

Hagen Graf

# Jetzt lerne ich Apache 2

## Apache installieren

In diesem Kapitel soll der Apache 2 auf Ihrem Rechner installiert werden. Unter Linux werden wir ihn kompilieren und installieren. Unter Windows wird er mit einem Setup-Programm installiert.

### 3.1 Vorüberlegungen

Ihr Rechner kann Software nur dann ausführen, wenn sie in einer binären Maschinsprache, bestehend aus Einsen und Nullen in den Hauptspeicher geladen wird. Da Menschen nicht effektiv in Maschinsprache programmieren können, werden Programme in einer höheren Programmiersprache wie beispielsweise C geschrieben. Diese für Menschen lesbaren Programmtexte werden dann von einer Übersetzungssoftware (Compiler) in Maschinsprache übersetzt. Jedes Betriebssystem hat eine eigene Maschinsprache und spezielle Methoden, um effektiv mit dem System umzugehen. Deswegen muss ein Programm für jede Betriebssystemplattform neu übersetzt werden.

#### 3.1.1 Binary oder übersetzen?

Sie haben zwei Möglichkeiten, den Apache auf Ihrem System zu installieren.

Sie können eine binäre Distribution (Binary-Version), in der die Software schon als ausführbares Programm enthalten ist, oder den Sourcecode (Quelltext) verwenden.

Wenn Sie die Sourcecode-Variante benutzen, müssen Sie sie vor der Installation übersetzen (kompilieren). Sie müssen also auf Ihrem Rechner aus dem Sourcecode eine eigene ausführbare Version zusammenbauen. Um zu kompilieren, benötigen Sie einen Compiler. Auf Linux-Systemen wird ein Compiler mitgeliefert, auf Windows-Systemen nicht. Kompilierungen sind gerade für Anfänger schwer nachvollziehbar und ein großer Quell für Bedienungsfehler. Um auch Anwendern, die keinen Compiler haben oder ihn nicht benutzen wollen, eine Installation zu ermöglichen, gibt es Binary-Versionen. Eine Binary-Version wurde bereits mit den gängigsten Optionen kompiliert und stellt die komfortabelste Art der Installation dar.

### Warum dann Sourcecode?

Aus dem Sourcecode können Sie sich Ihre ganz persönliche Apache-Version stricken. Das kann viele Vorteile haben. Manche Dinge werden durch eine eigene Kompilierung erst möglich, wie beispielsweise die sichere Übertragung von Daten (SSL-Unterstützung, siehe Kapitel 12.3). Sie können Geschwindigkeitsvorteile durch eine eigene Kompilierung erreichen und Sie können sich eine lauffähige Variante erstellen, wenn es für Ihr Betriebssystem keine binäre Distribution gibt.

Durch den modularen Aufbau des Apache und die Möglichkeit, zur Laufzeit neue Module hinzuladen, können Sie die Binary-Version auch um manche Funktionen erweitern.

## 3.2 Woher nehmen?

Alles, was wir benötigen, findet sich unter <http://httpd.apache.org>. Im Downloadbereich werden die verschiedenen Distributionen angeboten. Die Beispiele, die in diesem Buch Verwendung finden, stehen auch auf der Begleit-CD zur Verfügung.



Auch wenn die Beispiele auf der Buch-CD nicht die allerneuesten Versionen sind, empfehle ich Ihnen trotzdem, sie zu benutzen. Alle Beispiele in diesem Buch beruhen auf diesen Dateien. Wenn Sie nun teilweise andere Versionen nehmen, kann es sein, dass manche Beispiele nicht oder nur teilweise funktionieren. Wenn Sie nach einer Übungsphase die Zusammenhänge der verschiedenen Softwarepakete verstanden haben, können Sie sich daran machen, alle auf den neuesten Stand zu bringen.

## 3.3 Linux

Allen Distributionen gemein ist, dass sie den Apache bereits enthalten, leider in der falschen Version. In der SuSE 8.0 und in der redhat 7.3 finden Sie den Apache in der Version 1.3 vor.

Wir werden uns in diesem Buch mit der Version Apache 2.0 beschäftigen und daher müssen Sie auch unter Linux die Software zunächst installieren.

### 3.3.1 SuSE 8.0 – redhat 7.3

Für die Installation von Software hat sich unter Linux ein Standard etabliert. RPM steht für redhat packet manager. Dieser ermöglicht die Installation von vorkonfigurierten Paketen. Alle Distributionen können über so genannte Paket-Manager-Programme mit Software versorgt werden (siehe Anhang A – Wissenswertes). Da noch keine komfortablen rpm-Pakete von SuSE und redhat für den Apache 2 zur Verfügung stehen, kompilieren wir uns einfach eine Version. Das ist unter Linux nicht schwer.

In den Distributionen gibt es verschiedene Varianten der Linux-Installation (beispielsweise minimal). Welche Variante Sie gewählt haben, spielt für unser Vorhaben keine große Rolle. Es kann aber sein, dass das eine oder andere Softwarepaket fehlt und nachinstalliert werden muss.

Für die Kompilierung des Apache benötigen Sie den GNU-C-Compiler und den make-Befehl. Falls diese Programmpakete noch nicht auf Ihrem Rechner vorhanden ist, loggen Sie sich als User root ein und installieren diese mit dem entsprechenden Installationsprogramm.

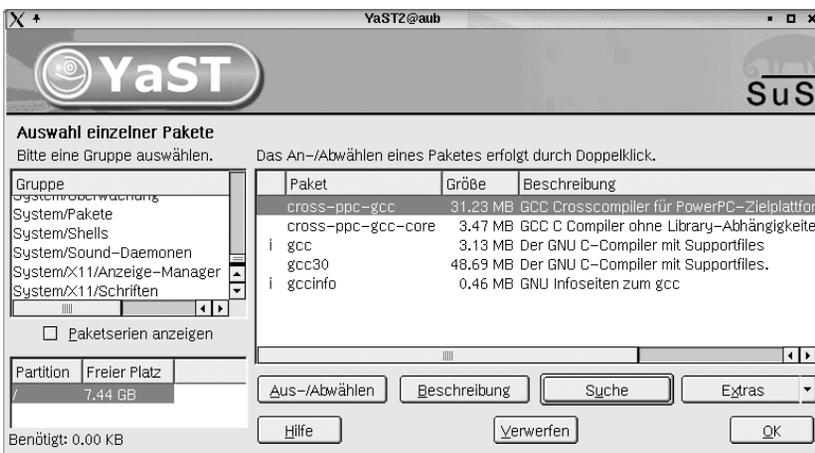


Abb. 3.1:  
SuSE 8.0  
YAST2 –  
Installation  
der Compiler-  
Pakete



Der Befehl `configure` kennt eine ganz Reihe von Optionen. Sie können sie sich mit dem Befehl

```
./configure --help
```

ansehen.

Die Ausführung der Befehle kann je nach Leistung Ihres Rechners ein wenig dauern. Beobachten Sie, was in der Konsole angezeigt wird. Alle drei Befehle sollten ohne Fehlermeldung durchlaufen.

Jetzt müssen Sie den Apache-Server als Dienst (Daemon) auf Ihrem Rechner starten. Dieser Start sorgt dafür, dass der Apache im Hauptspeicher geladen wird und auf Anfragen aus dem Netz hört. Sie stellen für Besucher Ihres Rechners damit die Dienstleistung zur Verfügung, bestimmte Dateien auf Ihrem Rechner abzurufen.

Starten Sie den Apache durch den Aufruf des Startscripts `apachectl` mit dem Befehl

```
/usr/local/apache2/bin/apachectl start
```

## 3.4 Windows

Apache für Windows wurde entwickelt für Windows NT 4.0 oder Windows 2000. Im Normalfall sollte er auch unter Win 95/98 und ME problemlos funktionieren. Allerdings hält die ASF die so genannten Consumersysteme Windows 95/98/ME für grundsätzlich ungeeignet, einen Webserver produktiv zu betreiben. Daher sollte man sich auf diesen Systemen auf Test- und Entwicklungsumgebungen beschränken.

Unter Windows NT 4.0 sollte das letzte Servicepack SP6a installiert sein (unter Windows 2000 entsprechend zurzeit das Servicepack 2). Unter Windows 95 muss die Winsock2-Umgebung installiert sein <http://www.microsoft.com/windows95/downloads/>.

Die hier benutzte und auf der Buch-CD enthaltene Version des Apache für Windows heißt `apache_2.0.35-win32-x86-no_ssl.msi`.

Diese Distribution ist eine Binary-Variante. Unter Windows wird leider kein Compiler mitgeliefert, sondern er muss separat erworben werden.

Um den Apache zu installieren, benötigen Sie den Microsoft Installer Version 1.10. Dieses Programm muss vor der Apache-Installation auf Ihrem PC installiert sein. Unter Windows 2000 und Windows ME ist der Installer bereits vorhanden

Download Installer Windows NT 4.0

<http://www.microsoft.com/downloads/release.asp?ReleaseID=17344>

Download Installer Windows 95/98

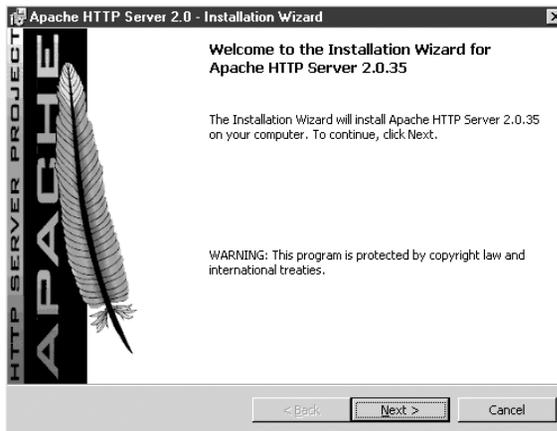
<http://www.microsoft.com/downloads/release.asp?ReleaseID=17343>

Nach einem Doppelklick auf die Datei `apache_2.0.35-win32-x86-no_ssl.msi` erscheinen die folgenden Bildschirmmasken (Abbildung 3.3 – 3.8)

Wählen Sie die Typical-Variante. Bei der Custom-Variante können Sie nur entscheiden, ob Sie die Dokumentation installieren wollen oder nicht.

So, nun ist die Installation beendet und Sie sollten den Rechner neu starten, damit alle Änderungen wirksam werden. Ein Neustart ist zwar nicht zwingend erforderlich, er bietet aber unter Windows (besonders unter 95/98/ME) die Gewissheit, dass alles korrekt funktioniert. Der Apache ist jetzt als Dienst installiert worden und startet nach dem Rechnerstart automatisch.

Abb. 3.3:  
Apache 2.0 –  
Start der  
Installation



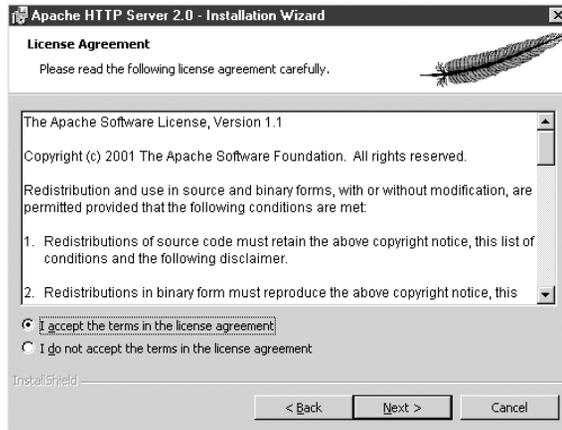


Abb. 3.4:  
Apache 2.0 –  
Lizenz-  
vereinbarung

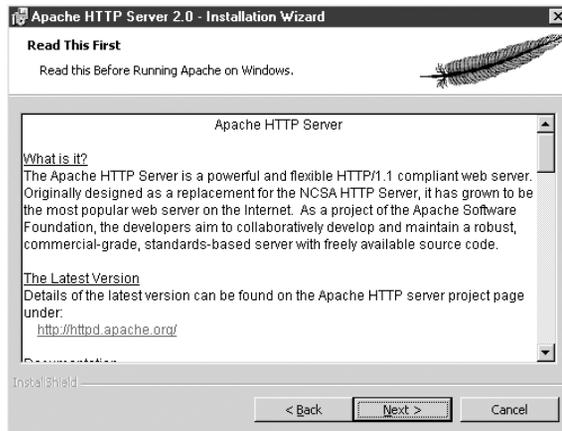


Abb. 3.5:  
Apache 2.0 –  
Hinweise

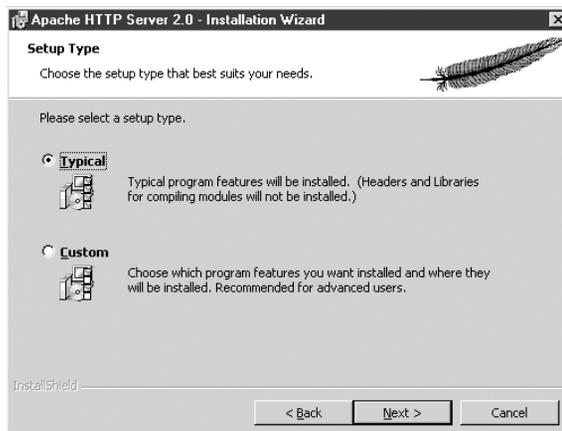


Abb. 3.6:  
Apache 2.0 –  
Setup-Methode

Abb. 3.7:  
Apache 2.0 –  
Server Infor-  
mation

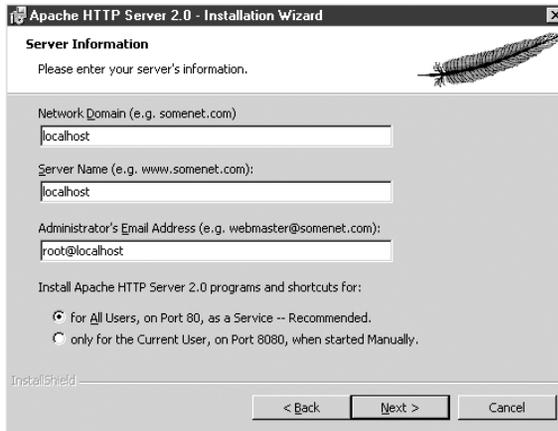


Abb. 3.8:  
Apache 2.0 –  
die Installation  
wurde erfolg-  
reich beendet



## 3.5 Test – ob die Installation geklappt hat

Jetzt wird es ernst! Der Test verläuft in allen Systemumgebungen gleich.

Öffnen Sie ein Browserfenster und geben Sie als Adresse `http://localhost` ein. Nach einem Druck auf  sollte folgende Seite erscheinen (Abbildung 3.9 und Abbildung 3.10).



Abb. 3.9:  
Die Startseite  
unter Linux-  
SuSE 8.0, der  
Benutzer-  
oberfläche  
KDE und dem  
Browser Kon-  
queror nach  
erfolgreicher  
Installation



Abb. 3.10:  
Die Startseite  
unter Win-  
dows mit dem  
Internet-Explo-  
rer nach erfolg-  
reicher Installa-  
tion

Wie Sie erfreut zur Kenntnis nehmen werden, haben Sie den Webserver erfolgreich installiert und können sich jetzt um den entsprechenden Webinhalt und die Konfiguration kümmern.

Schauen wir aber zunächst in den Tiefen des Dateisystems nach, was eigentlich installiert wurde (Abbildung 3.11 und Abbildung 3.12).

Abb. 3.11:  
Dateisystem  
des Apache unter  
Linux

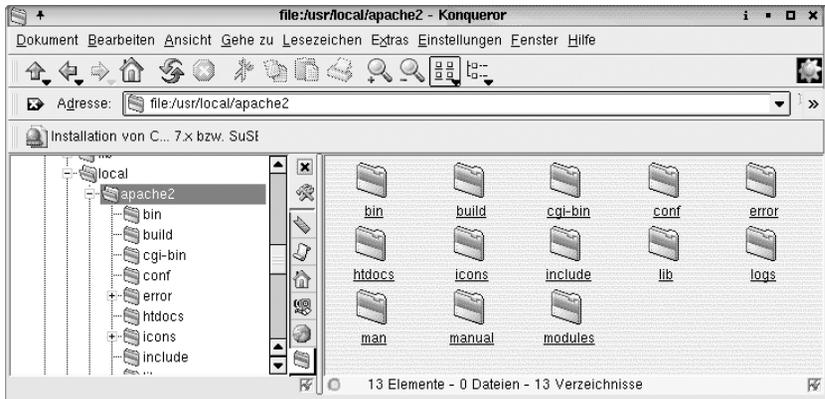
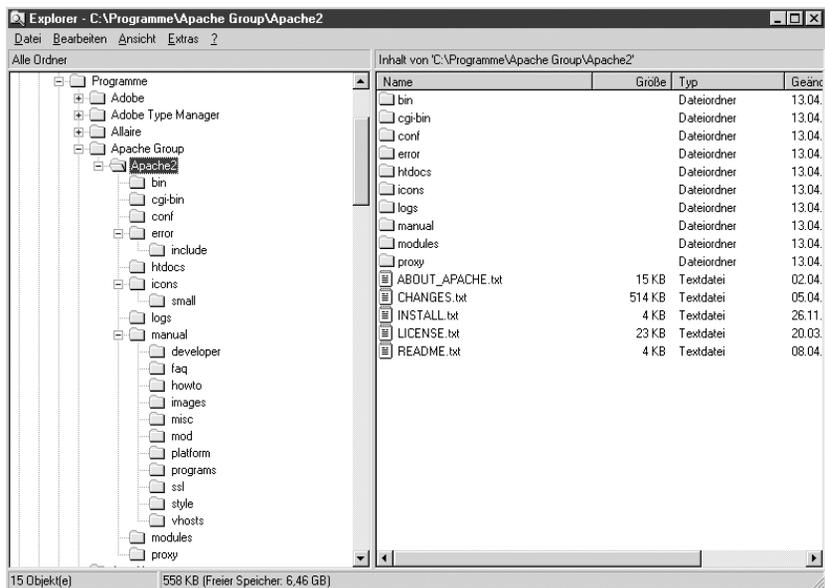


Abb. 3.12:  
Dateisystem  
des Apache unter  
Windows



Ein neuer Ordner und jede Menge Verzeichnisse.

apache2/bin

Hier befinden sich alle ausführbaren Programme, wie der Apache selbst und kleinere Hilfsprogramme wie htpasswd. Mit diesem Programm können Passwörter für User generiert werden.

apache2/cgi-bin

Hier findet sich ein Perl-Programm mit Namen `printenv` (bzw. `printenv.pl` unter Windows). Wenn wir die Skriptsprache Perl installiert haben, können wir uns damit die Umgebungsvariablen anschauen (siehe Kapitel 15.2 Bojen im Dickicht – Umgebungsvariablen).

`apache2/conf`

Hier befinden sich die Konfigurationsdateien des Apache. Uns interessiert zunächst die `httpd.conf`. Zu dieser Datei gibt es auch eine Sicherheitskopie mit dem Namen `httpd.conf.default`. Wenn Sie also mal die Konfiguration so durcheinander gebracht haben, dass gar nichts mehr geht, können Sie immer noch auf diese Standarddatei zurückgreifen. Sie enthält die Konfigurationsdirektiven zum Zeitpunkt der Installation

`apache2/error`

Hier liegen vordefinierte HTTP-Fehlerdokumente.

`apache2/htdocs`

Hier ist das Wurzelverzeichnis (Document Root), auf das jeder Benutzer, der eine Anfrage an den Apache stellt, zugreifen kann und in dem wir unsere HTML-Seiten ablegen werden. Erinnern Sie sich noch an das Beispiel Yahoo in der Einleitung? An dieser Stelle stehen die HTML-Seiten von Yahoo.

`apache2/icons`

Im Verzeichnis `Icons` sind die von Apache verwendeten Icons, beispielsweise für dynamisch erzeugte Inhaltsverzeichnisse. Mehr dazu in Kapitel 4.4.

`apache2/logs`

Hier befinden sich die Logfiles (Logbücher) des Apache (Kapitel 8).

`apache2/manual`

In diesem Verzeichnis befindet sich die englische Dokumentation des Apache.

`apache2/modules`

Hier sind die Module in dynamisch hinzuladbarer Form abgespeichert. Diese Module können nach Bedarf zum Apache dazugeladen und um weitere Module ergänzt werden (Kapitel 11).

`apache/proxy`

In diesem Verzeichnis legt der Apache-Proxy zwischengespeicherte Seiten ab, wenn er als so genannter Proxy-Server konfiguriert wurde. Ein Proxy-Server stellt die am häufigsten verlangten Dokumente in einem separaten Bereich zur Verfügung und verbessert dadurch die Zugriffsgeschwindigkeit des Webservers.

Damit ist die Behandlung der verschiedenen Betriebssystemplattformen zunächst abgeschlossen. Die Konfiguration verhält sich relativ neutral. Ein paar Direktiven funktionieren unter Linux anders als unter Windows. Das liegt an der unterschiedlichen Architektur der Betriebssysteme und soll uns momentan nicht stören.

## 3.6 Starten und Stoppen des Servers

Zum Testen haben Sie den Server bereits gestartet. Wenn Sie jetzt eine aktive Internetverbindung auf Ihrem Rechner haben, so könnten bereits Besucher über die Eingabe Ihrer aktuellen IP-Adresse im Browser auf das Document Root Ihres Webservers zugreifen. Im Normalfall muss der Server selten neu gestartet werden. Da wir jedoch die Konfiguration ständig ändern und der Apache die Konfigurationsdateien nur beim Start einliest, müssen wir den Server öfter starten.

### 3.6.1 Linux

Zum Beenden von Prozessen benutzt man unter Linux den Befehl `kill`. Damit man weiß, was man killen soll, hat jeder Prozess in Linux eine Nummer. Eine Auflistung der Prozesse erhalten Sie mit dem Befehl `ps ax`. Da der Apache jedoch mehrere Prozesse öffnet, ist es gar nicht so einfach, den richtigen Prozess zu finden. Deshalb schreibt der Apache bei seinem Start eine Datei mit dem Namen `PidFile` (Process-ID-File) in das Verzeichnis `logs`. In der Datei `httpd.pid` steht die Nummer des Prozesses, der als Erstes gestartet wurde.

Der Aufruf zum Beenden lautet

```
kill -TERM 'cat /usr/local/apache2/logs/httpd.pid'.
```

Apache schließt nun diesen und alle anderen Apache-Prozesse.

Wenn Sie den Apache neu starten wollen, weil Sie nur eine Änderung in der Konfiguration einspielen wollen, so lautet der Befehl

```
kill -HUP 'cat /usr/local/apache2/logs/httpd.pid'.
```

Hier sorgt das HUP-Signal dafür, dass der Apache alle übergeordneten Prozesse schließt, unabhängig davon, ob noch offene Anforderungen von Clients da sind. Dann liest der Hauptprozess die Konfigurationsdateien neu ein und generiert neue untergeordnete Prozesse. Da das in einer produktiven Umgebung zu Fehlermeldungen bei den Clients führt, wird meist der Befehl `kill -USR1 'cat /usr/local/apache2/logs/httpd.pid'` verwendet. Diese Methode beendet zunächst die offenen Anfragen und dann die untergeordneten Prozesse. Das ist freundlicher, weil alle Transaktionen sauber abgeschlossen werden, in manchen Situationen aber auch fehleranfällig. Wenn Sie also nach einem Restart

mit dem USR1-Signal seltsame Fehlermeldungen entdecken, starten Sie am besten Apache in der vorher beschriebenen Weise komplett neu.

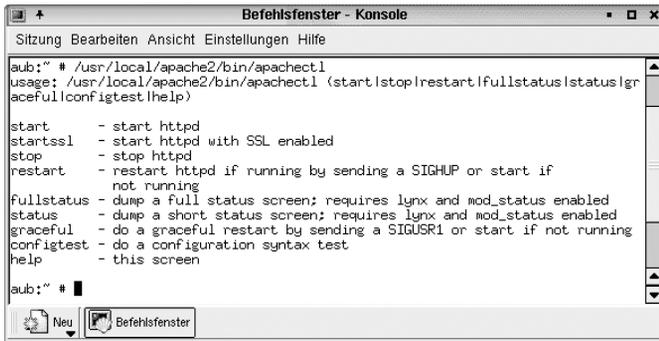


Abb. 3.13:  
Linux – Start-  
skript apache-  
ctl – Hilfe-  
bildschirm

Komfortabler geht es unter Linux mit dem Shell-Skript `apachectl`. Die vorher beschriebenen Befehle sind hier ordentlich verpackt und man hat alle Varianten an einer Stelle (Abbildung 3.13).

Innerhalb des Shellskripts wird der oben beschriebene `kill`-Befehl benutzt. Wenn Sie das Skript ohne Parameter aufrufen, erhalten Sie einen Hilfebildschirm.

```
/apache2/bin/apachectl start -> Startet den Server
/apache2/bin/apachectl stop -> Stoppt den Server
/apache2/bin/apachectl restart -> Restart mit der HUP-Methode
/apache2/bin/apachectl graceful -> Restart mit der USR1-Methode
```

### 3.6.2 Windows

In der Systemsteuerung von Windows unter dem Icon »Dienste« steht ein Dialogfenster zum Starten und Stoppen von Diensten zur Verfügung (Abbildung 3.14).

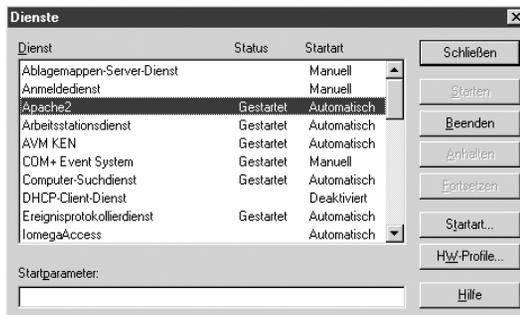
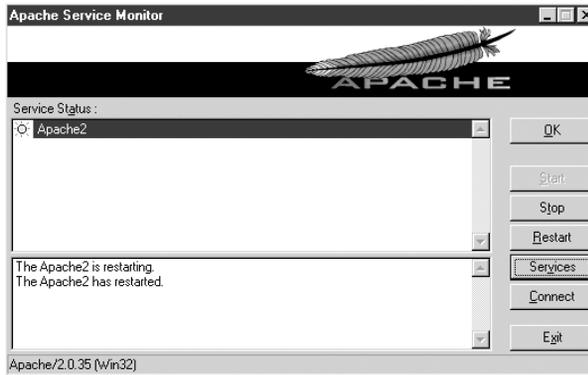


Abb. 3.14:  
Dienste-Menü  
in Windows  
NT4

Außerdem hat sich in der Statuszeile von Windows die Apache-Feder breit gemacht und mit einem rechten Mausklick können Sie den Apache Monitor aufrufen, in dem Sie den Apache starten, stoppen und neu starten können (Abbildung 3.15).

Abb. 3.15:  
Apache Service  
Monitor



## 3.7 Zusammenfassung

In diesem Kapitel wurde die Installation des Apache2-Webserver beschrieben. Unter den beiden Linux-Distributionen haben wir den Apache kompiliert und in einem eigenen Verzeichnis installiert. Unter Windows haben wir ihn mit dem Microsoft Installer installiert. Das Starten und Stoppen des Servers erfolgt unter Linux mit dem Skript `apachectl`, unter Windows mit dem Apache Monitor. Nach der Installation haben wir getestet, ob der Apache ordnungsgemäß gestartet wurde.

## 3.8 Fragen

1. Was ist der Vorteil einer eigenen Kompilierung des Apache-Webserver?
2. Warum ist die Kompilierung unter Windows für die meisten Benutzer ein Problem?
3. Was muss vor der Installation des Apache auf dem Windows-System installiert sein?
4. Mit welcher Methode starten Sie den Apache-Webserver unter Linux?
5. Mit welcher Methode starten Sie den Apache-Webserver unter Windows?
6. Wie testen Sie, ob der Apache läuft?

## 3.9 Antworten

1. Durch eine eigene Kompilierung kann der Apache auf die persönlichen Bedürfnisse angepasst werden.
2. Unter Windows wird kein C-Compiler kostenlos mitgeliefert. Er muss beispielsweise bei der Firma Microsoft separat erworben werden.
3. Das MSI-Installationssystem muss vor der Installation des Apache installiert sein. Windows ME/XP/2000 enthalten den Installer bereits.
4. Mit dem Skript `apachectl start`.
5. Mit dem Apache Service Monitor durch einen Klick auf den Button Start.
6. Durch den Aufruf von `http://localhost` im Browser. Wenn der Apache läuft, sollte eine Testseite angezeigt werden.