

Windows NT ist als Netzwerkbetriebssystem nicht zuletzt wegen seiner komfortablen graphischen Benutzeroberfläche sehr beliebt. Nichts desto weniger ergeben sich immer wieder Situationen, in denen Kommandozeilen-Werkzeuge Vorteile gegenüber graphischen Pendanten besitzen. Geringer Bandbreitenbedarf bei Fernwartung, Skriptfähigkeit und daraus resultierend eine geringere Fehleranfälligkeit durch die Automatisierung von Aufgaben zeichnen die textbasierenden Administrationswerkzeuge aus.

VON DIRK PELZER

Skriptbefehle unter Windows-NT

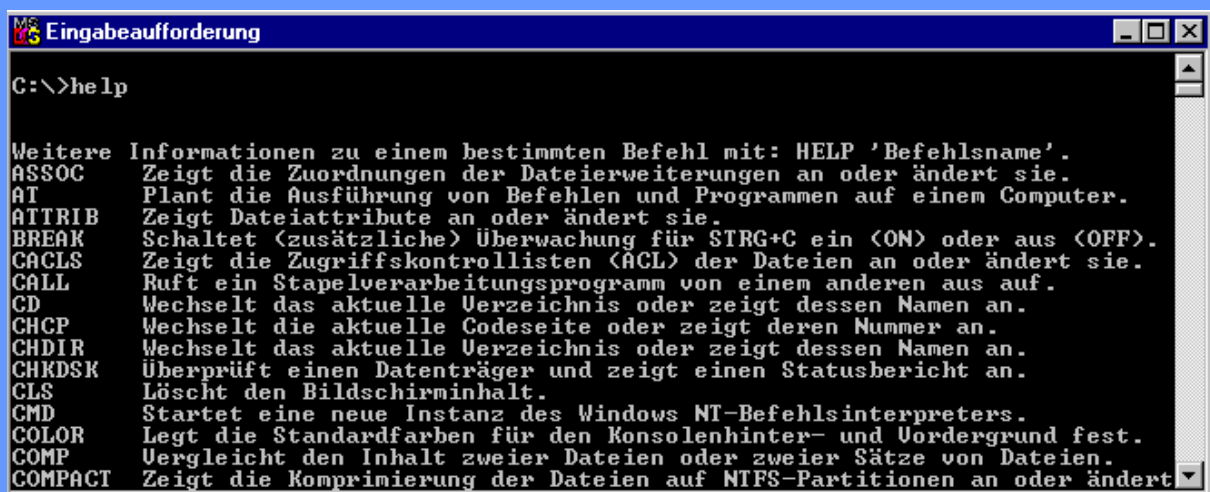
Die Skriptbefehle, die unter Windows-NT verwendet werden können, sind, was Leistungs-umfang und Syntax anbelangt der Version 5.0 von MS-DOS entlehnt. Wer also noch im Besitz von Literatur zu MS-DOS ist, kann sich dort schon einmal über die entsprechenden Kommandos informieren. Wer nicht mehr über solche historischen Quellen verfügt, dem stellt NT eine Hilfedatei zu Verfügung, in der alle unterstützten Befehle aufgelistet und beschrieben sind. Die Kombination dieser Skriptbefehle mit verschiedenen Kommando-zeilenwerkzeugen, die teilweise im Lieferumfang von NT, teilweise im NT Resource-Kit und teilweise von Drittanbietern verfügbar sind, ergibt ein Toolset, mit dem man be-

reits etliche administrative Arbeiten erledigen kann.

Um eine Liste aller verfügbaren Skriptbefehle zu erhalten, muß man auf einer Kommandozeile HELP eintippen. Dieses liefert neben der Liste der möglichen Befehle noch eine kurze Erklärung, was das jeweilige Kommando bewirkt. Wenn man das für den jeweiligen Einsatzzweck passende Kommando gefunden hat und genauere Informationen zur Syntax benötigt, gibt man in einem DOS-Fenster das Kommando HELP <Befehlsname>, wobei <Befehlsname> durch das entsprechende Kommando zu ersetzen ist. Beispiel: Die Eingabe des Kommandos HELP TYPE liefert folgende Information zurück:

[Zeigt den Inhalt einer oder mehrerer Textdateien an.](#)
TYPE [Laufwerk:][Pfad]Dateiname

Durch Eingabe von HELP erhält man eine Liste der unter NT verfügbaren Skriptbefehle



```

C:\>help

Weitere Informationen zu einem bestimmten Befehl mit: HELP 'Befehlsname'.
ASSOC   Zeigt die Zuordnungen der Dateierweiterungen an oder ändert sie.
AT      Plant die Ausführung von Befehlen und Programmen auf einem Computer.
ATTRIB  Zeigt Dateiattribute an oder ändert sie.
BREAK   Schaltet (zusätzliche) Überwachung für STRG+C ein (ON) oder aus (OFF).
CACLS   Zeigt die Zugriffskontrolllisten (ACL) der Dateien an oder ändert sie.
CALL    Ruft ein Stapelverarbeitungsprogramm von einem anderen aus auf.
CD      Wechselt das aktuelle Verzeichnis oder zeigt dessen Namen an.
CHCP    Wechselt die aktuelle Codeseite oder zeigt deren Nummer an.
CHDIR   Wechselt das aktuelle Verzeichnis oder zeigt dessen Namen an.
CHKDSK  überprüft einen Datenträger und zeigt einen Statusbericht an.
CLS     Löscht den Bildschirminhalt.
CMD     Startet eine neue Instanz des Windows NT-Befehlsinterpreters.
COLOR   Legt die Standardfarben für den Konsolenhinter- und Vordergrund fest.
COMP    Vergleicht den Inhalt zweier Dateien oder zweier Sätze von Dateien.
COMPACT Zeigt die Komprimierung der Dateien auf NTFS-Partitionen an oder ändert
  
```

Kommandozeilen-Befehle

Im Gegensatz zu den Skriptbefehlen, ist es bei den Kommandozeilenbefehlen nicht ganz so einfach, eine Liste der unterstützten Kommandos zu erhalten. Fündig wird man in der Hilfedatei zu Windows-NT, auf

die man über die Menüleiste durch Klicken auf Start und anschließend Hilfe zugreifen kann. Im sich daraufhin öffnenden Fenster, wählt man den Eintrag Windows-NT-Befehle und erhält eine Liste aller unterstützten Kommandozeilen-Befehle einschließlich der Skriptbefehle.

Für die netzwerkseitige Administration zum Beispiel existiert ein Kommandozeilen-Werkzeug namens NET.EXE, mit dem sich einige wichtige Aufgaben, wie etwa das Hinzufügen oder Löschen von Benutzern ohne graphische Werkzeuge ausführen lassen. Um eine Liste aller verfügbaren Optionen von NET.EXE zu erhalten, gibt man in einem DOS-Fenster das Kommando NET /? ein. Möchte man noch genauere Informationen haben, so gibt man NET HELP <Kommando> ein, wobei <Kommando> durch den gewünschten Befehl zu ersetzen ist. So liefert zum Beispiel NET HELP USE eine Erklärung für den Befehl NET USE, mit dem sich eine Netzwerkverbindung zu einem Server herstellen läßt.

NET.EXE stellt sicherlich eines der interessantesten Werkzeuge für die Administration über Kommandozeile dar, da es beinahe alle notwendigen Funktionen zur Benutzer-, Datei- und Dienstverwaltung beinhaltet. Darüber hinaus gibt es aber noch eine ganze Reihe weiterer interessanter Tools aus dem Bereich Netzwerke, wie zum Beispiel PING, TRACERT, NSLOOKUP oder IPCONFIG. Auf der anderen Seite fehlen jedoch Werkzeuge, um beispielsweise Meldungen aus der Ereignisanzeige oder der Registry über Kommandozeile auslesen zu können. Hierbei ist man entweder auf das NT Resource Kit oder Drittanbieter angewiesen.

Administration am praktischen Beispiel

Um die Administration über Kommandozeilen an einem praktischen Beispiel zu demonstrieren, wird nachfolgend ein kleines Skript entwickelt, mit dem der Benutzerdatenbank eines NT-Servers neue Anwender hinzugefügt werden sollen. Die Benutzer sollen dabei in einer Textdatei gespeichert sein, welche als Eingabedatei für das eigentliche Skript dient. Das Start-Kennwort für alle Anwender soll identisch mit der Benutzerkennung sein. Zusätzlich soll jedem Benutzer-namen ein vollständiger Name mitgegeben werden.

Ein solches Skript läßt sich im Wesentlichen mit zwei unterschiedlichen Befehlen realisieren. Es

handelt sich dabei um die Befehle NET USER und FOR.

Mit dem Kommando NET USER lassen sich auf einem NT-Rechner Benutzerkonten einrichten, oder löschen. NET HELP USER zeigt alle verfügbaren Parameter an. Für das zu realisierende Skript, wird folgende Syntax benötigt:

```
NET USER <Benutzer-ID> <Kennwort> /ADD /FULLNAME:<Benutzername>.
```

Alle in <> stehenden Werte sollen über die einzulesende Textdatei eingegeben werden, wobei Benutzer-ID und Kennwort identisch sein sollen. Dazu müssen die Parameter <Benutzer-ID>, <Kennwort> und <Benutzername> durch Variable ersetzt werden.

Das modifizierte Kommando sieht dann folgendermaßen aus:

```
NET USER %%i %%i /ADD /FULLNAME:%%j.
```

Es ist zu beachten, daß Variable, wenn sie in Skript-Dateien verwendet werden sollen mit zwei vorangestellten Prozent-Zeichen zu versehen sind.

Um mehrere Befehle hintereinander ausführen zu können, wird eine Kontrollstruktur, wie zum Beispiel das aus Programmiersprachen bekannte FOR oder WHILE benötigt. Die einzige unter der NT-Skriptsprache verfügbare Kontrollstruktur ist FOR. Wenn man in einem DOS-Fenster HELP FOR eingibt, erhält man eine sehr ausführliche Beschreibung der Syntax des FOR-Befehls. Diese mag auf den ersten Blick etwas verwirrend aussehen, entpuppt sich aber auf den zweiten Blick als sehr mächtige Funktion. Für das zu erstellende Skript wird FOR mit folgender Syntax benötigt:

```
FOR /F ["Optionen"] %%Variable IN (Fileset) DO Befehl [Kommando-Parameter]
```

Die Leistung, die dieses zugegebenermaßen abenteuerlich anmutende Konstrukt erledigt, besteht darin, die unter Fileset angegebenen Dateien der Reihe nach zu öffnen und deren Inhalt zeilenweise an %%Variable zu übergeben. %%Variable kann dann wiederum als Parameter für den auszuführenden Befehl dienen. Über die Optionen läßt sich festlegen, wie viele sogenannte Tokens einer Zeile als Variable zur Verfügung stehen sollen. In dem zu erstellenden

Skript werden für das NET USER-Kommando zwei Parameter benötigt, die aus der Eingabedatei zu übergeben sind.

Die Syntax des FOR-Befehls stellt sich damit folgendermaßen dar:

```
FOR /F "tokens=1,2" %%i in (USERS.TXT) DO  
NET USER %%i %%i /add /FULLNAME:%%j
```

Die Option tokens=1,2 bedeutet dabei nichts anderes, als daß zwei Einträge einer eingelesenen Zeile der Eingabedatei als Variable übergeben werden sollen. Die erste Variable hat dabei die Bezeichnung %%i, die zweite die Bezeichnung %%j.

In Pseudocode formuliert, macht das oben stehende FOR-Kommando folgendes:

- Datei USERS.TXT öffnen
- Wiederhole bis Dateiende von USERS.TXT erreicht
- Lies zeilenweise zwei Parameter ein und übergebe sie an die Variablen %%i und %%j
- Führe das Kommando NET USER aus und verwende %%i und %%j als Parameter

Um also die von dem Skript gewünschte Funktion zu implementieren, ist lediglich eine einzige Kommandozeile erforderlich, die folgendermaßen aussieht:

```
FOR /F "tokens=1,2" %%i in (USERS.TXT) DO  
NET USER %%i %%i /add /FULLNAME:%%j
```

Die Eingabedatei USERS.TXT muß folgendes Aussehen haben, damit sie von diesem Skript bearbeitet werden kann:

```
user1 Benutzer1  
user2 Benutzer2  
user3 Benutzer3  
user4 Benutzer4  
user5 Benutzer5
```

Dabei wird userx als Variable %%i für die Benutzer-ID und Kennwort verwendet und Benutzerx als Variable %%j.

Das ganze Skript läßt sich dahingehend erweitern, daß man die einzulesende Datei dem Skript als Parameter übergibt. Auf diese Weise kann man das Skript mit beliebigen Eingabedateien verwenden. Dabei gilt es zu beachten, daß sich einer Skriptdatei bis zu neun Parameter übergeben lassen, auf die

innerhalb des Skriptes über die Bezeichner %1, %2, %3, etc. zugegriffen werden kann. Dabei entspricht %1 dem ersten übergebenen Parameter ist, %2 dem zweiten und so weiter. In %0 ist der Name der ausgeführten Skripts gespeichert. Wenn als zu Beispiel ADDUSER.CMD USERS.TXT ausgeführt wird, so enthält %0 ADDUSER und %1 USERS.TXT. Wie sich das bisher erstellte Skript unter Berücksichtigung von Aufruf-Parametern realisieren läßt, ist in Listing 1 auf Seite 3/4 erkennbar.

Listing 1

```
CLS
@ECHO OFF
REM *****
REM
REM SKRIPTNAME:   ADDUSER.CMD
REM PARAMETER:   Eingabedatei
REM
REM Aufruf ADDUSER <Textdatei>
REM
REM Skript, das aus einer Textdatei
REM neu anzulegende NT-Benutzer liest
REM und entsprechende Konten erstellt.
REM
REM (c) Dirk Pelzer 1997
REM *****
IF "%1" == "" GOTO ERROR
@ECHO ON
FOR /F "tokens=1,2" %%i in (%1) DO NET
USER %%i %%i /add /FULLNAME:%%j
@ECHO OFF
GOTO ENDE
```

```
ERROR
```

```
REM *****
REM Kein Parameter übergeben
REM *****
```

```
@ECHO Fehler: Es wurde kein Parameter
mit uebergeben.
```

```
@ECHO Die korrekte Syntax lautet: %0
Textadtei
```

Erweiterung des Skriptes

Nun ist die administrative Arbeit zumeist mit dem Anlegen der Benutzer nicht erledigt. Vielmehr müssen Benutzer Gruppen zugeordnet werden und gegebenenfalls neue Gruppen angelegt werden. Eine auch immer wieder häufig anzutreffende Arbeit besteht darin, jedem Benutzer ein Basisverzeichnis (Home Directory) zuzuweisen. Um zu veranschaulichen, wie diese Schritte über die Kommandozeile gelöst werden können, soll das bereits erstellte Skript entsprechend erweitert werden. Folgende Aufgaben sind zu erledigen:

1. Jeder Benutzer soll Mitglied der globalen Gruppe Marketing werden, wobei diese Gruppe noch nicht existiert und zunächst angelegt werden muß.
2. Jeder Benutzer soll ein Basisverzeichnis erhalten, das seinem Benutzernamen entspricht. In diesem Verzeichnis soll der Benutzer selbst nur Änderungsrecht haben, nicht jedoch andere Benutzer.
3. Schließlich soll das Basisverzeichnis jedes Benutzers freigegeben werden, so daß der Anwender eine Netzwerkverbindung aufbauen kann.

Um unter Windows-NT eine globale Benutzergruppe anzulegen, verwendet man das Kommando NET GROUP mit entsprechenden Parametern. Im Beispiel würde sich folgender Befehl anbieten:

```
NET GROUP Marketing /ADD
/COMMENT:"Lokale Gruppe fuer Marketing"
/DOMAIN
```

Dieser erstellt die globale Gruppe Marketing mit dem Kommentar »Globale Gruppe fuer Marketing« in der Benutzerdatenbank der Domäne, in der das Skript ausgeführt wird.

Über NET GROUP lassen sich der globalen Gruppe auch die dazugehörigen Benutzer hinzufügen. Dazu verwendet man folgende Syntax:

```
NET GROUP <Gruppenname> <Benutzer-ID>
/ADD /DOMAIN.
```

Da der Gruppe mehrere Benutzer hinzugefügt werden sollen, bietet es sich an, wieder das FOR-Kommando mit USERS.TXT als Eingabedatei zu verwenden. Dazu muß beim NET GROUP-Kommando der Parameter Benutzer-ID durch eine Variable ersetzt werden. Die Syntax lautet damit:

```
FOR /F %i IN (USERS.TXT) DO NET GROUP
Marketing %i /ADD /DOMAIN
```

Möchte man anstelle einer globalen Gruppe eine lokale Gruppe erstellen, so verwendet man anstelle von NET GROUP das Kommando NET LOCALGROUP, welches aber ansonsten über identische Syntax verfügt.

Um für jeden Benutzer ein Basisverzeichnis anzulegen, verwendet man das Kommando MD oder das gleichwertige MKDIR. Die Benutzer sollen ein Verzeichnis gemäß ihrer Benutzer-ID bekommen. Alle Verzeichnisse sollen Unterverzeichnisse von D:\USERS werden. Auch hier bietet es sich wieder an, USERS.TXT als Eingabedatei in einer FOR-Schleife zu verwenden. Die Syntax lautet entsprechend:

```
FOR /F %i IN (USERS.TXT) DO MD D:\USERS
\%i
```

Um in den Verzeichnissen die entsprechenden Berechtigungen zu setzen, verwendet man das

Kommando CACLS. Mit dessen Hilfe lassen sich die Access-Control-Listen (ACL) von Dateien und Verzeichnissen modifizieren. Die Syntax für CACLS lautet folgendermaßen:

```
<Verzeichnis> /E /C /T /G <Benutzer-ID>:<Recht>
```

Dabei ist unter <Verzeichnis> anzugeben, auf welches Verzeichnis die Rechteänderung angewendet werden soll. Mit dem Parameter /E wird CACLS mitgeteilt, daß die ACL editiert werden soll anstatt sie zu ersetzen. Ohne /E würden vorhandene Berechtigungen einfach überschrieben werden. Die Option /C sorgt dafür, daß CACLS weiterläuft, auch wenn die Fehlermeldung »Zugriff verweigert« auftritt und ist in Batchprogrammen notwendig um zu verhindern, daß das Skript angehalten wird, bis ein Benutzer die Fehlermeldung quittiert. Der etwas ungewöhnlich Parameter /T bedeutet, daß die Rechteänderung auf das angegebene Verzeichnis, sowie alle Unterverzeichnisse angewendet werden soll. Der skripterfahrene Benutzer hätte hier wohl eher den Parameter /S erwartet. Über /G <Benutzer-ID>:<Recht> schließlich wird angegeben, welcher Benutzer welches Zugriffsrecht erhält. Dabei bedeuten »F« Vollzugriff, »C« Ändern, und »R« Lesen.

Die Syntax für die FOR-Schleife lautet somit:

```
FOR /F %%i IN (USERS.TXT) DO CACLS
D:\USERS\%%i /E /C /T /G %%i:C
```

Um schließlich die angelegten Verzeichnisse freizugeben, setzt man wieder ein NET-Kommando ein. Der Befehl zu Freigabe eines Verzeichnisses lautet:

```
NETSHARE<Freigabename>=<Laufwerk:Pfad>
REMARK:"<Bemerkung>"
```

Wenn man das ganze wieder in eine FOR-Schleife verpackt sieht der Befehl folgendermaßen aus:

```
FOR /F %%i IN (USERS.TXT) DO NET SHARE
%%i=D:\USERS\%%i /REMARK:"Home
Benutzer %%i"
```

Das vollständige Skript findet sich in Listing 2 auf Seite 5/6.

Listing 2

```
@ECHO OFF
CLS

REM *****
REM
REM SKRIPTNAME:   ADDUSER.CMD
REM PARAMETER:   Eingabedatei
REM
REM Aufruf ADDUSER <Textdatei>
REM
REM Skript, das eine globale Gruppe,
Benutzerkonten,
REM und Homedirectories anlegt, sowie
Verzeichnis-
REM rechte setzt und die
Homedirectories freigibt.
REM
REM (c) Dirk Pelzer 1997
REM
REM *****
IF "%1" == "" GOTO ERROR

REM *****
REM Lege globale Gruppe Marke-
REM ting in der Domäne an
REM *****

NET GROUP Marketing /ADD
/COMMENT:"Loakle Gruppe fuer
Marketing" /DOMAIN

REM *****
REM Lege Domänen-Benutzer aus
```

```

REM Eingabe-Datei an
REM *****

FOR /F "tokens=1,2" %%i in (%1) DO NET
USER %%i %%i /add /FULLNAME:%%j

REM *****
REM Füge Benutzer der Gruppe
REM Marketing hinzu
REM *****

FOR /F %%i IN (%1) DO NET GROUP
Marketing %%i /ADD /DOMAIN

REM *****
REM Lege Homedirectory für Benutzer an
REM *****

FOR /F %%i IN (%1) DO MD D:\USERS\%%i

REM *****
REM Setze Berechtigungen
REM *****

FOR /F %%i IN (%1) DO CACLS
D:\USERS\%%i /E /C /T /G %%i:C

REM *****
REM Verzeichnisfreigabe
REM *****

FOR /F %%i IN (%1) DO NET SHARE
%%i=D:\USERS\%%i /REMARK:"Home
Benutzer %%i"

```

```
GOTO ENDE
```

```
:ERROR
```

```

REM *****
REM Kein Parameter übergeben
REM *****

```

```
@ECHO Fehler: Es wurde kein Parameter
mit uebergeben.
```

```
@ECHO Die korrekte Syntax lautet: %0
Textadtei
```

```
:ENDE
```

Abschließend noch ein Anmerkung zum Thema Benutzerkennungen anlegen mit NET.EXE. Je nachdem, wo ein Skript ausgeführt wird, das mit Benutzerkennungen arbeitet und Rechtevergaben durchführt kann, es passieren, daß möglicherweise die Vergabe von Zugriffsrechten durch CACLS fehlschlägt, obwohl die Benutzer im selben Skript kurz zuvor angelegt worden sind. Wenn beispielsweise das Skript auf einem Backup-Domain-Controller (BDC) ausgeführt wird, werden die Benutzerkennungen zwar richtig angelegt, sind jedoch zunächst nur in der Benutzerdatenbank des Primary-Domain-Controllers (PDC) bekannt. Backup-Domain-Controller erhalten diese Änderungen erst beim nächsten Replikations-intervall. Da aber ein BDC für die Rechtevergabe seine eigene Benutzerdatenbank verwendet, kann es passieren, daß ein gerade hinzugefügter Benutzer noch nicht bekannt ist und es zu einer Fehlermeldung bei Versuch kommt, einem solchen Benutzer Rechte zuzuweisen. Um das Problem zu beseitigen, sollte man, nachdem man die Benutzer mit dem Kommando NET USER angelegt hat, den Befehl NET ACCOUNTS /SYNC ausführen. Dieser sorgt dafür, daß der Backup-Domain-Controller, auf dem er ausgeführt wird, seine Benutzerdatenbank mit dem

PDC abgleicht. Auf einem PC ausgeführt, sorgt NET ACCOUNTS /SYNC dafür, daß alle Backup-Domain-Controller ihre Benutzerdatenbank mit dem Primary abgleichen.

Fazit

Mit den Skriptbefehlen von Windows-NT kann man bereits eine ganze Menge administrative Arbeiten erledigen. Allerdings fehlt es zum Teil an wichtigen Werkzeugen, mit denen sich über die Kommandozeile Änderungen in der Registry vornehmen oder einfach nur Einträge auf dem Bildschirm anzeigen

lassen. Auch nach Tools zur Administration von DHCP und WINS sucht man vergeblich. Diese Lücke füllt teilweise das NT-Resource-Kit mit zahlreichen zusätzlichen Werkzeugen. Teilweise sind für Windows-NT auch Portierungen von Skriptsprachen anderer Betriebssysteme verfügbar, wie Perl oder REXX, die wesentlich mehr bieten, als die Original-Microsoft Skriptsprache. Sehr empfehlenswert ist auch ein Shareware-Toolset namens Kix95, das beispielsweise über www.softseek.com aus dem Internet heruntergeladen werden kann und eine ausgezeichnete Erweiterung der Skriptfähigkeit von Windows-NT und Windows 95 darstellt.

Zur Person

DIPL. ING. DIRK PELZER arbeitet als freier Consultant und Journalist in München. Er betreibt ein Storage Labor für verschiedene namhafte Fachzeitschriften. Zudem beschäftigt er sich mit Speichernetzen und Hochverfügbarkeit.