

# **Apache unter Windows**

## **Installation und Konfiguration des Apache Web Servers 1.3.x unter Windows**

**Projektarbeit zum Fach :  
Planung und Realisation von  
heterogenen Netzwerken (Fallstudie)**

**Bearbeitet von: Michael Engelke**

# Inhaltsverzeichnis:

<b>I. Einleitung</b> .....	<b>3</b>
<b>II. Installation</b> .....	<b>4</b>
<b>1. Installation von Apache</b> .....	<b>4</b>
<b>2. Kontrollieren des Serverprozesses</b> .....	<b>5</b>
<b>III. Die Konfigurationsdatei "httpd.conf"</b> .....	<b>5</b>
<b>1. Wichtige Optionen in der Konfigurationsdatei</b> .....	<b>6</b>
<b>2. Anpassen der Einstellungen</b> .....	<b>7</b>
<b>3. Virtuelle Hosts</b> .....	<b>7</b>
<b>IV. Verzeichnisschutz mit .htaccess</b> .....	<b>8</b>
<b>1. Allgemeine Einstellungen</b> .....	<b>8</b>
<b>2. Verwalten der Kennwortdatei</b> .....	<b>9</b>
<b>3. Erstellen der .htaccess</b> .....	<b>9</b>
<b>V. Zusammenfassung</b> .....	<b>10</b>

# I. Einleitung:

Das Thema dieses Scriptes ist es einen Webserver unter Windows 2000 einzurichten. Um dies zu lösen gibt es verschiedene möglichkeiten. Die einfachste lösung ist Apache einzusetzen. Apache wurde ursprünglich für Linux entwickelt , ist aber auch uunter anderen Plattformen verfügbar. (Amiga, BeOs, Windows, Unix, etc.)

Für dieses Script wurde Apache 1.3.20 vom 22. März 2001 für Windows verwendet. Alle Erklärungen und Einstellungen beziehen sich auf diese Version.

Die Beschreibung die Konfiguration beschränken sich auf die notwendigsten Einstellungen, da eine umfassende Erklärung aller Möglichkeiten den Rahmen sprengen würde.

Bevor wir mit der Installation anfangen, müssen wir Programmdateien herunterladen:

Offizielle Homepage: <http://httpd.apache.org/>

Programm-Archiv: [http://www.komplikator.cz/download/apache\\_1.3.20-win32-src-r2.msi](http://www.komplikator.cz/download/apache_1.3.20-win32-src-r2.msi)

Dokumentation: <http://httpd.apache.org/docs/>

MSI-Installer Win9x: <http://www.microsoft.com/downloads/release.asp?ReleaseID=17343>

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

# II. Installation:

## 1. Installation von Apache

Unter Windows 2000 ist es egal, welche Installer-Version man herunter geladen hat, da alle Varianten unterstützt werden. Wobei es bei anderen Windows-Versionen bei der Installation einiges zu beachten gibt:

- Bei Windows 95 muß noch, zusätzlich Winsock2 und falls notwendig, den MSI-Installer installiert werden.
- Unter Windows 98 muß nur, falls notwendig, der MSI-Installer installiert werden.

Falls man unter Windows 95 / 98 oder NT noch nicht den MSI-Installer installiert hat, sollte man jetzt nachholen, da sonst eine Installation nicht möglich ist.

Anschließend kann man jetzt das Apache MSI-Archiv starten.

Nach dem Begrüßungstext kann man auf NEXT klicken um auf die License Agreement Seite zu kommen. Diese sollte man sich durchlesen und entweder zustimmen oder ablehnen. (Bei Ablehnung kann die Installation nicht fortgesetzt werden!) Nachdem man auf NEXT geklickt hat, werden noch einige Informationen zu Apache angezeigt. Nach einem Klick auf NEXT kann man jetzt die Server-Informationen eingeben: (Diese Eingaben werden in `conf/httpd.conf` gespeichert)

Network Domain: Hier können Sie den Namen Ihrer Domain eingeben.  
z.B. `irgentwas.de`

Server Name: Hier tragen Sie den Namen Ihres Servers ein.  
z.B. `www.irgentwas.de`

Administrator's eMail: Hier tragen Sie die eMail-Adresse ein  
z.B. `webmaster@irgentwas.de`

Unten können Sie einstellen, ob Apache als Dienst oder manuell gestartet werden soll. Wenn Sie Apache als Dienst benutzen, wird Apache gleich nach dem Booten von Windows gestartet. Diese Einstellung wird empfohlen, wenn Sie Apache für Server-Zwecke nutzen wollen.

Oder Sie starten Apache immer manuell. Zum Beispiel für Testzwecke.

Anschließend können Sie wieder auf NEXT klicken, um zur Setup-Type Seite zu kommen. Hier können Sie wählen, ob Sie eine 08/15-Installation durchführen wollen oder eine benutzerdefinierte.

Bei einer kompletten Installation wird alles installiert, was im Installationspaket enthalten war. (ca. 9 MB)

Bei einer benutzerdefinierten Installation können Sie wählen, welche Pakete installiert werden sollen und wohin die Installation gespeichert werden soll.

Nachdem Sie alles eingestellt haben, klicken Sie auf NEXT und anschließend auf INSTALL. Nun werden die Dateien installiert. Dieser Vorgang kann einige Minuten dauern. Wenn Apache als Dienst gestartet werden soll, wird Apache gleich nach dem Kopieren gestartet.

Anschließend bekommen Sie eine Meldung, daß die Installation vollständig ausgeführt wurde und Sie das Setup-Programm beenden können.

## 2. Kontrollieren des Serverprozesses

Den Apache-Serverprozess steuert man mit dem Kommando Apache.exe, das sich im Apache Hauptverzeichnis befindet. Man kann diese Optionen im Startmenü anwählen oder direkt in der Console eingeben. Hier ist eine Übersicht der wichtigsten Optionen, die wir benötigen werden.

- start** Startet den Apache-Prozess. Dies muß nach jedem Neustart des Systems manuell erledigt werden, wenn Apache nicht als Dienst installiert wurde.  
Der Befehl für die Konsole: Apache.exe -w -n "Apache" -k start
- stop** Beendet den Apache-Prozess.  
Der Befehl für die Konsole: Apache.exe -w -n "Apache" -k stop
- restart** Startet den Apache-Prozess neu. Bewirkt im Prinzip das gleiche wie **start** und **stop** zusammen. Zusätzlich wird noch ein **configtest** ausgeführt, bevor der Prozess wieder gestartet wird. Dies sollte man nach jeder Änderung an der httpd.conf durchführen, damit die Änderungen wirksam werden.  
Der Befehl für die Konsole: Apache.exe -w -n "Apache" -k restart
- configtest** Testet die Syntax der Konfigurationsdatei und meldet ggfs. detaillierte Fehlerbeschreibungen. Dies sollte man nach jeder Änderung an der httpd.conf einmal ausführen, um Fehler zu finden und zu vermeiden.  
Der Befehl für die Konsole: Apache.exe -w -t -f "apache\conf\httpd.conf" -d "apache\  
(“apache“ steht für das Rootverzeichnis von Apache)

## III. Die Konfigurationsdatei ‘httpd.conf’:

Alle Einstellungen für den Apache Web Server werden in der Konfigurationsdatei **httpd.conf** vorgenommen, die sich im Unterverzeichnis **conf** befindet. Wie es beim Betriebssystem Linux üblich ist, handelt es sich hier um eine reine Textdatei, die mit einem beliebigen Texteditor bearbeitet werden kann. Die anderen Dateien im selben Verzeichnis bleiben üblicherweise ungenutzt bzw. unverändert.

Es fällt sofort auf, daß die Datei hauptsächlich aus Kommentaren besteht (fangen mit dem ‘#’-Zeichen an), die alle Abschnitte und Direktiven erläutern. Vor jeder Änderung sollte man sich den entsprechenden Kommentar durchlesen, dies hilft bei Problemen oft weiter. Außerdem sollte man sich zu Beginn die Hinweise und Tipps am Dateianfang durchlesen!

Die Konfigurationsdatei ist in drei Abschnitte eingeteilt:

1. Die ‘globalen Einstellungen’, die den Betrieb des Apache-Serverprozesses als solchen steuern und beeinflussen.
2. Einstellungen des ‘Haupt’-Servers, der auf Anfragen reagiert, die von keinem virtuellen Host verarbeitet werden. Diese Einstellungen legen auch Standardwerte für die Einstellungen der virtuellen Hosts ein, etwa Dateitypen, Zeichensätze und Erweiterungsmodule.
3. Einstellungen für virtuelle Hosts, die es erlauben, Anfragen an andere IP-Adressen oder Hostnamen umzuleiten, die aber vom selben Apache-Serverprozess verarbeitet werden.

# 1. Wichtige Optionen in der Konfigurationsdatei

Hier werden die wichtigsten Konfigurations-Direktiven für die **httpd.conf**-Datei erläutert:

**ServerRoot** Das Stammverzeichnis von Apache, unter dem alle anderen Verzeichnisse und Dateien abgelegt werden. Standardmäßig sind hier auch (in entsprechenden Unterverzeichnissen) die Web- und Protokoll-Dateien des haupt-Servers abgelegt. Diese Variable wird während der Installation bereits korrekt eingestellt.

**Port** Der TCP-Port auf den der Server reagiert. Dies ist standardmäßig Portnummer 80 für das HTTP-Protokoll.

**ServerAdmin** Die E-Mail-Adresse des Server-Administrators, an die Serverprobleme gemeldet werden sollten. Hierbei sollte es sich natürlich um eine gültige Adresse handeln. Diese Adresse wird auf einigen automatisch vom Server generierten Dokumenten, z.B. Fehlermeldungen, angezeigt. Ausßerdem kann die Adresse von CGI-Programmen ermittelt werden.

**ServerName** Dies ist der Hostname oder die IP-Adresse des Servers. Auch dieser Wert muß natürlich gültig sein, d.H. der Hostname muß im Netzwerk bekannt sein. Zum Testen auf dem lokalen System reicht es auch, die lokale Loop-Back-Adresse 127.0.0.1 oder localhost anzugeben.

**DocumentRoot** Dies ist das Stammverzeichnis für alle Web-Dokumente, d.H. hier liegt auch die Startdatei der Homepage. Diese Variable wird während der Installation bereits korrekt eingestellt.

**Options** Bezieht sich immer auf ein bestimmtes Verzeichnis und taucht daher immer in einem **<Directory>**-Block auf. Es können mehrere Optionen (jeweils durch ein Leerzeichen getrennt) angegeben werden. Alle Optionen werden im Kommentartext erläutert. Interessant ist die Option **+Indexes** bzw. **-Indexes**, mit der man das Auflisten von Verzeichnisinhalten erlauben bzw. verbieten kann, wenn keine **DirectoryIndex**-Datei in diesem Verzeichnis liegt.

**AllowOverride** Bezieht sich immer auf ein bestimmtes Verzeichnis und taucht daher immer in einem **<Directory>**-Block auf. Damit kann man die Verwendung von **.htaccess**-Zugriffsschutzdateien verbieten oder erlauben. (Siehe Kapitel zum Zugriffsschutz).

**DirectoryIndex** Name(n) der Standard-Datei(en), die an den Client zurückgeliefert werden, wenn in der Request-URL kein Dateiname angegeben wurde (dies kann man auch als "Startdatei" für ein Verzeichnis ansehen). Es können auch mehrere Dateinamen, jeweils durch Leerzeichen getrennt, angegeben werden.

**AccessFileName** Name der Zugriffsdatei, standardmäßig **.htaccess** (beachte den Punkt am Anfang des Dateinamens!).

**ErrorLog** Pfad zur Fehlerprotokolldatei. Hier werden fehlgeschlagene Requests protokolliert, z.B. ungültige und fehlgeschlagene Requests.

**CustomLog** Pfad und Format der Zugriffsprotokolldatei. Das Format **common** ist die Vorgabe. Zusätzlich können Referer- und Agent-Logging aktiviert werden, indem die zwei Anweisungen (mit dem Format **referer** bzw. **agent**) entkommentiert werden.

**ErrorDocument** Selbstdefinierte Fehlermeldungen für Serverfehler. Es wird zuerst die betroffene Fehlernummer angegeben (z.B. 404 für nicht gefundene Dateien), danach die Art der Fehlermeldung: Dies kann ein Text, der mit einem einzelnen Anführungszeichen eingeleitet wird. Nützlicher ist aber die Angabe einer URL zu einer HTML-Dokument oder CGI-Skript. Mehr erfährt man in dem Kommentaren zu dieser Direktive.

## 2. Anpassen der Einstellungen

Um Apache anfangs in Betrieb zu nehmen, müssen nur sehr wenige Veränderungen an der Standardkonfiguration vorgenommen werden. Die Verzeichnisse werden bei der Installation bereits korrekt eingestellt.

Üblicherweise wird als Startdatei in einem Verzeichnis nur **“index.html“** akzeptiert. Es kann aber auch sinnvoll sein, diese Angabe zu verändern oder zu erweitern. So kann man auch **“welcome.htm“** und **“index.php“** zulassen. Mit der **DirectoryIndex**-Direktive kann man mehrere Dateinamen durch Leerzeichen getrennt angeben.

Das Definieren eigener Fehlerdokumente mit **ErrorDocument** ist vor allem sinnvoll, um diese an das Design der restlichen Website anzupassen. Dazu erstellt man eine beliebige HTML-Datei und gibt deren relativen Pfad an. Am besten orientiert man sich an den Beispielen, die in der Konfigdatei angegeben sind.

## 3. Virtuelle Hosts:

Üblicherweise **“hört“** ein Webserver lediglich auf einen einzigen Domainnamen, der seiner IP-Adresse zugeordnet ist. Apache bietet mit dem Konzept der Virtual Hosts aber eine einfache Möglichkeit, dem Webserver mehrere Domainnamen zuzuordnen. So kann man für den selben Host (die selbe IP-Adresse) alternative Namen oder auch Subdomains definieren.

Zunächst stellt man die IP-Adresse des Serverrechners mit der Direktive **NameVirtualHost** ein. Beachte, daß diese Direktive standardmäßig auskommentiert ist und das **“#“** davor entfernt werden muß.

**Wichtig:** Die Namen der virtuellen Hosts müssen im Netzwerk natürlich bekannt sein. Am einfachsten zum Testen ist es, diese in der Datei **hosts** einzutragen. (%systemroot%/system32/drivers/etc/hosts)

```
192.168.96.16    www.test.de
192.168.96.16    www.muellhal.de
192.168.96.16    www.winzigweich.com
```

Für jeden virtuellen Host fügt man einen **<VirtualHost>**-Block ein, wie am Ende der Konfigurationsdatei beispielhaft dargestellt:

```
<VirtualHost 192.168.96.16>
    ServerAdmin webmaster@winzigweich.com
    DocumentRoot /www/docs/winzigweich.com
    ServerName winzigweich.com
    ErrorLog logs/winzigweich-error_log
    CustomLog logs/winzigweich-access_log common
</VirtualHost>
```

Die IP-Adresse am Blockanfang muß an dem zuvor mit **NameVirtualHost** eingestellten Wert angepaßt werden. Innerhalb dieses Blocks können nahezu alle Direktiven verwendet werden, um die Standardeinstellungen für jeden virtuellen Host gezielt zu verändern. Die Standardeinstellungen werden aus Abschnitt 2 übernommen. Die Minimalversion eines VirtualHost-Blocks besteht einzig aus der Einstellung des virtuellen Hostnames mit **ServerName**:

```
<VirtualHost 192.168.96.16>
    ServerName www.muellhal.de
</VirtualHost>
```

Sinnvoll ist es, jeden **VirtualHost** in ein eingendes Verzeichnis zu packen. Zusätzlich ist es sinnvoll, die **CustomLog** mit dem Namen des **VirtualHost** unzubennen.

# IV. Verzeichnisschutz mit .htaccess:

Apache kann den Zugriff auf einzelne Verzeichnisse mit Hilfe der sogenannten .htaccess-Dateien kontrollieren. Prinzipiell lassen sich alle Direktiven, die in einem **<Directory>**-Block gültig sind, auch in einer Schutzdatei verwenden. Besonders sinnvoll ist es, bestimmten Verzeichnisse mit einem Kennwortschutz zu versehen. Eine Dokumentation aller Möglichkeiten ist unter folgender Adresse zu finden:

<http://httpd.apache.org/docs/howto/auth.html>

## 1. Allgemeine Einstellungen

Zunächst muß der Verzeichnisschutz aktiviert werden. In der Konfigurationsdatei befindet sich der folgende Abschnitt, in dem der Wert nach **AllowOverride** von **none** auf **all** geändert wird:

```
#
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "C:/Programme/Apache/htdocs">

#
# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
    Options Indexes FollowSymLinks MultiViews

#
# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options", "FileInfo",
# "AuthConfig", and "Limit"
#
    AllowOverride None

#
# Controls who can get stuff from this server.
#
    Order allow,deny
    Allow from all
</Directory>
```

## 2. Verwalten der Kennwortdatei

Zunächst muß eine Kennwortdatei erstellt werden, in der Benutzernamen und Kennwörter verwaltet werden. Dies erledigt man mit dem Kommando **htpasswd**, das sich in den Verzeichnis `apache/bin` befindet. Eine Beschreibung aller Optionen findet man heraus, wenn man das Programm ohne Parameter in der Konsole startet.

```
htpasswd
```

```
Usage:
```

```
htpasswd [-cmdps] passwordfile username
htpasswd -b[cmdps] passwordfile username password
```

```
htpasswd -n[mdps] username
htpasswd -nb[mdps] username password
```

- c Create a new file.
- n Don't update file; display results on stdout.
- m Force MD5 encryption of the password (default).
- d Force CRYPT encryption of the password.
- p Do not encrypt the password (plaintext).
- s Force SHA encryption of the password.
- b Use the password from the command line rather than prompting for it.

On Windows, TPF and NetWare systems the '-m' flag is used by default.

On all other systems, the '-p' flag will probably not work.

Als Parameter erwartet das `htpasswd`-Kommando mindestens den Pfad zur Kennwortdatei sowie den Namen des neu einzurichtenden Benutzers.

z.B. "`htpasswd -c htdocs/secret/.htpasswd hannibal`"

Die wichtigsten Optionen und Parameter:

-c Optional. Erstellt eine neue Kennwortdatei. Diese Option ist beim ersten Aufruf des Kommandos zwingend notwendig. Bei weiteren Aufrufen darf man die Option nicht angeben, da die Datei dann neu erstellt wird (wodurch bisherige Einstellungen natürlich verloren gehen).

Datei Pfad zur Kennwortdatei.

Benutzer Name des neu einzurichtenden Benutzerkontos, z.B. "hannibal".

Kennwort Optional. Kennwort des angegebenen Benutzers. Diese Angabe kann nur angegeben werden wenn die Option "-b" gesetzt wurde. Dies bewirkt, daß man das Kennwort direkt als Parameter dazu Eingeben kann. z.B. für Scripts. Läßt man die Option weg, so wird man nach dem Kennwort gefragt, und muß es "Blind" eingeben, so daß niemand das Kennwort beim Eintippen sehen kann.

## 3. Erstellen der .htaccess-Datei

Zuletzt muß die `.htaccess`-Datei erstellt werden, was ebenfalls mit einem Texteditor erledigt wird. Die Datei wird in dem Verzeichnis abgelegt, das mit einem Kennwortschutz versehen werden soll. Der so eingerichtete Schutz gilt dann rekursiv auch für alle Unterverzeichnisse. Die Standard `.htaccess`-Datei sieht wie folgt aus:

```
AuthType Basic
AuthName "Authentication Required"
AuthUserFile htdocs/.htpasswd
Require valid-user
```

Es ist darauf zu achten, daß der Pfad für `AuthUserFile` anzupassen ist. Anstatt **valid-user** kann man bei **Require** einen bestimmten User angeben: z.B. "**Require user test**". Der Text hinter **AuthName** wird in dem Login-Dialogfenster angezeigt und kann beliebig gewählt werden.

# V. Zusammenfassung

Der Apache ist ein Server, der bis in kleinste Detail den Benutzerwünsche angepaßt werden kann. Dies hat aber auch die Folge, daß man sich tief einarbeiten muß, wenn man eine Optimale Konfiguration erreichen möchte. Jetzt am Anfang wird man erschlagen von den ganzen Möglichkeiten, aber da Apache schon mit der Standard-Konfiguration auf jeden Rechner läuft, kann man sich recht schnell einarbeiten.

Wie schon gesagt, ist Apache auf vielen Plattformen verfügbar, dennoch die Benutzung und Konfiguration auf allen Rechner gleich. d.H. Ein Linux-User kann ohne weiteres Apache unter Window Administrieren. Da Apache zum größten Teil aus reinen Textdateien besteht.

Interessant ist noch zu erwähnen, daß man Apache noch mehr erweitern kann. Im zusammen-spiel mit PHP (Das schon in Apache intrigiert ist) kann man auf MySQL Datenbanken zugreifen um Web-Seiten Dynamisch zu gestalten. Auf diese Weise sind z.B. Useranmeldungen, Persönliche Postfächer andere Dinge möglich.

Ich persönlich benutze Apache zum einen um meine Homepage "<http://www.MEngelke.de/>" zu testen, die unter mehrenden Domainen läuft und zum anderen benutze ich Apache um das Internet zu simulieren, da einige Webseiten nicht Lokal funktionieren.

Apache braucht man eigentlich nicht Installieren, in der Regel reicht es das Verzeichnis Apache auf dem Rechner zu kopieren. Dann noch einige Pfade an der httpd.conf korrigieren und schon ist Apache einsatzbereit. Mit einigen Tricks ist es sogar möglich Apache direkt von CD-ROM zu starten.

Ich habe mich jetzt schon so einige Monate lang mit Apache beschäftigt, und kann nicht sagen daß ich ein Oberguru geworden bin, da möglichkeiten unausschöpflich sind. Trotzdem traue es mir zu kleiner Web-Server mit Apache aufzusetzen.

Als Literatur aber ich ausschließlich die Offizielle Homepage "<http://httpd.apache.org/>" eingesetzt. Da dort wirklich beschrieben ist und mir sonst keine weitere Schriftliche Literatur zur verfügung stand.

© Freitag, 13 Dezember 2002 von Michael Engelke