.168.0.1
```

Listing 597: 2_07_018a.bat

Möchten Sie statt einer Server-Option eine Bereichsoption konfigurieren, so selektieren Sie nach dem Herstellen der Verbindung zum DHCP-Server mit `Scope IP-Adresse` (z.B. `Scope 192.168.0.0`) den entsprechenden Bereich und wenden erst dann den `Set OptionValue`-Subbefehl an, dem Sie wieder die Optionskennung und auch den Optionswert als Parameter übergeben.

```
@Echo off
Netsh Dhcp Server 192.168.0.10 Scope 192.168.0.0 Set OptionValue 003 _
    IPAddress 192.168.0.1
```

Listing 598: 2_07_018b.bat

Die Lösung mit VBScript

Um die Server- und Bereichsoptionen per VBScript zu konfigurieren, setzen Sie ebenfalls den Netsh-Befehl (siehe vorheriger Abschnitt) ein, den Sie mit der Run-Methode des Shell-Objektes jederzeit aus einem Skript heraus aufrufen können.

Wenn Sie beispielsweise die Server-Optionen bearbeiten möchten und die IP-Adresse 192.168.0.1 des zu verwendenden Routers hinterlegen wollen, so führen Sie mit der Run-Methode den Netsh-Befehl aus und übergeben dabei die erforderlichen Parameter, womit Sie zunächst in den DHCP-Kontext wechseln, dann die Verbindung zum DHCP-Server herstellen und zu guter Letzt die gewünschte Server-Option festsetzen können.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Netsh Dhcp Server 192.168.0.10 Set _
    OptionValue 003 IPAddress 192.168.0.1"
Set oShell = Nothing
```

Listing 599: 2_07_018a.vbs

Wenn Sie nicht die Server-Optionen, sondern die Optionen eines bestimmten DHCP-Bereichs konfigurieren wollen, so selektieren Sie nach dem Herstellen der Verbindung zum DHCP-Server mit Scope IP-Adresse noch den entsprechenden DHCP-Bereich und setzen erst dann mit Set OptionValue die gewünschte Bereichsoption fest.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Netsh Dhcp Server 192.168.0.10 Scope 192.168.0.0 _
    Set OptionValue 003 IPAddress 192.168.0.1"
Set oShell = Nothing
```

Listing 600: 2_07_018b.vbs

174 Anzahl der Konflikterkennungsversuche festlegen

Der DHCP-Server bietet die Möglichkeit einer Konflikterkennung an, womit das Risiko möglicher IP-Adresskonflikte gemindert werden kann. Mit Hilfe dieser Konflikterkennung wird eine IP-Adresse vor der dynamischen Zuweisung an einen DHCP-Client zunächst per Ping auf ihre Erreichbarkeit hin geprüft. Erfolgt ein Reply auf diesen Ping, so ist die IP-Adresse bereits in Benutzung, und eine andere IP-Adresse wird überprüft oder – nach Erreichen der maximalen Anzahl an Erkennungsversuchen – an den DHCP-Client zugewiesen.

Um die Konflikterkennung zu aktivieren und die Anzahl der Erkennungsversuche festzulegen, rufen Sie die Management-Konsole DHCP auf, die Sie im STARTMENÜ unter VERWALTUNG anklicken oder mit `Dhcpmgmt.msc` auch direkt aufrufen können. Innerhalb der Management-Konsole DHCP markieren Sie in der linken Fensterhälfte den DHCP-Server, für den Sie die Konflikterkennung konfigurieren möchten, und rufen aus dem Kontextmenü heraus die EIGENSCHAFTEN auf. Auf der Registerkarte ERWEITERT können Sie nun die Anzahl der Konflikterkennungsversuche festlegen, indem Sie einen Wert zwischen 1 und 5 (0 würde die Konflikterkennung wieder deaktivieren) eingeben (siehe Abbildung 242).

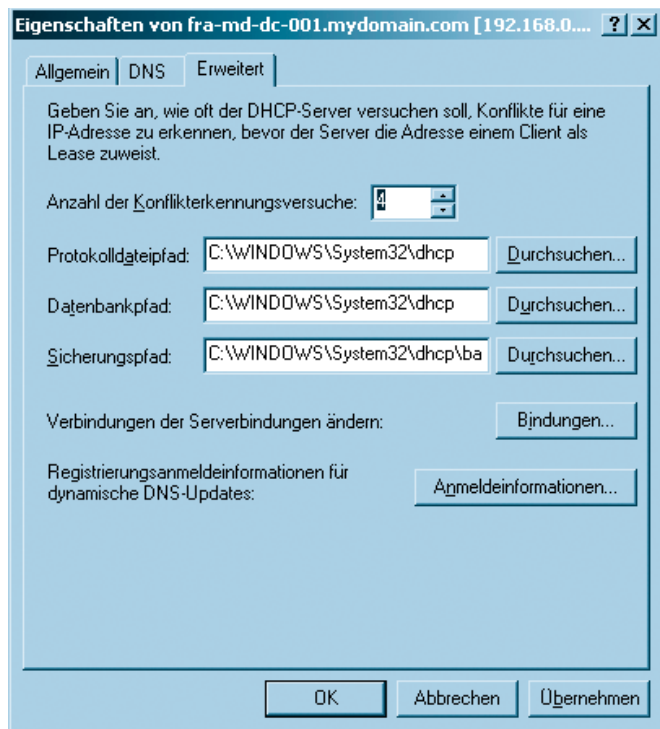


Abbildung 242: Die Anzahl der Konflikterkennungsversuche festlegen

Die Lösung per Eingabeaufforderung

Mit dem Netsh-Befehl lässt sich die Anzahl der Konflikterkennungsversuche auch von der Eingabeaufforderung aus festlegen. Hierzu rufen Sie durch Eingabe von Netsh die Netsh-eigene Eingabeaufforderung auf, wechseln dann mit `Dhcp` in den entsprechenden Kontext und stellen mit `Server` IP-Adresse (z.B. `Server 192.168.0.10`) eine Verbindung zu dem jeweiligen DHCP-Server her. Dann wenden Sie den Subbefehl `Set DetectConflictRetry` an und übergeben die Anzahl der Konflikterkennungsversuche (0–5).

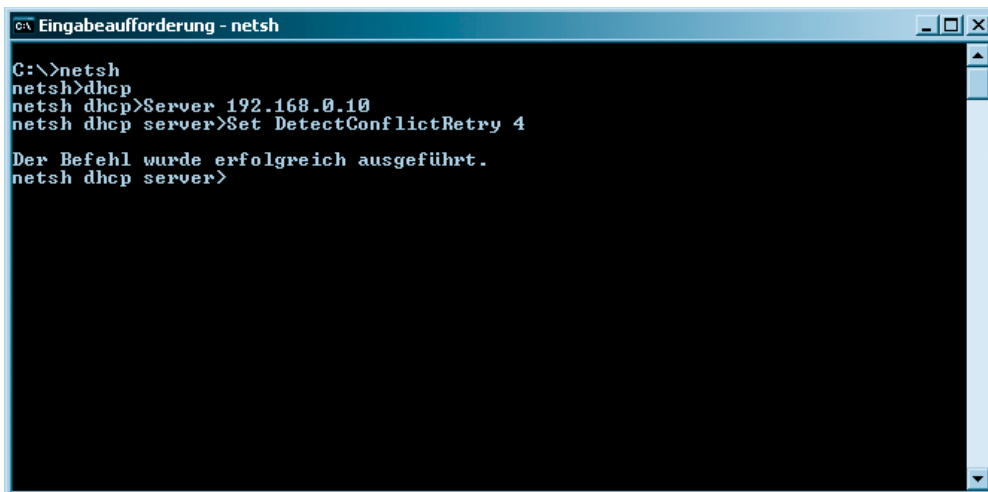


Abbildung 243: Mit Netsh die Konflikterkennungsversuche auf »4« setzen

Das nachfolgende Beispiel zeigt, wie Sie per Netsh eine Verbindung zum DHCP-Server mit der IP-Adresse *192.168.0.10* herstellen und dort die Anzahl der Konflikterkennungsversuche auf 4 setzen können.

```
@Echo off
Netsh Dhcp Server 192.168.0.10 Set DetectConflictRetry 4
```

Listing 601: *2_07_019.bat*

Die Lösung mit VBScript

Wenn Sie die Anzahl der Konflikterkennungsversuche per VBScript festlegen möchten, so wenden Sie den Netsh-Befehl der Eingabeaufforderung an und übergeben die erforderlichen Parameter (siehe vorheriger Abschnitt). Die folgenden Codezeilen zeigen ein Beispiel dafür, wie Sie die Konflikterkennungsversuche für den DHCP-Server mit der IP-Adresse *192.168.0.10* auf 4 setzen können:

```
Option Explicit

Dim oShell
Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Netsh Dhcp Server 192.168.0.10 _
    Set DetectConflictRetry 4"

Set oShell = Nothing
```

Listing 602: *2_07_019.vbs*

175 DHCP-Statistik abfragen

Wenn Sie auf die Schnelle herausfinden müssen, wie viele IP-Adressen ein DHCP-Server bereitstellt und wie viele davon noch verfügbar sind, so können Sie hierfür die DHCP-Statistik aufrufen. Denn die DHCP-Statistik gibt unter anderem Auskunft über die Betriebszeit (engl. = Uptime) des DHCP-Servers, über die Anzahl der bereitgestellten und der noch verfügbaren IP-Adressen, über die Anzahl der DHCP-Bereiche und vieles mehr.

Um die DHCP-Statistik aufzurufen, öffnen Sie zunächst die Management-Konsole DHCP, die Sie im STARTMENÜ unter VERWALTUNG nur anzuklicken brauchen oder mit `Dhcpmgmt.msc` auch direkt ausführen können. Markieren Sie anschließend den DHCP-Server, dessen Statistik Sie abfragen möchten, und wählen Sie im Kontextmenü den Eintrag STATISTIK ANZEIGEN... aus. Daraufhin wird ein Dialogfenster geöffnet, worüber die DHCP-relevanten Informationen angezeigt werde.

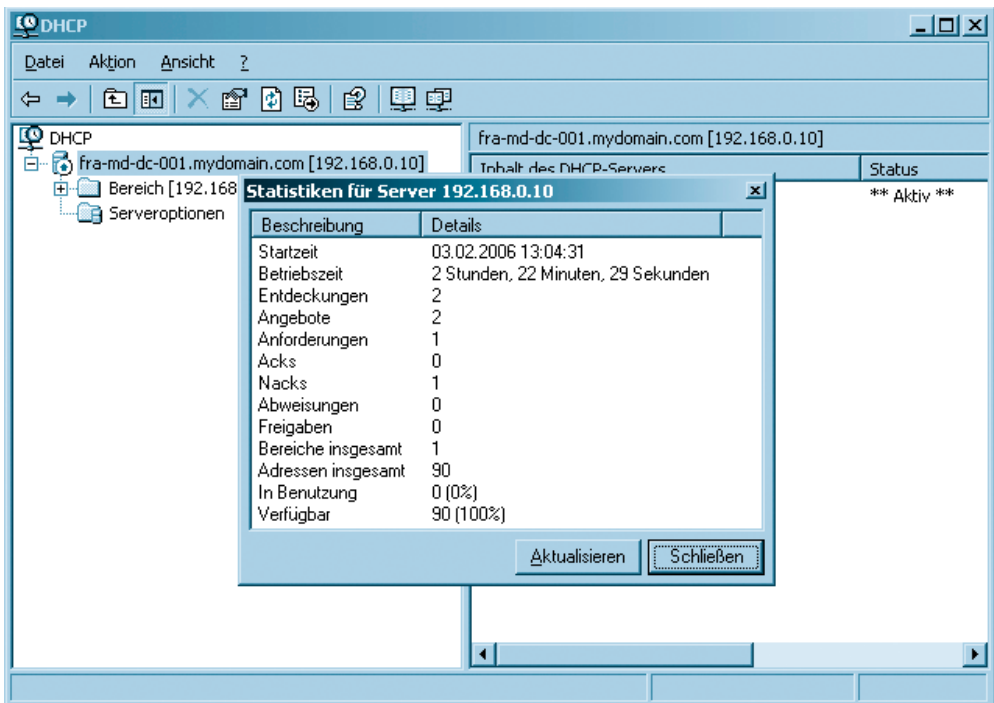
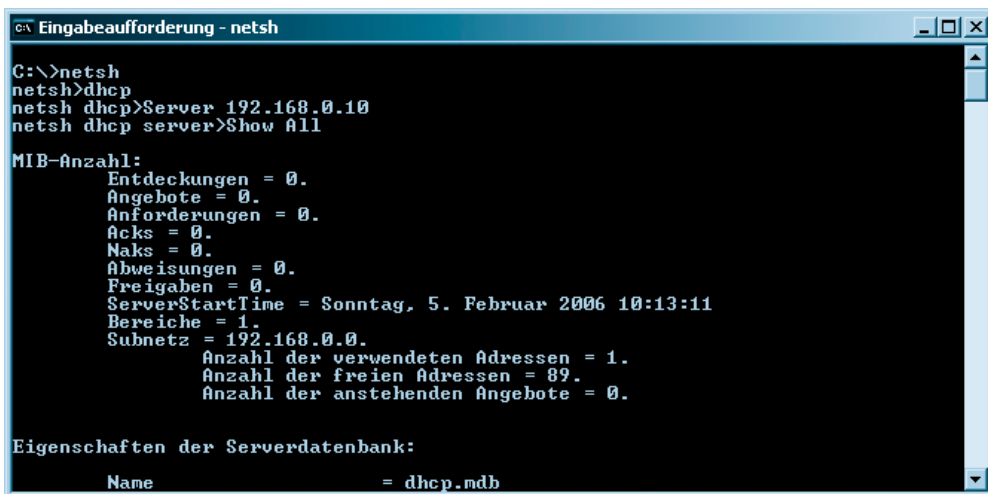


Abbildung 244: Die DHCP-Statistik abfragen

Die Lösung per Eingabeaufforderung

Die DHCP-Statistik lässt sich natürlich auch von der Eingabeaufforderung aus abfragen. Hierzu wenden Sie wieder den `Netsh`-Befehl an, um in die `Netsh`-eigene Eingabeaufforderung zu gelangen, wechseln dann mit `Dhcp` in den entsprechenden Kontext,

stellen mit Server IP-Adresse (z.B. *Server 192.168.0.10*) eine Verbindung zu dem jeweiligen DHCP-Server her und wenden dann den Subbefehl `Show all` an, woraufhin die Statistik am Bildschirm ausgegeben wird (siehe Abbildung 245).



```
C:\>netsh
netsh>dhcp
netsh dhcp>Server 192.168.0.10
netsh dhcp server>Show All

MIB-Anzahl:
    Entdeckungen = 0.
    Angebote = 0.
    Anforderungen = 0.
    Acks = 0.
    Naks = 0.
    Abweisungen = 0.
    Freigaben = 0.
    ServerStartTime = Sonntag, 5. Februar 2006 10:13:11
    Bereiche = 1.
    Subnetz = 192.168.0.0.
        Anzahl der verwendeten Adressen = 1.
        Anzahl der freien Adressen = 89.
        Anzahl der anstehenden Angebote = 0.

Eigenschaften der Serverdatenbank:

    Name                = dhcp.mdb
```

Abbildung 245: Die DHCP-Statistik mit Netsh abfragen

Gleiches lässt sich natürlich auch ohne den Einsatz der Netsh-eigenen Eingabeaufforderung realisieren, indem Sie dem Netsh-Befehl die Netsh-eigenen Subbefehle als Parameter übergeben. Und mit Hilfe einer Ausgabeumleitung lässt sich die Statistik natürlich auch in eine Datei ausgeben.

Die folgende Netsh-Befehlszeile zeigt ein Beispiel dafür, wie Sie die Statistik des DHCP-Servers mit der IP-Adresse 192.168.0.10 abfragen und per Ausgabeumleitung in die Datei *Result.txt* schreiben können.

```
@Echo off
Netsh Dhcp Server 192.168.0.10 Show All > Result.txt
```

Listing 603: 2_07_020.bat

Die Lösung mit VBScript

Wenn Sie die DHCP-Statistik aus einem VBScript heraus abfragen möchten, so führen Sie mit der `Run`-Methode des `Shell`-Objektes den Netsh-Befehl der Eingabeaufforderung aus und übergeben zeitgleich die entsprechenden Parameter (siehe vorheriger Abschnitt).

Das nachfolgende Code-Beispiel stellt eine Verbindung zum DHCP-Server *192.168.0.10* her und schreibt die DHCP-Statistik per Ausgabeumleitung in die Datei *Result.txt*:

```
Option Explicit
```

```
Dim oShell
```

```
Set oShell = CreateObject ("WScript.Shell")
```

```
oShell.Run "%comspec% /c Netsh Dhcp Server 192.168.0.10 _  
Show All > Result.txt"
```

```
Set oShell = Nothing
```

Listing 604: 2_07_020.vbs

176 Eine primäre Zone einrichten

Um auf einem DNS-Server eine neue primäre DNS-Zone einzurichten, rufen Sie die Management-Konsole DNS auf, die Sie im Menü VERWALTUNG anklicken oder mit *Dnsmgmt.msc* auch direkt aufrufen können. Markieren Sie in der linken Fensterhälfte den DNS-Server, und wählen Sie im Kontextmenü den Eintrag NEUE ZONE... aus. Daraufhin wird der ASSISTENT ZUM ERSTELLEN NEUER ZONEN gestartet. Schließen Sie den Willkommensbildschirm mit einem Klick auf WEITER, und legen Sie im nächsten Schritt fest, dass Sie eine primäre Zone erstellen möchten (siehe Abbildung 246).

Klicken Sie auf WEITER. Sie gelangen damit zur Dialogseite ACTIVE DIRECTORY-ZONEN-REPLIKATIONSBEREICH, worüber Sie den Replikationsbereich für die DNS-Daten festlegen können. Folgende Optionen stehen zur Auswahl:

- ▶ Auf allen DNS-Servern in der Active Directory-Gesamtstruktur
- ▶ Auf allen DNS-Servern in der Active Directory-Domäne
- ▶ Auf allen Domänencontrollern in der Active Directory-Domäne

Legen Sie fest, innerhalb welchem Bereich die DNS-Daten repliziert werden sollen, und klicken Sie auf WEITER. Im nächsten Schritt wählen Sie aus, ob Sie eine Forward- oder eine Reverse-Lookup-Zone einrichten möchten. Nehmen Sie die entsprechende Einstellung vor, und klicken Sie auf WEITER, um den Zonennamen (z.B. *Support.MyDomain.com*) eingeben zu können. Als Nächstes müssen Sie festlegen, ob Sie dynamische Updates (und wahlweise auch unsichere Updates) zulassen oder dynamische Aktualisierungen der DNS-Einträge unterbinden möchten. Wählen Sie die entsprechende Option aus, und klicken Sie auf WEITER und anschließend auf FERTIG STELLEN, woraufhin die neue Zone eingerichtet wird.

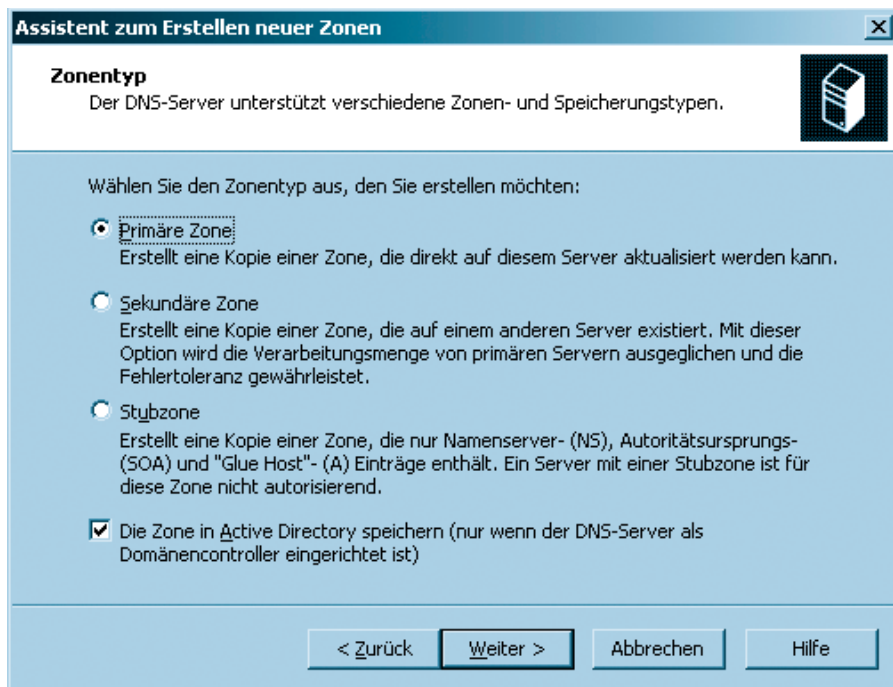


Abbildung 246: Eine primäre Zone erstellen

Die Lösung per Eingabeaufforderung

Um von der Eingabeaufforderung aus eine primäre DNS-Zone zu erstellen, können Sie den `Dnscmd`-Befehl verwenden, der Bestandteil der Windows Server 2003 Support-Tools (Ordner `\Support\Tools` der Installations-CD) ist. Wenn Sie beispielsweise auf dem DNS-Server mit dem Hostnamen `FRA-MD-DC-001` der Domäne `MyDomain.com` eine neue primäre Zone mit der Bezeichnung `Support.MyDomain.com` anlegen möchten, so können Sie hierfür folgende `Dnscmd`-Befehlszeile verwenden:

```
@Echo off
Dnscmd FRA-MD-DC-001.MyDomain.com /ZoneAdd Support.MyDomain.com /Primary
```

Listing 605: `2_07_021.bat`

Die Lösung mit VBScript

Mit der `CreateZone`-Methode der WMI-Klasse `MicrosoftDNS_Zone` (WMI-Namespace `root\MicrosoftDNS`) lassen sich primäre Zonen auch per VBScript erstellen. Hierzu stellen Sie im ersten Schritt eine Verbindung zum WMI-Namespace `root\MicrosoftDNS` des betreffenden DNS-Servers und zur WMI-Klasse `MicrosoftDNS_Zone` her. Anschließend wenden Sie die `CreateZone`-Methode an, die folgende Argumente erwartet:

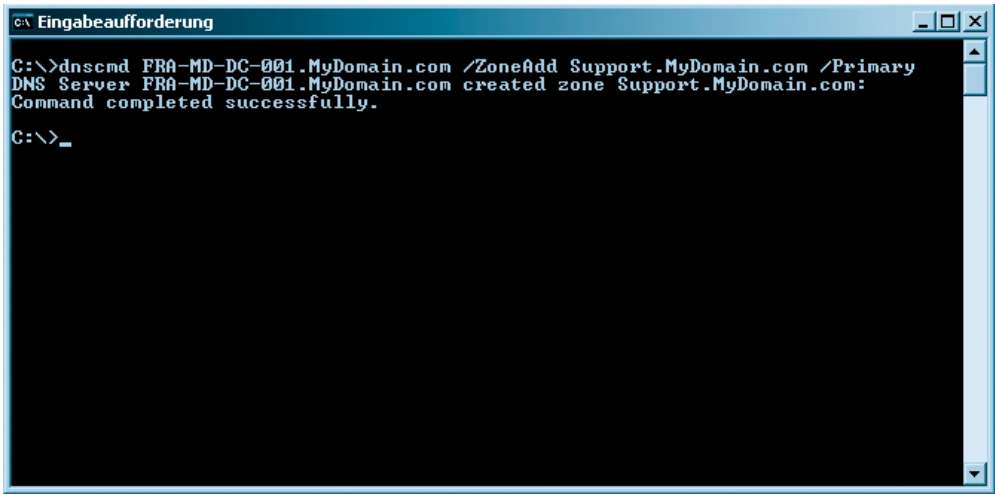


Abbildung 247: Mit »Dnscmd« eine neue Zone erstellen

Argument	Beschreibung
ZoneName	Name der neuen Zone
ZoneType	Typ der neuen Zone (0 = Primäre Zone, 1 = Sekundäre Zone, 2 = Stub Zone, 3 = Weiterleitung)
DSIntegrated	Wählen Sie den Wert True, wenn die Zonendaten im Active Directory gespeichert werden sollen, oder False, wenn die Zonendaten in einer Datei abgelegt werden sollen.
DataFileName	Name der Datei, in der die Zonendaten abgelegt sind
IPAddr	IP-Adresse des Master-DNS-Servers
AdminEmailName	E-Mail-Adresse des Administrators
RR	Microsoft DNS_Zone-Typ-Ausgabeparameter

Tabelle 81: Die Argumente der CreateZone-Methode

Das nachfolgende Listing zeigt ein Beispiel dafür, wie Sie auf dem DNS-Server *FRA-MD-DC-001* der Domäne *MyDomain.com* eine neue primäre Zone mit der Bezeichnung *Support.My Domain.com* anlegen können.

```

Option Explicit

Dim oDNS
Dim oDNSZone
Dim strDNSServer
Dim strPrimaryZone
  
```

Listing 606: 2_07_021.vbs

```
Dim strResult

strDNSServer = "FRA-MD-DC-001.MyDomain.com"
strPrimaryZone = "Support.MyDomain.com"

Set oDNS = GetObject("winMgmts:\\\" & strDNSServer & "\root\MicrosoftDNS")
Set oDNSZone = oDNS.Get("MicrosoftDNS_Zone")
strResult = oDNSZone.CreateZone(strPrimaryZone, 0 , True)

Set oDNS = Nothing
Set oDNSZone = Nothing
```

Listing 606: 2_07_021.vbs

177 Zonen löschen

Um eine bereits eingerichtete Zone wieder zu löschen, rufen Sie auf dem DNS-Server das Verwaltungstool DNS auf, das Sie im STARTMENÜ, genauer gesagt unter VERWALTUNG, nur anzuklicken brauchen oder mit Dnsmgmt.msc auch direkt aufrufen können.

Markieren Sie dann die Zone, die Sie löschen möchten, und wählen Sie im Kontextmenü der Zone den Eintrag LÖSCHEN aus. Sie werden daraufhin noch einmal gefragt, ob Sie die Zone auch wirklich entfernen wollen. Klicken Sie auf JA, um den Löschvorgang zu bestätigen. Die Zone wird daraufhin unwiderruflich gelöscht.

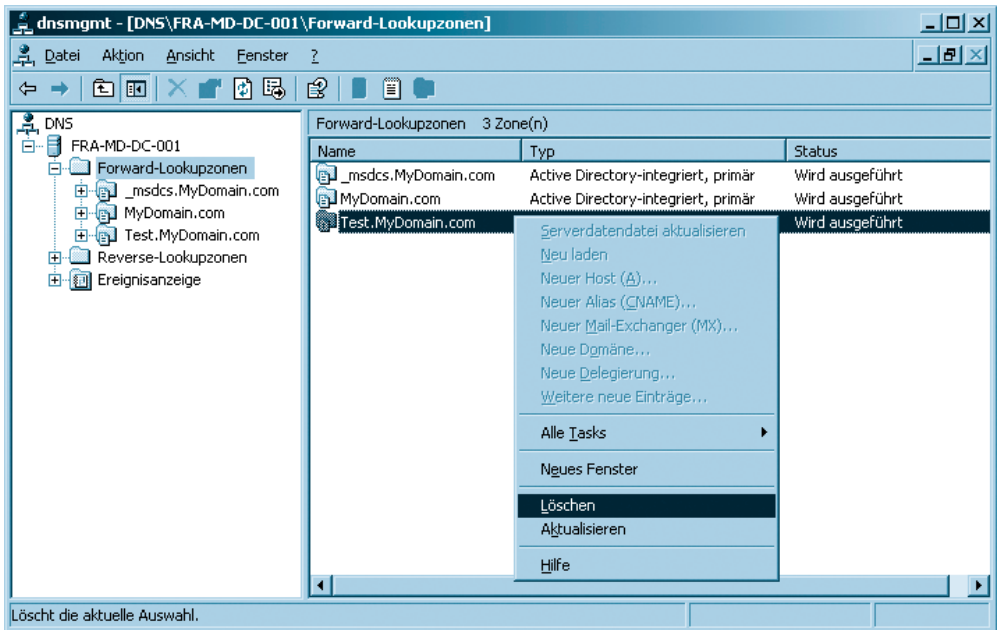


Abbildung 248: Eine Zone löschen

Die Lösung per Eingabeaufforderung

Vorhandene Zonen lassen sich auch von der Eingabeaufforderung aus löschen. Hierzu verwenden Sie wieder das *Windows Server 2003 Support-Tool* namens *Dnscmd.exe* und – wenn Sie den Befehl direkt auf dem DNS-Server ausführen – übergeben lediglich die Aktion */ZoneDelete* und den Namen der zu löschenden Zone als Parameter.

```
@Echo off
Dnscmd /ZoneDelete Test.MyDomain.com /Dsdel
```

Listing 607: 2_07_022a.bat

Der *Dnscmd*-Befehl fragt daraufhin noch einmal nach, ob Sie die Zone auch wirklich löschen wollen. Drücken Sie die ☐-Taste, um das Löschen zu bestätigen, woraufhin die Zone dann sofort gelöscht wird. Mit Hilfe des Parameters */F* können Sie die Rückfrage aber auch unterdrücken und den *Dnscmd*-Befehl beispielsweise auch in einer Batch-Datei verwenden.

Wenn Sie den Befehl nicht direkt auf dem DNS-Server ausführen, dann müssen Sie dem *Dnscmd*-Befehl den DNS-Namen des entsprechenden DNS-Servers übergeben. Die nachfolgende *Dnscmd*-Befehlszeile stellt eine Verbindung zum Server *FRA-MD-DC-001.Mydomain.com* her und löscht dort die Zone *Test.MyDomain.com*.

```
@Echo off
Dnscmd FRA-MD-DC-001.MyDomain.com /ZoneDelete Test.MyDomain.com /Dsdel
```

Listing 608: 2_07_022b.bat

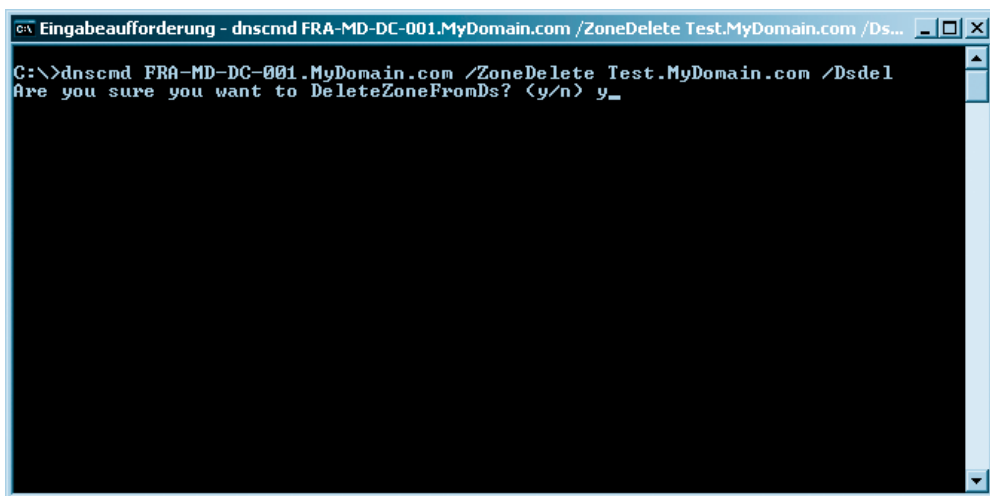


Abbildung 249: Eine Zone löschen

Die Lösung mit VBScript

Um per VBScript eine vorhandene Zone zu löschen, gehen Sie genauso vor wie auch in der Eingabeaufforderung. Sprich: Sie setzen den Dnscmd-Befehl mit dem Parameter /ZoneDelete ein und fügen den Namen der zu löschenden Zone an.

Das nachfolgende Skriptbeispiel zeigt, wie Sie den Dnscmd-Befehl ausführen und die Zone *Test.MyDomain.com* löschen können:

```
Option Explicit
```

```
Dim oShell
```

```
Set oShell = CreateObject ("WScript.Shell")
```

```
oShell.Run "%comspec% /c Dnscmd FRA-MD-DC-001.MyDomain.com _  
/ZoneDelete Test.MyDomain.com"
```

```
Set oShell = Nothing
```

Listing 609: 2_07_022.vbs

178 Einen neuen Host-Eintrag erzeugen

Um unter grafischer Oberfläche einen neuen DNS-Eintrag für einen Host zu erzeugen, rufen Sie wieder die Management-Konsole DNS auf, die Sie im Menü VERWALTUNG anklicken oder mit *Dnsmgmt.msc* auch direkt aufrufen können. Erweitern Sie in der linken Fensterhälfte die Zone, für die Sie einen neuen DNS-Eintrag einrichten möchten, und wählen Sie im Kontextmenü den Eintrag NEUER HOST (A)... aus.

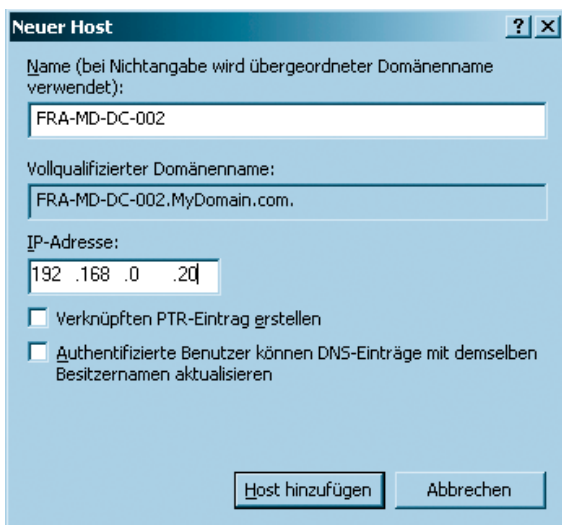


Abbildung 250: Einen neuen Host-Eintrag erzeugen

Sie gelangen daraufhin in das Dialogfenster NEUER HOST und können den DNS-Namen wie auch die dazugehörige IP-Adresse hinterlegen. Geben Sie den Hostnamen und die IP-Adresse ein, und klicken Sie anschließend auf OK, um den neuen Eintrag anzulegen.

Die Lösung per Eingabeaufforderung

Um von der Eingabeaufforderung aus einen neuen Host-Eintrag zu erstellen, können Sie wieder das Tool *Dnscmd.exe* einsetzen, das Bestandteil der *Windows Server 2003 Support-Tools* ist. Wenn Sie beispielsweise auf dem DNS-Server *FRA-MD-DC-001* für die Zone *MyDomain.com* einen neuen Host-Eintrag für *FRA-MD-DC-002* mit der IP-Adresse *192.168.0.2* einrichten möchten, so geben Sie hierfür folgende *Dnscmd*-Befehlszeile ein:

```
@Echo off
Dnscmd FRA-MD-DC-001.MyDomain.com /RecordAdd MyDomain.com FRA-MD-DC-002 _
A 192.168.0.2
```

Listing 610: 2_07_023.bat

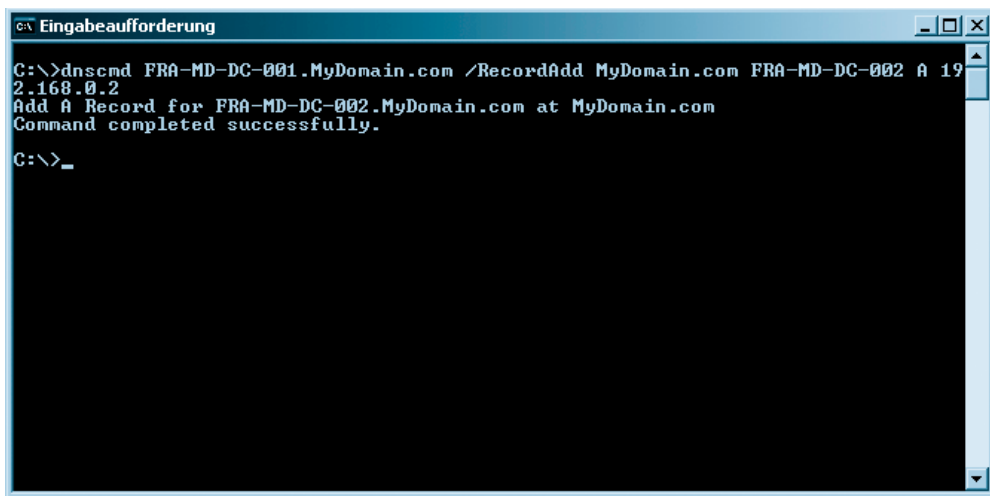


Abbildung 251: Einen neuen Host-Eintrag erzeugen

Die Lösung mit VBScript

Mit der *Run*-Methode des *Shell*-Objektes lässt sich die *Dnscmd*-Befehlszeile auch per VBScript ausführen, womit Sie den neuen DNS-Eintrag auf recht einfache Art und Weise auch aus einem VBScript heraus erzeugen können.

Option Explicit

Dim oShell

Set oShell = CreateObject ("WScript.Shell")

```
oShell.Run "%comspec% /c Dnscmd FRA-MD-DC-001.MyDomain.com /RecordAdd _
MyDomain.com FRA-MD-DC-002 A 192.168.0.2"
```

Set oShell = Nothing

Listing 611: 2_07_023.vbs

179 DNS-Cache leeren

Um den Cache des DNS-Servers zu leeren, rufen Sie zunächst die Management-Konsole DNS auf. Wechseln Sie hierzu in das STARTMENÜ, erweitern Sie VERWALTUNG, und klicken Sie dann auf DNS, oder wählen Sie im STARTMENÜ den Eintrag AUSFÜHREN, und geben Sie `Dnsmgmt.msc` ein, um die Management-Konsole direkt aufzurufen.

Markieren Sie anschließend in der linken Fensterhälfte den DNS-Server, öffnen Sie dessen Kontextmenü, und wählen Sie dort den Eintrag CACHE LÖSCHEN aus. Der DNS-Cache wird daraufhin vollständig geleert.

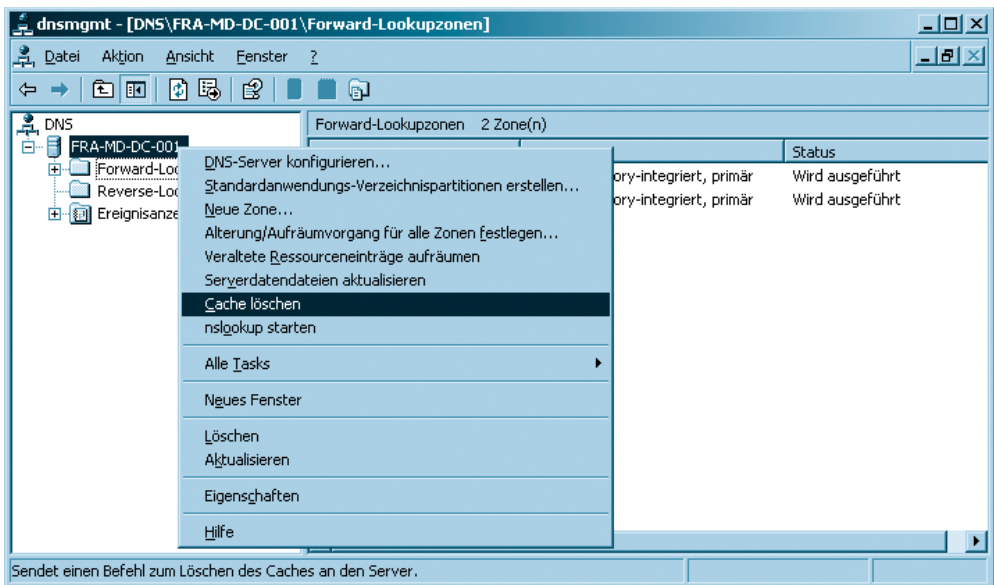


Abbildung 252: Den Cache leeren

Die Lösung per Eingabeaufforderung

Mit dem *Windows Server 2003 Support-Tool* namens *Dnscmd.exe* lässt sich der Cache auch von der Eingabeaufforderung aus leeren. Wenn Sie das Tool direkt auf dem DNS-Server ausführen, dessen Cache Sie leeren wollen, so brauchen Sie das Tool *Dnscmd* nur mit Parameter */ClearCache* aufzurufen.

```
@Echo off
Dnscmd /ClearCache
```

Listing 612: 2_07_024a.bat

Falls Sie das Tool jedoch von einem anderen System aus aufrufen, so müssen Sie dem *Dnscmd*-Befehl zunächst den DNS-Namen des DNS-Servers übergeben, dessen Cache Sie löschen wollen, und übergeben erst dann den Parameter */ClearCache*.

```
@Echo off
Dnscmd FRA-MD-DC-002.MyDomain.com /ClearCache
```

Listing 613: 2_07_024b.bat

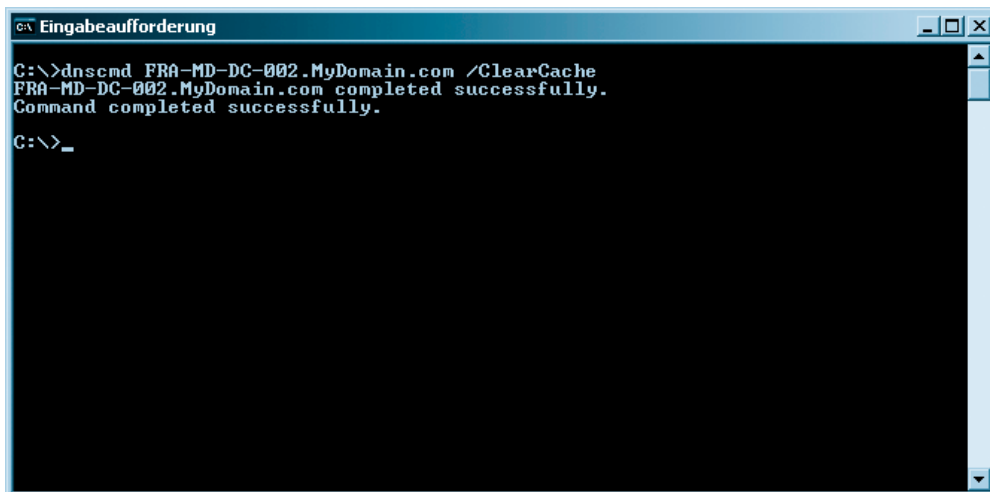


Abbildung 253: Den DNS-Cache eines anderen DNS-Servers löschen

Die Lösung mit VBScript

Mit der *ClearCache*-Methode der WMI-Klasse *MicrosoftDNS_Cache* lässt sich der Cache eines DNS-Servers auch per VBScript löschen. Stellen Sie hierzu im ersten Schritt eine Verbindung zum WMI-Namespace *root\MicrosoftDNS* des betreffenden DNS-Servers her, und selektieren Sie mit Hilfe einer *ExecQuery*-Abfrage die Instanzen der

WMI-Klasse `MicrosoftDNS_Cache`. Setzen Sie anschließend eine `For Each`-Schleife ein, und wenden Sie die `ClearCache`-Methode (die keinerlei Argumente erwartet) gegen die einzelnen Instanzen an.

```
Option Explicit

Dim oDNS
Dim strDNSServer
Dim colItems
Dim objItem

strDNSServer = "FRA-MD-DC-001.MyDomain.com"
Set oDNS = GetObject("winMgmts:\\\" & strDNSServer & "\root\MicrosoftDNS")
Set colItems = oDNS.ExecQuery("SELECT * FROM MicrosoftDNS_Cache")

For each objItem in colItems
    objItem.ClearCache()
Next

Set oDNS = Nothing
```

Listing 614: 2_07_024.vbs

180 Dynamische Updates zulassen und verweigern

Um die Einstellung für die dynamischen Updates vorzunehmen, starten Sie zunächst die Management-Konsole DNS, die Sie im STARTMENÜ (unter VERWALTUNG) anklicken oder mit `Dnsmgmt.msc` auch direkt aufrufen können. Markieren Sie die Zone, für die Sie die Einstellung vornehmen möchten, und rufen Sie aus dem Kontextmenü heraus die EIGENSCHAFTEN auf. Auf der Registerkarte ALLGEMEIN können Sie nun auswählen, ob ausschließlich sichere, sichere und auch unsichere oder keine dynamischen Updates verwendet werden sollen. Wählen Sie die gewünschte Einstellung aus, und übernehmen Sie die Änderung mit einem Klick auf ÜBERNEHMEN (siehe Abbildung 254).

Die Lösung per Eingabeaufforderung

Um die Einstellung für die dynamischen Updates von der Eingabeaufforderung aus zu verändern, verwenden Sie wieder den `Dnscmd`-Befehl, dem Sie zunächst den Parameter `/Config` übergeben, womit Sie das Ändern einer Konfigurationseinstellung einleiten. Anschließend wenden Sie den Parameter `/AllowUpdate` an und übergeben den Wert der neuen Einstellung. Folgende Einstellungen sind möglich:

Wert	Beschreibung
0	Keine dynamische Updates verwenden
1	Sichere und unsichere dynamische Updates
2	Nur sichere dynamische Updates

Tabelle 82: Die Werte für »AllowUpdate«

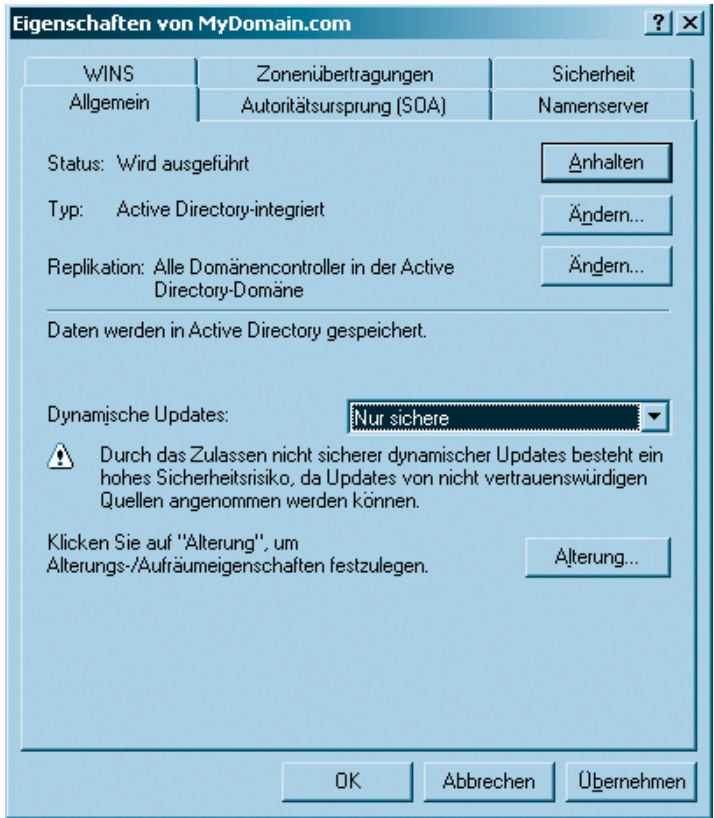


Abbildung 254: Nur sichere dynamische Updates zulassen

Die nachfolgende `Dnscmd`-Befehlszeile ändert die Einstellung der dynamischen Updates dahingehend, dass ausschließlich sichere dynamische Updates zugelassen werden. Nicht sichere dynamische Updates werden verweigert.

```
@Echo off
Dnscmd /Config /AllowUpdate 2
```

Listing 615: 2_07_025.bat

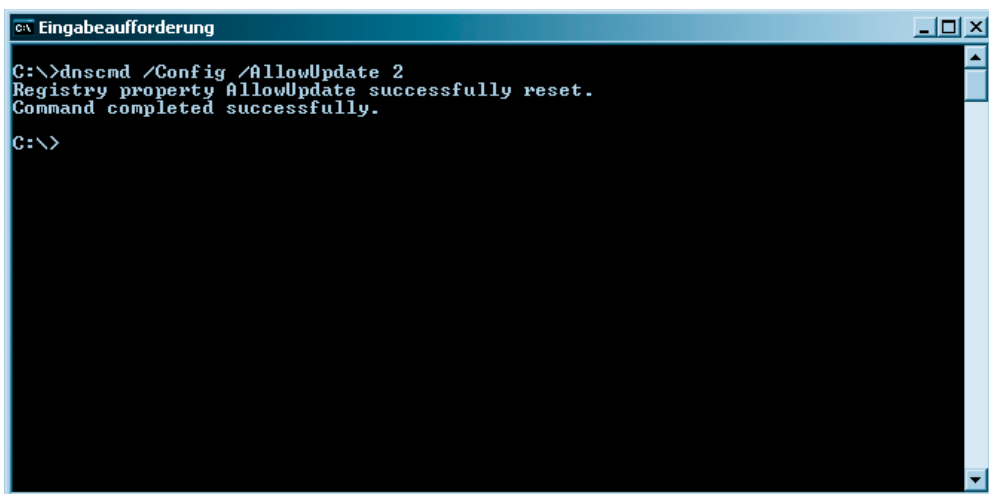


Abbildung 255: Den Wert für AllowUpdate auf »2« setzen

Die Lösung mit VBScript

Wenn Sie die Konfiguration der dynamischen Updates per VBScript vornehmen möchten, so führen Sie mit Hilfe der Run-Methode des Shell-Objektes den Dnscmd-Befehl der Eingabeaufforderung aus und weisen der Eigenschaft AllowUpdates den entsprechenden Wert (siehe vorheriger Abschnitt) zu.

Mit dem folgenden Skript wird der Eigenschaft AllowUpdates beispielsweise der Wert 2 zugewiesen, womit nur noch sichere dynamische Updates zugelassen werden. Der Wert 0 dagegen würde bewirken, dass keinerlei dynamische Updates mehr zugelassen werden.

```
Option Explicit
```

```
Dim oShell
```

```
Set oShell = CreateObject ("WScript.Shell")
```

```
oShell.Run "%comspec% /c Dnscmd /Config /AllowUpdate 2"
```

```
Set oShell = Nothing
```

Listing 616: 2_07_025.vbs

181 DNS-Ereignisprotokollierung konfigurieren

Mit dem Einrichten des DNS-Servers wird die Ereignisanzeige von Windows Server 2003 um ein Protokoll mit der Bezeichnung DNS-SERVER erweitert, worin die Ereignisse zum DNS-Server protokolliert werden. In diesem Protokoll werden per Default alle Informationen, Warnungen und auch Fehler des DNS-Servers hinterlegt.

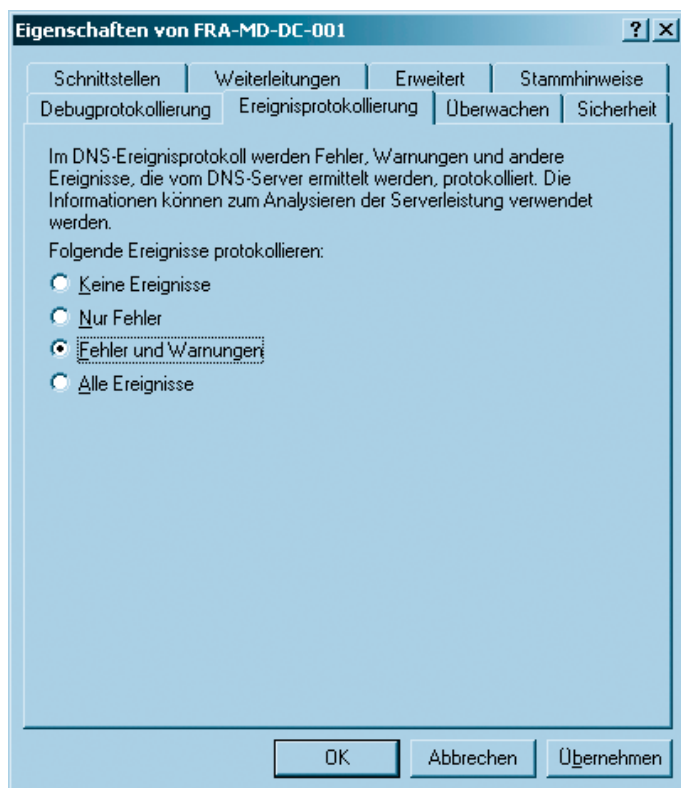


Abbildung 256: Die Ereignistypen festlegen

Für den Fall, dass Sie nicht alle Ereignisse aufzeichnen möchten, sondern beispielsweise nur *Fehler* oder *Fehler* und auch *Warnungen*, so können Sie dies in den Eigenschaften des DNS-Servers festlegen. Rufen Sie hierzu die Management-Konsole DNS auf, die Sie zum einen im STARTMENÜ (unter VERWALTUNG) anklicken oder zum anderen mit `Dnsmgmt.msc` auch direkt aufrufen können. Markieren Sie den DNS-Server, für den Sie die Einstellung vornehmen möchten, und rufen Sie aus dem Kontextmenü heraus die EIGENSCHAFTEN auf. Wechseln Sie dann auf die Registerkarte EREIGNISPROTOKOLLIERUNG. Dort können Sie nun auswählen, welche Ereignistypen künftig protokolliert werden sollen (siehe Abbildung 256).

Die Lösung per Eingabeaufforderung

Die Ereignisprotokollierung lässt sich natürlich auch von der Eingabeaufforderung aus konfigurieren, wofür Sie wieder den `Dnscmd`-Befehl einsetzen können. Der `Dnscmd`-Befehl erwartet dabei zunächst den Parameter `/Config`, womit Sie die Änderung der Konfigurationseinstellung einleiten. Im Anschluss daran verwenden Sie den Parameter `/EventLogLevel` und übergeben den Wert der gewünschten Konfigurationseinstellung. Folgende Einstellungen sind möglich:

Wert	Beschreibung
0	Keine Ereignisse protokollieren
1	Nur Fehler protokollieren
2	Nur Fehler und Warnungen protokollieren
4	Alle Ereignisse (Fehler, Warnungen und Informationen) protokollieren

Tabelle 83: Die Werte der Ereignistypen

Wenn Sie beispielsweise festlegen möchten, dass ausschließlich Ereignisse der Ereignistypen *Fehler* und *Warnungen* protokolliert werden, so wenden Sie den `Dnscmd`-Befehl mit dem Parameter `/Config` an und übergeben mit `/EventLogLevel` den Wert 2.

```
@Echo off
Dnscmd /Config /EventLogLevel 2
```

Listing 617: 2_07_026.bat

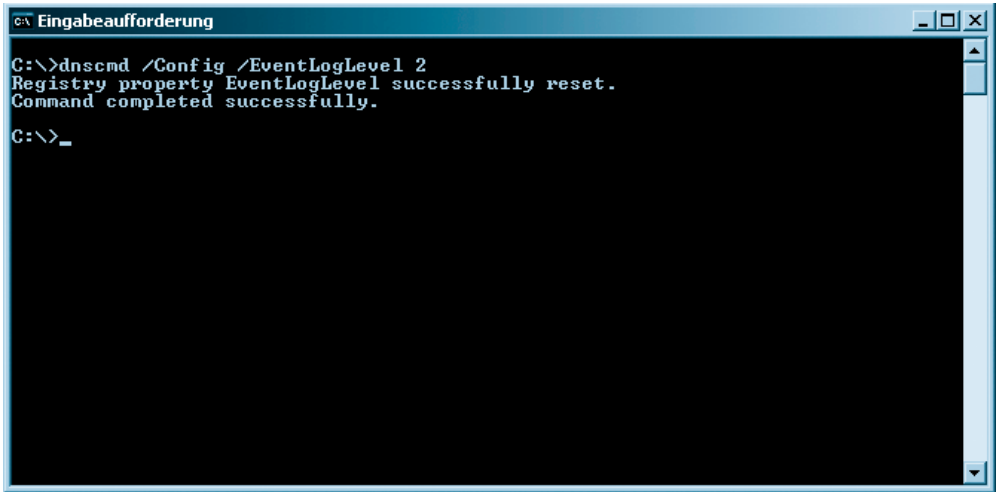


Abbildung 257: Den EventLogLevel auf »2« setzen

Die Lösung mit VBScript

Um die zu protokollierenden Ereignistypen per VBScript festzulegen, setzen Sie ebenfalls den `Dnscmd`-Befehl mit dem Parameter `/Config` ein und übergeben mit `/EventLogLevel` den Wert der gewünschten Konfigurationseinstellung (siehe *Tabelle 83: Die Werte der Ereignistypen* auf Seite 733).

Das nachfolgende Beispiel zeigt, wie Sie die Ereignisprotokollierung dahingehend ändern können, dass nicht mehr alle Ereignistypen (*Informationen*, *Warnungen* und *Fehler*) protokolliert werden, sondern ausschließlich *Warnungen* und *Fehler*.

Option Explicit

```
Dim oShell
Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Dnscmd /Config /EventLogLevel 2"
```

```
Set oShell = Nothing
```

Listing 618: 2_07_026.vbs

Automatisierung

Im Windows Server 2003-Umfeld gibt es eine ganze Reihe administrativer Aufgaben, die regelmäßig durchgeführt werden müssen. Dies können z.B. Wartungsaufgaben wie Datensicherung, Defragmentierung, Archivierung von Log-Dateien und dergleichen mehr sein, aber auch Konfigurationsarbeiten, die nach der Installation des Betriebssystems oder nach dem Redesign der IT-Infrastruktur durchgeführt werden müssen. Auch administrative Tätigkeiten aus der Benutzerverwaltung und aus dem Bereich des Systemmanagements sind oftmals regelmäßiger Natur.

Einen Großteil dieser Aufgaben lassen sich mit wenig Aufwand automatisieren, womit Sie einerseits viel Zeit einsparen und andererseits auch Fehler, die sich durch eine manuelle Administration ergeben können, schon im Vorfeld minimieren können. Und da der Windows Server 2003 bereits von Haus aus entsprechende Werkzeuge mitbringt, gilt es also nur, diese auch entsprechend einzusetzen.

Exkurs

Geplante Tasks, Sctasks und AT

Der Windows Server 2003 bietet mehrere Möglichkeiten der Automatisierung an. Um neue Aufgaben unter grafischer Oberfläche zu automatisieren, steht Ihnen in der Systemsteuerung der Assistent zur Einrichtung geplanter Tasks zur Verfügung. In der Eingabeaufforderung dagegen lassen sich neue Tasks sowohl mit `Sctasks.exe` als auch mit `At.exe` einrichten. Ersteres ist die neuere Variante und soll den aus Kompatibilitätsgründen mit älteren Systemen noch vorhandenen AT-Taskplaner langsam ablösen, wobei aber auch gleich erwähnt werden muss, dass diese beiden Werkzeuge nicht miteinander kompatibel sind. Das heißt, dass vorhandene AT-Tasks auch weiterhin mit `At.exe` bearbeitet werden müssen, da Sie den `Sctasks`-Befehl nur für die Tasks verwenden können, die entweder mit `Sctasks` oder aber mit dem Assistenten zur Einrichtung geplanter Tasks erstellt wurden.

182 Existiert der geplante Task bereits?

Bevor Sie einen neuen Task planen, sollten Sie zunächst überprüfen, ob der betreffende Task bereits existiert. Denn auch Tasks müssen – zumindest in ihrer Bezeichnung – eindeutig sein. Unter grafischer Oberfläche schauen Sie hierzu in der Übersicht der geplanten Tasks nach, ob der gewünschte Task bereits existiert. Öffnen Sie das Systemsteuerungsobjekt GEPLANTE TASKS, und prüfen Sie deren Einträge. Mit einem Doppelklick können Sie die einzelnen Tasks öffnen, wobei Sie auf der Registerkarte TASK die Befehlszeile oder das Programm, das automatisiert wurde, ansehen können.

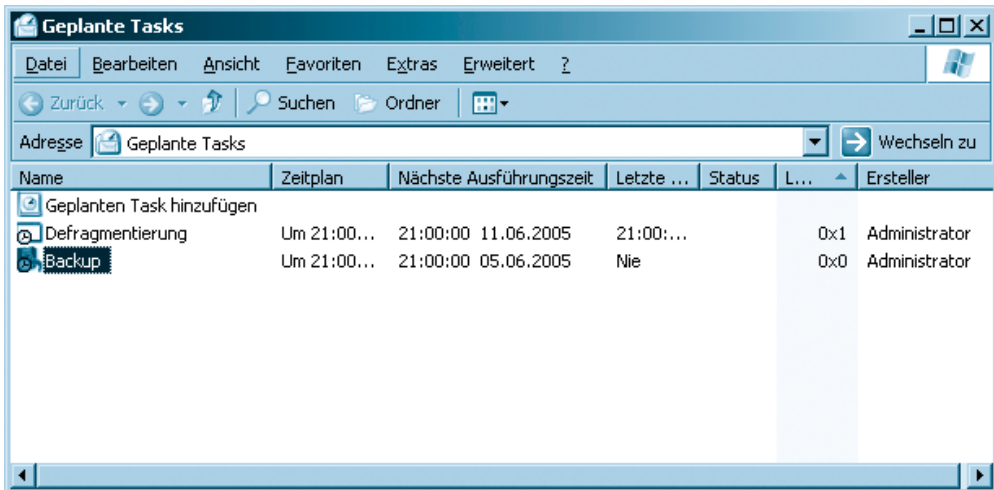


Abbildung 258: Der Task existiert bereits.

Möchten Sie diese Überprüfung nicht auf dem lokalen Server vornehmen, sondern auf einem Remote-System, so stellen Sie eine Verbindung zum Ordner `%System-root%\Tasks` des Zielsystems her und überprüfen die darin enthaltenen Einträge. Eine andere Möglichkeit wäre, eine Remotedesktop-Verbindung zu dem Zielsystem herzustellen und das Systemsteuerungsobjekt GEPLANTE TASKS zu öffnen.

Die Lösung per Eingabeaufforderung

Um in der Eingabeaufforderung das Vorhandensein eines mit `At.exe` eingerichteten Tasks zu überprüfen, lassen Sie sich mit dem Befehl `At.exe` (ohne Angabe weiterer Parameter) alle eingerichteten Tasks anzeigen und prüfen deren Befehlszeilen und wahlweise auch deren Ausführungszeiten. Sie können das Ergebnis der `AT`-Abfrage auch in eine Textdatei umleiten und mit Hilfe von `Findstr.exe` nach einer bestimmten Zeichenfolge suchen, um die Abfrage zu automatisieren.

Wenn Sie zum Beispiel herausfinden möchten, ob auf dem Computersystem bereits ein mit `At.exe` eingerichteter Task existiert, der die Befehlszeile `C:\Windows\System32\Defrag.exe C: /a` ausführt, so führen Sie `At.exe` aus, leiten das Ergebnis in eine beliebige Textdatei um und führen anschließend eine `Findstr`-Suche aus, um nach der zuvor genannten Befehlszeile zu suchen.

Die nachfolgenden Befehlszeilen zeigen eine solche Möglichkeit: Zunächst werden die bereits eingerichteten `AT`-Tasks abgefragt und per Ausgabeumleitung in die Datei `C:\Temp\Joblist.txt` geschrieben. Anschließend wird der Inhalt der Textdatei mit `Findstr.exe` durchsucht, wobei als Suchbegriff die Befehlszeile des `AT`-Tasks verwendet wird. Der Parameter `/c` gibt dabei an, dass nur nach einer exakten Übereinstimmung gesucht werden soll. Und konnte der Suchbegriff in der Textdatei gefunden werden, so erhält `ErrorLevel` den Wert 0, und die Meldung `Der Task existiert ...` wird ausgegeben.

```
@Echo off
AT > C:\Temp\Joblist.txt
Findstr /c:"C:\Windows\System32\Defrag.exe C:" _
    C:\Temp\Joblist.txt > NUL
IF %ERRORLEVEL% EQU 0 Goto Message
Goto Ende

:Message
Echo Der Task existiert ...

:Ende
```

Listing 619: 2_08_001a.bat

Um das Vorhandensein eines Tasks zu überprüfen, der unter grafischer Oberfläche mit dem Assistenten oder in der Eingabeaufforderung mit `Schtasks` eingerichtet wurde, gehen Sie ähnlich wie im vorherigen Beispiel vor. Sie lassen sich zunächst alle Tasks anzeigen, leiten die Ausgabe in eine Textdatei um, führen eine `Findstr`-Suche durch und fragen lediglich den Rückgabewert ab.

Die Abfrage der bereits eingerichteten Tasks erfolgt hierbei mit dem Befehl `Schtasks /Query`, der ohne Angabe weiterer Parameter den Namen des Tasks, die nächste Ausführungszeit wie auch den Status in Form einer Tabelle ausgibt. Mit dem Parameter `/v` können Sie die Ausgabe aller Informationen zum Task erzwingen, womit auch das Programm bzw. die Befehlszeile des Tasks ausgegeben wird. Und mit dem Parameter `/fo List` können Sie `Schtasks /Query` dazu veranlassen, die Task-Informationen nicht mehr als Tabelle, sondern in einer etwas übersichtlicheren Liste auszugeben.

```
@Echo off
Schtasks /Query /v /fo List > C:\Temp\Joblist.txt
Findstr /c:"C:\Windows\System32\Defrag.exe C:" _
    C:\Temp\Joblist.txt > NUL
If %ErrorLevel% EQU 0 Goto Message
Goto Ende

:Message
Echo Der Task existiert ...

:Ende
```

Listing 620: 2_08_001b.bat

Die Lösung mit VBScript

Um das Vorhandensein eines mit At.exe eingerichteten Tasks per VBScript zu prüfen, führen Sie mit Hilfe der ExecQuery-Methode der WMI-Klasse Win32_ScheduledJob (WMI-Namespace root\cimv2) eine Abfrage durch, woraufhin sämtliche Tasks in Form einer Kollektion zurückgeliefert werden. Mit einer For Each-Schleife können Sie dann auf die einzelnen Tasks zugreifen und deren Eigenschaften (siehe nachfolgende Tabelle) abfragen.

Eigenschaft	Beschreibung
Caption	Bezeichnung des Tasks
Command	Programm oder Befehlszeile, die durch den Task ausgeführt wird
DaysOfMonth	Tag (1–31), an dem der Task ausgeführt wird (siehe Tabelle 85: Die Werte der Tage auf Seite 739)
DaysOfWeek	Wochentag (Mo–So), an dem der Task ausgeführt wird (siehe Tabelle 86: Die Werte der Wochentage auf Seite 740)
Description	Beschreibung des Tasks
ElapsedTime	Dauer der letzten Task-Ausführung
InstallDate	Datum, wann der Task erstellt wurde
InteractWithDesktop	Task wird interaktiv ausgeführt (True) oder nicht (False)
JobID	Kennung des Tasks
JobStatus	Ergebnis (Success oder Failure) der letzten Task-Ausführung
Name	Name des Tasks
Notify	Benutzer, der über die erfolgreiche oder fehlerhafte Task-Ausführung benachrichtigt werden soll
Owner	Benutzer, der den Task erstellt hat
Priority	Priorität des Tasks
RunRepeatedly	Task wird regelmäßig ausgeführt (True) oder nicht (False)
StartTime	Uhrzeit, wann der Task das nächste Mal gestartet wird. Die Angabe erfolgt im Format YYYYMMDDHHMMSS.MMMMMM±UUU, wobei die ersten 8 Zeichen durch ein * ersetzt werden (z.B. 08:00 = *****080000.000000+120)
Status	Status des Tasks
TimeSubmitted	Uhrzeit, wann der Task erstellt wurde
UntilTime	Maximale Ausführungszeit des Task

Tabelle 84: Die Task-Eigenschaften und ihre Beschreibung

Beim Auslesen der Task-Eigenschaften werden die Tage, an denen der Task jeweils ausgeführt wird, durch folgende Werte repräsentiert:

Tag	Wert
1	1
2	2
3	4
4	8
5	16
6	32
7	64
8	128
9	256
10	512
11	1024
12	2048
13	4096
14	8192
15	16384
16	32768
17	65536
18	131072
19	262144
20	524288
21	1048576
22	2097152
23	4194304
24	8388608
25	16777216
26	33554432
27	67108864
28	134217728
29	268435456
30	536870912
31	1073741824

Tabelle 85: Die Werte der Tage

Auch die einzelnen Wochentage werden jeweils durch einen Wert dargestellt. Welcher Wochentag durch welchen Wert repräsentiert wird, können Sie der nachfolgenden Tabelle entnehmen:

Wochentag	Wert
Montag	1
Dienstag	2
Mittwoch	4
Donnerstag	8
Freitag	16
Samstag	32
Sonntag	64

Tabelle 86: Die Werte der Wochentage

Um nun das Vorhandensein eines AT-Tasks zu überprüfen, der die Befehlszeile `C:\Windows\System32\Defrag.exe C:` ausführt, stellen Sie zunächst eine Verbindung zum WMI-Service her und setzen den Impersonation-Level auf `Impersonate`. Anschließend führen Sie die `ExecQuery`-Abfrage gegen die `Win32_ScheduledJob`-Klasse aus, woraufhin die eingerichteten Tasks als Kollektion zurückgeliefert werden. Mit einer `For Each`-Schleife prüfen Sie dann für jeden einzelnen Task, ob die Eigenschaft `Command` den Wert `C:\Windows\System32\Defrag.exe C:` besitzt. Ist dies der Fall, so wird in dem folgenden Beispiel die Meldung `Task existiert ...` ausgegeben, und die Schleife wird mit `Exit For` vorzeitig verlassen:

Option Explicit

```
Dim oWMI
Dim strComputer
Dim strCommand
Dim colTasks
Dim strTask
```

```
strComputer = "."
strCommand = "C:\Windows\System32\Defrag.exe C:"
```

```
Set oWMI=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer)
Set colTasks = oWMI.ExecQuery ("SELECT * FROM _
    Win32_ScheduledJob")
```

```
For Each strTask in colTasks
    If strTask.Command = strCommand Then
```

Listing 621: 2_08_001a.vbs


```
WScript.Echo "Task existiert ..."  
Exit For  
End if  
Next  
  
Set oWMI=Nothing
```

Listing 621: 2_08_001a.vbs (Forts.)

Wenn Sie dagegen die Tasks ausgeben möchten, die unter grafischer Oberfläche mit dem Assistenten zur Einrichtung geplanter Tasks oder in der Eingabeaufforderung mit Shtasks erstellt wurden, so bleibt nur der Einsatz der `Run`-Methode des `Shell`-Objektes und die Ausführung des `Shtasks`-Befehls der Eingabeaufforderung, da es hierfür keine entsprechende API gibt. Führen Sie also mit Hilfe der `Run`-Methode des `Shell`-Objektes die Batch-Datei Listing 620: 2_08_001b.bat auf Seite 737 aus, die im folgenden Code-Beispiel als *Findjob.bat* angegeben ist.

```
Option Explicit  
Dim oShell  
  
Set oShell = CreateObject ("WScript.Shell")  
oShell.Run "%comspec% /c FindJob.bat"  
  
Set oShell = Nothing
```

Listing 622: 2_08_001b.vbs

183 Geplante Tasks einrichten

Wenn Sie unter grafischer Oberfläche die Ausführung bestimmter Programme (z.B. Wartungsaufgaben, Skripte, Batch-Dateien ...) automatisieren möchten, dann verwenden Sie hierfür den Windows Server 2003-eigenen Taskplaner und richten entsprechende Tasks ein.

Um eine neue Aufgabe zu automatisieren, stellt das Betriebssystem den Assistenten zur Einrichtung geplanter Tasks (Systemsteuerungsobjekt **GEPLANTE TASKS**) bereit, der folgende Angaben für den neuen Task erwartet:

- ▶ **Programmname:** Angabe des Programms, das mit Hilfe des Taskplaners automatisiert werden soll.
- ▶ **Name:** Die Bezeichnung für den neuen Task, die später auch in der Liste der eingerichteten Tasks angezeigt wird
- ▶ **Zeitplan:** Die Angabe darüber, wann der Task ausgeführt werden soll. Dies kann ein Systemereignis (*Systemstart*, *Anmeldung*, *Leerlaufzeit*) sein, aber auch ein zeitlicher Rhythmus (*Täglich*, *Wöchentlich*, *Monatlich*, *Einmalig* ...). Bei Auswahl

eines zeitlichen Rhythmus können der Tag und auch die Uhrzeit exakt spezifiziert werden, wobei auch mehrere Zeitpläne mit unterschiedlichen Tag- und Zeitangaben festgelegt werden können. Des Weiteren ist das Start-Datum (und optional das Ende-Datum) anzugeben, ab wann der Task das erste Mal ausgeführt werden soll

- **Benutzername:** Benutzer und Passwort, unter dessen Anmeldung der Task später ausgeführt wird

Um beispielsweise die Aufgabe der Datenträger-Defragmentierung zu automatisieren, starten Sie zunächst das Systemsteuerungsobjekt GEPLANTE TASKS und doppelklicken auf NEUEN TASK HINZUFÜGEN. Damit wird der Assistent zur Einrichtung neuer Tasks gestartet, und Sie müssen im ersten Stepp die auszuführende Anwendung festlegen. Taucht die gewünschte Anwendung nicht in der Liste der Standardanwendungen auf, was bei der Defragmentierung der Fall ist, so klicken Sie auf DURCHSUCHEN und wählen die gewünschte Anwendung von Hand aus. In diesem Beispiel wäre das die Datei *Defrag.exe*, die im Pfad `C:\Windows\System32` abgelegt ist.

Im nächsten Schritt geben Sie eine aussagekräftige Bezeichnung für den neuen Task ein, zum Beispiel *Defragmentierung* oder Ähnliches. Im gleichen Schritt legen Sie auch fest, wann dieser Task ausgeführt werden soll. Wählen Sie entweder ein Systemereignis (*Systemstart*, *Anmeldung*) oder einen zeitlichen Rhythmus (*Täglich*, *Wöchentlich*, *Monatlich*) aus.

Die Defragmentierung sollte regelmäßig ausgeführt werden, legen Sie also *wöchentlich* oder *monatlich* fest, und bestimmen Sie im nächsten Schritt den Tag und auch die genaue Uhrzeit. Anschließend hinterlegen Sie den Benutzernamen und auch das dazugehörige Passwort, mit Hilfe dessen der Task später ausgeführt werden soll.

Achtung

Der Benutzer muss natürlich über ausreichende Berechtigungen verfügen, um diesen Task auch ausführen zu können. Es reicht nicht aus, auf dem Server einen neuen Benutzer anzulegen und ein Passwort zu bestimmen. Denn Benutzer mit eingeschränkten Berechtigungen dürfen keine Wartungsaufgaben ausführen, weshalb es zu Problemen bei der Task-Ausführung kommen wird. Erstellen Sie am besten eine Art Wartungsaccount, der über ausreichende Rechte verfügt, und berücksichtigen Sie die regelmäßige Kennwortänderung.

Haben Sie den Benutzer und auch das Passwort hinterlegt, so müssen Sie im letzten Schritt die Option ERWEITERTE EIGENSCHAFTEN FÜR DIESEN TASK BEIM KLICKEN AUF »FERTIG STELLEN« ÖFFNEN aktivieren, woraufhin die Konfigurationseinstellungen für den Task geöffnet werden. Denn auf der Registerkarte TASK müssen Sie dem auszuführenden Befehl `C:\Windows\System32\Defrag.exe` noch die erforderlichen Parameter übergeben, damit es später nicht zu einer Unterbrechung oder gar zu einem Abbruch der Befehlsausführung kommt. Soll z.B. das Laufwerk C defragmentiert werden, so muss die vollständige Befehlszeile

```
C:\Windows\System32\Defrag.exe C:
```

lauten. Wenn Sie diese Parameter nicht angeben, würde der Task zwar gestartet werden, die Defragmentierung selbst aber nicht, da das zu defragmentierende Laufwerk nicht angegeben wurde. Das Defragmentierungsprogramm würde sonst nur die Hilfe zum Befehl ausgeben.

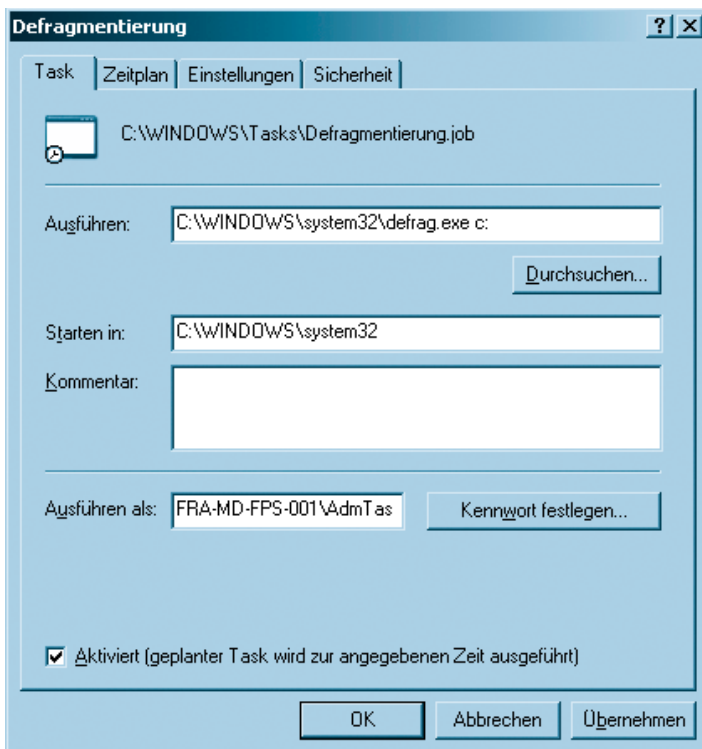


Abbildung 259: Die Parameter für die Defragmentierung hinterlegen

Auf der Registerkarte EINSTELLUNGEN können Sie weitere Optionen zur Task-Ausführung setzen. Beispielsweise ob der Task nach einer gewissen Dauer von alleine beendet bzw. unterbrochen werden soll; oder ob der Task bereits in gewissen Leerlaufzeiten gestartet werden kann.

Des Weiteren können Sie auf der Registerkarte SICHERHEIT die Benutzer und Gruppen definieren, die diesen Task ausführen dürfen. Per Default haben alle Administratoren, wie auch das System selbst, die Berechtigung zur Ausführung dieses Tasks. Haben Sie aber einen speziellen Benutzer erstellt, unter dessen Anmeldung der Task ausgeführt werden soll, der jedoch kein Mitglied der Gruppe *Administratoren* ist, so müssen Sie diesen Benutzer für die Ausführung des Tasks noch entsprechend berechtigen.

Tipp

Wenn Sie einen neuen Task eingerichtet haben, dann sollten Sie den neuen Task entsprechend testen und dessen Funktionstüchtigkeit sicherstellen. Sie müssen dabei nicht auf das festgelegte Start-Datum warten, da Sie jeden Task auch von Hand ausführen können. Wählen Sie im Kontextmenü des entsprechenden Tasks den Eintrag AUSFÜHREN aus, der Task wird daraufhin manuell gestartet und im Hintergrund ausgeführt.

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung lassen sich ebenfalls neue Tasks erstellen, wobei es natürlich auch hier wieder zwischen AT-Tasks und Schtasks zu unterscheiden gilt. Um beispielsweise einen neuen AT-Task einzurichten, verwenden Sie den Befehl `At.exe` und übergeben dabei folgende Parameter:

- ▶ **Zeit:** Der Ausführungszeitpunkt des neuen Tasks, der im Format *HH:MM* angegeben wird
- ▶ **Interaktiv (/interactive):** Legt fest, dass der Task interaktiv ausgeführt werden soll
- ▶ **Zeitplan (/Every):** Legt den Tag der Woche oder den Tag des Monats fest, wann der Task ausgeführt werden soll
- ▶ **Datum (/Next):** Legt den nächsten Ausführungszeitpunkt fest
- ▶ **Programm:** Das auszuführende Programm oder eine Befehlszeile, die ausgeführt werden soll

Das nachfolgende Beispiel erstellt einen neuen AT-Task, der jeden Samstag um 21:00 Uhr die Defragmentierung für das Laufwerk C startet.

```
@Echo off
At 21:00 /Every:Sa C:\Windows\System32\Defrag.exe C:
```

Listing 623: 2_08_002a.bat

Möchten Sie dagegen einen Task vom Typ Schtasks erstellen, so verwenden Sie hierfür den Befehl `Schtasks /Create`. Der Befehl `Schtasks /Create` erwartet dabei zumindest die folgenden Angaben, die Sie dem Befehl als Parameter übergeben müssen:

- ▶ **Name (/tn):** Die Bezeichnung des geplanten Tasks, der erstellt werden soll. Die Bezeichnung darf die maximale Länge von 238 Zeichen nicht überschreiten.
- ▶ **Programm (/tr):** Das Programm oder die Befehlszeile, die durch den geplanten Task automatisiert werden soll. Der Programmpfad bzw. die Befehlszeile darf die maximale Länge von 262 Zeichen nicht überschreiten.
- ▶ **Zeitplan (/sc):** Das Systemereignis oder der zeitliche Rhythmus, wann der geplante Task ausgeführt werden soll. Gültige sind *MINUTE*, *STÜNDLICH*, *TÄGLICH*, *WÖCHENTLICH*, *MONATLICH*, *EINMAL*, *BEIMSTART*, *BEIANMELDUNG* oder *BEILEERLAUF*.

Ohne Angabe weiterer Parameter würde `Schtasks /Create` nach Eingabe des Kennwortes versuchen, den neuen geplanten Task unter Verwendung des Benutzerkontos anlegen, der gerade angemeldet ist. Falls der Benutzer aber nicht Mitglied der Gruppen *Administratoren*, *Sicherungs-Operatoren* oder *Server-Operatoren* ist, so fehlt ihm die ausreichende Berechtigung, und das Erstellen des Tasks bricht ab. Sie sollten dem Befehl also noch mit dem Parameter `/ru` den Benutzernamen und mit `/rp` das dazugehörige Passwort übergeben, damit das Erstellen den neuen Tasks ordnungsgemäß ausgeführt werden kann.

Die folgenden Befehlszeilen erstellen einen neuen geplanten Task mit der Bezeichnung *Defragmentierung*, der die Befehlszeile `C:\Windows\System32\Defrag.exe C:` ausführt. Der Task wird jeden *Samstag* um *21:00 Uhr* ausgeführt. Für das Erstellen des Tasks wird das Benutzerkonto `AdmTask` verwendet, das dazugehörige Passwort wird ebenfalls übergeben. Und für den Fall, dass der Task schon existieren sollte, wird das Überschreiben des vorhandenen Tasks mit dem Parameter `/f` erzwungen.

```
@Echo off
Schtasks /Create /tn Defragmentierung _
/tr "C:\Windows\System32\Defrag.exe C:" /sc WÖCHENTLICH _
/d SA /ST 21:00 /ru FRA-MD-FPS-001\AdmTask _
/rp 12345678 /f
```

Listing 624: 2_08_002b.bat

Mit dem `Schtasks /Create`-Befehl können Sie natürlich auch auf einem Remote-System einen Task erstellen. Hierzu erweitern Sie die Befehlszeile um die Parameter `/s` (IP-Adresse oder Hostnamen), `/u` (Benutzername auf dem Remote-System) und `/p` (dazugehöriges Passwort). Die Befehlszeile würde dann folgendermaßen lauten:

```
@Echo off
Schtasks /Create /tn Defragmentierung _
/tr "C:\Windows\System32\Defrag.exe C:" /s FRA-MD-FPS-005_
/u FRA-MD-FPS-005\AdmTask /p 12345678 /sc WÖCHENTLICH _
/d SA /ST 21:00 /ru FRA-MD-FPS-001\AdmTask _
/rp 12345678 /f
```

Listing 625: 2_08_002c.bat

Die Lösung mit VBScript

Auch mit VBScript ist das Erstellen neuer Tasks kein Problem, wobei Sie hier natürlich wieder zwischen den beiden Task-Typen (AT-Task und `Schtasks`) unterscheiden müssen. Möchten Sie aus einem VBScript heraus einen neuen AT-Task erstellen, so stellen Sie mit Hilfe der `GetObject`-Methode eine Verbindung zum WMI-Service her, führen anschließend die `Create`-Methode aus und übergeben dabei folgende Parameter:

- ▶ **Programm:** Das Programm, das durch den Task automatisiert werden soll
- ▶ **Startzeit:** Die Uhrzeit, zu der die Task-Ausführung stattfinden soll. Die Angabe der Uhrzeit erfolgt hierbei im UTC-Format (YYYYMMDDHHMMSS.MMMMMM(+/-)000), wobei die ersten acht Zeichen stets durch ein * ersetzt werden, da keine Datumsangabe erlaubt ist. Die Uhrzeit 06:30 Uhr wird also in *****063000.000000+120 dargestellt.
- ▶ **Wiederholen:** Legt fest, ob der Task immer an den angegebenen Tagen ausgeführt werden soll (True) oder nur einmal (False)
- ▶ **Wochentag:** Legt den Tag der Woche fest, an dem der Task ausgeführt werden soll (siehe Tabelle 86: Die Werte der Wochentage auf Seite 740)
- ▶ **Tag:** Legt den Tag des Monats fest, an dem der Task ausgeführt werden soll (siehe Tabelle 85: Die Werte der Tage auf Seite 739)
- ▶ **Interaktiv:** Legt fest, ob der Task interaktiv gestartet werden soll
- ▶ **ID:** Gibt die vergebene Task-ID zurück

Das nachfolgende Beispiel legt einen neuen Task an, der jeden Samstag um 21:00 Uhr die Befehlszeile `C:\Windows\System32\Defrag.exe C:` ausführt: Zunächst werden die erforderlichen Variablen definiert, damit das Skript jederzeit ohne großen Aufwand auch neuen Anforderungen angepasst werden kann und auch etwas überschaubarer bleibt. Dann wird mit der `GetObject`-Methode eine Verbindung zum WMI-Service und zur WMI-Klasse `Win32_ScheduledJob` hergestellt und der `Impersonation-Level` auf `Impersonate` gesetzt. Anschließend wird der neue Task mit Hilfe der `Create`-Methode angelegt, wobei das auszuführende Programm (`strCommand`), die Startzeit (`strTime`), die regelmäßige Wiederholung (`Repeat`), der Wochentag (`intDay`) und die Variable der zurückgelieferten Task-ID (`intID`) übergeben werden.

Option Explicit

```
Dim oWMI
Dim oTask
Dim strComputer
Dim strTask
Dim strCommand
Dim strTime
Dim intDay
Dim intID

Const Repeat = True
strComputer = "."
strCommand = "C:\Windows\System32\Defrag.exe C:"
strTime = "*****210000.000000+120"
intDay = "32"
```

Listing 626: 2_08_002a.vbs

```

Set oWMI=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\" & strComputer)
Set oTask=oWMI.Get("Win32_ScheduledJob")
strTask = oTask.Create (strCommand, strTime, Repeat, _
    intDay, , , intID)

If Err = 0 Then
    Wscript.Echo "Der Task wurde erfolgreich erstellt"
Else
    Wscript.Echo "Fehler beim Erstellen des Tasks"
End If

Set oWMI=Nothing
Set oTask=Nothing

```

Listing 626: 2_08_002a.vbs (Forts.)

Möchten Sie stattdessen einen Task erzeugen, der zu einem späteren Zeitpunkt mit dem Schtasks-Befehl oder auch unter grafischer Oberfläche bearbeitet werden kann, so rufen Sie mit Hilfe der Run-Methode des Shell-Objektes den Schtasks /Create-Befehl der Eingabeaufforderung auf.

```

Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c NewTask.bat"

Set oShell = Nothing

```

Listing 627: 2_08_002b.vbs

Das obige Skript-Beispiel führt eine Batch-Datei mit der Bezeichnung *NewTask.bat* aus, die sich im aktuellen Ordner befindet, und folgende Befehlszeilen enthält:

```

@Echo off
Schtasks /Create /tn Defragmentierung _
    /tr "C:\Windows\System32\Defrag.exe C:" /sc WÖCHENTLICH _
    /d SA /ST 21:00 /ru FRA-MD-FPS-001\AdmTask /rp 12345678 /f

```

Listing 628: NewTask.bat

184 Geplante Tasks bearbeiten

Eingerichtete Tasks können Sie natürlich jederzeit wieder ändern. Und Gründe dafür gibt es genügend: Sei es, dass Sie andere Parameter für das betreffende Programm verwenden möchten, die Berechtigungen anpassen müssen oder einfach nur andere Zeitpläne für den Start des Tasks verwenden wollen. Auch wenn Sie einen bestimmten Task zu unterschiedlichen Zeiten ausführen lassen möchten, brauchen Sie den betreffenden Task nicht mehrfach einzurichten, sondern nur den Task zu bearbeiten. Denn in den Konfigurationseinstellungen eines jeden Tasks können Sie mehrere, sprich: unterschiedliche, Zeitpläne hinterlegen.

Um einen Task unter grafischer Oberfläche zu bearbeiten, rufen Sie zunächst das Systemsteuerungsobjekt GEPLANTE TASKS auf, woraufhin Sie in die Übersicht der bereits eingerichteten Tasks gelangen. Anschließend markieren Sie den zu bearbeitenden Task und führen entweder einen Doppelklick auf den Task aus oder wählen im Kontextmenü den Eintrag BEARBEITEN aus. Und je nachdem, welche Konfigurationseinstellungen Sie ändern möchten, wechseln Sie auf die entsprechende Registerkarte, nehmen die Einstellungen vor und speichern die Änderungen mit einem Klick auf ÜBERNEHMEN ab.

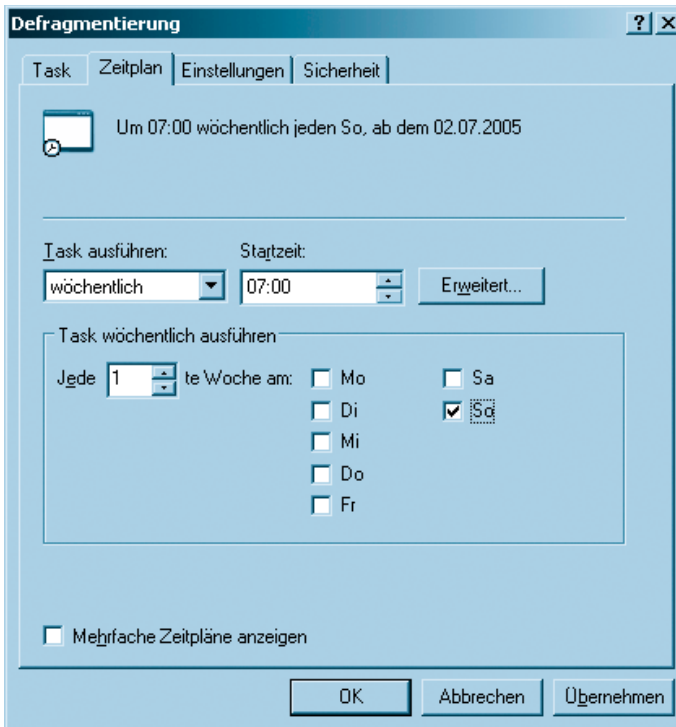


Abbildung 260: Mehrere Zeitpläne für einen Task einrichten

Für den Fall, dass Sie – wie im Rezept Geplante Tasks einrichten auf Seite 741 beschrieben – einen Task zur Defragmentierung eingerichtet haben und nun die Ausführungszeit von Samstag, 21:00 Uhr auf Sonntag, 07:00 Uhr ändern möchten, so brauchen Sie lediglich den Zeitplan anzupassen: Starten Sie also das Systemsteuerungsobjekt GEPLANTE TASKS, und öffnen Sie die Konfigurationseinstellungen des entsprechenden Tasks. Wechseln Sie auf die Registerkarte ZEITPLAN, und legen Sie den neuen Tag wie auch die neue Uhrzeit fest.

Die Lösung per Eingabeaufforderung

In der Eingabeaufforderung gestaltet sich das Ändern eines vorhandenen Tasks leider nicht ganz so einfach wie unter grafischer Oberfläche. Wenn Sie z.B. einen vorhandenen AT-Task ändern möchten, so müssen Sie den betreffenden Task zunächst löschen und anschließend unter Angabe der neuen Parameter wieder neu erstellen. Vorhandene Tasks, die mit `Schtasks` erstellt wurden, können Sie dagegen ändern, wobei es aber auch hier einige Einschränkungen gibt. Dazu aber gleich mehr.

Einmal angenommen, Sie haben einen AT-Task eingerichtet, der jeden Samstag um 21:00 Uhr eine Defragmentierung des Laufwerks C durchführt. Nun möchten Sie aber, dass der AT-Task nicht mehr samstags um 21:00 Uhr gestartet wird, sondern jeweils sonntags um 06:00 Uhr. Hierzu ermitteln Sie zunächst die ID des AT-Tasks, indem Sie `At.exe` eingeben, sich die Tasks anzeigen lassen und die betreffende ID notieren. Anschließend löschen Sie den Task mit `At ID /Delete` und erstellen einen neuen Task unter Angabe des neuen Zeitplans:

```
@Echo off
AT 10 /Delete
AT 06:00 /Every:So C:\Windows\System32\Defrag.exe C:
```

Listing 629: 02_08_003a.bat

Möchten Sie dagegen einen Task ändern, der mit `Schtasks` erstellt wurde, so können Sie mit Hilfe des `Schtasks /Change`-Befehls die folgenden Task-Eigenschaften ändern:

- ▶ Benutzer (`/ru /rp`): Benutzername, unter dessen Verwendung der Task ausgeführt wird
- ▶ Programm (`/tr`): Das Programm oder die Befehlszeile, die durch den Task ausgeführt werden soll
- ▶ Start-Zeit (`/st`): Die Uhrzeit, wann der Task ausgeführt werden soll
- ▶ Intervall (`/ri`): Das Intervall zur Wiederholung des Tasks
- ▶ Ende-Zeit (`/et`): Die Uhrzeit, wann der Task automatisch beendet werden soll
- ▶ Dauer (`/du`): Die maximale Dauer der Task-Ausführung

- ▶ Start-Datum (/sd): Das Datum, an dem der Task das erste Mal ausgeführt werden soll
- ▶ Ende-Datum (/ed): Das Datum, an dem Task das letzte Mal ausgeführt werden soll
- ▶ Aktivieren (/enable): Aktiviert den Task
- ▶ Deaktivieren (/disable): Deaktiviert den Task
- ▶ Interaktiv (/it): Legt fest, dass der Task interaktiv gestartet werden soll.
- ▶ Automatisch löschen (/z): Legt fest, dass der Task nach Erreichen des Endes-Datums automatisch gelöscht werden soll.

Angenommen, Sie haben einen Task eingerichtet, der jeden Samstag um 21:00 Uhr die Defragmentierung des Laufwerks C durchführt. Nun möchten Sie, dass der Task nicht mehr um 21:00 Uhr, sondern um 06:00 des gleichen Tages gestartet wird. Diese Änderung lässt sich mit folgender Befehlszeile bewerkstelligen:

```
@Echo off
Schtasks /Change /tn Defragmentierung /st 06:00 _
/ru FRA-MD-FPS-001\AdmTask /rp 12345678
```

Listing 630: 2_08_003b.bat

Wenn Sie die Ausführung des Tasks dagegen von Samstag auf Sonntag verlegen möchten, so löschen Sie den Task und erstellen ihn neu, da Sie hinsichtlich des Zeitplans lediglich das Start-/Ende-Datum, die Start-/Ende-Zeit, das Intervall, die Dauer und auch die Wiederholung bestimmen können, nicht aber den eigentlichen Zeitplan selbst:

```
@Echo off
Schtasks /Delete /tn Defragmentierung /f
Schtasks /Create /tn Defragmentierung _
/tr "C:\Windows\System32\Defrag.exe C:" /sc WÖCHENTLICH _
/d SO /ST 06:00 /ru FRA-MD-FPS-001\AdmTask _
/rp 12345678 /f
```

Listing 631: 2_08_003c.bat

Die Lösung mit VBScript

Um einen vorhandenen AT-Task per VBScript zu ändern, müssen Sie den Task zunächst löschen und anschließend neu erstellen, da die WMI-Klasse `Win32_ScheduledJob` keine entsprechende Methode bereitstellt, womit Sie einen vorhandenen Task ändern können. Wenn Sie einen Task, der jeden Samstag um 21:00 Uhr gestartet wird, auf Sonntag 06:00 Uhr verlegen möchten, dann ermitteln zunächst die ID des Tasks, setzen die `Delete`-Methode ein, um den Task mit der betreffenden ID zu löschen, und erstellen den Task dann unter Angabe der neuen Task-Eigenschaften mit der `Create`-Methode neu:

Option Explicit

```
Dim oWMI
Dim oTask
Dim strComputer
Dim strTask
Dim strCommand
Dim strTime
Dim intDay
Dim intID

Const Repeat = True
strComputer = "."
strCommand = "C:\Windows\System32\Defrag.exe C:"
strTime = "*****060000.000000+120"
intDay = "64"
intID = "1"

Set oWMI=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer)
Set oTask = oWMI.Get("Win32_ScheduledJob.JobID=" & intID)
oTask.Delete
strTask = oTask.Create (strCommand, strTime, Repeat, _
    intDay, , , intID)

Set oWMI=Nothing
Set oTask=Nothing
```

Listing 632: 2_08_003a.vbs

Wenn Sie dagegen einen Task ändern möchten, der mit `Schtasks /Create` oder unter grafischer Oberfläche mit dem Assistenten zur Einrichtung geplanter Tasks erstellt wurde, so können Sie den `Schtasks`-Befehl der Eingabeaufforderung einsetzen, den Sie mit der `Run`-Methode des `Shell`-Objektes auch aus einem VBScript heraus aufrufen können.

Der `Schtasks /Change`-Befehl erlaubt zwar das Ändern vieler Task-Eigenschaften, nur leider lässt sich damit nicht der eigentliche Zeitplan ändern. Das heißt, wenn Sie die Task-Ausführung von Samstag, 21:00 Uhr, auf Sonntag, 06:00 Uhr, verlegen möchten, so müssen Sie auch hier wieder einen kleinen Umweg gehen, indem Sie den betreffenden Task zunächst löschen und anschließend unter Angabe des neuen Zeitplans neu erstellen:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c ChangeJob.bat"

Set oShell = Nothing
```

Listing 633: 2_08_003b.vbs

Das obige Skript-Beispiel startet eine Batch-Datei namens *ChangeJob.bat*, worüber der Task mit der Bezeichnung *Defragmentierung* mit `Schtasks /Delete` gelöscht und anschließend per `Schtasks /Create` mit den neuen Task-Eigenschaften neu erstellt wird:

```
@Echo off
Schtasks /Delete /tn Defragmentierung /f
Schtasks /Create /tn Defragmentierung _
    /tr "C:\Windows\System32\Defrag.exe C:" /sc WÖCHENTLICH _
    /d SO /ST 06:00 /ru FRA-MD-FPS-001\AdmTask _
    /rp 12345678 /f
```

Listing 634: ChangeJob.bat

185 Geplante Tasks vorübergehend deaktivieren

Für den Fall, dass Sie einen geplanten Task eingerichtet haben, diesen aber für eine gewisse Zeit nicht brauchen und daher auch nicht ausführen lassen möchten, so brauchen Sie diesen Task nicht zu löschen. Denn Sie können einzelne Tasks vorübergehend deaktivieren und die entsprechende Programmausführung damit für unbestimmte Zeit außer Kraft setzen.

Um einen Task zu deaktivieren, wechseln Sie zunächst in die Übersicht der geplanten Tasks. Öffnen Sie hierzu das Systemsteuerungsobjekt **GEPLANTE TASKS**, und markieren Sie den Task, den Sie deaktivieren möchten. Rufen Sie aus dem Kontextmenü des Tasks die Eigenschaften auf, und deaktivieren Sie auf der Registerkarte **TASK** die Option **AKTIVIERT (GEPLANTER TASK WIRD ZUR ANGEgebenEN ZEIT AUSGEFÜHRT)**. Klicken Sie anschließend auf **ÜBERNEHMEN**, um die Änderung zu speichern. Das Fenster wird daraufhin geschlossen, und der Task wird nicht mehr ausgeführt, bis Sie ihn wieder aktivieren (siehe Abbildung 261).

Wenn Sie den Task zu einem späteren Zeitpunkt wieder aktivieren und damit automatisch ausführen lassen möchten, so brauchen Sie lediglich die Option **AKTIVIERT (GEPLANTER TASK WIRD ZUR ANGEgebenEN ZEIT AUSGEFÜHRT)** zu aktivieren.

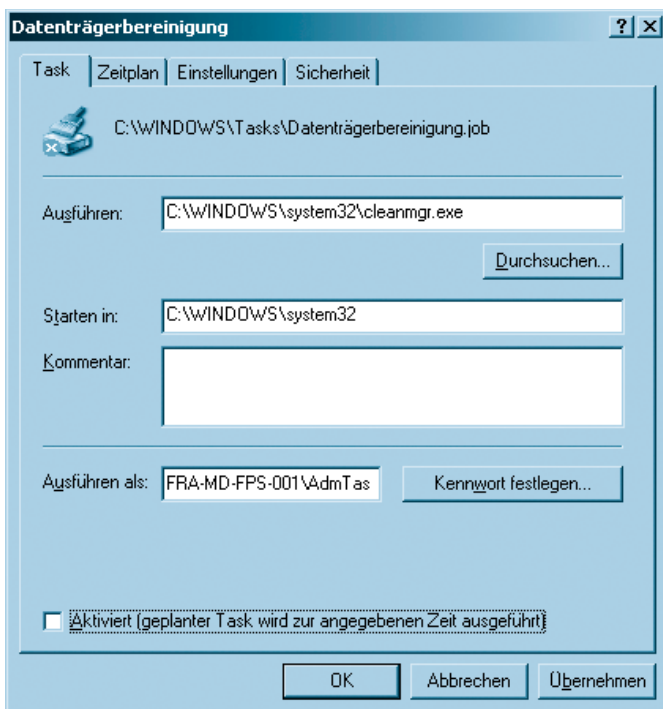


Abbildung 261: Einen Task vorübergehend deaktivieren

Die Lösung per Eingabeaufforderung

Um in der Eingabeaufforderung einen vorhandenen AT-Task vorübergehend zu deaktivieren, müssen Sie etwas improvisieren, da `At.exe` leider keine Möglichkeiten bietet, womit Sie einen Task deaktivieren und anschließend wieder aktivieren können. Die einfachste Möglichkeit ist, das Suffix der eigentlichen Task-Datei umzubenennen (z.B. `At1.job` nach `At1.old`), womit der Task vom Betriebssystem nicht mehr als ein solcher erkannt und demzufolge auch nicht mehr ausgeführt wird.

Eine andere Möglichkeit wäre, den Task einfach zu löschen. Dies hat aber den Nachteil, dass Sie den Task neu einrichten müssen, wenn Sie ihn wieder benötigen. Des Weiteren könnten Sie den Dienst *Taskplaner* deaktivieren, womit der AT-Task ebenfalls nicht mehr ausgeführt wird. Letzteres hat aber den Nachteil, dass sich das Deaktivieren auf alle Tasks auswirken würde, was in vielen Fällen aber nicht erwünscht ist.

Wenn Sie nun einen einzelnen AT-Task vorübergehend deaktivieren möchten, so benennen Sie die Task-Datei einfach um. Hierzu benötigen Sie die ID des Tasks, die Sie sich in der Eingabeaufforderung durch Eingeben des Befehls `At` (ohne weitere Parameter) anzeigen lassen können. Anschließend benennen Sie die Datei `C:\Windows\Tasks\ATID.job` mit dem `Ren`-Befehl nach `C:\Windows\Tasks\ATID.old` um, wobei Sie die Zeichenfolge ID durch die jeweilige Task-ID ersetzen:

```
@Echo off
Ren C:\Windows\Tasks\At1.job At1.old
```

Listing 635: 2_08_004a.bat

Wenn Sie den Task zu einem späteren Zeit wieder aktivieren möchten, so benennen Sie die Datei `C:\Windows\Tasks\At1.old` wieder in eine gültige Task-Datei um (z.B. `At1.job`), wobei Sie vorher jedoch das Vorhandensein der Task-Datei überprüfen müssen. Denn wenn zwischenzeitlich ein neuer Task eingerichtet wurde, so ist die ID des deaktivierten Tasks bereits für den neuen Task vergeben, und Sie müssen eine neue Task-ID verwenden.

Welche Task-ID verwendet werden kann, können Sie der Registrierung entnehmen, indem Sie den DWORD-Wert `NextATJobID` des Schlüssels `HKLM\SYSTEM\CURRENT-CONTROLSET\SERVICES\SCHEDULE` abfragen.

```
@Echo off
If not exist C:\Windows\Tasks\At1.job Goto Rename
Goto Ende

:Rename
Ren C:\Windows\Tasks\At1.old At1.job
Goto Ende

:Ende
```

Listing 636: 2_08_004b.bat

Wenn Sie dagegen einen Task vorübergehend deaktivieren möchten, der mit dem Assistenten zur Einrichtung geplanter Tasks unter grafischer Oberfläche oder aber mit Schtasks erstellt wurde, so haben Sie es in diesem Fall etwas leichter, da der Schtasks-Befehl entsprechende Möglichkeiten bietet, einen Task zu deaktivieren und auch wieder zu aktivieren. Um einen Task vorübergehend zu deaktivieren, verwenden Sie den Schtasks `/Change`-Befehl, dem Sie den Namen des Tasks, die Aktion `/Disable` wie auch den Benutzernamen und das Passwort als Parameter übergeben:

```
@Echo off
Schtasks /Change /tn Defragmentierung /Disable _
/ru FRA-MD-FPS-001\AdmTask /rp 12345678
```

Listing 637: 2_08_004c.bat

Um den zuvor deaktivierten Task wieder zu aktivieren, verwenden Sie die gleiche Befehlszeile und tauschen lediglich die Aktion `/Disable` durch `/Enable` aus:

```
@Echo off
Schtasks /Change /tn Defragmentierung /Enable _
/ru FRA-MD-FPS-001\AdmTask /rp 12345678
```

Listing 638: 2_08_004d.bat

Die Lösung mit VBScript

Um einen vorhandenen AT-Task aus einem VBScript heraus kurzzeitig zu deaktivieren, gehen Sie ähnlich vor wie in der Eingabeaufforderung. Sprich: Sie benennen die Task-Datei um und ändern dabei das Suffix der Datei von *.job auf *.old oder Ähnliches.

Stellen Sie mit Hilfe der `CreateObject`-Methode eine Verbindung zum `FileSystem`-Objekt her, und wenden Sie die `MoveFile`-Methode an, der Sie den Pfad und auch den Dateinamen der alten Datei (`strOldFile`) wie auch der neuen Datei (`strNewFile`) übergeben:

```
Dim oFS
Dim strOldFile
Dim strNewFile
Dim intID

intID = "1"
strOldFile = "C:\Windows\Tasks\At" & intID & ".job"
strNewFile = "C:\Windows\Tasks\At" & intID & ".old"

Set oFS = CreateObject ("Scripting.FileSystemObject")
oFS.MoveFile strOldFile, strNewFile

Set oFS = Nothing
```

Listing 639: 2_081_004a.vbs

Wenn Sie dagegen einen mit `Schtasks` oder unter grafischer Oberfläche eingerichteten Task kurzzeitig deaktivieren möchten, so setzen Sie den `Schtasks /Disable`-Befehl der Eingabeaufforderung ein, den Sie mit der `Run`-Methode des `Shell`-Objektes auch aus einem VBScript heraus aufrufen können:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Schtasks /Change _
```

Listing 640: 2_08_004b.vbs

```
/tn Defragmentierung /Disable _  
/ru FRA-MD-FPS-001\AdmTask /rp 12345678"
```

```
Set oShell = Nothing
```

Listing 640: 2_08_004b.vbs (Forts.)

186 Geplante Tasks starten und beenden

Für den Fall, dass Sie einen (oder auch mehrere) Tasks eingerichtet haben und diese nun zu Testzwecken außerhalb des Zeitplans von Hand starten möchten, so brauchen Sie lediglich in die Übersicht der geplanten Tasks (Systemsteuerungsobjekt GEPLANTE TASKS) zu wechseln, den betreffenden Task zu markieren und im Kontextmenü des Tasks AUSFÜHREN zu wählen. Der Task wird daraufhin sofort gestartet, und in der Spalte STATUS der Task-Übersicht wird GESTARTET angezeigt.

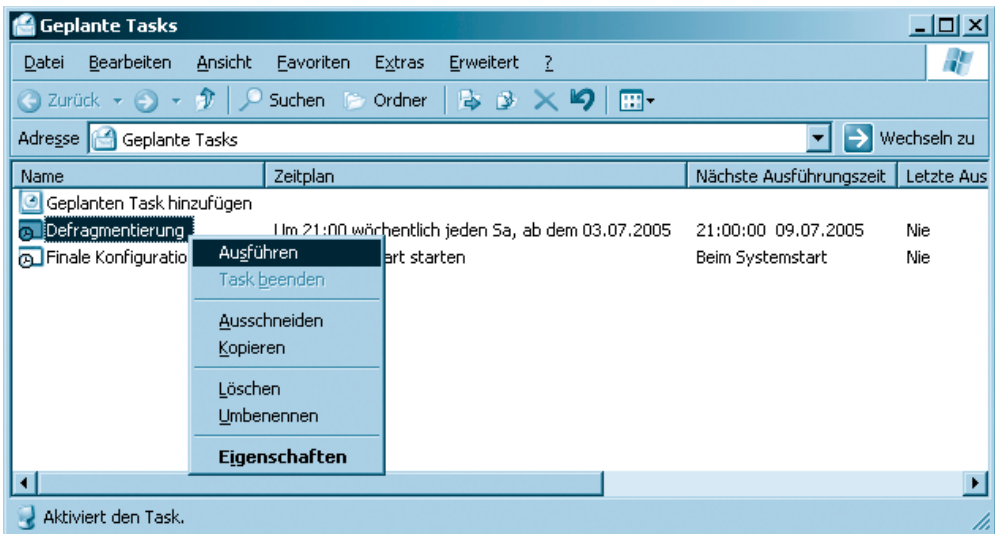


Abbildung 262: Einen Task manuell starten

Führt der manuell gestartete Task eine etwas größere Wartungsaufgabe aus, so müssen Sie nicht bis zum Ende der Taskausführung warten, denn Sie können den Task auch wieder von Hand beenden. Öffnen Sie hierzu das Kontextmenü des geplanten Tasks, den Sie gerade gestartet haben, und wählen Sie TASK BEENDEN aus. Die Ausführung des Tasks wird damit sofort gestoppt, und der Task wird dann zum festgelegten Zeitplan wieder von neuem gestartet.

Die Lösung per Eingabeaufforderung

Von der Eingabeaufforderung lassen sich leider nur solche Tasks starten, die entweder mit dem Assistenten zur Einrichtung geplanter Tasks unter grafischer Oberfläche erstellt worden sind oder aber mit Hilfe des `Schtasks /Create`-Befehls der Eingabeaufforderung. AT-Tasks dagegen können nur dann manuell gestartet werden, wenn Sie unter grafischer Oberfläche das Kontextmenü des Tasks öffnen und `AUSFÜHREN` auswählen. Ein direkter Aufruf des AT-Tasks von der Eingabeaufforderung aus ist leider nicht möglich.

Um einen Task von der Eingabeaufforderung aus manuell zu starten, verwenden Sie den `Schtasks /Run`-Befehl, dem Sie mit dem Parameter `/tn` lediglich die Bezeichnung des zu startenden Tasks übergeben müssen:

```
@Echo off
Schtasks /Run /tn Defragmentierung
```

Listing 641: 2_08_005a.bat

Auf ähnliche Art und Weise lässt sich auch ein Task auf einem Remote-System starten. Möchten Sie beispielsweise den Task *Defragmentierung* auf dem Server *FRA-MD-FPS-005* von der Eingabeaufforderung aus starten, so übergeben Sie vor der Task-Bezeichnung den Hostnamen oder die IP-Adresse des Remote-Systems (`/s`), den Benutzernamen (`/u`) wie auch das Passwort (`/p`), unter dessen Verwendung der Task ausgeführt werden soll:

```
@Echo off
Schtasks /Run /s FRA-MD-FPS-005 /u FRA-MD-FPS-005\AdmTask _
/p 12345678 /tn Defragmentierung
```

Listing 642: 2_08_005b.bat

Die Lösung mit VBScript

Wie bereits im vorherigen Abschnitt erwähnt, lassen sich einzelne AT-Tasks leider nicht manuell starten; weder von der Eingabeaufforderung aus noch per VBScript. Lediglich die mit dem `Schtasks /Create`-Befehl oder unter grafischer Oberfläche eingerichteten Tasks können aus einem VBScript heraus von Hand gestartet und wieder beendet werden.

Und da es für diesen Task-Typ leider keine entsprechende API gibt, womit Sie den Task per VBScript steuern können, bleibt Ihnen leider nur das Ausführen des `Schtasks /Run`-Befehls der Eingabeaufforderung, den Sie mit Hilfe der `Run`-Methode des `Shell`-Objektes auch aus einem VBScript heraus aufrufen können:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Schtasks /Run /tn Defragmentierung"

Set oShell = Nothing
```

Listing 643: 2_08_005a.vbs

Um den manuell gestarteten Task wieder zu beenden, verwenden Sie den gleichen Quellcode und tauschen lediglich den Parameter /Run durch /End aus:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Schtasks /End /tn Defragmentierung"

Set oShell = Nothing
```

Listing 644: 2_08_005b.vbs

187 Geplante Tasks löschen

Für den Fall, dass Sie einen Task eingerichtet haben, diesen aber nicht mehr benötigen, so können Sie den Task natürlich auch dauerhaft vom Computersystem verbannen, indem Sie den Task einfach löschen. Hierzu brauchen Sie nur die Übersicht der geplanten Tasks (Systemsteuerungsobjekt GEPLANTE TASKS) aufzurufen, den zu löschenden Task zu markieren und die **Entf**-Taste zu drücken. Oder Sie öffnen das Kontextmenü des Tasks und wählen LÖSCHEN aus. Das Betriebssystem fragt daraufhin noch einmal nach, ob Sie den Task auch wirklich löschen möchten. Klicken Sie auf JA, um den Task endgültig zu löschen (siehe Abbildung 263).

Die Lösung per Eingabeaufforderung

Um einen vorhandenen AT-Task von der Eingabeaufforderung aus zu löschen, führen Sie den Befehl `At` aus und übergeben dem Befehl die ID des zu löschenden Tasks (diese können Sie sich mit `At` ohne Angabe weiterer Parameter anzeigen lassen) wie auch den Parameter `/Delete`. Der Task mit der angegebenen ID wird daraufhin ohne Rückfrage vom Computersystem gelöscht:

```
@Echo off
At 1 /Delete
```

Listing 645: 2_08_006a.bat

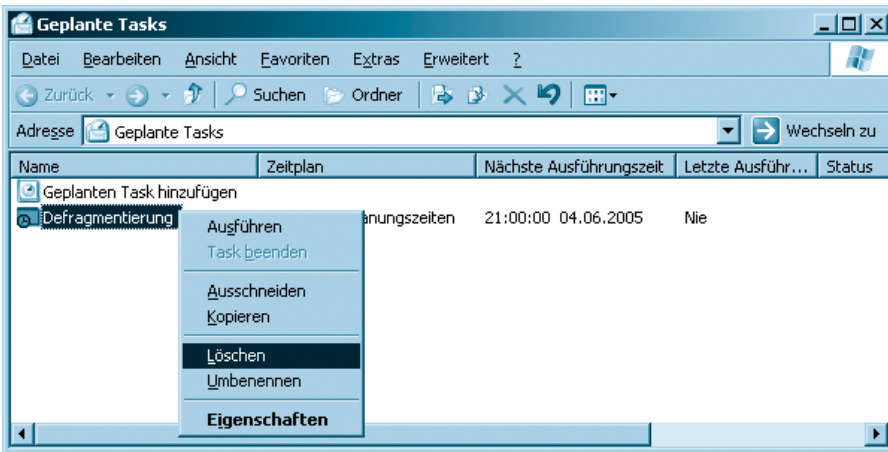


Abbildung 263: Einen vorhandenen Task löschen

Möchten Sie statt eines einzelnen AT-Tasks alle AT-Tasks löschen, so geben Sie keine ID an, sondern führen lediglich den Befehl `At /Delete` aus, wobei Sie in diesem Fall aber den Parameter `/Yes` anfügen sollten, um das Löschen aller AT-Tasks zu bestätigen (dies ist beim Löschen eines einzelnen AT-Tasks nicht erforderlich):

```
@Echo off
At /Delete /Yes
```

Listing 646: 2_08_006b.bat

Um dagegen einen unter grafischer Oberfläche oder mit `Schtasks` eingerichteten Task von der Eingabeaufforderung aus zu löschen, nutzen Sie den Befehl `Schtasks /Delete`, dem Sie zumindest den Parameter `/tn` mit der Bezeichnung des zu löschenden Task übergeben müssen. Alternativ können Sie mit dem Parameter `/tn *` auch das Löschen aller geplanten Tasks veranlassen. Der `Schtasks /Delete`-Befehl versucht daraufhin, den geplanten Task zu löschen, und möchte dies aber zuvor noch einmal manuell bestätigt haben. Diese Nachfrage können Sie vermeiden, in dem Sie mit dem Parameter `/f` das Löschen des geplanten Tasks erzwingen.

Die folgenden Befehlszeilen löschen den geplanten Task mit der Bezeichnung *Defragmentierung* vom lokalen Computersystem, wobei der Account des gerade angemeldeten Benutzers verwendet wird:

```
@Echo off
Schtasks /Delete /tn Defragmentierung /f
```

Listing 647: 2_08_006c.bat

Einen geplanten Task können Sie natürlich auch auf einem Remote-System löschen. Sie müssen lediglich die IP-Adresse oder den Hostnamen des Remote-Systems wie auch den zu verwendenden Benutzer und das dazugehörige Passwort angeben:

```
@Echo off
Schtasks /Delete /tn Defragmentierung /f /s FRA-MD-FPS-005 _
/u FRA-MD-FPS-005\AdmTask /p 12345678
```

Listing 648: 2_08_006d.bat

Die Lösung mit VBScript

Um einen einzelnen AT-Task per VBScript zu löschen, stellen Sie mit der `GetObject`-Methode eine Verbindung zum WMI-Service her und selektieren das entsprechende Task-Objekt anhand der Task-ID. Anschließend wenden Sie die `Delete`-Methode an, um den AT-Task zu löschen.

```
Option Explicit
```

```
Dim oWMI
Dim oTask
Dim strComputer
Dim intID
```

```
strComputer = "."
intID = "4"
```

```
Set oWMI=GetObject("winmgmts:{impersonationLevel=_
impersonate}!\\" & strComputer)
Set oTask = oWMI.Get("Win32_ScheduledJob.JobID=" & intID)
oTask.Delete
```

```
Set oWMI = Nothing
Set oTask = Nothing
```

Listing 649: 2_08_006a.vbs

Möchten Sie dagegen alle eingerichteten AT-Tasks auf einmal löschen, so führen Sie erst eine `ExecQuery`-Abfrage gegen die WMI-Klasse `Win32_ScheduledJob` aus, woraufhin die Tasks in Form einer Kollektion zurückgeliefert werden. Mit einer `For Each`-

Schleife können Sie dann pro Task die Task-ID ermitteln, eine Verbindung zur jeweiligen Instanz herstellen und die jeweilige Instanz dann mit der `Delete`-Methode löschen.

```
Option Explicit

Dim oWMI
Dim oTask
Dim colTasks
Dim Task
Dim strComputer
Dim intID

strComputer = "."

Set oWMI=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\" & strComputer)
Set colTasks = oWMI.ExecQuery ("SELECT * FROM _
    Win32_ScheduledJob")

For Each Task in colTasks
    intID = Task.JobID
    Set oTask = oWMI.Get("Win32_ScheduledJob.JobID=\"" _
        & intID & "\"")
    oTask.Delete
Next

Set oWMI = Nothing
Set oTask = Nothing
```

Listing 650: 2_08_006b.vbs

Wenn Sie stattdessen einen Task löschen möchten, der mit `Schtasks` oder mit dem Assistenten zur Einrichtung geplanter Tasks erstellt wurde, so verwenden Sie den Befehl `Schtasks /Delete`, den Sie mit der `Run`-Methode des `Shell`-Objektes auch direkt aus einem VBScript heraus aufrufen können. Sie müssen lediglich den Parameter `/tn` mit der Bezeichnung des Tasks und auch den Parameter `/f` zum Erzwingen des Löschens übergeben:

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c Schtasks /Delete _
```

Listing 651: 2_08_006c.vbs

```
/tn Defragmentierung /f"
```

```
Set oShell = Nothing
```

Listing 651: 2_08_006c.vbs (Forts.)

188 Benutzerkonto für die Task-Ausführung ändern

Wenn Sie unter grafischer Oberfläche einen neuen Task einrichten, so legen Sie während der Task-Einrichtung auch das Benutzerkonto und das dazugehörige Passwort fest, unter dessen Verwendung der Task später ausgeführt werden soll. Und für den Fall, dass sich Änderungen an der Berechtigungsstruktur ergeben, weil vielleicht Benutzerkonten oder Gruppenmitgliedschaften reorganisiert wurden, so müssen Sie gegebenenfalls auch das Benutzerkonto für die Task-Ausführung ändern.

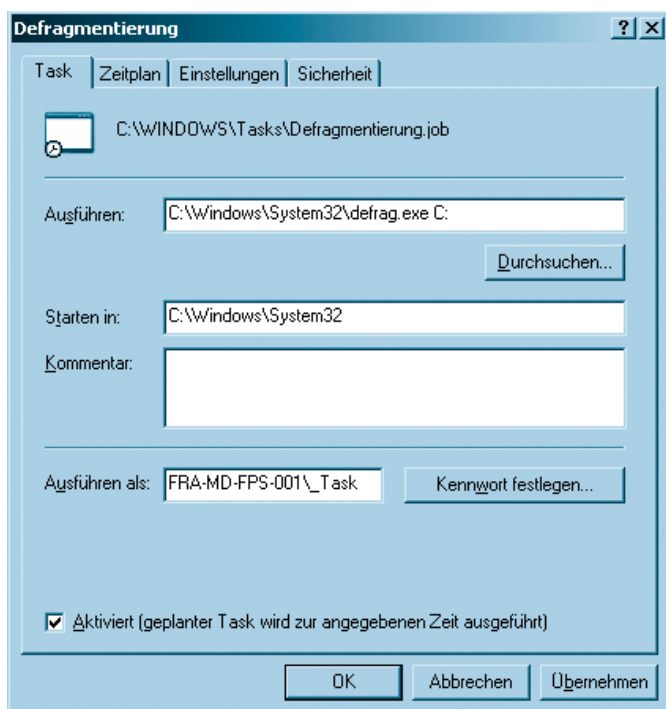


Abbildung 264: Einen neuen Benutzernamen festlegen

Hierzu wechseln Sie in die Übersicht der geplanten Tasks (Systemsteuerungsobjekt GEPLANTE TASKS), markieren den betreffenden Task und öffnen im Kontextmenü des Tasks die Eigenschaften. Auf der Registerkarte TASK geben Sie im Feld AUSFÜHREN ALS zunächst den Benutzernamen (Format *Domänenname\Benutzername*) an, unter dessen Verwendung der Task später ausgeführt werden soll. Anschließend klicken Sie

auf die Schaltfläche KENNWORT FESTLEGEN, um das dazugehörige Passwort einzugeben und auch noch einmal zu bestätigen. Schließen Sie das Fenster wieder, und speichern Sie die Änderung mit einem Klick auf ÜBERNEHMEN ab.

Ein AT-Task dagegen wird per Default unter Verwendung des Systemkontos ausgeführt. Sie haben aber auch hier die Möglichkeit, einen Benutzernamen zu hinterlegen, unter dessen Verwendung die AT-Tasks später ausgeführt werden sollen.

Wechseln Sie hierzu in die Übersicht der geplanten Tasks (Systemsteuerungsobjekt GEPLANTE TASKS), und wählen Sie im Menü ERWEITERT den Eintrag AT-DIENSTKONTO. Im Dialogfenster KONFIGURATION DES AT-DIENSTKONTOS aktivieren Sie die Option DIESES KONTO und geben den Benutzernamen (Format *Domänenname\Benutzername*) wie auch das dazugehörige Passwort ein. Das Passwort bestätigen Sie noch einmal und schließen das Dialogfenster daraufhin mit einem Klick auf OK.

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus das Benutzerkonto, unter dessen Verwendung ein mit `Schtasks` oder unter grafischer Oberfläche eingerichteteter Task ausgeführt wird, ändern möchten, so verwenden Sie hierfür den `Schtasks /Change`-Befehl und übergeben neben der Task-Bezeichnung (`/tn`) auch den neuen Benutzernamen (`/ru`) und das dazugehörige Passwort (`/rp`):

```
@Echo off
Schtasks /Change /tn Defragmentierung _
/ru FRA-MD-FPS-001\_Task /rp 12345678
```

Listing 652: 2_08_007.bat

Die Lösung mit VBScript

Wenn Sie aus einem VBScript heraus das Benutzerkonto für einen mit `Schtasks` oder unter grafischer Oberfläche eingerichteten Task ändern möchten, dann gehen Sie genauso vor wie in der Eingabeaufforderung.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c _
Schtasks /Change /tn Defragmentierung _
/ru FRA-MD-FPS-001\_Task /rp 12345678"

Set oShell = Nothing
```

Listing 653: 2_08_007.vbs

189 Automatischer Programmstart vor der Windows-Anmeldung

In einigen Fällen kann es erforderlich sein, bestimmte Programme noch vor der Windows-Anmeldung automatisiert starten zu lassen. Beispielsweise dann, wenn Sie die finale Konfiguration nach einer Betriebssystem-Installation per Batch-Datei oder sonstigem Skript vornehmen möchten, ohne dass eine automatische Administratoren-Anmeldung erfolgt. Oder wenn Sie unmittelbar nach dem Computerneustart bestimmte Programme starten möchten, die auch dann ihren Dienst im Hintergrund tun, wenn kein Benutzer angemeldet ist.

Um ein Programm direkt nach dem Systemstart und damit noch vor der Windows-Anmeldung automatisch starten zu lassen, haben Sie nun mehrere Möglichkeiten:

- ▶ **Geplanter Task:** Erstellen Sie einen neuen Task, legen Sie das gewünschte Programm, die Batch-Datei oder das VBScript fest, und wählen Sie im Zeitplan das Systemereignis BEIM SYSTEMSTART aus. Der Task wird daraufhin unmittelbar nach dem Starten des Dienstes TASKPLANER aufgerufen und im Hintergrund ausgeführt.
- ▶ **Start-Skript:** Erstellen Sie ein Start-Skript in Form von einer Batch-Datei und kopieren Sie das Skript in den Ordner `C:\Windows\System32\GroupPolicy\Machine\Scripts\Startup`. Anschließend öffnen Sie die Gruppenrichtlinien, wechseln in den Pfad `COMPUTERKONFIGURATION\WINDOWS-EINSTELLUNGEN\STARTEN` und fügen das Skript der Liste der Start-Skripte hinzu.
- ▶ **AutoExNt:** Im *Windows Server 2003 Resource Kit* ist ein Tool namens `Autoexnt.exe` enthalten, womit Sie eine Batch-Datei (*Autoexnt.bat*) erstellen können, die während dem Systemstart automatisch ausgeführt wird. In dieser Batch-Datei können Sie Programme, Batch-Dateien, einzelne Befehlszeilen wie auch VBScript-Aufrufe hinterlegen.
- ▶ **Neuer Dienst:** Mit Hilfe des Tools `Sc.exe`, das fester Bestandteil des Befehlssatzes von Windows Server 2003 ist, können Sie neue Dienste erstellen und damit eine gewöhnliche Applikation auch als Dienst ausführen lassen (siehe *Systemmanagement*). Das Starten der Dienste erfolgt ebenfalls mit dem Systemstart und noch vor der Windows-Anmeldung.

Einmal angenommen, Sie haben eine Batch-Datei oder ein VBScript erstellt und möchten die Befehlszeilen unmittelbar nach dem Systemstart, aber noch vor der Windows-Anmeldung, automatisch ausführen lassen. Die Batch-Datei oder das VBScript soll aber nur nach dem nächsten Systemstart aufgerufen werden, weitere, sprich: regelmäßige Aufrufe sollen nicht erfolgen.

Um diese Aufgabe zu lösen, erstellen Sie zunächst einen neuen Task und hinterlegen die Batch-Datei oder das VBScript. Öffnen Sie hierzu das Systemsteuerungsobjekt GEPLANTE TASKS, und klicken Sie auf GEPLANTEN TASK HINZUFÜGEN, um den Assistenten zum Einrichten neuer Tasks aufzurufen. Schließen Sie den Willkommensbildschirm mit einem Klick auf WEITER, und legen Sie im nächsten Dialogfenster die Batch-Datei oder das VBScript als auszuführendes Programm fest. Im nächsten Schritt geben Sie eine

Bezeichnung für den Task ein und wählen BEIM STARTEN DES COMPUTERS als Ausführungszeitpunkt aus. Mit einem Klick auf WEITER wird ein Dialogfenster geöffnet, worüber Sie den Benutzernamen und auch das Passwort des Benutzers eingeben müssen, unter dessen Verwendung der Task später ausgeführt wird. Schließen Sie die Einrichtung des neuen Tasks ab, und aktivieren Sie im letzten Dialogfenster die Option ERWEITERTE EIGENSCHAFTEN FÜR DIESEN TASK BEIM KLICKEN AUF »FERTIG STELLEN« ÖFFNEN. Auf der Registerkarte TASK überprüfen Sie nun die Befehlszeile und fügen gegebenenfalls den erforderlichen Kommandozeileninterpreter wie auch die Pfad- und Parameterangaben hinzu.

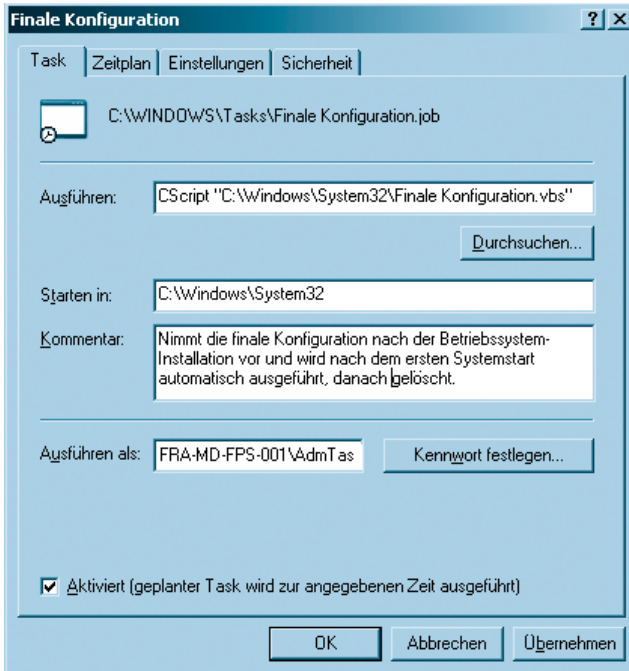


Abbildung 265: Die Eigenschaften des neuen Tasks

Damit der soeben erstellte Task nach der ersten Ausführung von selbst gelöscht wird, müssen Sie nun aus der Batch-Datei oder aus dem VBScript heraus eine weitere Batch-Datei aufrufen, womit der Task zunächst beendet und anschließend vom System gelöscht wird. Diese Batch-Datei muss folgende Befehlszeilen enthalten:

```
Schtasks /End /tn "Finale Konfiguration"  
Schtasks /Delete /tn "Finale Konfiguration" /f
```

Listing 654: Task beenden und löschen

Wenn Sie diese beiden Befehlszeilen direkt aus dem Skript heraus aufrufen, das im eigentlichen Task hinterlegt ist, so wird das Löschen des Tasks misslingen, da Sie keine Tasks löschen können, die gerade ausgeführt werden. Sie müssen das Skript also erst beenden und damit freigeben, um den Task löschen zu können.

Die Lösung per Eingabeaufforderung

Gleiches lässt sich natürlich auch in der Eingabeaufforderung realisieren, in dem Sie mit dem `Schtasks /Create`-Befehl zunächst einen neuen Task einrichten, worüber die Batch-Datei *Finale Konfiguration.bat* bei Systemstart ausgeführt wird:

```
@Echo off
Schtasks /Create /SC BEIMSTART /tn "Finale Konfiguration" _
/tr "C:\Windows\System32\Finale Konfiguration.bat" _
/ru FRA-MD-FPS-001\AdmInstall /rp 12345678
```

Listing 655: 2_08_008a.bat

Wenn die angegebene Batch-Datei nur einmal ausgeführt und nicht wiederholt werden soll, so müssen Sie aus dieser Batch-Datei heraus eine weitere Batch-Datei aufrufen, worüber der Task mit `Schtasks /End` zunächst beendet und anschließend mit `Schtasks /Delete` gelöscht wird:

```
@Echo off
Schtasks /End /tn "Finale Konfiguration"
Schtasks /Delete /tn "Finale Konfiguration" /f
```

Listing 656: 2_08_008b.bat

Die Lösung mit VBScript

Auch mit VBScript lässt sich ein Programm noch vor der Windows-Anmeldung starten. Der Einfachheit halber erstellen Sie – wie auch in den vorangegangenen beiden Beispielen – einen entsprechenden Task, der während des Systemstarts gestartet wird und das betreffende Programm aufruft.

Nun lassen sich mit Hilfe der `Create`-Methode der WMI-Klasse `Win32_ScheduledJob` zwar problemlos neue AT-Tasks erstellen, allerdings können diese Tasks nur zu bestimmten Uhrzeiten ausgeführt werden, nicht aber beim Auftreten bestimmter Systemereignisse (z.B. während des Systemstarts). Sie müssen daher den `Schtasks /Create`-Befehl der Eingabeaufforderung einsetzen, den Sie mit der `Run`-Methode des `Shell`-Objektes auch aus einem VBScript heraus aufrufen können. Denn dieser Befehl stellt die Möglichkeit bereit, einen Task auch mit dem Auftreten eines bestimmten Systemereignisses (z.B. *BeimStart*) zu starten.

```
Option Explicit
Dim oShell

Set oShell = CreateObject ("WScript.Shell")
oShell.Run "%comspec% /c _
    Schtasks /Create /sc BEIMSTART /tn FinaleKonfiguration _
    /tr C:\Windows\System32\FinaleKonfiguration.bat _
    /ru FRA-MD-FPS-001\AdmTask /rp 12345678"

Set oShell = Nothing
```

Listing 657: 2_08_008.vbs

Auch hier gilt: Wenn der Task nur einmal – und nicht regelmäßig – ausgeführt werden soll, so muss nach der Programmausführung ein weiteres VBScript oder eine Batch-Datei aufgerufen werden, womit der Task beendet und anschließend gelöscht wird.

190 Automatischer Programmstart nach der Windows-Anmeldung

Für den Fall, dass bestimmte Programme erst nach der Anmeldung am Windows Server 2003 automatisch ausgeführt werden sollen, so fügen Sie die betreffenden Programme entweder der *Autostart*-Gruppe hinzu, hinterlegen es im *Run*-Schlüssel der Registrierung oder erstellen einen neuen Task (siehe Geplante Tasks einrichten auf Seite 735). Dabei haben Sie bei den ersten beiden Varianten jeweils die Wahl, ob der automatische Programmstart generell bei jeder Windows-Anmeldung erfolgen soll oder nur bei der Anmeldung durch einen bestimmten Benutzer.

Soll das Programm beispielsweise generell nach jeder Windows-Anmeldung ausgeführt werden (und damit unabhängig von einem Benutzer), dann erstellen Sie in der allgemeinen *Autostart*-Gruppe, die im Ordner *C:\Dokumente und Einstellungen\All Users\Programme\Startmenü\Autostart* abgelegt ist, eine Verknüpfung auf das betreffende Programm. Alternativ können Sie auch das Programm selbst in den *Autostart*-Ordner kopieren. Soll das Programm dagegen nur bei der Anmeldung durch einen bestimmten Benutzer gestartet werden, so verwenden Sie stattdessen den *Autostart*-Ordner im jeweiligen Benutzerprofil (z.B. *C:\Dokumente und Einstellungen\Administrator\Programme\Startmenü\Autostart*) (siehe Abbildung 266).

Statt den Programmaufruf mit Hilfe der Autostart-Gruppe zu automatisieren, können Sie auch einen entsprechenden Eintrag im *Run*-Schlüssel der Registrierung erzeugen (siehe auch Kategorie *Registrierung*). Soll das Programm mit jeder Windows-Anmeldung gestartet werden, dann legen Sie im Schlüssel *HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN* eine neue Zeichenfolge an, geben eine aussagekräftige Bezeichnung ein und hinterlegen den Programmpfad als eigentliche Zeichenfolge. Soll das Programm dagegen nur mit der Anmeldung eines bestimmten Benutzers gestartet werden, so legen Sie die Zeichenfolge stattdessen im Schlüssel *HKU\SID\SOFTWARE*

MICROSOFT\WINDOWS\CURRENTVERSION\RUN an, wobei SID durch die Security-ID des gewünschten Benutzerprofils zu ersetzen ist (siehe *Welche SID hat der Benutzer?* in der Kategorie *Benutzer und Gruppen*).

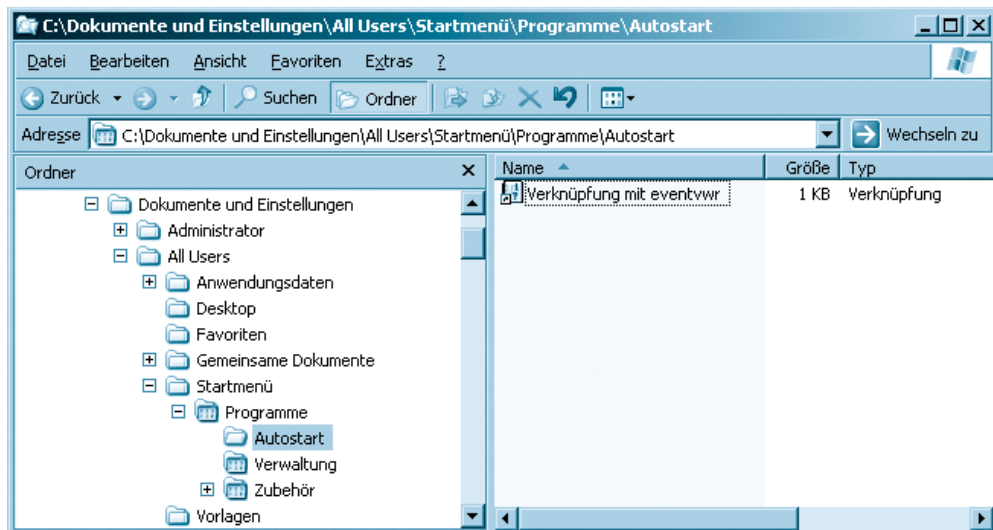


Abbildung 266: Einen Autostart-Eintrag für alle Benutzer erzeugen

Die Lösung per Eingabeaufforderung

Um von der Eingabeaufforderung aus bestimmte Programme oder Befehlszeilen mit der Windows-Anmeldung automatisch ausführen zu lassen, gehen Sie ähnlich wie unter der grafischen Oberfläche vor. Will heißen, dass Sie das Programm oder die Befehlszeile entweder in die Autostart-Gruppe kopieren oder im Run-Schlüssel der Registrierung einen neuen Eintrag erzeugen.

Wenn Sie zum Beispiel mit der Windows-Anmeldung, gleich durch welchen Benutzer, automatisch die Ereignisanzeige starten lassen möchten, so brauchen Sie nur die Datei *Eventvwr.msc*, die sich im Ordner *C:\Windows\System32* befindet, mit Hilfe des Copy-Befehls in den Ordner *C:\Dokumente und Einstellungen\All Users\Startmenü\Programme\Autostart* zu kopieren. Wenn der automatische Start der Ereignisanzeige dagegen nur mit der Anmeldung eines bestimmten Benutzers erfolgen soll, so wählen Sie stattdessen den Autostart-Ordner im Benutzerprofil des Benutzers (*C:\Dokumente und Einstellungen\%Username%\Programme\Startmenü\Programme\Autostart*).

```
@Echo off
Copy C:\Windows\System32\Eventvwr.msc _
"C:\Dokumente und Einstellungen\
All Users\Startmenü\Programme\Autostart"
```

Listing 658: 2_08_009a.bat

Alternativ können Sie auch im Run-Schlüssel der Registrierung einen neuen Eintrag erzeugen und den Pfad zur Datei *Eventvwr.msc* als Zeichenfolge hinterlegen, wobei Sie den Run-Schlüssel unter HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN verwenden, wenn der automatische Start der Ereignisanzeige für alle Benutzer gelten soll. Anderenfalls verwenden Sie den Run-Schlüssel in den Benutzereinstellungen, den Sie unter HKU\SID\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN finden, wobei Sie SID durch die SID des jeweiligen Benutzers ersetzen müssen.

Um einen neuen Eintrag im Run-Schlüssel zu erzeugen, verwenden Sie den `Reg Add`-Befehl (siehe Kategorie *Registrierung*), dem Sie folgende Parameter übergeben müssen:

- ▶ Schlüssel: Der Schlüssel, der um den neuen Wert erweitert werden soll
- ▶ Wert (/v): Die Bezeichnung des Wertes, der neu angelegt werden soll
- ▶ Daten-Typ (/t): Der Datentyp des neuen Wertes, der in diesem Fall `REG_SZ` (= Zeichenfolge) lautet
- ▶ Daten (/d): Die Daten, die dem Wert zugewiesen werden sollen. In diesem Beispiel ist der Pfad zur Ereignisanzeige anzugeben.

Das nachfolgende Beispiel legt unter HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN einen neuen Wert mit der Bezeichnung *EventViewer* an und weist dem Wert den Pfad zur Ereignisanzeige, die sich unter *C:\Windows\System32* befindet, zu:

```
@Echo off
Reg Add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run _
/v "EventViewer" /t REG_SZ _
/d C:\Windows\System32\Eventvwr.msc
```

Listing 659: 2_08_009b.bat

Die Lösung mit VBScript

Auch mit VBScript lassen sich Programme nach der Windows-Anmeldung zum automatischen Start bewegen, indem Sie das betreffende Programm entweder in den Autostart-Ordner kopieren oder aber im Run-Schlüssel der Registrierung hinterlegen.

Wenn z.B. nach jeder Windows-Anmeldung (gleich durch welchen Benutzer) die Ereignisanzeige gestartet werden soll, so brauchen Sie nur die Datei *Eventvwr.msc*, die sich im Ordner *C:\Windows\System32* befindet, in den allgemeinen Autostart-Ordner (*C:\Dokumente und Einstellungen\All Users\Startmenü\Programme\Autostart*) zu kopieren. Hierfür verwenden Sie die `CopyFile`-Methode des `Filesystem`-Objektes, der Sie folgende Parameter übergeben:

- ▶ `strSourceFile`: Pfad und Dateiname der Quelldatei
- ▶ `strDestinationFile`: Pfad und Dateiname der Zieldatei
- ▶ `Overwrite`: Legt fest, ob die Zieldatei überschrieben werden soll (`True`), wenn sie bereits vorhanden ist, oder nicht (`False`)

Das nachfolgende Beispiel zeigt eine solche Möglichkeit. Zunächst werden die Variablen und Konstanten deklariert, damit das Skript überschaubar bleibt und leichter angepasst werden kann. Danach wird mit der `CreateObject`-Methode das Filesystem-Objekt referenziert und die `CopyFile`-Methode gegen das in `oFS` angegebene Filesystem-Objekt angewendet, wobei auch die erforderlichen Parameter (`strSourceFile`, `strDestinationFile` und `Overwrite`) übergeben werden.

```
Dim oFS
Dim strSourceFile
Dim strDestinationFile

Const Overwrite = True
strSourceFile = "C:\Windows\System32\Eventvwr.msc"
strDestinationFile = "C:\Dokumente und Einstellungen\All _
    Users\Startmenü\Programme\Autostart\Eventvwr.msc"

Set oFS = CreateObject ("Scripting.FileSystemObject")
oFS.CopyFile strSourceFile, strDestinationFile, Overwrite

Set oFS = Nothing
```

Listing 660: 2_08_009a.vbs

Alternativ dazu können Sie mit Hilfe der `SetStringValue`-Methode (siehe *Registrierung*) der WMI-Klasse `StdRegProv` auch einen neuen Eintrag im `Run`-Schlüssel der Registrierung erzeugen und den Pfad, wie auch die ausführbare Datei, als Zeichenfolge hinterlegen:

Stellen Sie mit der `GetObject`-Methode zunächst eine Verbindung zur WMI-Klasse `StdRegProv` her, und setzen Sie den `Impersonation-Level` auf `Impersonate`. Anschließend setzen Sie die `SetStringValue`-Methode gegen den Schlüssel (`strKey`) des angegebenen Hauptzweiges (`HKLM`) ein und übergeben die Bezeichnung der neuen Zeichenfolge (`strValueName`) wie auch die eigentliche Zeichenfolge (`strValue`) selbst.

```
Option Explicit

Dim oReg
Dim strComputer
Dim strKey
Dim strValueName
Dim strValue

Const HKLM = &H80000002
strComputer = "."
strKey = "SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
```

Listing 661: 2_08_009b.vbs

```

strValueName = "Ereignisanzeige"
strValue = "C:\Windows\System32\Eventvwr.msc"

Set oReg=GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")
oReg.SetStringValue HKLM, strKey, strValueName, strValue

Set oReg = Nothing

```

Listing 661: 2_08_009b.vbs (Forts.)

191 Automatischer Programmstart vor der Windows-Abmeldung

Wenn Sie bestimmte Befehlszeilen oder Programme unmittelbar vor der Windows-Abmeldung ausführen lassen möchten, so bietet der Taskplaner leider keine entsprechenden Möglichkeiten an. Sie können aber in den Gruppenrichtlinien ein Abmeldeskript definieren, worüber die Befehlszeilen oder Programme dann automatisch aufgerufen werden, sobald das Systemereignis *Abmelden* eintritt.

Allerdings gilt dabei zu berücksichtigen, dass Programme oder Befehlszeilen, die über ein Abmeldeskript gestartet werden, nicht interaktiv ausgeführt werden können. Das heißt, dass Sie im Abmeldeskript nur solche Befehlszeilen oder Programme verwenden können, die keine Interaktion mit dem Benutzer erfordern. Außerdem werden die Abmeldeskripte unter Verwendung des Benutzerkontos ausgeführt, der die Abmeldung gerade durchführt, weswegen die erforderlichen Berechtigungen zu berücksichtigen sind.

Bestes Beispiel für den Einsatz eines Abmeldeskriptes ist das Löschen temporärer Dateien vor der Windows-Abmeldung. Um dies zu realisieren, erstellen Sie zunächst eine Batch-Datei (oder wahlweise auch ein VBScript), in der Sie die erforderlichen Befehlszeilen hinterlegen. Möchten Sie beispielsweise alle Dateien mit dem Suffix **.tmp* (unabhängig von ihrem Ablageort) und alle temporären Internetdateien löschen, dabei auch die benutzerspezifischen Temp-Ordner leeren, dann erstellen Sie im einfachsten Fall eine Batch-Datei mit folgendem Inhalt:

```

@Echo off
%SystemDrive%
CD\
Del *.tmp /s /f /q
Del %Temp%\*.* /s /f /q
Del "%UserProfile%\Lokale _
    Einstellungen\Temporary Internet Files\*.*" /s /f /q

```

Listing 662: 2_08_010a.bat

Speichern Sie diese Datei im Ordner `C:\Windows\System32\GroupPolicy\User\Scripts\Logoff` als Batch-Datei (z.B. als *Abmeldeskript.bat*) ab, und öffnen Sie daraufhin die Gruppenrichtlinien. Auf einem Standalone-Server öffnen Sie hierzu das STARTMENÜ, klicken auf AUSFÜHREN und geben `Gpedit.msc` ein. Auf einem Member-Server oder auf einem Domänen-Controller dagegen öffnen Sie ACTIVE DIRECTORY-BENUTZER UND -COMPUTER, wählen im Konsolenstamm die entsprechende Domäne oder die Organisationseinheit (OU) aus, öffnen die Eigenschaften und wechseln auf die Registerkarte GRUPPENRICHTLINIE. Dort klicken Sie auf BEARBEITEN.

Innerhalb der Gruppenrichtlinien wechseln Sie zu BENUTZERKONFIGURATION\WINDOWS-EINSTELLUNGEN\SKRIPTS (ANMELDEN/ABMELDEN) und öffnen in der rechten Fensterhälfte die Richtlinie ABMELDEN. Klicken Sie auf HINZUFÜGEN und anschließend auf DURCHSUCHEN, und wählen Sie das soeben erstellte Abmeldeskript aus. Schließen Sie die Fenster wieder, die Gruppenrichtlinie ist nun definiert und – sofern Sie die Einstellung in den lokalen Richtlinien vorgenommen haben – auch sofort aktiv.

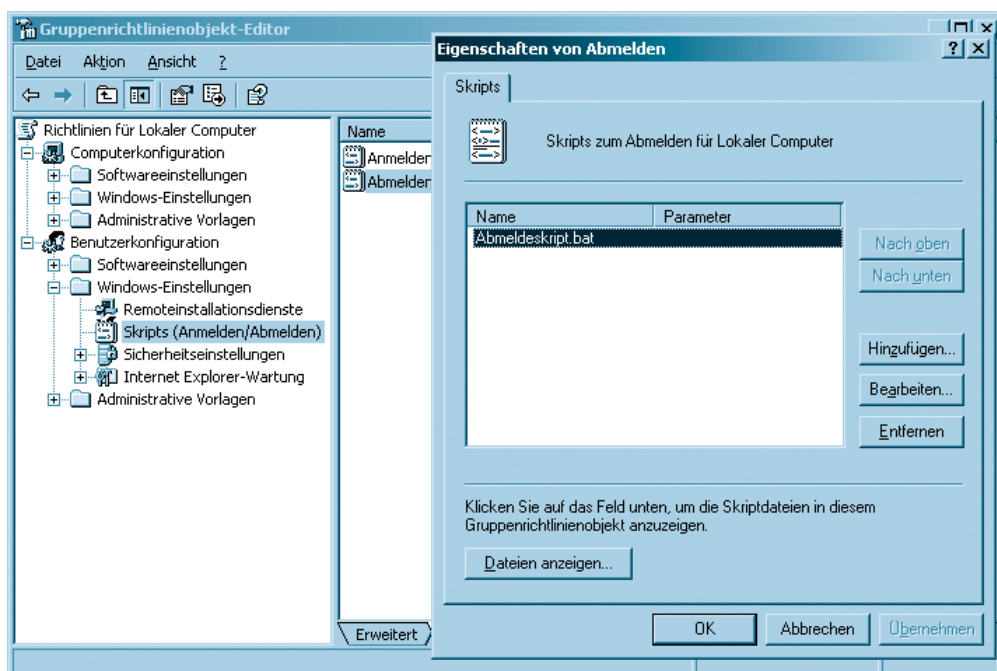


Abbildung 267: Das Abmeldeskript aktivieren

Die Lösung per Eingabeaufforderung

Um ein Abmeldeskript zu hinterlegen, womit bestimmte Befehlszeilen oder Programme unmittelbar vor der Windows-Abmeldung ausgeführt werden, reicht es nicht aus, das Abmeldeskript nur in den Ordner `C:\Windows\System32\GroupPolicy\User\Scripts\Logoff` zu kopieren. Denn das Abmeldeskript wird nur dann ausgeführt, wenn auch die

entsprechende Richtlinie konfiguriert wurde, wofür es in der Eingabeaufforderung aber leider keine entsprechenden Befehle gibt.

Über einen kleinen Umweg können Sie das Abmeldeskript aber zumindest in den lokalen Gruppenrichtlinien definieren. Denn wenn Sie in den lokalen Gruppenrichtlinien ein Abmeldeskript hinterlegen, so wird im Schlüssel HKCU\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\SYSTEM\SCRIPTS\LOGOFF\0 ein neuer Schlüssel mit fortlaufender Nummerierung erstellt, worin die Werte und Daten des Abmeldeskriptes hinterlegt werden.

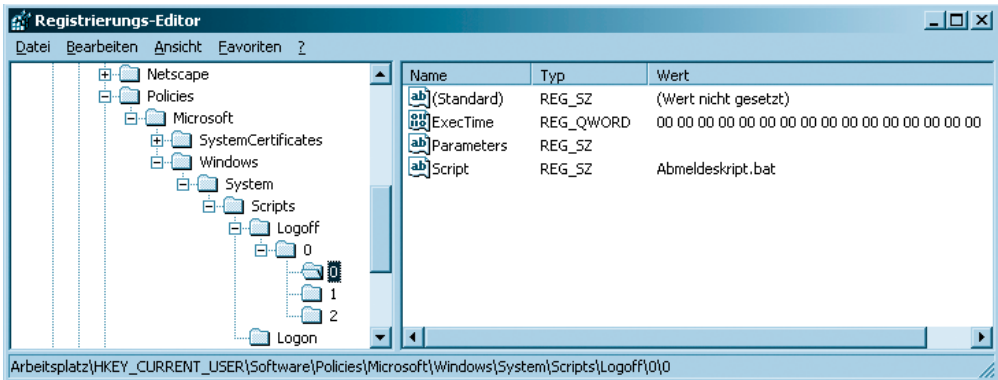


Abbildung 268: Die Registrierungseinstellungen für das Abmeldeskript

Das nachfolgende Code-Beispiel zeigt, wie Sie mit Hilfe des `Reg Add`-Befehls ein Abmeldeskript hinterlegen können. Zuerst wird das Abmeldeskript in den Zielordner (`C:\Windows\System32\GroupPolicy\User\Scripts\Logoff`) kopiert. Anschließend wird mit `Reg Add` die erforderliche Schlüsselstruktur erstellt, und die erforderlichen Werte und Daten werden in die Registrierung geschrieben:

```
@Echo off
Copy Abmeldeskript.bat C:\Windows\System32\GroupPolicy\
  User\Scripts\Logoff /y

Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
  Scripts\Logoff
Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
  Scripts\Logon
Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
  Scripts\Logoff\0
Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
  Scripts\Logoff\0\0
Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
  Scripts\Logoff\0 /v "DisplayName" /t REG_SZ _
```

Listing 663: 2_08_010b.bat

```

/d "Richtlinien der lokalen Gruppe"
Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
Scripts\Logoff\0 /v "FileSysPath" /t REG_SZ _
/d "C:\WINDOWS\System32\GroupPolicy\User"
Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
Scripts\Logoff\0 /v "GPO-ID" /t REG_SZ _
/d "LocalGPO"
Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
Scripts\Logoff\0 /v "GPOName" /t REG_SZ _
/d "Richtlinien der lokalen Gruppe"
Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
Scripts\Logoff\0 /v "SOM-ID" /t REG_SZ _
/d "Local"
Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
Scripts\Logoff\0\0 /v "ExecTime" /t REG_BINARY _
/d 00000000000000000000000000000000

Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
Scripts\Logoff\0\0 /v "Parameters" /t REG_SZ /d ""
Reg Add HKCU\Software\Policies\Microsoft\Windows\System\
Scripts\Logoff\0\0 /v "Script" /t REG_SZ _
/d "Abmeldeskript.bat"

```

Listing 663: 2_08_010b.bat (Forts.)

Das obige Code-Beispiel setzt voraus, dass die lokale Richtlinie ABMELDEN noch nicht konfiguriert wurde. Denn für den Fall, dass die Richtlinie schon konfiguriert ist und ein oder auch mehrere Abmeldeskripte hinterlegt wurden, so würden die Einstellungen des ersten Skripts überschrieben werden. Im produktiven Einsatz sollten Sie also erst prüfen, ob die Schlüsselstruktur HKEY_CURRENT_USER\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\SYSTEM\SCRIPTS\LOGOFF bereits existiert und wie viele Schlüssel im untergeordneten Schlüssel 0 enthalten sind. Denn anhand dieser Anzahl können Sie herausfinden, wie viele Skripte bereits hinterlegt wurden und welche Nummer das nächste zu aktivierende Skript erhält.

Die Lösung mit VBScript

Um in den lokalen Gruppenrichtlinien ein Abmeldeskript per VBScript zu hinterlegen, gehen Sie ähnlich vor wie in der Eingabeaufforderung. Will heißen, dass Sie erst das Abmeldeskript in den Ordner *C:\Windows\System32\GroupPolicy\User\Scripts\Logoff* kopieren und dann die entsprechenden Schlüssel, Werte und Daten in die Registrierung schreiben.

Das nachfolgende Code-Beispiel zeigt eine solche Lösung, wobei auch hier wieder davon ausgegangen wird, dass die lokale Richtlinie ABMELDEN noch nicht konfiguriert wurde (siehe vorheriger Abschnitt). Zunächst wird das Abmeldeskript mit Hilfe der CopyFile-Methode des Filesystem-Objektes von *C:\Daten\Skripte\Abmeldeskript.vbs*

(strSourceFile) nach C:\Windows\System32\GroupPolicy\User\Scripts\Logoff\Abmel-deskript.vbs (strDestinationFile) kopiert, wobei ein eventuell vorhandenes Abmelde-skript mit dem gleichen Dateinamen überschrieben wird (Overwrite).

Anschließend werden die benötigten Schlüssel (strKey1 – strKey4) in die Registrierung geschrieben, wofür die CreateKey-Methode der WMI-Klasse StdRegProv (WMI-Namespace root\default) verwendet wird. Und nach dem Erstellen der Schlüssel werden die Werte und Daten in die Registrierung geschrieben, was mit Hilfe der Set-StringValue- und SetBinaryValue-Methode der WMI-Klasse StdRegProv erfolgt.

Option Explicit

```
Dim oReg
Dim oFS
Dim strComputer
Dim strSourceFile
Dim strDestinationFile
Dim strKey1
Dim strKey2
Dim strKey3
Dim strKey4
Dim strValueName1, strValue1
Dim strValueName2, strValue2
Dim strValueName3, strValue3
Dim strValueName4, strValue4
Dim strValueName5, strValue5
Dim strValueName6, strValue6
Dim strValueName7, strValue7
Dim strValueName8, strValue8

Const HKCU = &H80000001
Const Overwrite = True
strComputer = "."
strSourceFile = "C:\Daten\Skripte\AbmeldeSkript.vbs"
strDestinationFile = "C:\Windows\System32\GroupPolicy\User\
    Scripts\Logoff\Abmel-deskript.vbs"

strKey1 = "Software\Policies\Microsoft\Windows\System\
    Scripts\Logoff"
strKey2 = "Software\Policies\Microsoft\Windows\System\
    Scripts\Logon"
strKey3 = "Software\Policies\Microsoft\Windows\System\
    Scripts\Logoff\0"
strKey4 = "Software\Policies\Microsoft\Windows\System\
    Scripts\Logoff\0\0"
```

```

strValueName1 = "DisplayName"
strValue1 = "Richtlinien der lokalen Gruppe"
strValueName2 = "FileSysPath"
strValue2 = "C:\WINDOWS\System32\GroupPolicy\User"
strValueName3 = "GPO-ID"
strValue3 = "LocalGPO"
strValueName4 = "GPOName"
strValue4 = "Richtlinien der lokalen Gruppe"
strValueName5 = "SOM-ID"
strValue5 = "Local"
strValueName6 = "ExecTime"
strValue6 = Array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
strValueName7 = "Parameters"
strValue7 = ""
strValueName8 = "Script"
strValue8 = "Abmeldeskript.bat"

Set oFS = CreateObject("Scripting.FileSystemObject")
oFS.CopyFile strSourceFile, strDestinationFile, Overwrite
Set oReg = GetObject("winmgmts:{impersonationLevel=_
    impersonate)!\" & strComputer & _
    "\root\default:StdRegProv")

oReg.CreateKey HKCU, strKey1
oReg.CreateKey HKCU, strKey2
oReg.CreateKey HKCU, strKey3
oReg.CreateKey HKCU, strKey4

oReg.SetStringValue HKCU, strKey3, strValueName1, strValue1
oReg.SetStringValue HKCU, strKey3, strValueName2, strValue2
oReg.SetStringValue HKCU, strKey3, strValueName3, strValue3
oReg.SetStringValue HKCU, strKey3, strValueName4, strValue4
oReg.SetStringValue HKCU, strKey3, strValueName5, strValue5
oReg.SetBinaryValue HKCU, strKey4, strValueName6, strValue6
oReg.SetStringValue HKCU, strKey4, strValueName7, strValue7
oReg.SetStringValue HKCU, strKey4, strValueName8, strValue8

Set oReg = Nothing
Set oFS = Nothing

```

Listing 664: 2_08_010.vbs (Forts.)

192 Automatischer Programmstart vor dem Herunterfahren

Ähnlich wie Sie ein Abmeldeskript definieren können, das automatisch vor der Windows-Abmeldung ausgeführt wird, so können Sie mit Hilfe der Gruppenrichtlinien auch ein Skript hinterlegen, das unmittelbar vor dem Herunterfahren des Computersystems ausgeführt wird.

Um beispielsweise das Abmeldeskript von Seite 771 nicht nach der Abmeldung, sondern direkt vor dem Herunterfahren des Computersystems auszuführen, kopieren Sie das Skript erst in den Ordner `C:\Windows\System32\GroupPolicy\Machine\Scripts\Shutdown`. Anschließend öffnen Sie die Gruppenrichtlinien und führen einen Doppelklick auf die Richtlinie HERUNTERFAHREN aus, die sich unter COMPUTERKONFIGURATION\WINDOWS-EINSTELLUNGEN\SKRIPTS (START/HERUNTERFAHREN) befindet. Daraufhin wird das Dialogfenster EIGENSCHAFTEN VON HERUNTERFAHREN geöffnet, und Sie können das Skript mit einem Klick auf HINZUFÜGEN und anschließend auf DURCHSUCHEN auswählen und der Liste der Skripte hinzufügen. Übernehmen Sie die neuen Einstellungen, und schließen Sie die Gruppenrichtlinien wieder.

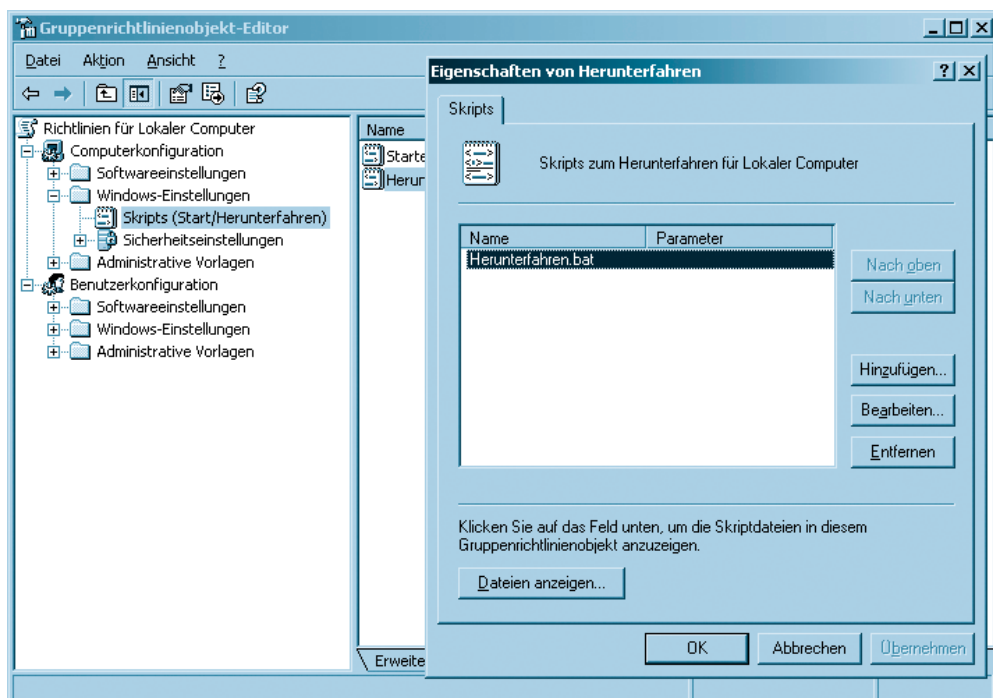


Abbildung 269: Das Skript hinzufügen

Die Lösung per Eingabeaufforderung

Wenn Sie von der Eingabeaufforderung aus ein Skript einrichten möchten, das direkt vor dem Herunterfahren des Computersystems gestartet wird, so gehen Sie auch wieder ähnlich wie im Rezept Automatischer Programmstart vor der Windows-Abmeldung auf Seite 772 vor: Kopieren Sie das Skript (z.B. *Herunterfahren.bat*) in den Ordner *C:\Windows\System32\GroupPolicy\Machine\Scripts\Shutdown*, und erstellen Sie dann mit Hilfe des *Reg Add*-Befehls die erforderlichen Schlüssel, Werte und Daten, um die lokale Richtlinie *HERUNTERFAHREN* zu aktivieren und das angegebene Skript als erstes Skript zu hinterlegen:

```
@Echo off
Copy Herunterfahren.bat C:\Windows\System32\GroupPolicy\
Machine\Scripts\Shutdown /y

Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
Scripts\Shutdown
Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
Scripts\Shutdown
Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
Scripts\Shutdown\0
Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
Scripts\Shutdown\0\0

Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
Scripts\Shutdown\0 /v "DisplayName" /t REG_SZ _
/d "Richtlinien der lokalen Gruppe"
Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
Scripts\Shutdown\0 /v "FileSysPath" /t REG_SZ _
/d "C:\WINDOWS\System32\GroupPolicy\Machine"
Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
Scripts\Shutdown\0 /v "GPO-ID" /t REG_SZ _
/d "LocalGPO"
Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
Scripts\Shutdown\0 /v "GPOName" /t REG_SZ _
/d "Richtlinien der lokalen Gruppe"
Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
Scripts\Shutdown\0 /v "SOM-ID" /t REG_SZ _
/d "Local"
Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
Scripts\Shutdown\0\0 /v "ExecTime" /t REG_BINARY _
/d 00000000000000000000000000000000

Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
Scripts\Shutdown\0\0 /v "Parameters" /t REG_SZ /d ""
Reg Add HKLM\Software\Policies\Microsoft\Windows\System\
```

Listing 665: 2_08_011.bat

```
Scripts\Shutdown\0\0 /v "Script" /t REG_SZ _
/d "Herunterfahren.bat"
```

Listing 665: 2_08_011.bat (Forts.)

Die Lösung mit VBScript

Wenn Sie das Skript, das unmittelbar vor dem Herunterfahren des Computersystems gestartet werden soll, mit VBScript einrichten und aktivieren möchten, so kopieren Sie erst die Skriptdatei mit der `CopyFile`-Methode des `FileSystem`-Objektes in den Ordner `C:\Windows\System32\Group Policy\Machine\Scripts\Shutdown`, erzeugen dann mit der `CreateKey`-Methode der WMI-Klasse `StdRegProv` die erforderliche Schlüsselstruktur unter `HKLM\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\SYSTEM\SCRIPTS`, um die Richtlinie `HERUNTERFAHREN` zu aktivieren, und legen zu guter Letzt die Werte und Daten mit der `SetStringValue`- und `SetBinaryValue`-Methode an, um das angegebene Skript als erstes Skript zu registrieren.

Option Explicit

```
Dim oReg
Dim oFS
Dim strComputer
Dim strSourceFile
Dim strDestinationFile
Dim strKey1
Dim strKey2
Dim strKey3
Dim strKey4
Dim strValueName1, strValue1
Dim strValueName2, strValue2
Dim strValueName3, strValue3
Dim strValueName4, strValue4
Dim strValueName5, strValue5
Dim strValueName6, strValue6
Dim strValueName7, strValue7
Dim strValueName8, strValue8

Const HKLM = &H80000002
Const Overwrite = True
strComputer = "."
strSourceFile = "C:\Herunterfahren.vbs"
strDestinationFile = C:\Windows\System32\GroupPolicy\
    Machine\Scripts\Shutdown\Herunterfahren.vbs"

strKey1 = "Software\Policies\Microsoft\Windows\System\_"
```

Listing 666: 2_08_011.vbs

```

Scripts\Shutdown"
strKey2 = "Software\Policies\Microsoft\Windows\System\_
Scripts\Shutdown"
strKey3 = "Software\Policies\Microsoft\Windows\System\_
Scripts\Shutdown\0"
strKey4 = "Software\Policies\Microsoft\Windows\System\_
Scripts\Shutdown\0\0"

strValueName1 = "DisplayName"
strValue1 = "Richtlinien der lokalen Gruppe"
strValueName2 = "FileSysPath"
strValue2 = "C:\WINDOWS\System32\GroupPolicy\Machine"
strValueName3 = "GPO-ID"
strValue3 = "LocalGPO"
strValueName4 = "GPOName"
strValue4 = "Richtlinien der lokalen Gruppe"
strValueName5 = "SOM-ID"
strValue5 = "Local"
strValueName6 = "ExecTime"
strValue6 = Array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
strValueName7 = "Parameters"
strValue7 = ""
strValueName8 = "Script"
strValue8 = "Herunterfahren.vbs"

Set oFS = CreateObject("Scripting.FileSystemObject")
oFS.CopyFile strSourceFile, strDestinationFile, Overwrite
Set oReg = GetObject("winmgmts:{impersonationLevel=_
    impersonate}!\\" & strComputer & _
    "\root\default:StdRegProv")

oReg.CreateKey HKLM, strKey1
oReg.CreateKey HKLM, strKey2
oReg.CreateKey HKLM, strKey3
oReg.CreateKey HKLM, strKey4

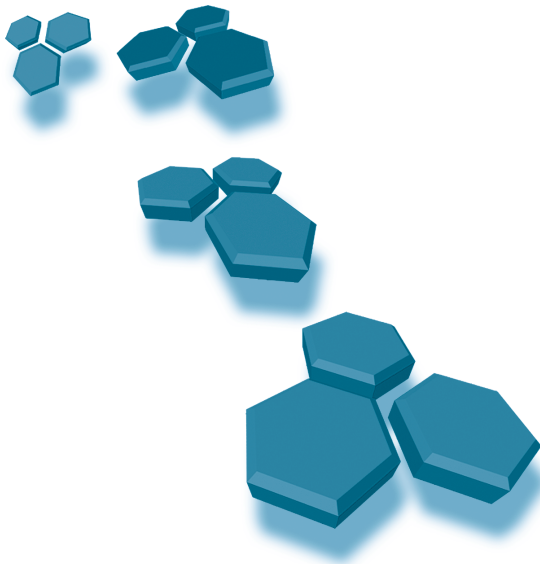
oReg.SetStringValue HKLM, strKey3, strValueName1, strValue1
oReg.SetStringValue HKLM, strKey3, strValueName2, strValue2
oReg.SetStringValue HKLM, strKey3, strValueName3, strValue3
oReg.SetStringValue HKLM, strKey3, strValueName4, strValue4
oReg.SetStringValue HKLM, strKey3, strValueName5, strValue5
oReg.SetBinaryValue HKLM, strKey4, strValueName6, strValue6
oReg.SetStringValue HKLM, strKey4, strValueName7, strValue7
oReg.SetStringValue HKLM, strKey4, strValueName8, strValue8

Set oReg = Nothing
Set oFS = Nothing

```

Listing 666: 2_08_011.vbs (Forts.)

Teil III Anhang



Referenz

Die Parameter für die DCPromo-Antwortdatei

Nachfolgend finden Sie eine Aufstellung der Parameter, die Sie in der DCPromo-Antwortdatei verwenden können:

Parameter	Beschreibung
AdministratorPassword = Kennwort	Setzt das Kennwort für den lokalen Administrator-Account (wird nur beim Herunterstufen eines Domänen-Controllers zum Member-Server benötigt).
AllowAnonymousAccess = Yes No	Legt fest, ob Prä-Windows 2000-Systeme zur Authentifizierung von Benutzerkonten verwendet werden (Yes) oder nicht (No).
AutoConfigDNS = Yes No	Legt fest, ob die DNS-Konfiguration automatisch vorgenommen werden soll (Yes), wenn dynamische Aktualisierungen nicht möglich sind oder nicht (No).
ChildName = NameChildDomäne	Legt die DNS-Bezeichnung für die Child-Domäne fest, die dem DNS-Suffix vorangestellt werden soll.
ConfirmGc = Yes No	Legt fest, ob der Domänen-Controller die Funktion des globalen Katalog-Servers einnimmt (Yes) oder nicht (No).
CreateOrJoin = Create Join	Legt fest, ob die Domäne Teil einer bestehenden Gesamtstruktur (Join) oder die Root-Domäne einer neuen Gesamtstruktur (Create) wird.
CriticalReplicationOnly = Yes No	Legt fest, ob während des Prozesses des Heraufstufens nur wichtige und kritische Daten repliziert werden sollen (Yes) oder ob alle Daten repliziert werden sollen (No).
DatabasePath = Ppfad	Legt den Pfad zur Active Directory-Datenbank fest.
DisableCancelForDnsInstall = Yes No	Legt fest, ob die Schaltfläche ABBRECHEN während der DNS-Installation deaktiviert werden soll (Yes) oder nicht (No).
DNSOnNetwork = Yes No	Legt fest, dass der DNS-Client automatisch konfiguriert werden soll (Yes) oder nicht (No).
DomainNetBiosName = NetBIOSName	Legt den NetBIOS-Namen der Domäne fest.

Tabelle 87: Die Parameter für die Dcpromo-Antwortdatei

Parameter	Beschreibung
IsLastDCInDomain = Yes No	Legt fest, ob dies der letzte Domänen-Controller der Domäne ist (Yes) oder nicht (No). Dieser Parameter wird nur beim Herunterstufen eines Domänen-Controllers zu einem Member- bzw. Standalone-Server benötigt.
LogPath = Pfad	Legt den Pfad für die Logdateien fest.
NewDomain = Tree Child Forest	Legt den Typ der neuen Domäne fest.
NewDomainDNSName = DNSName	Legt den DNS-Namen für die neue Domäne fest.
ParentDomainDNSName = DNSName	Legt den DNS-Namen der übergeordneten Domäne fest, wenn eine neue Child-Domäne eingerichtet wird.
Password = Kennwort	Legt das Kennwort des Benutzers fest, der über ausreichende Berechtigungen zum Heraufstufen des Standalone- oder Member-Servers zu einem Domänen-Controller verfügt.
RebootOnSuccess = Yes No	Legt fest, ob das Computersystem im Anschluss automatisch neu gestartet werden soll (Yes) oder nicht (No).
RemoveApplicationPartitions = Yes NO	Legt fest, ob die Anwendungspartition nach dem Herabstufen eines Domänen-Controllers gelöscht werden soll (Yes) oder nicht (No).
ReplicaDomainDNSName = DNSName	Legt den DNS-Namen der Domäne fest, die als Replikationsquelle genutzt werden soll.
ReplicaOrMember = Replica Member	Legt fest, ob ein NT-System, das ein Upgrade erfährt, zu einem Domänen-Controller konvertiert (Replica) oder zu einem Member-Server (Member) herabgestuft werden soll.
ReplicaOrNewDomain = Replica Domain	Legt fest, ob der Domänen-Controller als zusätzlicher Domänen-Controller einer bereits vorhandenen Domäne (Replica) oder als erster Domänen-Controller einer neuen Domäne (Domain) eingerichtet werden soll.
ReplicationSourceDC = DNSName	Legt den vollständigen DNS-Namen des Domänen-Controllers fest, von dem die Domäneninformationen repliziert werden sollen.
ReplicationSourcePath = Pfad	Legt den Pfad der Dateien fest, die für das Heraufstufen zu einem Domänen-Controller verwendet werden sollen.
SafeModeAdminPassword = Kennwort	Legt das Kennwort für den Wiederherstellungsmodus fest.

Tabelle 87: Die Parameter für die Dcpromo-Antwortdatei (Forts.)

Parameter	Beschreibung
SetForestVersion = Yes No	Legt fest, ob die Gesamtstrukturfunktionsebene angehoben (Yes) werden soll oder nicht (No).
SiteName = StandortName	Legt den Namen des Standorts fest.
Syskey = Key	Legt den Syskey fest.
SysVolPath = Pfad	Legt den Pfad des SysVol-Ordners fest.
TreeOrChild = Tree Child	Legt fest, ob die Domäne als Root-Domäne einer neuen Gesamtstruktur (Tree) oder als Child-Domäne einer bestehenden Domänen-Struktur (Child) eingerichtet werden soll.
UserDomain = Domänenname	Legt den Namen der Domäne fest, in der das Benutzerkonto, das für diesen Vorgang genutzt wird, eingerichtet wurde.
UserName = Benutzername	Legt den Namen des zu verwendenden Benutzerkontos fest.

Tabelle 87: Die Parameter für die Dcpromo-Antwortdatei (Forts.)

Reguläre Ausdrücke (VBScript)

Nachfolgend finden Sie eine Aufstellung der regulären Ausdrücke unter Angabe einer kurzen Beschreibung, die Sie in VBScript verwenden können.

Regulärer Ausdruck	Beschreibung
\	Markiert das folgende Zeichen als Sonderzeichen oder als wortgenaues Zeichen
^	Entspricht dem Zeilenanfang
\$	Entspricht dem Zeilenende
*	Entspricht dem vorherigen Zeichen kein oder mehrere Male
+	Entspricht dem vorherigen Zeichen ein oder mehrere Male
?	Entspricht dem vorherigen Zeichen nicht oder einmal
.	Entspricht allen Zeichen mit Ausnahme des Zeilenumbruchzeichens
(Muster)	Entspricht dem angegebenen Zeichenmuster
x y	Entspricht entweder <i>x</i> oder <i>y</i>
{n}	Entspricht <i>n</i> Mal
{n,}	Entspricht mindestens <i>n</i> Mal
{n,m}	Entspricht mindestens <i>n</i> und höchstens <i>m</i> Mal

Tabelle 88: Die regulären Ausdrücke von VBScript (Quelle: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/42add7d6-4490-4d51-a5a7-5d654a317b07.asp>)

Regulärer Ausdruck	Beschreibung
[xyz]	Entspricht einem beliebigen Zeichen im angegebenen Bereich
[^xyz]	Entspricht nicht einem beliebigen Zeichen im angegebenen Bereich
[a-z]	Entspricht allen Zeichen im angegebenen Bereich
[^m-z]	Entspricht nicht allen Zeichen im angegebenen Bereich
\b	Entspricht einer Wortgrenze
\B	Entspricht nicht einer Wortgrenze
\d	Entspricht einer Ziffer
\D	Entspricht nicht einer Ziffer
\f	Entspricht dem Seitenvorschubzeichen
\n	Entspricht dem Zeilenumbruchzeichen
\r	Entspricht dem Wagenrücklaufzeichen
\s	Entspricht beliebigen Leerräumen (Leerzeichen, Tabulator ...)
\S	Entspricht nicht beliebigen Leerräumen
\t	Entspricht dem Tab-Zeichen
\v	Entspricht einem vertikalen Tab-Zeichen
\w	Entspricht allen alphabetischen Zeichen (inkl. Unterstrich)
\W	Entspricht nicht allen alphabetischen Zeichen (inkl. Unterstrich)
\Zahl	Entspricht <i>Zahl</i> , wenn <i>Zahl</i> eine Ganzzahl ist
\n	Entspricht <i>n</i> , wobei <i>n</i> ein oktaler Escape-Wert ist
\xn	Entspricht <i>n</i> , wobei <i>n</i> ein hexadezimaler Escape-Wert ist

Tabelle 88: Die regulären Ausdrücke von VBScript (Quelle: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/42add7d6-4490-4d51-a5a7-5d654a317b07.asp>) (Forts.)

Die Eigenschaften der Win32_WindowsProductActivation-Klasse

Die nachfolgende Tabelle enthält die Eigenschaften der Win32_WindowsProductActivation-Klasse, unter Angabe einer kurzen Beschreibung.

Eigenschaft	Beschreibung
ActivationRequired	Enthält den Wert 1, wenn das Produkt aktiviert werden muss, anderenfalls ist der Wert gleich 0.
Caption	Kurze Beschreibung
Description	Beschreibung
IsNotificationOn	Enthält den Wert 0, wenn alle Erinnerungsfunktionen (Info-Balloons) zur anstehenden Produktaktivierung deaktiviert sind, anderenfalls ist der Wert ungleich 0.
ProductID	Produkt-ID von Windows Server 2003
RemainingEvaluationPeriod	Enthält die Anzahl der noch verbleibenden Tage bis die Test- oder Betaversion ungültig wird und lizenziert werden muss.
RemainingGracePeriod	Enthält die Anzahl der noch verbleibenden Tage bis die Produktaktivierung durchgeführt werden muss.
ServerName	Hostname des Computersystems, auf dem die Produktaktivierung vorgenommen werden muss.
SettingID	ID des WPA-Objekts

Tabelle 89: Die Eigenschaften der WMI-Klasse »Win32_WindowsProductActivation«

Die Eigenschaften des ADSI-User-Objekts

Die nachfolgende Tabelle enthält die Eigenschaften eines ADSI-User-Objekts (IADs-User) unter Angabe einer kurzen Beschreibung und zeigt, welche dieser Eigenschaften von WinNT- und LDAP-Providern unterstützt werden.

Eigenschaft	WinNT	LDAP	Beschreibung
AccountDisabled	•	•	Legt fest, ob das angegebene Benutzerkonto aktiviert (no) oder deaktiviert (yes) ist.
AccountExpirationDate	•	•	Datum, an dem das Benutzerkonto seine Gültigkeit verliert
BadLoginAddress		•	Liefert die Adresse des Computersystems zurück, über das die letzte fehlerhafte Anmeldung durchgeführt wurde.
BadLoginCount	•	•	Liefert die Anzahl der fehlerhaften Anmeldungen zurück.

Tabelle 90: Die Eigenschaften des ADSI-User-Objekts

Eigenschaft	WinNT	LDAP	Beschreibung
ChangePassword	•	•	Überschreibt das aktuelle Kennwort mit einem neuen Kennwort.
Department		•	Abteilung des Benutzers
Description	•	•	Kurze Beschreibung zum Benutzerkonto
Division		•	Abteilung des Benutzers
EmailAddress		•	E-Mail-Adresse des Benutzers
EmployeeID		•	Personalnummer des Benutzers
FaxNumber		•	Faxnummer des Benutzers
FirstName		•	Vorname des Benutzers
FullName	•	•	Vollständiger Name des Benutzers (z.B. Vor- und Nachname)
GraceLoginsAllowed		•	Anzahl der möglichen Anmeldungen, nachdem ein Kennwort abgelaufen ist
GraceLoginsRemaining		•	Anzahl der möglichen Anmeldungen, bevor ein Benutzerkonto gesperrt wird
Groups	•	•	Gruppenmitgliedschaften des Benutzers
HomeDirectory	•	•	Pfad für das Homelaufwerk
HomePage		•	Homepage des Benutzers
IsAccountLocked	•	•	Gibt an, ob das Benutzerkonto gesperrt ist (<i>True</i>) oder nicht (<i>False</i>).
Languages		•	Gibt den Ländercode an, der verwendet werden soll.
LastFailedLogin		•	Liefert das Datum und auch die Uhrzeit der letzten fehlerhaften Anmeldung zurück.
LastLogin	•	•	Liefert das Datum und auch die Uhrzeit der letzten erfolgreichen Anmeldung zurück.
LastLogoff	•	•	Liefert das Datum und auch die Uhrzeit der letzten Abmeldung zurück.
LastName		•	Nachname des Benutzers
LoginHours	•	•	Legt fest, zu welchen Uhrzeiten eine Anmeldung erlaubt ist.
LoginScript	•	•	Pfad zum Anmeldeskript
LoginWorkstations	•	•	Legt fest, von welchen Computersystemen aus eine Anmeldung erlaubt ist.
Manager		•	Manager des Benutzers
MaxLogins	•	•	Legt die maximale Anzahl der gleichzeitigen Anmeldungen fest.

Tabelle 90: Die Eigenschaften des ADSI-User-Objekts (Forts.)

Eigenschaft	WinNT	LDAP	Beschreibung
MaxStorage	•	•	Festplattenkontingent des Benutzers
NamePrefix		•	Namenszusatz
NameSuffix		•	Namenszusatz
OfficeLocations		•	Bürostandort des Benutzers
OtherName		•	Initialen des zweiten Vornamens
PasswordExpirationDate	•	•	Legt das Datum und die Uhrzeit fest, wann das Kennwort seine Gültigkeit verliert.
PasswordLastChanged		•	Liefert das Datum und auch die Uhrzeit der letzten Kennwortänderung zurück.
PasswordMinimum-Length	•	•	Gibt die minimale Kennwortlänge an.
PasswordRequired	•	•	Legt fest, ob ein Kennwort erforderlich ist (<i>True</i>) oder nicht (<i>False</i>).
Picture		•	Legt fest, ob für den Benutzer ein Bild hinterlegt wurde.
PostalAddresses		•	Postalische Adresse des Benutzers
PostalCodes		•	Postleitzahl des Benutzers
Profile	•	•	Pfad zum Benutzerprofil
RequireUniquePassword		•	Legt fest, ob ein Kennwort eindeutig sein muss (<i>True</i>) oder nicht (<i>False</i>).
SeeAlso		•	Verweise auf andere Objekte
SetPassword	•	•	Kennwort für den Benutzer
TelephoneHome		•	Private Telefonnummer des Benutzers
TelephoneMobile		•	Mobilfunknummer des Benutzers
TelephoneNumber		•	Telefonnummer des Benutzers
TelephonePager		•	Pager-Telefonnummer des Benutzers
Title		•	Titel des Benutzers

Tabelle 90: Die Eigenschaften des ADSI-User-Objekts (Forts.)

Die Eigenschaften der Win32_OperatingSystem-Klasse

Die nachfolgende Tabelle enthält die Eigenschaften der Win32_OperatingSystem-Klasse unter Angabe einer kurzen Beschreibung.

Eigenschaft	Beschreibung
BootDevice	Name der Partition, von der das Betriebssystem gebootet wird
BuildNumber	Build-Nummer des Betriebssystems
BuildType	Typ des Betriebssystems-Build
Caption	Kurze Beschreibung zum Betriebssystem
CodeSet	Verwendete Codepage
CountryCode	Verwendeter Ländercode
CreationClassName	Name der Klasse, über die das Betriebssystem-Objekt erstellt wurde (Win32_OperatingSystem)
CSCreationClassName	Name der Klasse, über die das Computersystem-Objekt erstellt wurde (Win32_ComputerSystem)
CSDVersion	Bezeichnung des zuletzt installierten Service Pack
CSName	Name des Computersystems
CurrentTimeZone	Verwendete Zeitzone
DataExecutionPrevention_32BitApplications	Diese Eigenschaft wird unter Windows Server 2003 nicht unterstützt.
DataExecutionPrevention_Available	Diese Eigenschaft wird unter Windows Server 2003 nicht unterstützt.
DataExecutionPrevention_Drivers	Diese Eigenschaft wird unter Windows Server 2003 nicht unterstützt.
Debug	Enthält den Wert <i>True</i> , wenn die Debugging-Version der Datei <i>User.exe</i> installiert wurde, anderenfalls ist der Wert gleich <i>False</i> .
Description	Beschreibung zum Betriebssystem
Distributed	Enthält den Wert <i>True</i> , wenn das Computersystem Teil eines Clusters ist, anderenfalls ist der Wert gleich <i>False</i> .
EncryptionLevel	Transaktionsverschlüsselungslevel
ForegroundApplicationBoost	Priorität einer Anwendung (0 = None, 1 = Minimum, 2 = Maximum)
FreePhysicalMemory	Größe des freien und noch ungenutzten Speichers in Kbyte.

Tabelle 91: Die Eigenschaften der WMI-Klasse »Win32_OperatingSystem«

Eigenschaft	Beschreibung
FreeSpaceInPagingFiles	Größe des noch freien Speichers in der Auslagerungsdatei in Kbyte.
FreeVirtualMemory	Größe des noch freien virtuellen Speichers in Kbyte.
InstallDate	Datum, wann das Objekt eingerichtet wurde.
LargeSystemCache	Legt fest, ob die Systemleistung für Anwendungen oder Hintergrunddienste optimiert werden soll (0 = Applications, 1 = System).
LastBootupTime	Zeitpunkt, zu dem das Betriebssystem zuletzt gestartet wurde.
LocalDateTime	Systemdatum/-uhrzeit
Locale	ID des verwendeten Ländercodes
Manufacturer	Bezeichnung des Betriebssystemherstellers
MaxNumberOfProcesses	Maximale Anzahl der Prozesse, die betriebssystemseitig unterstützt werden
MaxProcessMemorySize	Maximale Speichernutzung einzelner Prozesse in Kbytes.
Name	Name des Objekts bzw. des Betriebssystems
NumberOfLicensedUsers	Anzahl der Benutzerlizenzen
NumberOfProcesses	Anzahl der aktiven Prozesse
NumberOfUsers	Anzahl der Benutzer, für die das Betriebssystem Statusinformationen vorhält.
Organization	Name der Organisation, für die das Betriebssystem registriert wurde.
OSLanguage	Sprachversion-ID des Betriebssystems (z.B. 1031 = German, 1033 = English (US))
OSProductSuite	ID der installierten Server-Add Ons (1 = Small Business, 2 = Enterprise, 4 = BackOffice, 8 = Communication Server, 16 = Terminal Server, 32 = Small Business (Restricted), 64 = Embedded NT, 128 = Data Center)
OSType	Betriebssystemtyp (z.B. 17 = Win98, 18 = WinNT)
OtherTypeDescription	Zusätzliche Beschreibung der Betriebssystemversion (z.B. R2 für den Release Candidate 2)
PAEEnabled	Enthält den Wert <i>True</i> , wenn PAE (Physical Address Extension) aktiviert ist, anderenfalls ist der Wert gleich <i>False</i> .
PlusProductID	ID des Windows Plus!-Pakets, falls installiert
PlusVersionNumber	Version des Windows Plus!-Pakets, falls installiert

Tabelle 91: Die Eigenschaften der WMI-Klasse »Win32_OperatingSystem« (Forts.)

Eigenschaft	Beschreibung
Primary	Enthält den Wert <i>True</i> , wenn das Betriebssystem das primäre Betriebssystem ist, anderenfalls ist der Wert gleich <i>False</i> .
ProductType	Zusätzliche Information zum Produkttyp (1 = Work Station, 2 = Domain Controller, 3 = Server)
QuantumLength	Diese Eigenschaft wird unter Windows Server 2003 nicht genutzt.
QuantumType	Diese Eigenschaft wird unter Windows Server 2003 nicht genutzt.
RegisteredUser	Name des Benutzers, für den das Betriebssystem registriert wurde
SerialNumber	Seriennummer des Betriebssystems
ServicePackMajorVersion	Höchste Service Pack-Versionsnummer, die installiert ist
ServicePackMinorVersion	Kleinste Service Pack-Versionsnummer, die installiert ist
SizeStoredInPagingFiles	Nutzbare Größe der Auslagerungsdatei in Kbytes (nicht identisch mit der eigentlichen Größe der Auslagerungsdatei)
Status	Status des Betriebssystemobjektes. Mögliche Werte sind <i>OK</i> , <i>Error</i> , <i>Degraded</i> , <i>Unknown</i> , <i>Pred Fail</i> , <i>Starting</i> , <i>Stopping</i> , <i>Service</i> .
SuiteMask	Bit-Muster, worüber der Server-Typ und insbesondere zusätzliche Server-Add Ons identifiziert werden können (1 = Small Business, 2 = Enterprise, 4 = Back Office, 8 = Communications, 16 = Terminal, 32 = Small Business Restricted, 64 = Embedded NT, 128 = Data Center, 256 = Single User, 512 Personal, 1024 = Blade)
SystemDevice	Partition, auf der das Betriebssystem installiert ist.
SystemDirectory	Systemordner
SystemDrive	Systemlaufwerk
TotalSwapSpaceSize	Größe des Swapfile in Kbytes
TotalVirtualMemorySize	Größe des virtuellen Speichers in Kbytes
TotalVisibleMemorySize	Größe des vom Betriebssystem erkannten Speichers
Version	Betriebssystemversion
WindowsDirectory	Windows-Ordner

Tabelle 91: Die Eigenschaften der WMI-Klasse »Win32_OperatingSystem« (Forts.)

Die Eigenschaften der Win32_UserAccount-Klasse

Die nachfolgende Tabelle enthält die Eigenschaften der Win32_UserAccount-Klasse, unter Angabe einer kurzen Beschreibung.

Eigenschaft	Beschreibung
AccountType	Typ des Benutzerkontos (256 = UF_TEMP_DUPLICATE_ACCOUNT, 512 = UF_NORMAL_ACCOUNT, 2048 = UF_INTERDOMAIN_TRUST_ACCOUNT, 4096 = UF_WORKSTATION_TRUST_ACCOUNT, 8192 = UF_SERVER_TRUST_ACCOUNT)
Caption	Kurze Beschreibung zum Benutzerkonto
Description	Beschreibung des Benutzerkontos
Disabled	Enthält den Wert <i>True</i> , wenn das Benutzerkonto deaktiviert ist, anderenfalls ist der Wert gleich <i>False</i> .
Domain	Name der Domäne, in der das Benutzerkonto angelegt wurde.
FullName	Vollständiger Name des Benutzers
InstallDate	Datum und Uhrzeit, wann das Benutzerkonto-Objekt angelegt wurde
LocalAccount	Enthält den Wert <i>True</i> , wenn das Benutzerkonto ein lokales Benutzerkonto ist, anderenfalls ist der Wert gleich <i>False</i> .
Lockout	Enthält den Wert <i>True</i> , wenn das Benutzerkonto gesperrt ist (z.B. nach mehrmaligem falschen Anmelden), anderenfalls ist der Wert gleich <i>False</i> .
Name	Name des Benutzerkontos in der in <i>Domain</i> hinterlegten Domäne
PasswordChangeable	Enthält den Wert <i>True</i> , wenn der Benutzer das Kennwort ändern kann, anderenfalls ist der Wert gleich <i>False</i> .
PasswordExpires	Enthält den Wert <i>True</i> , wenn das Kennwort abgelaufen ist und erneuert werden muss. Anderenfalls ist der Wert gleich <i>False</i> .
PasswordRequired	Enthält den Wert <i>True</i> , wenn das Benutzerkonto kennwortgeschützt sein muss, anderenfalls ist der Wert gleich <i>False</i> .
SID	Enthält die SID des Benutzerkontos
SIDType	Typ der SID (1 = SIDTypeUser, 2 = SIDTypeGroup, 3 = SIDTypeDomain, 4 = SIDTypeAlias, 5 = SIDTypeWellknownGroup, 6 = SIDTypeDeletedAccount, 7 = SIDTypeInvalid, 8 = SIDTypeUnknown, 9 = SIDTypeComputer)
Status	Status des Benutzerkontos. Mögliche Werte sind <i>OK</i> , <i>Error</i> , <i>Degraded</i> , <i>Unknown</i> , <i>Pred Fail</i> , <i>Starting</i> , <i>Stopping</i> , <i>Service</i> , <i>Stressed</i> , <i>NonRecover</i> , <i>No Contact</i> , <i>Lost Comm</i> .

Tabelle 92: Die Eigenschaften der WMI-Klasse »Win32_UserAccount«

Die Eigenschaften der Win32_Group-Klasse

Die nachfolgende Tabelle enthält die Eigenschaften der Win32_Group-Klasse, unter Angabe einer kurzen Beschreibung.

Eigenschaft	Beschreibung
Caption	Kurze Beschreibung zum Gruppenkonto (einzeilig)
Description	Beschreibung des Gruppenkontos
Domain	Name der Domäne, in der das Gruppenkonto angelegt wurde
InstallDate	Datum und Uhrzeit, wann das Gruppenkonto-Objekt angelegt wurde
LocalAccount	Enthält den Wert <i>True</i> , wenn das Gruppenkonto ein lokales Gruppenkonto ist, anderenfalls ist der Wert gleich <i>False</i> .
Name	Name des Gruppenkontos in der in <i>Domain</i> hinterlegten Domäne
SID	SID des Gruppenkontos
SIDType	Typ der SID (1 = SIDTypeUser, 2 = SIDTypeGroup, 3 = SIDTypeDomain, 4 = SIDTypeAlias, 5 = SIDTypeWellknownGroup, 6 = SIDTypeDeletedAccount, 7 = SIDTypeInvalid, 8 = SIDTypeUnknown, 9 = SIDTypeComputer)
Status	Status des Gruppenkontos. Mögliche Werte sind <i>OK</i> , <i>Error</i> , <i>Degraded</i> , <i>Unknown</i> , <i>Pred Fail</i> , <i>Starting</i> , <i>Stopping</i> , <i>Service</i> , <i>Stressed</i> , <i>NonRecover</i> , <i>No Contact</i> , <i>Lost Comm</i> .

Tabelle 93: Die Eigenschaften der WMI-Klasse »Win32_Group«

Die Eigenschaften der CIM_DataFile- und Win32_Directory-Klasse

Die nachfolgende Tabelle enthält die Eigenschaften der CIM_DataFile- und Win32_Directory-Klasse, unter Angabe einer kurzen Beschreibung.

Eigenschaft	Beschreibung
AccessMask	Enthält die NTFS-Berechtigung der Datei oder des Ordners in Form eines Bit-Musters (0 = <i>File_Read_Data</i> bzw. <i>File_List_Directory</i> , 1 = <i>File_Write_Data</i> bzw. <i>File_Add_File</i> , 4 = <i>File_Append_Data</i> bzw. <i>File_Add_Subdirectory</i> , 8 = <i>File_Read_EA</i> , 16 = <i>File_Write_EA</i> , 32 = <i>File_Execute</i> bzw. <i>File_Traverse</i> , 64 = <i>File_Delete_Child</i> , 128 = <i>File_Read_Attributes</i> , 256 = <i>File_Write_Attributes</i> , 65536 = <i>Delete</i> , 131072 = <i>Read_Control</i> , 262144 = <i>Write_DAC</i> , 524288 = <i>Write_Owner</i> , 1048576 = <i>Synchronize</i>).
Archive	Enthält den Wert <i>True</i> , wenn das Archivattribut gesetzt ist, anderenfalls ist der Wert gleich <i>False</i> .
Caption	Kurze Beschreibung zur Datei oder zum Ordner

Tabelle 94: Die Eigenschaften der CIM_DataFile- und Win32_Directory-Klasse

Eigenschaft	Beschreibung
Compressed	Komprimierungsattribut. Diese Eigenschaft enthält den Wert <i>True</i> , wenn die Datei oder der Ordner komprimiert ist, anderenfalls ist der Wert <i>False</i> .
CompressionMethod	Enthält die Methode oder das Tool, womit die Datei oder der Ordner komprimiert wurde. Die Eigenschaft enthält den Wert <i>Compressed</i> oder <i>Unknown</i> , wenn die Methode oder das Tool nicht ermittelt werden konnte.
CreationClassName	Name der Klasse, durch die das Objekt erstellt wurde
CreationDate	Datum und Uhrzeit des Erstellungszeitpunkts
CSCreationClassName	Name der Klasse des Computersystems
CSName	Name des Computersystems
Description	Beschreibung zur Datei oder zum Ordner
Drive	Laufwerksbuchstabe des Pfads
EightDotThreeFileName	Datei- bzw. Ordnername nach der 8.3-Konvention
Encrypted	Verschlüsselungsattribut. Die Eigenschaft besitzt den Wert <i>True</i> , wenn die Datei oder der Ordner verschlüsselt ist, anderenfalls ist der Wert gleich <i>False</i> .
EncryptionMethod	Enthält die Methode oder das Tool, mit der bzw. dem die Datei oder der Ordner verschlüsselt wurde. Die Eigenschaft enthält <i>Unknown</i> , wenn die Methode oder das Tool nicht ermittelt werden konnte.
Extension	Suffix der Datei
FileName	Name der Datei
FileSize	Größe der Datei
FileType	Typ der Datei
FSCreationClassName	Name der Klasse des Dateisystems
FSName	Bezeichnung des Dateisystems
Hidden	Hidden-Attribut. Diese Eigenschaft besitzt den Wert <i>True</i> , wenn die Datei oder der Ordner versteckt ist, ansonsten ist der Wert gleich <i>False</i> .
InstallDate	Datum und Uhrzeit des Einrichtungszeitpunkts
InUseCount	Anzahl der gleichzeitigen Zugriffe, die gerade auf die Datei oder auf den Ordner stattfinden
LastAccessed	Datum und Uhrzeit des letzten Zugriffs
LastModified	Datum und Uhrzeit des letzten Änderungszeitpunkts
Manufacturer	Hersteller der Datei oder des Ordners
Name	Pfad und Datei-/Ordnername

Tabelle 94: Die Eigenschaften der CIM_DataFile- und Win32_Directory-Klasse (Forts.)

Eigenschaft	Beschreibung
Path	Pfad der Datei bzw. des Ordners
Readable	Enthält den Wert <i>True</i> , wenn die Datei oder der Ordner geöffnet werden kann, anderenfalls ist der Wert gleich <i>False</i> .
Status	Gibt Auskunft über den Status der Datei oder des Ordners. Mögliche Werte sind <i>OK</i> , <i>Error</i> , <i>Degraded</i> , <i>Unknown</i> , <i>Pred Fail</i> , <i>Starting</i> , <i>Stopping</i> , <i>Service</i> , <i>Stressed</i> , <i>NonRecover</i> , <i>No Contact</i> , <i>Lost Comm.</i>
System	Enthält den Wert <i>True</i> , wenn die Datei eine Systemdatei ist, anderenfalls ist der Wert gleich <i>False</i> .
Version	Gibt Auskunft (falls vorhanden) über die Datei- bzw. Ordner-version
Writeable	Enthält den Wert <i>True</i> , wenn die Datei oder der Ordner verändert werden kann, anderenfalls ist der Wert <i>False</i> .

Tabelle 94: Die Eigenschaften der CIM_DataFile- und Win32_Directory-Klasse (Forts.)

Die Eigenschaften der Win32_LogicalDisk-Klasse

Die nachfolgende Tabelle enthält die Eigenschaften der Win32_LogicalDisk-Klasse unter Angabe einer kurzen Beschreibung.

Eigenschaft	Beschreibung
Access	Unterstützter Zugriffsmodus für den Datenträger (0 = <i>Unknown</i> , 1 = <i>Readable</i> , 2 = <i>Writeable</i> , 3 = <i>Read/Write supported</i> , 4 = <i>Write Once</i>)
Availability	Datenträgerverfügbarkeit und -status (1 = <i>Other</i> , 2 = <i>Unknown</i> , 3 = <i>Running/Full Power</i> , 4 = <i>Warning</i> , 5 = <i>In Test</i> , 6 = <i>Not Applicable</i> , 7 = <i>Power off</i> , 8 = <i>Off Line</i> , 9 = <i>Off Duty</i> , 10 = <i>Degraded</i> , 11 = <i>Not Installed</i> , 12 = <i>Install Error</i> , 13 = <i>Power Save – Unknown</i> , 14 = <i>Power Save – Low Power Mode</i> , 15 = <i>Power Save – Standby</i> , 16 = <i>Power Cycle</i> , 17 = <i>Power Save – Warning</i>)
BlockSize	Blockgröße des Datenträgers
Caption	Kurze Beschreibung des Datenträgers
Compressed	Enthält den Wert <i>False</i> , wenn eine dateisystemspezifische Komprimierung unterstützt wird, anderenfalls ist der Wert gleich <i>True</i> .
ConfigManagerErrorCode	Enthält im Falle eines Datenträgerfehlers den Error-Code
ConfigManagerUserConfig	Enthält den Wert <i>True</i> , falls der Datenträger über eine benutzerspezifische Konfiguration verfügt, anderenfalls ist der Wert <i>False</i> .
CreationClassName	Name der Klasse, mit der dieses Objekt erstellt wurde

Tabelle 95: Die Eigenschaften der WMI-Klasse »Win32_LogicalDisk«

Eigenschaft	Beschreibung
Description	Beschreibung des Datenträgers
DeviceID	Eindeutige Datenträger-ID
DriveTyp	ID des Datenträgertyps (0 = <i>Unknown</i> , 1 = <i>No Root Directory</i> , 2 = <i>Removable Disk</i> , 3 = <i>Local Disk</i> , 4 = <i>Network Drive</i> , 5 = <i>Compact Disk</i> , 6 = <i>RAM Disk</i>)
ErrorCleared	Enthält den Wert <i>True</i> , wenn der Wert aus <i>LastErrorCode</i> zurückgesetzt wurde.
ErrorDescription	Enthält erweiterte Informationen zu dem Fehler, der in <i>LastErrorCode</i> reportet wurde
ErrorMethodology	Typ bzw. Methode der Fehlererkennung
FileSystem	Verwendetes Dateisystem
FreeSpace	Verfügbarer Speicherplatz auf dem Datenträger
InstallDate	Datum und Uhrzeit, wann das Datenträgerobjekt erstellt wurde
LastErrorCode	Letzter Fehlercode
MaximumComponentLength	Maximale Dateinamenlänge, die unterstützt wird bzw. verwendet werden darf
MediaType	Typ des Datenträgers (0 = <i>Unknown</i> , 1 – 10 = <i>Disketten-Laufwerkstyp</i> , 11 = <i>Removable Media</i> , 12 = <i>Festplatte</i> , 13 – 22 <i>Disketten-Laufwerkstyp</i>)
Name	Datenträgername
NumberofBlocks	Anzahl der vorhandenen Blöcke
PNPDeviceID	Enthält die Plug&Play-Geräte-ID
PowerManagementCapabilities	Enthält eine Aufstellung der Power-Management-Funktionen, die seitens des Datenträgers unterstützt werden (0 = <i>Unknown</i> , 1 = <i>Not Supported</i> , 2 = <i>Disabled</i> , 3 = <i>Enabled</i> , 4 = <i>Power Saving Modes Entered Automatically</i> , 5 = <i>Power State Settable</i> , 6 = <i>Power Cycling Supported</i> , 7 = <i>Timed Power On Supported</i>)
PowerManagementSupported	Enthält den Wert <i>True</i> , wenn der Datenträger in einen Energiesparmodus versetzt werden kann.
ProviderName	Netzwerkpfad für den logischen Datenträger
Purpose	Zusätzliche Beschreibung für den Datenträger
QuotasDisabled	Enthält den Wert <i>True</i> , wenn die Kontingentverwaltung deaktiviert ist.
QuotasIncomplete	Enthält den Wert <i>True</i> , wenn die Kontingentverwaltung genutzt wird, aber vorübergehend deaktiviert wurde.

Tabelle 95: Die Eigenschaften der WMI-Klasse »Win32_LogicalDisk« (Forts.)

Eigenschaft	Beschreibung
QuotasRebuilding	Enthält den Wert <i>True</i> , wenn die Kontingentverwaltung aktiviert wurde und die quotaspezifischen Informationen zusammengestellt werden.
Size	Größe bzw. Gesamtkapazität des Datenträgers
Status	Aktueller Datenträgerstatus (<i>Ok</i> , <i>Error</i> , <i>Degraded</i> , <i>Unknown</i> , <i>Pred Fail</i> , <i>Starting</i> , <i>Stopping</i> , <i>Service</i> , <i>Stressed</i> , <i>NonRecover</i> , <i>No Contact</i> , <i>Lost Comm</i>)
StatusInfo	Information zum Datenträgerstatus (1 = <i>Other</i> , 2 = <i>Unknown</i> , 3 = <i>Enabled</i> , 4 = <i>Disabled</i> , 5 = <i>Not Applicable</i>)
SupportsDiskQuotas	Enthält den Wert <i>True</i> , wenn der Datenträger (oder zumindest die aktuelle Partition) eine Kontingentverwaltung unterstützt.
SupportsFileBasedCompression	Enthält den Wert <i>True</i> , wenn die Partition oder das Volume eine dateibasierende Kompression unterstützt.
SystemCreationClassName	Name der Klasse, mit der das Computersystemobjekt erstellt wurde
SystemName	Aktueller Computername
VolumeDirty	Enthält den Wert <i>True</i> , wenn das Dirty Bit-Flag gesetzt ist und mit dem nächsten Computerneustart eine Dateisystemüberprüfung (<i>Chkdsk</i>) durchgeführt wird.
VolumeName	Name der Partition bzw. des Volume
VolumeSerialNumber	Seriennummer des Datenträgers, der Partition oder des Volume

Tabelle 95: Die Eigenschaften der WMI-Klasse »Win32_LogicalDisk« (Forts.)

Die Eigenschaften der Win32_DiskPartition-Klasse

Die nachfolgende Tabelle enthält die Eigenschaften der Win32_DiskPartition-Klasse unter Angabe einer kurzen Beschreibung, die Sie mit Hilfe der WMI-Technologie von der Eingabeaufforderung aus und auch aus einem VBScript heraus abfragen können.

Eigenschaft	Beschreibung
Access	Zugriffsmodus der Partition (0 = <i>Unknown</i> , 1 = <i>Readable</i> , 2 = <i>Writeable</i> , 3 = <i>Read/Write supported</i> , 4 = <i>Write Once</i>)
Availability	Status und Verfügbarkeit der Partition
BlockSize	Blockgröße der Partition
Bootable	Enthält den Wert <i>True</i> , wenn von dieser Partition aus gebootet werden kann.

Tabelle 96: Die Eigenschaften der WMI-Klasse »Win32_DiskPartition«

Eigenschaft	Beschreibung
BootPartition	Enthält den Wert <i>True</i> , wenn die Partition als aktiv markiert ist und das Computersystem von dieser Partition aus gebootet wird.
Caption	Kurze Partitionsbezeichnung
ConfigManagerErrorCode	Betriebssystemseitiger Fehlercode des Konfigurationsmanagers
ConfigManagerUserConfig	Enthält den Wert <i>True</i> , wenn die Partition eine benutzerdefinierte Konfiguration enthält.
CreationClassName	Name der Klasse, mit der das Partitionsobjekt erstellt wurde
Description	Beschreibung zur Partition
DeviceID	Eindeutige Geräteerkennung
DiskIndex	Nummer des Datenträgers, auf dem das Partitionsobjekt angelegt ist
ErrorCleared	Enthält den Wert <i>True</i> , wenn der Wert aus <i>LastErrorCode</i> zurückgesetzt wurde.
ErrorDescription	Enthält erweiterte Informationen zu dem Fehler, der in <i>LastErrorCode</i> reportet wurde.
ErrorMethodology	Typ bzw. Methode der Fehlererkennung
HiddenSectors	Anzahl der versteckten Sektoren auf der Partition
Index	Nummer der Partition auf dem Datenträger
InstallDate	Datum und Uhrzeit, wann das Partitionsobjekt angelegt wurde
LastErrorCode	Letzter Fehlercode
Name	Name der Partition
NumberOfBlocks	Anzahl der vorhandenen Blöcke
PNPDeviceID	Enthält die Plug and Play-Geräte-ID
PowerManagementCapabilities	Enthält eine Aufstellung der Power-Management-Funktionen, die seitens des Datenträgers unterstützt werden (0 = <i>Unknown</i> , 1 = <i>Not Supported</i> , 2 = <i>Disabled</i> , 3 = <i>Enabled</i> , 4 = <i>Power Saving Modes Entered Automatically</i> , 5 = <i>Power State Settable</i> , 6 = <i>Power Cycling Supported</i> , 7 = <i>Timed Power On Supported</i>)
PowerManagementSupported	Enthält den Wert <i>True</i> , wenn der Datenträger in einen Energiesparmodus versetzt werden kann.
PrimaryPartition	Enthält den Wert <i>True</i> , wenn die Partition eine primäre Partition ist.
Purpose	Zusätzliche Beschreibung für den Datenträger

Tabelle 96: Die Eigenschaften der WMI-Klasse »Win32_DiskPartition« (Forts.)

Eigenschaft	Beschreibung
RewritePartition	Enthält den Wert <i>True</i> , wenn sich die Partitionsinformationen geändert haben und die Informationen aktualisiert werden müssen.
Size	Gesamtkapazität der Partition
StartingOffset	Gibt Auskunft über den Offset, an dem die Partition beginnt
Status	Aktueller Datenträgerstatus (<i>Ok, Error, Degraded, Unknown, Pred Fail, Starting, Stopping, Service, Stressed, NonRecover, No Contact, Lost Comm</i>)
StatusInfo	Information zum Datenträgerstatus (1 = <i>Other</i> , 2 = <i>Unknown</i> , 3 = <i>Enabled</i> , 4 = <i>Disabled</i> , 5 = <i>Not Applicable</i>)
SystemCreationClassName	Name der Klasse, mit der das Computersystemobjekt erstellt wurde
SystemName	Name des Computersystems
Type	Partitionstyp (<i>Unused, 12-bit FAT, Xenix Type 1, Xenix Type 2, 16-bit FAT, Extended Partition, MS-DOS V4 Huge, Installable File System, PowerPC Reference Platform, UNIX, NTFS, Win95 w/Extended Int 13, Extended w/Extended Int 13, Logical Disk Manager, Unknown</i>)

Tabelle 96: Die Eigenschaften der WMI-Klasse »Win32_DiskPartition« (Forts.)

Die Eigenschaften der Win32_NetworkAdapterConfiguration-Klasse

Die nachfolgende Tabelle enthält die Eigenschaften der Win32_NetworkAdapterConfiguration-Klasse und nennt ihre jeweiligen Bedeutungen.

Eigenschaft	Beschreibung
ARPAwaysSourceRoute	Enthält den Wert <i>True</i> , wenn ARP-Anfragen mit aktiviertem Quell-Routing in Token Ring-Netzwerke übertragen werden, anderenfalls ist der Wert gleich <i>False</i> .
ARPUseEtherSnap	Enthält den Wert <i>True</i> , wenn Ethernet-Pakete die IEEE 802.3 SNAP-Codierung verwenden, anderenfalls ist der Wert gleich <i>False</i> .
Caption	Kurzbeschreibung des Netzwerkadapters unter Angabe des Index und auch der Bezeichnung.
DatabasePath	Pfad zu den Datenbankdateien LMHOSTS, HOSTS, Networks, Protocol und Services

Tabelle 97: Die Eigenschaften der WMI-Klasse »Win32_NetworkAdapterConfiguration«

Eigenschaft	Beschreibung
DeadGWDetectEnabled	Enthält den Wert <i>True</i> , wenn die »Dead-Gateway«-Erkennung aktiviert ist, anderenfalls ist der Wert gleich <i>False</i> .
DefaultIPGateway	IP-Adresse des Standardgateways
DefaultTOS	TOS-Wert (Type of Service), der im Header ausgehender IP-Pakete hinterlegt ist
DefaultTTL	TTL-Wert (Time to Live), der im Header ausgehender IP-Pakete hinterlegt ist
Description	Bezeichnung des Netzwerkadapters
DHCPEnabled	Enthält den Wert <i>True</i> , wenn die DHCP-Funktion für die Netzwerkkarte oder für den Netzwerkport aktiviert wurde und die IP-Adresse entweder automatisch konfiguriert oder dynamisch zugewiesen wird. Anderenfalls ist der Wert gleich <i>False</i> .
DHCPLeaseExpires	Zeitpunkt, an dem eine dynamisch zugewiesene IP-Adresse ihre Gültigkeit verliert
DHCPLeaseObtained	Zeitpunkt, an dem eine dynamische IP-Adresse zugewiesen wurde
DHCPServer	IP-Adresse des DHCP-Servers
DNSDomain	Name der verwendeten Domäne
DNSDomainSuffixSearchOrder	Suchreihenfolge der DNS-Domänensuffixe
DNSEnabledForWINSResolution	Enthält den Wert <i>True</i> , wenn die Namensauflösung zunächst mit DNS erfolgen soll und erst dann an den WINS-Service übergeben wird, wenn die Auflösung per DNS erfolglos war. Anderenfalls ist der Wert gleich <i>False</i> .
DNSHostname	Hostname des lokalen Computersystems, der auch in der DNS-Tabelle verwendet wird
DNSServerSearchOrdner	Suchreihenfolge der DNS-Server
DefaultIPGateway	IP-Adresse des Standardgateway
DomainDNSRegistrationEnabled	Enthält den Wert <i>True</i> , wenn die IP-Adresse der aktiven Netzwerkverbindung im DNS registriert werden soll, anderenfalls ist der Wert gleich <i>False</i> .
ForwardBufferMemory	Größe des Datenpuffers für die Router-Warteschlange
FullDNSRegistrationEnabled	Enthält den Wert <i>True</i> , wenn das Domänen-Suffix der aktiven Netzwerkverbindung im DNS registriert werden soll, anderenfalls ist der Wert <i>False</i> .
GatewayCostMetric	Gateway-Metric

Tabelle 97: Die Eigenschaften der WMI-Klasse »Win32_NetworkAdapter-Configuration« (Forts.)

Eigenschaft	Beschreibung
IGMPLevel	IGMP-Level (0 = No Multicast, 1 = IP Multicast, 2 = IP & IGMP Multicast)
Index	Eindeutiger Index der Netzwerkadapterkonfiguration
InterfaceIndex	Eindeutiger Index des Netzwerkadapters
IPAddress	IP-Adresse(n) des lokalen Computersystems
IPConnectionMetric	IP-Verbindungsmetrik
IPEnabled	Enthält den Wert <i>True</i> , wenn das TCP/IP-Protokoll an den Netzwerkadapter gebunden und auch aktiviert ist. Anderenfalls ist der Wert <i>False</i> .
IPFilterSecurityEnabled	Enthält den Wert <i>True</i> , wenn die TCP/IP-Filterfunktion aktiviert ist, anderenfalls ist der Wert gleich <i>False</i> .
IPPortSecurityEnabled	Enthält den Wert <i>True</i> , wenn der Netzwerkverkehr durch eine Port-Security geschützt ist, anderenfalls ist der Wert gleich <i>False</i> .
IPSecPermitIPProtocols	Enthält die IP-Protokolle, die bei Aktivieren der Filterfunktion erlaubt werden.
IPSecPermitTCPPorts	Enthält die TCP-Ports, die bei Aktivieren der Filterfunktion zugelassen werden.
IPSecPermitUDPPorts	Enthält die UDP-Ports, die bei Aktivieren der Filterfunktion zugelassen werden.
IPSubnet	Enthält die Subnetzmaske, die im TCP/IP-Protokoll des Netzwerkadapters hinterlegt ist.
IPUseZeroBroadCast	Enthält den Wert <i>True</i> , wenn »Null-IP-Broadcasts« (0.0.0.0) verwendet werden, anderenfalls ist der Wert gleich <i>False</i> .
IPXAddress	IPX-Adresse
IPXEnabled	Enthält den Wert <i>True</i> , wenn IPX aktiviert ist, anderenfalls ist der Wert gleich <i>False</i> .
IPXFrameType	IPX-Rahmentyp (0 = Ethernet II, 1 = Ethernet 802.3, 2 = Ethernet 802.2, 3 = Ethernet SNAP, 255 = Auto)
IPXMediaType	IPX-Medientyp (1 = Ethernet, 2 = Token Ring, 3 = FDDI, 4 = Arcnet)
IPXNetworkNumber	IPX-Netzwerknummer
IPXVirtualNetNumber	Virtuelle IPX-Netzwerknummer
KeepAliveInterval	Intervall zwischen einer erneuten Keepalive-Übertragung bis zum Empfang einer Antwort

Tabelle 97: Die Eigenschaften der WMI-Klasse »Win32_NetworkAdapter-Configuration« (Forts.)

Eigenschaft	Beschreibung
KeepAliveTime	Anzahl der Keepalive-Übertragungen während einer Leerlaufzeit
MACAddress	MAC-Adresse des Netzwerkadapters
MTU	Maximale Paketgröße
NumForwardPackets	Legt die Anzahl der für die Routerpaketwarteschlange zugeordneten IP-Paketheader an.
PMTUBHDetectEnabled	Enthält den Wert <i>True</i> , wenn »Black Hole-Router« erkannt werden, anderenfalls ist der Wert gleich <i>False</i> .
PMTUDiscoveryEnabled	Enthält den Wert <i>True</i> , wenn der MTU-Pfad über den Pfad zum Remotehost identifiziert wird, anderenfalls ist der Wert gleich <i>False</i> .
ServiceName	Dienstname des Netzwerkadapters
SettingID	Eindeutige Objekt-ID
TCPNetBIOSOptions	NetBIOS over TCP/IP-Einstellungen (0 = <i>EnableNetBIOSViaDHCP</i> , 1 = <i>EnableNetBIOS</i> , 2 = <i>DisableNetBIOS</i>)
TCPMaxConnectRetransmissions	Anzahl der Verbindungsaufbauwiederholungen
TCPMaxDataRetransmissions	Anzahl der Übertragungswiederholungen für den Versand eines Datensegments
TCPNumConnections	Maximale Anzahl der gleichzeitigen TCP-Verbindungen
TCPUseRFC1122UrgentPointer	Enthält den Wert <i>True</i> , wenn TCP die RFC 1122-Spezifikation nutzt, anderenfalls ist der Wert gleich <i>False</i> .
TCPWindowSize	Spezifiziert die Anzahl der Bytes, die eine Quellstation an die Zielstation schicken kann, ohne dass hierfür eine Bestätigung (ACK) angefordert wird.
WINSEnableLMHostsLookup	Enthält den Wert <i>True</i> , wenn zur Auflösung der Hostnamen zur IP-Adresse eine lokale HOST-Datei verwendet werden soll, anderenfalls ist der Wert <i>False</i> .
WINSHostLookupFile	Pfad zur lokalen HOST-Datei, über die die lokale Auflösung der Hostnamen zur IP-Adresse vorgenommen wird.
WINSPrimaryServer	IP-Adresse des primären WINS-Servers
WINSScopeID	Wert, der dem NetBIOS-Namen angehängt wird, um eine bestimmte Computergruppe zu isolieren, damit sie ausschließlich untereinander kommunizieren können
WINSSecondaryServer	IP-Adresse des sekundären WINS-Servers

Tabelle 97: Die Eigenschaften der WMI-Klasse »Win32_NetworkAdapter-Configuration« (Forts.)

Die Eigenschaften der Win32_Share-Klasse

Die nachfolgende Tabelle zeigt die Eigenschaften der WMI-Klasse Win32_Share unter Angabe einer kurzen Beschreibung.

Eigenschaft	Beschreibung
AccessMask	Enthält die Freigabeberechtigung in Form eines Bit-Musters (0 = <i>File_Read_Data</i> bzw. <i>File_List_Directory</i> , 1 = <i>File_Write_Data</i> bzw. <i>File_Add_File</i> , 4 = <i>File_Append_Data</i> bzw. <i>File_Add_Subdirectory</i> , 8 = <i>File_Read_EA</i> , 16 = <i>File_Write_EA</i> , 32 = <i>File_Execute</i> bzw. <i>File_Traverse</i> , 64 = <i>File_Delete_Child</i> , 128 = <i>File_Read_Attributes</i> , 256 = <i>File_Write_Attributes</i> , 65536 = <i>Delete</i> , 131072 = <i>Read_Control</i> , 262144 = <i>Write_DAC</i> , 524288 = <i>Write_Owner</i> , 1048576 = <i>Synchronize</i>).
AllowMaximum	Enthält den Wert <i>True</i> , wenn die Anzahl gleichzeitiger Zugriffe nicht eingeschränkt wurde, anderenfalls ist der Wert gleich <i>False</i> .
Caption	Kurze Beschreibung zur Freigabe (einzeilig)
Description	Beschreibung der Freigabe
InstallDate	Datum und Uhrzeit, wann das Objekt erstellt wurde
MaximumAllowed	Maximale Anzahl der gleichzeitigen Verbindungen zur Freigabe
Name	Name der Freigabe
Path	Pfad zur Freigabe
Status	Status der Freigabe. Mögliche Werte sind <i>OK</i> , <i>Error</i> , <i>Degraded</i> , <i>Unknown</i> , <i>Pred Fail</i> , <i>Starting</i> , <i>Stopping</i> , <i>Service</i> , <i>Stressed</i> , <i>Nonrecover</i> , <i>NoContact</i> , <i>LostComm</i>
Type	Typ der Freigabe (0 = <i>Disk Drive</i> , 1 = <i>Print Queue</i> , 2 = <i>Device</i> , 3 = <i>IPC</i> , 2147483648 = <i>Disk Drive Admin</i> , 2147483649 = <i>Print Queue Admin</i> , 2147483650 = <i>Device Admin</i> , 2147483651 = <i>IPC Admin</i>)

Tabelle 98: Die Eigenschaften der WMI-Klasse »Win32_Share«

Die Eigenschaften der Win32_NTLogEvent-Klasse

Die nachfolgende Tabelle zeigt die Eigenschaften der WMI-Klasse Win32_NTLogEvent unter Angabe einer kurzen Beschreibung.

Eigenschaft	Beschreibung
Category	Name der Unterkategorie (die Unterkategorie ist quellspezifisch)
CategoryString	Zeichenfolge, die ebenfalls die quellspezifische Unterkategorie bezeichnet
ComputerName	Name des Computersystems, das den Ereigniseintrag erzeugt hat
Data	Binäre Daten zum Ereignis

Tabelle 99: Die Eigenschaften der WMI-Klasse »Win32_NTLogEvent«

Eigenschaft	Beschreibung
EventCode	EventID des Ereignisses (uint16)
EventIdentifier	EventID des Ereignisses (uint32)
EventType	Typ des Ereignisses (1 = Error, 2 = Warning, 3 = Information, 4 = Security audit success, 5 = Security audit failure)
InsertionStrings	InsertionString des Ereignisses
Logfile	Name der Protokolldatei
Message	Ereignisbeschreibung bzw. Meldungstext
RecordNumber	Eindeutiger Index des Ereignisses
SourceName	Name der Quelle (Anwendung, Dienst, Treiber ...), die das Ereignis ausgelöst hat
TimeGenerated	Datum und Uhrzeit, wann das Ereignis aufgetreten ist
TimeWritten	Datum und Uhrzeit, wann das Ereignis aufgezeichnet wurde
Type	Ereignistyp (1 = <i>Error</i> , 2 = <i>Warning</i> , 3 = <i>Information</i> , 4 = <i>Security audit success</i> , 5 = <i>Security audit failure</i>)
User	Benutzer, der zu dem Zeitpunkt angemeldet war, als das Ereignis aufgetreten ist

Tabelle 99: Die Eigenschaften der WMI-Klasse »Win32_NTLogEvent« (Forts.)

Die Eigenschaften der Win32_ComputerSystem-Klasse

Die nachfolgende Tabelle zeigt die Eigenschaften der WMI-Klasse Win32_Computersystem, die mit dem WMIC-Befehl und auch per VBScript ausgelesen und teilweise geändert werden können.

Eigenschaft	Beschreibung
AdminPasswordStatus	Status des Administratorenkennworts (1 = <i>Disabled</i> , 2 = <i>Enabled</i> , 3 = <i>Not implemented</i> , 4 = <i>Unknown</i>)
AutomaticResetBootOption	Status der automatischen Reset-Option (<i>True</i> oder <i>False</i>)
AutomaticResetCapability	Enthält <i>True</i> , wenn die automatische Reset-Funktion aktiviert ist, anderenfalls ist der Wert gleich <i>False</i> .
BootOptionOnLimit	Legt die Aktion fest, die beim Erreichen des Boot-Limits durchgeführt wird (1 = <i>Reserved</i> , 2 = <i>Operating system</i> , 3 = <i>System utilities</i> , 4 = <i>do not reboot</i>).
BootOptionOnWatchDog	Bestimmt die Aktion, die nach Ablauf des Watchdog-Timers durchgeführt wird (1 = <i>Reserved</i> , 2 = <i>Operating system</i> , 3 = <i>System utilities</i> , 4 = <i>do not reboot</i>).
BootROMSupported	Boot-ROM-Unterstützung (<i>True</i> oder <i>False</i>)

Tabelle 100: Die Eigenschaften der WMI-Klasse »Win32_Computersystem«

Eigenschaft	Beschreibung
BootupState	Letzter Boot-Status (<i>Normal boot</i> , <i>fail-safe boot</i> oder <i>fail-safe with network-boot</i>)
Caption	Kurze Beschreibung
ChassisBootupState	Chassis-Status des letzten Boot-Vorgangs (1 = <i>Other</i> , 2 = <i>Unknown</i> , 3 = <i>Safes</i> , 4 = <i>Warning</i> , 5 = <i>Critical</i> , 6 = <i>non-recoverable</i>)
CreationClassName	Name der Klasse, mit der das Objekt erstellt wurde
CurrentTimeZone	Aktuelle Zeitzone
DaylightInEffect	Status der Sommerzeiteinstellung (<i>True</i> oder <i>False</i>)
Description	Beschreibung
DNSHostName	DNS-Name
Domain	Name der Domäne
DomainRole	Rolle innerhalb der Domäne
EnableDaylightSavingsTime	Aktiviert oder deaktiviert die Verwendung der Sommerzeit (<i>True</i> oder <i>False</i>).
FrontPanelResetStatus	Status des Reset-Buttons (0 = <i>Disabled</i> , 1 = <i>Enabled</i> , 2 = <i>Not implemented</i> , 3 = <i>Unknown</i>)
InfraredSupported	Infrarotunterstützung (<i>True</i> oder <i>False</i>)
InitialLoadInfo	Parameter zur Ermittlung der Startsequenz
InstallDate	Datum und Uhrzeit, wann das Objekt erzeugt wurde
KeyboardPasswordStatus	Sicherheitseinstellung für das Tastaturkennwort (0 = <i>Disabled</i> , 1 = <i>Enabled</i> , 2 = <i>Not implemented</i> , 3 = <i>Unknown</i>)
LastLoadInfo	InitialLoadInfo-Eintrag mit dem Start-Parameter für das Betriebssystem
Manufacturer	Name des Herstellers
Model	Name des Modells
Name	Name des Computersystemobjekts
NameFormat	Legt die Reihenfolge zur Vergabe des heur. Computernamens fest (<i>IP</i> , <i>Dial</i> , <i>Hid</i> , <i>Nwa</i> , <i>Hwa</i> , <i>X25</i> , <i>Isdn</i> , <i>Ipx</i> , <i>Dcc</i> , <i>Icd</i> , <i>E.164</i> , <i>Sna</i> , <i>Oid/Osi</i> , <i>Other</i>).
NetworkServerModeEnabled	Netzwerk-Server (<i>True</i> oder <i>False</i>)
NumberOfProcessors	Anzahl der installierten Prozessoren
OEMLogoBitmap	Datenliste für das Herstellerlogo
OEMStringArray	Definition der OEM-Informationen
PartOfDomain	Domänenmitgliedschaft (<i>True</i> oder <i>False</i>)

Tabelle 100: Die Eigenschaften der WMI-Klasse »Win32_Computersystem« (Forts.)

Eigenschaft	Beschreibung
PauseAfterReset	Wartezeit in Millisekunden, bis der Reset durchgeführt wird
PowerManagementCapabilities	PowerManagement-Funktionalitäten eines Geräts (0 = <i>Unknown</i> , 1 = <i>Not supported</i> , 2 = <i>Disabled</i> , 3 = <i>Enabled</i> , 4 = <i>Power saving modes entered automatically</i> , 5 = <i>Power state settable</i> , 6 = <i>Power cycling supported</i> , 7 = <i>Time power on supported</i>)
PowerManagementSupported	Unterstützung für die Energieverwaltung (<i>True</i> oder <i>False</i>)
PowerOnPasswordStatus	Status des PowerOn-Kennworts (0 = <i>Disabled</i> , 1 = <i>Enabled</i> , 2 = <i>Not implemented</i> , 3 = <i>Unknown</i>)
PowerState	Aktueller Energiemodus (0 = <i>Unknown</i> , 1 = <i>Full Power</i> , 2 = <i>Power Save – Low Power Save</i> , 3 = <i>Power Save – Standby</i> , 4 = <i>Power Save – Unknown</i> , 5 = <i>Power Cycle</i> , 6 = <i>Power off</i> , 7 = <i>Power Save – Warning</i>)
PowerSupplyState	Status des Netzteils nach dem letzten Boot-Vorgang (1 = <i>Other</i> , 2 = <i>Unknown</i> , 3 = <i>Safe</i> , 4 = <i>Warning</i> , 5 = <i>Critical</i> , 6 = <i>Non-recoverable</i>)
PrimaryOwnerContact	Kontaktinformationen des Computersystembesitzers
PrimaryOwnerName	Name des Computersystembesitzers
ResetCapability	Reset-Möglichkeiten (1 = <i>Other</i> , 2 = <i>Unknown</i> , 3 = <i>Disabled</i> , 4 = <i>Enabled</i> , 5 = <i>Non-recoverable</i>)
ResetCount	Anzahl der automatischen Resets
ResetLimit	Anzahl der erlaubten Resets
Roles	Rolle des Computersystems
Status	Status des Objekts (<i>OK</i> , <i>Error</i> , <i>Degraded</i> , <i>Unknown</i> , <i>Pred Fail</i> , <i>Starting</i> , <i>Stopping</i> , <i>Service</i>)
SupportContactDescription	Kontaktinformationen für den Windows-Support
SystemStartupDelay	Wartezeit bis Windows gestartet wird
SystemStartupOptions	Windows-Startoptionen
SystemStartupSetting	Index des Startprofils
SystemType	Typ des Computersystems (<i>X86-based PC</i> , <i>MIPS-based PC</i> , <i>Alpha-based PC</i> , <i>Power PC</i> , <i>SH-x PC</i> , <i>StrongARM PC</i> , <i>64-bit Intel PC</i> , <i>64-bit Alpha PC</i> , <i>Unknown</i> , <i>X86-Nec98 PC</i>)
ThermalState	Temperaturstatus während des letzten Boot-Vorgangs (1 = <i>Other</i> , 2 = <i>Unknown</i> , 3 = <i>Safe</i> , 4 = <i>Warning</i> , 5 = <i>Critical</i> , 6 = <i>Non-recoverable</i>)
TotalPhysicalMemory	Vorhandener Hauptspeicher

Tabelle 100: Die Eigenschaften der WMI-Klasse »Win32_Computersystem« (Forts.)

Eigenschaft	Beschreibung
UserName	Name des angemeldeten Benutzers
WakeUpType	WakeUp-Ereignis (0 = <i>Reserved</i> , 1 = <i>Other</i> , 2 = <i>Unknown</i> , 3 = <i>APM Timer</i> , 4 = <i>Modem Ring</i> , 5 = <i>LAN Remote</i> , 6 = <i>Power Switch</i> , 7 = <i>PCI PME</i> , 8 = <i>AC Power Restored</i>)
Workgroup	Name der Arbeitsgruppe

Tabelle 100: Die Eigenschaften der WMI-Klasse »Win32_Computersystem« (Forts.)

Die Eigenschaften der Win32_Process-Klasse

Die nachfolgende Tabelle zeigt die Eigenschaften der WMI-Klasse Win32_Process mit Angabe einer kurzen Beschreibung.

Eigenschaft	Beschreibung
Caption	Kurze Beschreibung bzw. Anzeigename des Prozesses
CommandLine	Befehlszeile, über die der Prozess gestartet wird
CreationClassName	Name der Klasse, mit der das Objekt erstellt wurde
CreationDate	Zeitpunkt der Prozessausführung
CSCreationClassName	Name der Klasse, mit der das Computersystemobjekt erstellt wurde
CSName	Name des Computersystems
Description	Beschreibung des Prozesses
ExecutablePath	Pfad zur ausführbaren Datei des Prozesses
Handle	Prozess-ID
HandleCount	Anzahl der geöffneten Handles pro Prozess
InstallDate	Datum und Uhrzeit, wann das Prozessobjekt eingerichtet wurde
KernelModeTime	Ausführungszeit im Kernelmodus
MaximumWorkingSetSize	Maximale Größe des für den Prozess zur Verfügung stehenden Speichers
MinimumWorkingSetSize	Minimale Größe des für den Prozess zur Verfügung stehenden Speichers
Name	Name des Prozessobjekts
OSCreationClassName	Name der Klasse mit der das Betriebssystemobjekt erstellt wurde
OSName	Name des Betriebssystems
OtherOperationCount	Anzahl der IO-Operationen (keine Lese- und Schreiboperationen)

Tabelle 101: Die Eigenschaften der WMI-Klasse »Win32_Process«

Eigenschaft	Beschreibung
OtherTransferCount	Übertragene Daten während der IO-Operationen (keine Lese- und Schreiboperationen)
PageFaults	Anzahl der Seitenfehler
PageFileUsage	Größe der Arbeitsseiten des Prozesses
ParentProcessID	Prozess-ID des übergeordneten Prozesses
PeakPageFileUsage	Maximale Speichernutzung der Auslagerungsdatei
PeakVirtualSize	Maximale Speichernutzung des virtuellen Speichers
PeakWorkingSetSize	Maximale Speichernutzung des für den Prozess zur Verfügung stehenden Speichers
Priority	Priorität des Prozesses
PrivatePageCount	Anzahl der Arbeitsseiten, die ausschließlich von dem Prozess verwendet werden
ProcessID	Eindeutige Prozess-ID
QuotaNonPagedPoolUsage	Größe des nichtausgelagerten Speicherpools
QuotaPagedPoolUsage	Größe des ausgelagerten Speicherpools
QuotaPeakNonPagedPoolUsage	Maximale Größe des nichtausgelagerten Speicherpools
QuotaPeakPagedPoolUsage	Maximale Größe des ausgelagerten Speicherpools
ReadOperationCount	Anzahl der Lesevorgänge
ReadTransferCount	Anzahl der Bytes, die gelesen wurden
SessionId	ID der Windows-Sitzung
Status	Diese Eigenschaft wird nicht genutzt
TerminationDate	Zeitpunkt, zu dem ein Prozess angehalten oder unterbrochen wurde
ThreadCount	Anzahl der Threads
UserModeTime	Ausführungszeit im User-Modus
VirtualSize	Größe des virtuellen Adressspeichers
WindowsVersion	Betriebssystemversion
WorkingSetSize	Größe des vom Prozess benötigten Speichers
WriteOperationCount	Anzahl der Schreibvorgänge
WriteTransferCount	Anzahl der Bytes, die geschrieben wurden

Tabelle 101: Die Eigenschaften der WMI-Klasse »Win32_Process« (Forts.)

Die Eigenschaften der Win32_Service-Klasse

Die nachfolgende Tabelle zeigt die Eigenschaften der WMI-Klasse Win32_Service unter Angabe einer kurzen Beschreibung.

Eigenschaft	Beschreibung
AcceptPause	Enthält <i>True</i> , wenn der Dienst die Pause-Anforderung akzeptiert.
AcceptStop	Enthält <i>True</i> , wenn der Dienst die Stop-Anforderung akzeptiert.
Caption	Kurze Beschreibung des Dienstes
CheckPoint	Eine Art »Timeout-Zähler«, über den sich eine längere Ausführungsdauer beim Starten/Beenden eines Prozesses ermitteln lässt
CreationClassName	Name der Klasse, mit der das Dienstobjekt erstellt wurde
Description	Beschreibung des Dienstes
DesktopInteract	Enthält <i>True</i> , wenn der Dienst interaktiv ausgeführt wird
DisplayName	Anzeigename des Dienstes
ErrorControl	Gibt Auskunft über die Schwere eines Fehlers, wenn der Dienst nicht automatisch gestartet werden konnte (<i>Ignore, Normal, Severe, Critical, Unknown</i>).
ExitCode	Betriebssystemseitiger Fehlercode, der beim fehlerhaften Starten oder Beenden eines Dienstes zurückgegeben wird
InstallDate	Datum und Uhrzeit, wann das Dienstobjekt erstellt wurde
Name	Name des Dienstes
PathName	Pfad zur ausführbaren Datei des Dienstes
ProcessID	Prozess-ID
ServiceSpecificExitCode	Dienstseitiger Fehlercode, der beim fehlerhaften Starten oder Beenden eines Dienstes zurückgegeben wird
ServiceType	Diensttyp (<i>Kernel driver, File system driver, Adapter, Recognizer driver, Own process, Share process</i>)
Started	Enthält <i>True</i> , wenn der Dienst gestartet wurde
StartMode	Start-Modus des Dienstes (<i>Boot, System, Auto, Manual, Disabled</i>)
StartName	Benutzerkontext, unter dessen Verwendung der Dienst ausgeführt wird
State	Aktueller Status des Dienstes (<i>Stopped, Start Pending, Stop Pending, Running, Continue Pending, Pause Pending, Paused, Unknown</i>)

Tabelle 102: Die Eigenschaften der WMI-Klasse »Win32_Service«

Eigenschaft	Beschreibung
Status	Status des Dienstes (<i>Ok, Error, Degraded, Unknown, Pred Fail, Starting, Stopping, Service</i>)
SystemCreationClassName	Name der Klasse, mit der das Computersystemobjekt erstellt wurde
SystemName	Name des Computersystems, auf dem der Dienst eingerichtet ist
TagID	Eindeutige Tag-ID, anhand derer die Position des Dienstes innerhalb einer Dienstgruppe bestimmt wird
WaitHint	Zulässiger Timeout-Wert für das Starten und Stoppen eines Dienstes

Tabelle 102: Die Eigenschaften der WMI-Klasse »Win32_Service« (Forts.)

Die Eigenschaften der Win32_ScheduledJob-Klasse

Die nachfolgende Tabelle zeigt die Eigenschaften der WMI-Klasse `Win32_ScheduledJob` unter Angabe einer kurzen Beschreibung.

Eigenschaft	Beschreibung
Caption	Kurze Beschreibung (einzeilig) zum Task
Command	Befehlszeile oder Programm (inklusive der erforderlichen Parameter), das automatisiert wird
DaysOfMonth	Tag im Monat (1-31), an dem der Task ausgeführt wird. Der Tag wird als Integer-Wert dargestellt (1 = erster Tag, 2 = zweiter Tag, 4 = dritter Tag, 8 = vierter Tag usw.)
DaysOfWeek	Wochentag (1-7), an dem der Task ausgeführt wird (1 = Montag, 2 = Dienstag usw.)
Description	Beschreibung des Tasks
ElapsedTime	Ausführungszeit des Tasks (gemessen an der letzten Task-Ausführung)
InstallDate	Datum und Uhrzeit, wann das Objekt installiert wurde
InteractWithDesktop	Enthält den Wert <i>True</i> , wenn der Task eine Interaktion mit dem Desktop erlaubt, anderenfalls ist der Wert gleich <i>False</i> .
JobID	Eindeutige ID des Tasks
JobStatus	Status des Tasks (<i>Success</i> oder <i>Failure</i>)
Name	Bezeichnung des Tasks
Notify	Benutzer, der benachrichtigt werden soll, wenn die Task-Ausführung beendet wurde oder fehlgeschlagen ist
Owner	Benutzer, der den Task eingerichtet hat

Tabelle 103: Die Eigenschaften der WMI-Klasse »Win32_ScheduledJob«

812 >> Die Eigenschaften der Win32_ScheduledJob-Klasse

Eigenschaft	Beschreibung
Priority	Priorität des Tasks
RunRepeatedly	Enthält den Wert <i>True</i> , wenn der Task mehrmalig ausgeführt wird, anderenfalls ist der Wert gleich <i>False</i> .
StartTime	Datum und Uhrzeit (im UTC-Format), wann die nächste Task-Ausführung beginnt
Status	Status des Tasks. Mögliche Werte sind <i>OK</i> , <i>Error</i> , <i>Degraded</i> , <i>Unknown</i> , <i>Pred Fail</i> , <i>Starting</i> , <i>Stopping</i> und <i>Service</i> .
TimeSubmitted	Uhrzeit der Task-Einreichung
UntilTime	Dauer, nach der die Task-Ausführung beendet sein muss

Tabelle 103: Die Eigenschaften der WMI-Klasse »Win32_ScheduledJob« (Forts.)

Stichwortverzeichnis

A

- Abmeldeskript 771
- ACE 233, 243
- Active Directory-Benutzer und
 - Computer 126
 - Abfrage erstellen 130, 192
- Active Directory-Domänen und
 - Vertrauensstellungen 675, 683
- Active Directory-Schema 139, 675
 - registrieren 675
- Active Directory-Standorte und -Dienste 666
- ADO Recordset-Objekt 352
 - AddNew-Methode 352
 - Open-Methode 352
 - Sort-Methode 352
 - Update-Methode 352
- ADOR.Recordset-Objekt
 - MoveFirst-Methode 338
 - Move-Methode 338
- Assoc 309
- At.exe 735f., 744, 749, 753, 758
- AT-Dienstkonto 763
 - Benutzer ändern 763
- Attrib.exe 280
- Attribute 278
 - Archive 278
 - Hidden 278
 - Read-Only 278
 - System 278
- Ausführen als-Funktion 173
- Auslagerungsdatei 637
 - dynamische Größe 637
 - Größe ändern 637
 - Größe anpassen 637
 - statische Größe 637
- Autoexnt.exe 764
- Automatische Updates 62, 591
 - Gruppenrichtlinien 591
 - konfigurieren 591
- Automatisierung 735, 741
- Autostart 767
 - beim Systemstart 764
 - vor dem Herunterfahren 777

- vor der Abmeldung 771
- vor der Anmeldung 764

Autostart-Gruppe 767

B

- Backup 355
- Batch-Datei 84
 - erzeugen 84
 - Kommentare einfügen 84
 - Sprungmarken verwenden 84
- Benutzer 125
 - abfragen 126
 - abmelden 170
 - ändern 153
 - aktivieren 159
 - anlegen 134
 - Anmeldung überprüfen 168
 - auflisten 130
 - aus Gruppe entfernen 217
 - bearbeiten 153
 - deaktivieren 159
 - Eigenschaften auslesen 145
 - erstellen 134
 - exportieren 130
 - Gruppen-Mitgliedschaften anzeigen 209
 - in Gruppe aufnehmen 205
 - in Organisationseinheit verschieben 178
 - Kennwort zurücksetzen 183
 - kopieren 139
 - löschen 165
 - SID ermitteln 180
 - Template erstellen 139
 - überprüfen 126
 - umbenennen 163
 - verschieben 178
- Benutzerverwaltung 125
- Berechtigungen
 - Dump erstellen 256
 - für Dateien verweigern 243
 - für Dateien zulassen 243
 - für Ordner verweigern 243
 - für Ordner zulassen 243
- Besitzer 253

Betriebssystem-Update 57f.
 Automatische Updates 62
 Feature Pack 57
 Hotfixes 57
 installierte Updates ermitteln 74
 manuell aktualisieren 60
 mehrere Updates installieren 61
 Service Pack 57
 Sicherheitsupdate 57
 Update-Pakete genehmigen 71
 Updaterollup 57
 Updates 57
 Updates anzeigen lassen 74
 Updates deinstallieren 76
 Updates entfernen 76
 Wichtiges Update 57
 WSUS-Clients konfigurieren 72
 WSUS-Server einrichten 67
 WSUS-Server konfigurieren 69
 Betriebszeit 579
 BGInfo.exe 589
 Boot.ini 366
 Pfadangaben 370
 Start-Optionen 371
 Bootcfg.exe 367, 373
 Bubble sort 352

C

Cacs.exe 238, 247, 258
 CD-R/RW 355
 CIM_DataFile-Klasse 232, 295
 AccessMask-Eigenschaft 284
 Archive-Eigenschaft 229, 284
 Caption-Eigenschaft 229, 284
 Compressed-Eigenschaft 229, 284
 Compress-Eigenschaft 295
 CompressionMethod-Eigenschaft 284
 CreationClassName-Eigenschaft 284
 CreationDate-Eigenschaft 229, 270, 284
 CSName-Eigenschaft 284
 Description-Eigenschaft 229, 284
 Drive-Eigenschaft 229, 284
 EightDotThreeFileName-Eigenschaft 229, 284
 Encrypted-Eigenschaft 229, 284
 EncryptionMethod-Eigenschaft 284
 Extension-Eigenschaft 229, 284
 FileName-Eigenschaft 229, 284
 FileSize-Eigenschaft 229, 267, 284

 FileType-Eigenschaft 229, 284
 FSCreationClassName-Eigenschaft 284
 Hidden-Eigenschaft 229, 284
 InstallDate-Eigenschaft 284
 InUseCount-Eigenschaft 284
 LastAccessed-Eigenschaft 229, 284
 LastModified-Eigenschaft 229, 275, 284
 Manufacturer-Eigenschaft 284
 Name-Eigenschaft 267, 284
 Path-Eigenschaft 229, 284
 Readable-Eigenschaft 284
 Rename-Methode 291
 System-Eigenschaft 284
 StatusEigenschaft 284
 TakeOwnership-Methode 255
 Übersicht der Eigenschaften 794
 Uncompress-Eigenschaft 295
 Version-Eigenschaft 262, 284
 Writeable-Eigenschaft 284
 Cipher.exe 299
 Comp.exe 318
 Compact.exe 294
 Computer
 aus Domäne entfernen 572
 herunterfahren 646
 in Domäne aufnehmen 568
 neu starten 646
 zur Domäne hinzufügen 568
 Computerkonto
 anlegen 563
 deaktivieren 573
 entfernen 566
 erstellen 563
 hinzufügen 563
 löschen 566, 573
 Convert.exe 417
 Copy 287, 302, 590
 Copy.exe 768, 772, 778
 Create Partition 394
 Creatfil.exe 305
 CScript.exe 110, 114

D

DACL 233
 DAT 355
 Date 584
 DateDiff-Funktion 581
 Date-Funktion 585
 Dateien 227

ACL 233, 243
Änderungszeitpunkt 272
Anzahl der Wörter 347
Anzahl der Zeilen 348
Attribute 278
aufteilen 301
Berechtigungen anzeigen 233
Berechtigungen erteilen 243
Berechtigungs-Dump erstellen 256
Besitzer ermitteln 251
Besitzrechte übernehmen 253
bestimmter Größe anlegen 305
Datums- und Zeitstempel 276
Datums- und Zeitstempel ändern 276
dekomprimieren 294
doppelte Einträge löschen 339
entschlüsseln 299
ersetzen 306
Erstellungszeitpunkt 268
erweiterte Eigenschaften 283
Fußzeilen entfernen 312
geöffnete Dateien ersetzen 306
Größe 263
Größe auf dem Datenträger 263
Größe ermitteln 263
Inhalt sortieren 350
komprimieren 292
Kopfzeilen entfernen 312
kopieren 286
Leerzeilen entfernen 333
letzte n Zeilen ausgeben 336
mit Applikation verknüpfen 308
nach Größe suchen 266
registrieren 308
sichern 355
sortieren 350
splitten 301
suchen 227
Testdatei anlegen 305
umbenennen 289
vergleichen 316
verschieben 286
verschlüsseln 298
Version ermitteln 260
Wortanzahl ermitteln 347
Zeichenfolgen suchen 321
Zeichenfolgen suchen und ersetzen 330
Zeilenanzahl ermitteln 348
Zeilennummerierung einfügen 343

Dateisystem
Auswahl 406
FAT 406
FAT32 406
NTFS 406
Dateityp, registrieren 308
Dateiverwaltung 227
Datensicherung 355
Datenträger
Auslastung ermitteln 376
Bezeichnung festlegen 414
Bezeichnung löschen 414
defragmentieren 418
Freier Speicher ermitteln 376
Gesamtkapazität ermitteln 376
in Basisdatenträger konvertieren 386
in dynamische Datenträger
konvertieren 382
Kontingente aktivieren 421
Kontingente anzeigen lassen 427
Kontingente einrichten 424
Kontingente überwachen 427
Konvertierung 382, 386, 398
Laufwerkbuchstabe ändern 409
Laufwerkbuchstabe zuweisen 409
Laufwerkinformationen abfragen 363
Laufwerkinformationen auslesen 363
Laufwerkpfad zuweisen 409
Partitionen anzeigen 379
Partitionsformat ermitteln 374
Status ermitteln 374
Typ ermitteln 374
verwalten 363
Volumes anzeigen 379
Datenträgerbezeichnung
festlegen 414
löschen 414
Datenträgerverwaltung 363
Datenträgerliste einblenden 374
Datum
abfragen 582
ändern 582
Dcpromo.exe 652, 656
Antwortdatei 652
Antwortdatei erstellen 654
Parameter für Antwortdatei 653, 656
DDS 355
Defrag.exe 420, 741
Defragmentierung 418

- DHCP 695
 - Bereich aktivieren 701
 - Bereich ausschließen 704
 - Bereich deaktivieren 701
 - Bereich definieren 697
 - Bereichsoptionen 711
 - IP-Adresse reservieren 708
 - Konflikterkennung 715
 - Server autorisieren 695
 - Server-Optionen 711
 - Statistik abfragen 718
- Dhcpmgmt.msc 695
- Dictionary-Objekt, Add-Methode 248
- Dienste
 - Abhängigkeiten 634
 - anhalten 628
 - anzeigen lassen 619
 - beenden 631
 - Eigenschaften 634
 - entfernen 626
 - erstellen 623, 764
 - fortsetzen 628
 - hinzufügen 623
 - löschen 626
 - registrieren 623
 - starten 631
 - Starttyp 634
 - Übersicht 619
 - Wiederherstellungsparemeter 634
- Diff.exe 318
- Dir 229, 251, 264, 272, 377
- Disconnected Recordset 341, 352
 - AddNew-Methode 341
 - Append-Methode 341
 - Delete-Methode 341
 - MoveFirst-Methode 341
 - MoveNext-Methode 341
 - Open-Methode 341
 - Sort-Methode 341
 - Update-Methode 341
- Disk Quota 421
- Diskpart.exe
 - Activ 397
 - Assign Letter 411
 - Assign Mount 412
 - Convert Basic 386
 - Convert Dynamic 384
 - Create Partition 389, 392
 - Create Volume Simple 400
 - Delete 404
 - Exit 392
 - List Disk 375
 - List Partition 380
 - Select Disk 380, 384, 386, 389, 392, 394, 397, 400, 404, 411
 - Select Partition 397, 404, 411
- DiskQuotaControl-Objekt 423
 - AddUser-Methode 426
 - Initialize-Methode 423
 - LogonName-Eigenschaft 428
 - QuotaLimit-Eigenschaft 426, 428
 - QuotaState-Eigenschaft 423
 - QuotaUsed-Eigenschaft 428
- DLT 355
- DNS 450
 - Cache leeren 727
 - Dynamische Updates konfigurieren 729
 - Ereignisprotokollierung 731
 - Host-Eintrag erzeugen 725
 - Primäre Zone einrichten 720
 - Zonen löschen 723
- Dnscmd.exe 721, 724, 726, 728, 732
- Domäne
 - Computer aufnehmen 568
 - Computer entfernen 572
 - Computer hinzufügen 568
- Domänenfunktionsebene 659
- heraufstufen 659
- Domain.msc 675
- Drive-Objekt
 - FreeSpace-Eigenschaft 378
 - IsReady-Eigenschaft 376
 - TotalSize-Eigenschaft 378
- Dsa.msc 126
- Dsadd.exe
 - Computer 564
 - Group 198
 - User 136, 141
- Dsget.exe
 - Group 201, 215, 225
 - User 148, 182, 210
- Dsmod.exe
 - Group 207
 - Server 672
 - User 156, 161, 185
- Dsmove.exe 163, 179

Dsquery.exe
 Group 189, 193
 Server 668, 676
 User 132
Dsrm.exe 167, 221f., 567
DumpSec 256
DVD-R/RW 355
Dynamische Datenträger
 einfaches Volume 382, 398
 gespiegeltes Volume 382, 398
 RAID5-Volume 382, 398
 Stripeset-Volume 382, 398
 übergreifendes Volume 382, 398

E

Echo 93
Edit.exe 312, 339
Eingabeaufforderung 79f.
 Ausgabeumleitung 82
 Befehlserweiterung 104
 Eigenschaften 88
 Eingabeumleitung 82
 Konfiguration 88
 Übersicht 79
Empty.exe 612
Ereignisanzeige 174, 600
 Anwendungsprotokoll 600
 Ereignisse erstellen 608
 Größe des Protokolls 603
 Protokolle exportieren 606
 Protokolle sichern 606
 Protokoll-Eigenschaften 603
 Sicherheitsprotokoll 533, 600
 Systemprotokoll 600
Ereignisprotokollierung für
 Herunterfahren 598, 646
Eventcreate.exe 608
Eventvwr.msc 174, 600

F

FAT 406
 nach NTFS konvertieren 416
FAT32 406
 nach NTFS konvertieren 416
Fc.exe 318
Festplatten-Cloning 45
File Transfer Protocol 459
FileSystem-Objekt
 AtEndOfStream-Eigenschaft 349

Attributes-Eigenschaft 280
CopyFile-Methode 288, 769, 774, 779
CopyFolder-Methode 288
DateCreated-Eigenschaft 270
DateLastModified-Eigenschaft 274
GetFile-Methode 265, 274
GetFolder-Methode 266, 274
Line-Eigenschaft 349
MoveFile-Methode 288, 291, 755
MoveFolder-Methode 288, 291
OpenTextFile-Methode 142, 319
ReadLine-Methode 314, 319, 345, 349
Size-Eigenschaft 265
SkipLine-Methode 314
WriteBlankLines-Methode 319
WriteLine-Methode 314, 345
Write-Methode 305
Findstr.exe 324, 335, 344, 736f.
Finger 459
For 141
Format 407
Freigaben 466
 abfragen 466
 ändern 474
 aktive Verbindungen abfragen 479
 anzeigen lassen 466
 aufheben 486
 Berechtigungen erteilen 476
 einrichten 469
 entfernen 486
 Laufwerkbuchstaben lösen 484
 Laufwerkbuchstaben zuweisen 483
 Sitzung schließen 481
 Verbindung trennen 481
 veröffentlichen 489
Freigegebene Ordner 469
FSMO-Rollen, übertragen 679
Fsutil.exe
 Quota 422
 Quota Modify 425
 Quota Query 427
FTP 459

G

Geplante Tasks 735
 abfragen 735
 ändern 748
 bearbeiten 748
 beenden 756

Benutzer ändern 762
 Benutzerkonto 762
 deaktivieren 752
 einrichten 741
 hinzufügen 741
 löschen 758
 NextATJobID 754
 Programm ändern 748
 starten 756
 Task-ID 754
 überprüfen 735
 Geräte-Manager 363, 431
 Gesamtstrukturfunktionsebene 663
 heraufstufen 663
 Globaler Katalog 666
 aktivieren 671
 deaktivieren 671
 finden 666
 GNU-Utilities für Win32 104, 106
 Gruppen 125
 abfragen 186
 anlegen 195
 auflisten 191
 Eigenschaften 200
 erstellen 195
 exportieren 191
 Globale Gruppe 196, 206
 löschen 221
 Lokale Gruppe 196, 206
 Mitglieder anzeigen 214
 Mitglieder aufnehmen 205
 Mitglieder hinzufügen 205
 SID ermitteln 224
 überprüfen 186
 Universale Gruppe 196, 206
 Gruppenbereich 196
 Gruppentyp 196
 Gruppenverwaltung 125

H

Head.exe 313
 HTTP 459
 HTTPS 459

I

Imaging 45
 IMAP 459
 Infrastruktur-Master, ermitteln 673

Installation 17, 30
 Automatische Installation 35
 Installation kopieren 41
 Installationsdateien aktualisieren 24
 Installationsmethode 17
 Konfigurationsparameter bereitstellen 21
 Produktaktivierung 53
 Unattend Setup 35
 versiegeln 42
 von CD installieren 30
 vorbereiten 19
 WPA 53
 Installations-Manager 36
 InStr-Funktion 326
 Inuse.exe 307
 IPConfig.exe 436, 438, 448

J

Jaz-Drive 355
 JScript 109

K

Komprimierung 292
 Kontingente
 aktivieren 421
 anzeigen lassen 427
 einrichten 424
 überwachen 427

L

Label.exe 415
 Laufwerkbuchstabe
 ändern 409
 zuweisen 409
 Laufwerkpfad, zuweisen 409
 LDAP-Provider
 AccountDisabled-Eigenschaft 151, 162
 AccountExpirationDate-Eigenschaft 151
 Create-Methode 565
 Delete-Methode 168, 223, 567
 Description-Eigenschaft 151, 199, 203
 DisplayName-Eigenschaft 138
 Eigenschaften des User-Objektes 787
 Filter-Eigenschaft 129
 FirstName-Eigenschaft 151
 FullName-Eigenschaft 151
 GetObject-Methode 565, 567
 Groups-Eigenschaft 151

GroupType-Eigenschaft 199
 HomeDirectory-Eigenschaft 151
 LastFailedLogin-Eigenschaft 151
 LastLogin-Eigenschaft 151, 177
 LastLogoff-Eigenschaft 151
 LastName-Eigenschaft 151
 LoginHours-Eigenschaft 151
 LoginScript-Eigenschaft 151, 158
 LoginWorkstations-Eigenschaft 151
 Member-Eigenschaft 203, 208
 Members-Eigenschaft 216
 MoveHere-Methode 165, 179
 Name-Eigenschaft 195, 203
 PasswordExpirationDate-Eigenschaft 151
 PasswordLastChanged-Eigenschaft 151
 Profile-Eigenschaft 151
 PutEx-Methode 208, 219
 Put-Methode 138, 199, 565
 SamAccountName-Eigenschaft 129, 133, 138, 190, 203
 SetPassword-Eigenschaft 151
 SetPassword-Methode 186
 TrustPartner-Eigenschaft 685
 Ldifde.exe 661, 664
 Left-Funktion 581
 Lemmy.exe 343, 350
 Listener 458
 Logisches Laufwerk
 anlegen 393
 erstellen 393
 formatieren 405
 löschen 402
 Logoff.exe 171
 Lokale Benutzer und Gruppen 126
 Lusrmgr.msc 126

M

MAC-Adresse 435
 abfragen 435
 ermitteln 435
 Magneto-optische Laufwerke 355
 Mailserver 459
 Microsoft_DomainTrustStatus-Klasse 691
 TrustDirection-Eigenschaft 691
 TrustedDomain-Eigenschaft 691
 TrustIsOk-Eigenschaft 691
 TrustStatus-Eigenschaft 691
 MicrosoftDNS_Cache-Klasse 728
 ClearCache-Methode 728

MicrosoftDNS-Klasse, CreateZone-Methode 721
 Mid-Funktion 581
 MinUpTime.exe 579
 MO-Laufwerk 355
 MO-Laufwerke 355
 Move 287
 Msinfo.exe 639
 Msinfo32.exe 363, 575

N

Net.exe
 Continue 629
 Group 188, 193, 197, 201, 215, 221
 LocalGroup 188, 193, 197, 201, 207, 215, 221
 Pause 629
 Session 480f.
 Share 466, 471, 475, 487
 Start 620, 631
 Stop 632
 Time 585
 Use 484
 User 127, 131, 136, 141, 148, 155, 160, 167, 176, 184, 210
 NetBIOS 459
 NetDom.exe 684
 Join 569
 Remove 573
 Trust 689, 691, 694
 Netsh.exe 696, 700, 703, 706, 709, 713, 715f., 718
 Netstat.exe 465
 Network-Objekt 170
 UserName-Eigenschaft 170
 Netzlaufwerk
 einrichten 483
 trennen 484
 Netzwerk 431
 Aktive Verbindungen abfragen 464
 DHCP-Funktion aktivieren 446
 IP-Adresse erneuern 448
 IP-Adresse lösen 448
 IP-Adresse vergeben 440
 Netzlaufwerk einrichten 483
 Netzlaufwerk trennen 484
 Schnittstellenmetrik 443
 Standardgateway festlegen 443
 TCP/IP-Filter 458

Netzwerkkarten
 abfragen 431
 anzeigen lassen 431
 NewSID.exe 45
 Nl.exe 344
 NTBackup 355
 Ntfsutil.exe 681
 NTFS 406
 NTP-Client, aktivieren 583
 NTP-Server, aktivieren 583
 NTVDM 80

O

Ordner 227
 ACL 233, 243
 Änderungszeitpunkt 272
 Attribute 278
 Berechtigungen anzeigen 233
 Berechtigungen erteilen 243
 Berechtigungs-Dump erstellen 256
 Besitzer ermitteln 251
 Besitzrechte übernehmen 253
 Datums- und Zeitstempel 276
 Datums- und Zeitstempel ändern 276
 dekomprimieren 294
 entschlüsseln 299
 Erstellungszeitpunkt 268
 erweiterte Eigenschaften 283
 Größe ermitteln 263
 komprimieren 292
 kopieren 286
 sichern 355
 suchen 227
 umbenennen 289
 verschieben 286
 verschlüsseln 298
 Ordnerverwaltung 227

P

Partition
 aktivieren 396
 als aktiv markieren 396
 anlegen 388, 391
 erstellen 388, 391
 erweiterte 391
 formatieren 405
 löschen 402
 primäre 388
 PDC-Emulator, ermitteln 673

POP3 459
 Primäre Gruppe 233
 Produktaktivierung 53
 Prozesse
 anzeigen 610
 auslesen 610
 beenden 614
 Priorität festlegen 617
 Prozessstruktur beenden 614
 Speichernutzung ermitteln 612

Q

Qchain.exe 62
 Query.exe, Session 169
 Quick Sort 352
 Quota 421

R

Reg.exe
 Add 309, 501, 513, 527, 529, 543, 584, 599,
 638, 644, 772, 778
 Copy 502
 Delete 502, 504, 519, 529, 545
 Export 537, 541, 553
 Import 539, 553
 Query 495, 497, 506, 509, 521, 523, 547,
 596
 REG_BINARY 512
 REG_DWORD 512
 REG_DWORD_BIG_ENDIAN 512
 REG_DWORD_LITTLE_ENDIAN 512
 REG_EXPAND_SZ 512
 REG_FULL_RESOURCE_DESCRIPTOR 512
 REG_LINK 512
 REG_MULTI_SZ 512
 REG_NONE 512
 REG_RESOURCE_LIST 512
 REG_RESOURCE_REQUIREMENTS_LIST 512
 REG_SZ 512
 RegExpr-Objekt 328, 332
 Execute-Methode 328, 332
 Global-Eigenschaft 328, 332
 Pattern-Eigenschaft 328, 332
 Replace-Methode 332
 Regfind.exe 509
 Registrierung 493
 ausdrucken 540
 Backup-Möglichkeiten 549
 Berechtigungen 556

- Besitzer festlegen 531
- Dateien 547
- Daten abfragen 520
- Daten ändern 526
- Daten auflisten 523
- Daten löschen 529
- Daten überprüfen 520
- Dump erstellen 540
- Einstellungen exportieren 551
- Einstellungen importieren 551
- Ereignisse überwachen 533
- Exportieren 537
- Favoriten entfernen 544
- Favoriten hinzufügen 542
- Importieren 538
- Lesezeichen setzen 542
- Referenz 511
- Registrierungsdateien 547
- Run-Schlüssel 767
- Schlüssel abfragen 494
- Schlüssel auflisten 496
- Schlüssel erstellen 500
- Schlüssel löschen 504
- Schlüssel überprüfen 494
- Schlüssel umbenennen 502
- Schlüsselstruktur 496
- Sicherheitsprotokoll 533
- sichern 549
- Überwachung 533
- Werte abfragen 506
- Werte auflisten 509
- Werte erstellen 511
- Werte löschen 518
- Werte überprüfen 506
- Werte umbenennen 515
- wiederherstellen 549
- Zugriffe überwachen 533
- Zugriffskontrollliste 556
- Registry Reference 511
- Reguläre Ausdrücke 321, 330
- Remote Procedure Call 459
- Remotedesktop
 - Verbindung erstellen 647
 - Verbindung herstellen 647
- Remoteinstallationsdienste 47
 - Abbild erstellen 49
 - Abbild restaurieren 51
 - Bootdiskette erstellen 51
 - einrichten 47
 - installieren 47
 - Remoteinstallation starten 51
- Ren 290
- Ren.exe 753
- Rename 290
- Replmon.exe 667
- Rgfg.exe 51
- RID-Master, ermitteln 673
- Riprep.exe 49
- RPC 459
- Runas.exe 174
- S**
- SACL 233
- Sc.exe 620, 764
 - Config 635
 - Continue 629
 - Create 624
 - Delete 627
 - Pause 629
 - Query 620
 - Start 631
 - Stop 633
- Schemamaster, ermitteln 675
- Schmmgmt.dll 139
- Schtasks.exe 735
 - Change 754, 763
 - Create 744, 766
 - Delete 759, 766
 - End 766
 - Query 737
 - Run 757
- Scriptomatic 117
- Secure Socket Layer 459
- Security Descriptor 233
- Security-Identifier 180
- Sed.exe 331
- Server-Rollen 651
- Service, erstellen 764
- Set 93, 640, 643
- Setupmgr.exe 36
- Setx.exe 93, 644
- Shell-Objekt
 - LogEvent-Methode 609
 - Run-Methode 761
- Shutdown.exe 647
- SID 180, 224
 - ändern 45
 - ermitteln 180, 224

- Slipstreaming 23, 26
 - der Treiberdateien 29
 - mit Update.exe 27
 - ohne Update.exe 27
 - Slipstream-Versionen erstellen 23
- SMTP 459
- Software
 - Installationen abfragen 596
 - Installierte Produkte ermitteln 596
- Software Update-Service 591
- Sort.exe 339, 351
- Split.exe 302
- Srvany.exe 623
- Srvinfo.exe 577, 580
- Srvinstw.exe 623
- SSL 459
- Standardgateway
 - festlegen 443
 - Schnittstellenmetrik 443
- Standard-Input 82
- Standard-Output 82
- Startkonfiguration
 - ändern 369
 - ermitteln 366
- StdIn 82
- StdOut 82
- StdRegProv-Klasse
 - CreateKey-Methode 311, 501, 774, 779
 - DeleteKey-Methode 505
 - DeleteKey-Methode (rekursiv) 505
 - DeleteValue-Methode 516, 519, 546
 - EnumKey-Methode 495, 497
 - EnumKey-Methode (rekursiv) 499
 - EnumValues-Methode 507, 510, 524, 548
 - GetBinaryValue-Methode 516, 521
 - GetDWORDValue-Methode 516, 521
 - GetExpandedStringValue-Methode 516, 521
 - GetMultiStringValue-Methode 516, 521
 - GetStringValue-Methode 516, 521, 548
 - SetBinaryValue-Methode 514, 516, 530, 774, 779
 - SetDWORDValue-Methode 514, 516, 530, 588, 594, 599
 - SetExpandedStringValue-Methode 514, 516, 530
 - SetMultiStringValue-Methode 514, 516, 530, 639

- SetStringValue-Methode 311, 514, 516, 528, 530, 543, 588, 594, 774, 779
- Streamer
 - DAT 355
 - DDS 355
 - DLT 355
- Subinac.exe 258, 478, 532f., 558
- SUS-Server
 - Client-Konfiguration 592
 - Gruppenrichtlinien 592
- Sysprep.exe 42
- Systeminformationen 363, 575, 639
- Systemkonto 763
- Systemstatusdatenfeature 598
 - für das Herunterfahren 646
- Systemvoraussetzungen 20
- Systemvorbereitungsprogramm 42

T

- Tail.exe 313, 337
- Takeown.exe 253
- Taskkill.exe 615
- Tasklist.exe 611, 613
- Task-Manager
 - Benutzer überwachen 168
 - Übersicht der Prozesse 610
- Taskplaner 735
- TCP/IP-Protokoll 437
 - Aktive Verbindungen abfragen 464
 - Aktuelle Konfiguration ermitteln 437
 - ARP-Protokoll 460
 - DHCP-Funktion aktivieren 446
 - DNS-Einstellungen 450
 - ICMP-Protokoll 460
 - IGMP-Protokoll 460
 - IP-Adresse erneuern 448
 - IP-Adresse lösen 448
 - IP-Adresse statisch vergeben 440
 - IP-Protokoll 460
 - Standardgateway festlegen 443
 - TCP/IP-Filter 458
 - TCP-Protokoll 460
 - UDP-Protokoll 460
 - WINS-Einstellungen 455
- Telnet 459
- Time 584
- Time-Funktion 585
- Touch.exe 277

U

Uhrzeit

- abfragen 582
- ändern 582

Umgebungsvariablen 90

- abfragen 93, 639
- ändern 642
- anzeigen 639
- auslesen 639
- bearbeiten 93, 642
- hinzufügen 642
- löschen 93, 642

Unattend.txt 36, 40

Uniq.exe 339

Uptime 579

- ermitteln 579

Uptime.exe 580

V

VBScript 109

Verschlüsselung 298

Vertrauensstellungen 683

- abfragen 683
- einrichten 687
- entfernen 692
- überprüfen 690

Volltextsuche 321, 330

Volume 382, 398

- anlegen 398
- einfaches Volume 382, 398
- erstellen 398
- formatieren 405
- gespiegeltes Volume 382, 398
- löschen 402
- RAID5-Volume 382, 398
- Stripeset-Volume 382, 398
- übergreifendes Volume 382, 398

W

Wc.exe 347ff.

Well-Known Ports 458

Whoami.exe 169, 182

Win32_BaseService-Klasse, Create-Methode 624

Win32_BootConfiguration-Klasse 368

- BootDirectory-Eigenschaft 368
- Caption-Eigenschaft 368
- ConfigurationPath-Eigenschaft 368
- Description-Eigenschaft 368

LastDrive-Eigenschaft 368

Name-Eigenschaft 368

ScratchDirectory-Eigenschaft 368

SettingID-Eigenschaft 368

TempDirectory-Eigenschaft 368

Win32_ComputerSystem-Klasse 578

Caption-Eigenschaft 578

CurrentTimeZone-Eigenschaft 578

DayLightInEffect-Eigenschaft 578

Description-Eigenschaft 578

Domain-Eigenschaft 578

DomainRole-Eigenschaft 578

EnableDaylightSavingsTime-Eigenschaft 578

InstallDate-Eigenschaft 578

KeyboardPasswordStatus-Eigenschaft 578

Manufacturer-Eigenschaft 578

Modell-Eigenschaft 578

Name-Eigenschaft 578

NumberOfProcessors-Eigenschaft 578

OEMStringArray-Eigenschaft 578

PartOfDomain-Eigenschaft 578

PowerManagementSupported-Eigenschaft 578

PrimaryOwnerContact-Eigenschaft 578

StartupOptions-Eigenschaft 578

SystemStartupDelay-Eigenschaft 578

SystemStartupSetting-Eigenschaft 578

SystemType-Eigenschaft 578

TotalPhysicalMemory-Eigenschaft 578

UserName-Eigenschaft 578

Win32_Computersystem-Klasse

JoinDomainOrWorkgroup-Methode 570

Übersicht der Eigenschaften 805

UnJoinDomainOrWorkgroup-Methode 574

Win32_Directory-Klasse 232

Compress-Eigenschaft 295

CreationDate-Eigenschaft 271

LastModified-Eigenschaft 275

Rename-Methode 291

TakeOwnership-Methode 255

Übersicht der Eigenschaften 794

Uncompress-Eigenschaft 295

Win32_DiskDrive-Klasse 376

Caption-Eigenschaft 376

Partitions-Eigenschaft 376

Size-Eigenschaft 376

Status-Eigenschaft 376

StatusInfo-Eigenschaft 376

- Win32_DiskPartition-Klasse 381
 - Caption-Eigenschaft 381
 - Size-Eigenschaft 381
 - Type-Eigenschaft 381
 - Übersicht der Eigenschaften 798
- Win32_Environment-Klasse 641
 - Name-Eigenschaft 641
 - VariableValue-Eigenschaft 641
 - VariableValue-Methode 645
- Win32_Group-Klasse 225
 - SID-Eigenschaft 225
 - Übersicht der Eigenschaften 794
- Win32_LocalTime-Klasse 585
 - Day-Eigenschaft 585
 - DayOfWeek-Eigenschaft 585
 - Hour-Eigenschaft 585
 - Minute-Eigenschaft 585
 - Month-Eigenschaft 585
 - Quarter-Eigenschaft 585
 - Second-Eigenschaft 585
 - WeekInMonth-Eigenschaft 585
 - Year-Eigenschaft 585
- Win32_LogicalDisk-Klasse 364f.
 - Caption-Eigenschaft 364f., 378
 - Description-Eigenschaft 364f.
 - FileSystem-Eigenschaft 364f., 378
 - FreeSpace-Eigenschaft 364f., 378
 - Size-Eigenschaft 364f., 378
 - Übersicht der Eigenschaften 796
 - VolumeName-Eigenschaft 364f., 415
 - VolumeSerialNumber-Eigenschaft 364f.
- Win32_LogicalFileSecuritySetting
 - GetSecurityDescriptor-Methode 240
 - SetSecurityDescriptor-Methode 248
- Win32_NetworkAdapterConfiguration
 - EnableIPFilterSec-Methode 461f.
 - EnableIPSec-Methode 461f.
- Win32_NetworkAdapterConfiguration-Klasse 431
 - Caption-Eigenschaft 431, 434, 436, 439
 - DefaultIPGateway-Eigenschaft 431, 439
 - Description-Eigenschaft 431
 - DHCPEnabled-Eigenschaft 431, 439
 - DHCPLeaseExpires-Eigenschaft 439
 - DHCPLeaseObtained-Eigenschaft 439
 - DHCPServer-Eigenschaft 431, 439
 - DNSDomain-Eigenschaft 431
 - DNSDomainSuffixSearchOrder-Eigenschaft 431
- DNSEnabledForWINSResolution-Eigenschaft 431
- DNSHostname-Eigenschaft 431
- DomainDNSRegistrationEnabled-Eigenschaft 431
- EnableDHCP-Methode 447
- EnableDNS-Methode 452, 454
- EnableStatic-Methode 441f.
- EnableWINS-Methode 456f.
- FullDNSRegistrationEnabled-Eigenschaft 431
- Index-Eigenschaft 431
- IPAddress-Eigenschaft 431, 439
- IPEnabled-Eigenschaft 431
- IPFilterSecurityEnabled-Eigenschaft 431
- IPPortSecurityEnabled-Eigenschaft 431
- IPSecPermitIPProtocols-Eigenschaft 431
- IPSecPermitTCPPorts-Eigenschaft 431
- IPSecPermitUDPPorts-Eigenschaft 431
- IPSubnet-Eigenschaft 431, 439
- MACAddress-Eigenschaft 431, 436
- ReleaseDHCPLease-Methode 449
- RenewDHCPLease-Methode 449
- SetDynamicDNSRegistration-Methode 452, 454
- SetGateways-Methode 444f.
- SetTCPIPNetBIOS-Methode 456f.
- SetWINSServer-Methode 456f.
- Übersicht der Eigenschaften 800
- WINSEnableLMHostsLookup-Eigenschaft 431
- WINSHostLookupFile-Eigenschaft 431
- WINSPrimaryServer-Eigenschaft 431
- WINSSecondaryServer-Eigenschaft 431
- Win32_NTEventLogFile-Klasse
 - BackupEventLog-Methode 607
 - MaxFileSize-Eigenschaft 605
 - OverwriteOutDated-Eigenschaft 605
- Win32_NTLogEvent-Klasse 602
 - Category-Eigenschaft 536
 - ComputerName-Eigenschaft 536
 - EventCode-Eigenschaft 536
 - Message-Eigenschaft 536
 - SourceName-Eigenschaft 536
 - TimeGenerated-Eigenschaft 536
 - Übersicht der Eigenschaften 804
 - User-Eigenschaft 536
- Win32_OperatingSystem-Klasse
 - LastBootUpTime-Eigenschaft 581

- Übersicht der Eigenschaften 790
- Win32ShutDown-Methode 172, 648
- Win32_Process-Klasse 611
 - Caption-Eigenschaft 614
 - ExecutablePath-Eigenschaft 611
 - Name-Eigenschaft 611
 - ProcessID-Eigenschaft 611
 - SetPriority-Methode 619
 - Terminate-Methode 616
 - Übersicht der Eigenschaften 808
 - WorkingSetSize-Eigenschaft 614
- Win32_Proxy, SetProxySetting-Methode 55
- Win32_ScheduledJob-Klasse
 - Create-Methode 745, 750
 - Delete-Methode 750, 760
 - ExecQuery-Methode 738
 - Übersicht der Eigenschaften 811
- Win32_Service-Klasse 621
 - AcceptPause-Eigenschaft 621
 - AcceptStop-Eigenschaft 621
 - Caption-Eigenschaft 621
 - Change-Methode 636
 - Checkpoint-Eigenschaft 621
 - CreationClassName-Eigenschaft 621
 - Delete-Methode 627
 - Description-Eigenschaft 621
 - DesktopInteract-Eigenschaft 621
 - DisplayName-Eigenschaft 621
 - ErrorControl-Eigenschaft 621
 - ExitCode-Eigenschaft 621
 - InstallDate-Eigenschaft 621
 - Name-Eigenschaft 621
 - PathName-Eigenschaft 621
 - PauseService-Methode 630
 - ProcessID-Eigenschaft 621
 - ResumeService-Methode 630
 - ServiceSpecificExitCode-Eigenschaft 621
 - ServiceType-Eigenschaft 621
 - Started-Eigenschaft 621
 - StartMode-Eigenschaft 621
 - StartName-Eigenschaft 621
 - StartService-Methode 633
 - State-Eigenschaft 621
 - Status-Eigenschaft 621
 - StopService-Methode 627, 633
 - SystemCreationClassName-Eigenschaft 621
 - SystemName-Eigenschaft 621
 - TagID-Eigenschaft 621
 - Übersicht der Eigenschaften 810
 - WaitHint-Eigenschaft 621
- Win32_Share-Klasse 467f.
 - AccessMask-Eigenschaft 467
 - AllowMaximum-Eigenschaft 467f.
 - Caption-Eigenschaft 467f.
 - Create-Methode 473
 - Delete-Methode 488
 - Description-Eigenschaft 467
 - InstallDate-Eigenschaft 467
 - MaximumAllowed-Eigenschaft 467f.
 - Name-Eigenschaft 467f.
 - Path-Eigenschaft 467f.
 - SetShareInfo-Methode 475
 - Status-Eigenschaft 467f.
 - Type-Eigenschaft 467f.
 - Übersicht der Eigenschaften 804
- Win32_TimeZone-Klasse
 - Caption-Eigenschaft 586
 - DayLightName-Eigenschaft 586
- Win32_UserAccount-Klasse 182
 - SID-Eigenschaft 182
 - Übersicht der Eigenschaften 793
- Win32_Volume-Klasse
 - AddMountPoint-Methode 413
 - Defrag-Methode 420
 - DeviceID-Eigenschaft 412
 - DriveLetter-Eigenschaft 412
 - Format-Methode 408
- Win32_WindowsProductActivation
 - ActivateOnline-Methode 55
 - ActivationRequired-Eigenschaft 55
- Win32_WindowsProductActivation-Klasse,
 - Übersicht der Eigenschaften 787
- Windiff.exe 316
- Windows Explorer
 - Datenträger-Eigenschaften 376
 - Details einblenden 283
 - Spalten auswählen 283
- Windows Script Host 109
- Windows Server 2003 Resource Kit-
 - Tools 104f.
- Windows Server 2003 Support-Tools 104f.
- Windows Server Update-Service 66
 - Systemvoraussetzungen 66
- Windows Services für Unix 104, 106
- Windows Update-Service 591
- Winnt.sif 40
- WinNT-Provider
 - AccountDisabled-Eigenschaft 151, 161
 - AccountExpirationDate-Eigenschaft 151
 - Add-Methode 208

- Delete-Methode 167, 223
 - Description-Eigenschaft 151, 198, 203
 - Eigenschaften des User-Objekts 787
 - Filter-Eigenschaft 128, 189
 - FullName-Eigenschaft 137, 151
 - Groups-Eigenschaft 151
 - HomeDirectory-Eigenschaft 151
 - LastLogin-Eigenschaft 151, 177
 - LastLogoff-Eigenschaft 151
 - LoginHours-Eigenschaft 151
 - LoginScript-Eigenschaft 151, 157
 - LoginWorkstations-Eigenschaft 151
 - Members-Eigenschaft 203, 216
 - MoveHere-Methode 164
 - Name-Eigenschaft 133, 189, 194, 203
 - PasswordExpirationDate-Eigenschaft 151
 - Profile-Eigenschaft 151
 - Put-Methode 137, 198
 - Remove-Methode 219
 - SetPassword-Eigenschaft 151
 - SetPassword-Methode 137, 185
 - WINS 455
 - WinVi32.exe 321, 330, 333, 343
 - WMI 117
 - Wmic.exe 95
 - Alias 96
 - BootConfig-Alias 368
 - Call 101
 - Datafile 229
 - DataFile-Alias 229, 262, 266, 273, 284
 - Environment-Alias 640
 - FSDir-Alias 229, 273, 284
 - Get 100
 - Group-Alias 224
 - List 100
 - LogicalDisk 364
 - LogicalDisk-Alias 377
 - NicConfig 456, 461
 - NicConfig-Alias 431, 436, 438, 441, 444, 447, 449, 452
 - NTEvent-Alias 535, 602
 - NTEventLog-Alias 605, 607
 - OS-Alias 172, 580
 - Partition-Alias 381
 - Process-Alias 611, 613, 616, 618
 - Product-Alias 597
 - Schalter 103
 - Service-Alias 621
 - Share 472, 487
 - Share-Alias 467, 475
 - TimeZone-Alias 584
 - UserAccount-Alias 163, 182
 - Volume-Alias 415
 - Where-Klausel 101
 - WPA 53
 - WQL-Abfrage 96
 - WScript.exe 110, 114
 - WSH 109
 - aktivieren 120
 - deaktivieren 120
 - Skript erzeugen 111
 - Skriptdateien ausführen 113
 - Skript-Eigenschaften 114
 - Standard-Host festlegen 114
 - WSHNetwork-Objekt
 - MapNetworkDrive-Methode 485
 - RemoveNetworkDrive-Methode 486
 - WSH-Referenz 109
- X**
- Xcopy.exe 287, 302
- Z**
- Zeitzone
 - abfragen 582
 - auswählen 582
 - Zip-Drive 355
 - Zugriffskontrollliste 233, 243
 - Zwischenablage 286