

Die Konfigurationsdatei des Apache

Diese Datei basiert auf den originalen Angaben von Rob McCool, mit denen der NCSA-Server konfiguriert wurde (NCSA = National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign).

Es handelt sich hier um die zentrale Konfigurationsdatei für den Apache. Sie enthält Konfigurations-Anweisungen, mit denen der Server seine Instruktionen bekommt (siehe <http://www.apache.org/docs/> für detailliertere Informationen).

Du solltest diese Anweisungen nicht einfach nur lesen, ohne genau zu verstehen, was sie bewirken. Sie stehen hier eigentlich auch nur als kurze Notizen zur Erinnerung. Wenn Du dich unsicher fühlst, solltest Du die Online-Dokumentation gründlich studieren.

<DU BIST GEWARNT !>

Wenn diese Datei abgearbeitet worden ist, sucht der Server nach den Dateien srm.conf und access.conf, falls Du sie hier nicht mit den Einstellungen für ResourceConfig und/oder AccessConfig überschrieben hast.

Die Anweisungen zur Konfiguration sind in drei Haupt-Abschnitte gegliedert:

1. [Anweisungen, die die gesamten Verhaltensweisen des Apache Web Servers festlegen \(die "globale Umgebung"\).](#)
2. [Anweisungen, die die Parameter für den Standard-Server definieren, der alle die Anfragen bearbeitet, die nicht von virtuellen Hosts entgegengenommen werden. Diese Angaben gelten auch als Standardwerte für virtuelle Hosts, sofern sie nicht in den entsprechenden Anweisungen dort überschrieben werden.](#)
3. [Festlegungen für virtuelle Hosts. Damit werden Web-Anfragen für verschiedene IP-Adressen oder Hostnamen ermöglicht, die jeweils von demselben Server-Prozess behandelt werden sollen.](#)

Namen für Konfigurations- und Protokolldateien: Wenn die von Dir angegebenen Dateinamen für viele der Kontrolldateien mit einem Slash "/" (oder "Laufwerk:/" bei Win32) beginnen, wird der Server das als den ausdrücklich zugewiesenen Pfad ansehen. Sofern die Dateinamen NICHT mit "/" beginnen, wird das ServerRoot-Verzeichnis davorgeschrieben. So wird beispielsweise die Angabe "logs/access.log" von Apache als "/usr/local/apache/logs/access.log" behandelt, wenn das ServerRoot-Verzeichnis auf "/usr/local/apache" (UNIX-Syntax) gesetzt ist.

Beachte bitte: Wenn Dateinamen notiert werden, musst Du einfache Schrägstriche (forward slashes - "/") anstelle von Backslashes verwenden (z.B. "c:/apache" statt "c:\apache"). Wenn kein Laufwerksbuchstabe angegeben wird, verwendet Apache das Laufwerk, auf dem die Datei Apache.exe liegt, als Standard-Laufwerk. Um Konflikte zu vermeiden, solltest Du immer einen Laufwerksbuchstaben bei absoluten Pfadangaben verwenden.

1. Abschnitt: Globale Umgebung

Die Anweisungen in diesem Abschnitt bestimmen die grundsätzliche Arbeitsweise Deines Apache-Servers.

"ServerType": Der Server-Typ ist entweder "inetd" oder "standalone". "inetd" wird allerdings nur von UNIX-Plattformen unterstützt bzw. verwendet.

ServerType standalone

"ServerRoot" bezeichnet die Spitze des Verzeichnisbaums, unter dem die Server-Konfiguration, Fehleranzeigen und Protokolle zu finden sind. Am Ende dieser Zeile darf kein Slash stehen.

ServerRoot "C:/Apache"

"PidFile": das ist die Datei, in der der Server jedesmal dann seine "process identification number" niederlegen sollte, wenn er gestartet wird.

PidFile logs/httpd.pid

"ScoreBoardFile": die Datei, die interne Informationen über Server-Prozesse aufnimmt. Nicht alle Architekturen verlangen eine solche Datei. Aber wenn Deine Architektur sie vorsieht (Du erfährst das, sobald Du den Apache startest, womit die Datei erforderlichenfalls generiert wird), musst Du dafür sorgen, daß zwei (oder mehr) Apache-Aufrufe nicht auf dieselbe Scoreboard-Datei zugreifen.

ScoreBoardFile logs/apache_status

In der Standard-Konfiguration wird der Server die Dateien httpd.conf (diese Datei), srm.conf, und access.conf in dieser Reihenfolge abarbeiten. Die beiden letztgenannten Dateien werden jetzt leer belassen, da empfohlen wird, zur besseren Übersicht sämtliche Anweisungen in einer einzigen Konfigurationsdatei zusammenzufassen. Die auskommentierten Werte unten sind jeweils die vorgegebenen Standardwerte. (Auskommentierte Werte erkennt man daran, das sie kein #-Zeichen am Zeilenanfang besitzen).

```
# ResourceConfig conf/srm.conf  
# AccessConfig conf/access.conf
```

"Timeout" bestimmt die Anzahl Sekunden, ehe ein Timeout gesendet wird.

Timeout 300

"KeepAlive": legt fest, ob persistente Verbindungen (mehr als eine Anfrage pro Verbindung) zulässig sind. Wird mit "Off" deaktiviert

KeepAlive On

"MaxKeepAliveRequests": die Höchstzahl der während einer persistenten Verbindung zulässigen Anfragen. Wird hier 0 angegeben, ist unbegrenzter Zugriff möglich. Empfohlen

wird ein hoher Wert, um eine hohe Performance zu erhalten.

MaxKeepAliveRequests 200

"KeepAliveTimeout": Zeitspanne (Zahl an Sekunden), um auf die nächste Abfrage desselben Clients mit derselben Verbindung zu warten.

KeepAliveTimeout 15

Apache erstellt unter WINDOWS grundsätzlich einen "Kind"-Prozeß zum Abarbeiten von Abfragen. Wenn dieser Prozeß beendet wird ("stirbt"), wird automatisch sofort der nächste derartige Prozeß gestartet. Innerhalb dieses "Kind"-Prozesses können zahlreiche Threads die eingehenden Anfragen behandeln. Die nächsten beiden Anweisungen legen das Verhalten von Threads und Prozessen fest.

"MaxRequestsPerChild": Höchstzahl an Anfragen, die jeder "Kind-Prozeß" behandeln darf, ehe er beendet wird. Das Beenden von "Kind-Prozessen" ist wichtig, um Probleme zu vermeiden, falls Apache (und möglicherweise die von ihm verwendeten Bibliotheken) den Speicher oder andere Ressourcen aufgebraucht hat. Auf den meisten Plattformen wird das zwar kaum einmal vorkommen, aber ein paar, wie zum Beispiel Solaris, verfügen über beachtliche "Lecks" in ihren Bibliotheken. Für Win32 kannst Du diesen Wert ruhig auf Null setzen, wenn Dir nicht ausdrücklich etwas anderes empfohlen wird.

Beachte: Mit Ausnahme der Initial-Anfrage gilt dieser Wert NICHT für "Keepalive"-Anfragen. Wenn ein "Kind-Prozeß" beispielsweise mit einer Initial-Anfrage startet, der weitere zehn Anfragen folgen, so würde hier lediglich 1 gezählt werden.

MaxRequestsPerChild 0

"ThreadsPerChild": Anzahl der konkurrierenden Threads (z. B. Anfragen), die der Server zulassen soll. Setze diesen Wert entsprechend den Möglichkeiten Deines Servers (je mehr aktive Anfragen Du gleichzeitig zulässt, desto langsamer wird die Abarbeitung für alle) und der Menge an Systemressourcen, die Du Deinem Server zumuten willst.

ThreadsPerChild 150

"Listen": Diese Anweisung gestattet es, Apache mit spezifischen IP-Adressen und/oder Ports zu verbinden, zusätzlich zu den Standardvorgaben. Bitte auch Abschnitt 3 (<VirtualHost>) vergleichen. 80 ist z.B. der Port für http und 21 wäre der Port für ftp.

Listen 3000

Listen 192.168.0.1:80

"BindAddress": mit dieser Option können virtuelle Hosts unterstützt werden. Die Anweisung wird genutzt, um dem Server mitzuteilen, welche IP er wählen soll. Sie kann entweder einen Stern (*) oder einen vollständigen Internetnamen enthalten. Siehe auch die Anweisungen "<VirtualHost>" und "Listen".

BindAddress *

Module der Standard-Windows-Distribution

Die folgenden Module sind in der Standard-Distribution für WINDOWS bereits enthalten. Wenn Du das Standard-Verhalten ändern möchtest, kannst Du die folgenden Zeilen auskommentieren bzw. Deinen Anforderungen gemäß modifizieren, so daß nur die Module angesprochen werden, die Du wirklich benötigst.

WARNUNG: Dies ist eine umfassende Option, mit der Dein Server irreparabel konfiguriert wird! Wenn Du nicht absolut sicher bist, solltest Du die folgende Liste unverändert lassen.

```
# ClearModuleList  
# AddModule mod_so.c mod_mime.c mod_access.c mod_auth.c mod_negotiation.c  
# AddModule mod_include.c mod_autoindex.c mod_dir.c mod_cgi.c mod_userdir.c  
# AddModule mod_alias.c mod_env.c mod_log_config.c mod_asis.c mod_imap.c  
# AddModule mod_actions.c mod_setenvif.c mod_isapi.c
```

Unterstützung für dynamisch verbundene Objekte (Module)

Um die Funktionalität eines als dynamisch verbundenes Objekt eingebundenen Moduls nutzen zu können, muß die korrespondierende Zeile 'LoadModule' hier eingetragen sein, damit die in diesem Modul vorhandenen Anweisungen aktuell verfügbar sind, bevor sie aufgerufen werden. Bitte lese in der Datei README.DSO Deiner Apache-Distribution nach, wenn Du Genaueres über das Konzept der dynamisch verbundenen Objekte (DSO) erfahren möchtest, und lasse Dir mit 'apache -l' auf der Befehlszeile eine Liste der aktuell verfügbaren Module anzeigen.

Beachte: Die Reihenfolge, in der die Module geladen werden, ist wichtig. Wenn es nicht absolut nötig ist oder wenn Du nicht absolut präzise weisst, was Du ändern möchtest, solltest Du auch diese Liste unverändert lassen.

```
# LoadModule anon_auth_module modules/mod_auth_anon.so  
# LoadModule dbm_auth_module modules/mod_auth_dbm.so  
# LoadModule digest_auth_module modules/mod_auth_digest.so  
# LoadModule cern_meta_module modules/mod_cern_meta.so  
# LoadModule digest_module modules/mod_digest.so  
# LoadModule expires_module modules/mod_expires.so  
# LoadModule headers_module modules/mod_headers.so  
# LoadModule proxy_module modules/mod_proxy.so  
# LoadModule rewrite_module modules/mod_rewrite.so  
# LoadModule speling_module modules/mod_speling.so  
# LoadModule info_module modules/mod_info.so  
# LoadModule status_module modules/mod_status.so  
# LoadModule usertrack_module modules/mod_usertrack.so
```

"ExtendedStatus" legt fest, ob Apache eine vollständige Status-Information (On) oder nur die Basis-Informationen ausgeben soll, wenn der "server-status" im Browser abgerufen wird. Der Standard ist Off.

ExtendedStatus On

2. Abschnitt: Generelle Server-Konfiguration

Die Anweisungen in diesem Abschnitt legen die generell vom Server benötigten Werte fest, mit denen auf alle Anfragen reagiert wird, die nicht von virtuellen Hosts abgearbeitet werden. Alle diese Anweisungen können auch innerhalb von <VirtualHost>-Containern stehen, womit dann die hier vorgenommenen Standard-Einstellungen für den jeweiligen virtuellen Host überschrieben werden.

"Port": Der Port, den der standalone-Server benutzt. Manche Firewalls müssen korrekt konfiguriert sein, wenn Apache auf einen bestimmten Port zugreifen soll.

Es können auch andere aktive HTTPD-Server versuchen auf diesen Port zuzugreifen, falls Du solche Software auf Deinem Rechner installiert hast. Wenn Du Probleme bekommst, solltest Du zuerst sämtliche Firewalls, Sicherheitssoftware und ähnliches deaktivieren. Um solche Probleme zu diagnostizieren, kannst Du unter WINDOWS den Befehl "netstat -a" in der Befehlszeile (DOS-Prompt) eingeben.

Port 80

"ServerAdmin" gibt die Adresse (eMail-Adresse) an, an die eventuelle Probleme mit dem Server gemeldet werden können. Diese Adresse erscheint auf allen Dokumenten, die vom Server generiert werden, zum Beispiel Fehlermeldungen.

ServerAdmin webmaster@localhost

"ServerName" erlaubt Dir, einen Hostnamen festzulegen, der an Clients zurückgegeben wird - falls er sich von dem unterscheiden soll, den das Programm ausgeben würde (zum Beispiel kannst Du "www" statt des realen Hostnamens festlegen).

ACHTUNG: Du kannst nicht einfach irgendeinen Hostnamen erfinden und erwarten, daß er funktioniert. Ein Name, den Du hier angibst, muß ein gültiger DNS-Name sein (er sollte bei der zuständigen NIC registriert sein). Wenn Dir nicht klar ist, was damit gemeint ist, frage Deinen Netzwerkadministrator bzw. Provider. Wenn Dein Host nicht über einen DNS-Namen verfügt, gebe hier seine IP-Adresse an. Du kannst dann auf jeden Fall über diese IP-Adresse Zugriff bekommen (z.B. <http://192.168.0.1>).

127.0.0.1 ist die lokale Loopback-Adresse für das TCP/IP-Protokoll, meist als "localhost" bezeichnet. Dein Computer müßte sich mit dieser Adresse immer selbst erkennen. Wenn Du Apache ausschließlich dazu einsetzt, lokale Tests und Entwicklungsarbeiten durchzuführen, kannst Du hier 127.0.0.1 als Servernamen eintragen.

ServerName localhost

"DocumentRoot": Das Verzeichnis, von dem aus die zur Publikation vorgesehenen Dokumente erreicht werden können. Im Standardfall werden sämtliche Server-Anfragen von diesem Verzeichnis aus beantwortet, und symbolische links sowie Aliase können auf andere Verzeichnisse verweisen (voreingestellt ist das Standardverzeichnis "htdocs").

DocumentRoot "C:/Apache/htdocs"

Jedes Verzeichnis, auf das Apache zugreifen kann, kann unter Berücksichtigung dessen,

welche Dienste und Features hier (und in Unterverzeichnissen) erlaubt und/oder nicht zugelassen sind, konfiguriert werden.

Zuerst wird ein "Standard" so konfiguriert, daß sich ein sehr restriktiver Satz an Berechtigungen ergibt.

```
<Directory /> Options Indexes FollowSymLinks MultiViews AllowOverride None
</Directory>
```

ACHTUNG !!

Von hier an muß zwingend für jedes einzelne Verzeichnis festgeschrieben werden, welche Berechtigungen ihm zugestanden werden sollen. Falls irgendetwas nicht wie erwartet funktioniert, muß sichergestellt werden, daß die dafür gültigen Bedingungen in den folgenden Abschnitten korrekt festgelegt wurden. Wenn das nicht erfolgt, gelten die soeben bestimmten restriktiven Standardbedingungen.

Zunächst folgen die Festlegungen für das "DocumentRoot"-Verzeichnis. (Der Pfad muß mit dem weiter oben angegebenen Pfad übereinstimmen).

```
<Directory "C:/Apache/htdocs">
```

die folgenden Werte können auch "None" oder "All" sein, oder irgendeine Kombination aus "Indexes", "Includes", "FollowSymLinks", "ExecCGI" und "MultiViews".

Denke daran, daß der Wert "MultiViews" explizit angegeben werden muß - "Options All" enthält diese Funktion nicht.

```
Options Indexes FollowSymLinks MultiViews
```

Jetzt werden die Optionen festgelegt, die von der Datei .htaccess überschrieben werden dürfen. Auch hier kann "All" oder irgendeine Kombination von "Options", "FileInfo", "AuthConfig" und "Limit" eingetragen werden.

```
AllowOverride None
```

Festlegung, wer auf den Server zugreifen darf:

```
Order allow,deny
```

```
Allow from all
```

```
</Directory>
```

"UserDir": Das Verzeichnis, das aufgerufen wird, wenn eine "~user"-Anforderung eintrifft.

Unter Win32 ist derzeit nicht versucht worden, ein "home"-Verzeichnis für ein Windows-Login festzulegen. Daher musst Du ein Format wie das untenstehende verwenden. Lese die Dokumentation "UserDir" nach.

```
#<IfModule mod_userdir.c>
```

```
UserDir "D:/apache/users/"
```

```
#</IfModule>
```

Zugriffsberechtigungen für UserDir-Verzeichnisse. Das Folgende ist ein Beispiel, mit dem diese Verzeichnisse nur gelesen werden dürfen (read-only).

```
<Directory "C:/apache/users">  
AllowOverride FileInfo AuthConfig Limit  
Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec  
<Limit GET POST OPTIONS PROPFIND>  
Order allow,deny  
Allow from all  
</Limit>  
<LimitExcept GET POST OPTIONS PROPFIND>  
Order deny,allow  
Deny from all  
</LimitExcept>  
#</Directory>
```

"DirectoryIndex": Der Name/die Namen der Datei(en), der/die als Index-Datei(en) akzeptiert wird/werden. Verschiedene Eintragungen bitte durch Leerzeichen voneinander trennen.

```
<IfModule mod_dir.c>  
DirectoryIndex index.htm index.html index.shtml index.asp index.php  
</IfModule>
```

"AccessFileName": Name der Datei, die in jedem Verzeichnis Informationen über Zugriffsberechtigungen zur Verfügung stellt.

```
# AccessFileName .htaccess
```

Die folgenden Zeilen schützen ".htaccess"-Dateien davor, daß Clients deren Inhalt auslesen können. Aus Sicherheitsgründen ist der Zugriff normalerweise nicht gestattet, da ".htaccess" Informationen zur Autorisierung enthält. Entfernen Sie die Kommentarzeichen, wenn Sie Web-Besuchern gestatten möchten, die Informationen in ".htaccess" zu lesen. Wenn Sie weiter oben die Anweisung "AccessFileName" mit irgendeinem Wert versehen haben, müssen auch hier entsprechende Veränderungen notiert werden.

Analog solltest Du Dateinamen wie ".htpasswd" für eine Datei mit dem aktuellen Paßwort verwenden, die dann ebenfalls vor Zugriffen geschützt ist.

```
#<Files ~ "\.ht">  
Order allow,deny  
Deny from all  
#</Files>
```

"CacheNegotiatedDocs": In der Regel sendet Apache ein "Pragma: no-cache" mit jedem Dokument. Damit werden Proxy-Server veranlaßt, das Dokument nicht zwischenspeichern. Wenn Sie die folgende Zeile auskommentieren, wird dieses Verhalten deaktiviert und Proxyserver können dann Dokumente zwischenspeichern.

```
# CacheNegotiatedDocs
```

"UseCanonicalName": (seit Version 1.3.X) Wenn Du hier "On" angibst, wird immer dann

das, was Du weiter oben als "ServerName" und "Port" eingetragen hast, auf Server-generierten Dokumenten auftauchen, wenn Apache veranlaßt wird, eine sich selbst referenzierende URL (ein Dokument, das auf den Server, von dem die Antwort kommt, zurückverweist) zu erstellen. Wenn Du "Off" angibst, wird Apache versuchen, die für den Client gültige "hostname:port"-Angabe in solche Dokumente aufzunehmen. Diese Angaben beeinflussen auch die Variablen SERVER_NAME und SERVER_PORT in CGI-Scripts.

UseCanonicalName On

"TypesConfig" zeigt an, ob/wo die Datei mime.types (oder ihr äquivalent) gefunden werden kann.

```
<IfModule mod_mime.c>  
TypesConfig conf/mime.types  
</IfModule>
```

"DefaultType" ist der Standard-MIME-Typ, den der Server einem Dokument zuweist, falls es nicht von einer Extension her genauer bestimmt wird. Wenn Ihr Server überwiegend Text- oder HTML-Dokumente enthält ist "text/plain" ein guter Wert. Wenn die meisten Dateien Binärdateien sind (z. B. Applikationen oder Bildformate wie .GIF) solltest Du "application/octet-stream" angeben - statt den Browsern zu erlauben, diese Binärdateien so darzustellen, als ob sie Text wären.

DefaultType text/plain

Das Modul mod_mime_magic gestattet dem Server, verschiedene Hinweise aus einem Dateiinhalt direkt zu nutzen, um den Dateityp zu erkennen. Die Anweisung "MIMEMagicFile" teilt dem Modul mit, wo diese Hinweise zur Typ-Definition zu finden sind. Das Modul mod_mime_magic ist allerdings kein Bestandteil des Standardservers. Sie müssen es selbst mit "LoadModule" einbinden [siehe der Absatz über dynamisch verbundene Objekte im Abschnitt "globale Umgebung" oben] oder den Server neu kompilieren und dabei mod_mime_magic als Teil der Konfiguration einbeziehen (gilt für UNIX-Systeme). Da es nicht zur Standard-Konfiguration gehört, wird es hier in einem eigenen <IfModule>-Container notiert. Das bedeutet, daß die Anweisung "MIMEMagicFile" nur dann befolgt werden kann, wenn das Modul in den Server eingebunden worden ist.

```
#<IfModule mod_mime_magic.c>  
#MIMEMagicFile conf/magic  
#</IfModule>
```

"HostnameLookups": Zeichnet die Namen von Clients auf oder auch nur deren IP-Adresse; z.B. www.apache.org (on) oder 204.62.129.132 (off). Der Standard ist off, weil es insgesamt für das Netz besser ist, wenn die Leute dieses Feature bewußt nutzen. Es bedeutet nämlich, daß jede Anfrage eines Clients mindestens eine lookup-Rückfrage an den Nameserver weitergibt.

HostnameLookups Off

"ErrorLog": Die Ablagedatei für Fehlermeldungen. Wenn Du innerhalb Deines <VirtualHost>-Container keine speziellen ErrorLog-Anweisungen niederlegst, werden Fehlermeldungen dieser virtuellen Hosts hier eingetragen. Wenn Du allerdings eine eigene

Protokolldatei für einen virtuellen Host festgelegt hast, werden Fehler auch dort protokolliert und nicht hier.

ErrorLog logs/error.log

"LogLevel": Legt die Art und Menge der Eintragungen fest, die in der Protokolldatei abgelegt werden sollen. Mögliche Werte sind: debug, info, notice, warn, error, crit, alert, emerg.

LogLevel warn

Die nächsten Anweisungen bestimmen, was bei der Verwendung einer CustomLog-Anweisung in die Protokolldatei eingetragen wird (siehe weiter unten).

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined

LogFormat "%h %l %u %t \"%r\" %>s %b" common

LogFormat "%{Referer}i -> %U" referer

LogFormat "%{User-agent}i" agent

Ablageort und Format des Zugriffsprotokolls (Common Logfile Format). Falls Du nicht innerhalb von <VirtualHost>-Container eigene Zugriffs-Protokolldateien definiert hast, werden Zugriffe hier aufgezeichnet. Wenn es aber in Deinen <VirtualHost>-Containern definierte Zugriffs-Protokolldateien gibt, werden alle Berichte dort niedergelegt und nicht hier.

CustomLog logs/access.log common

Wenn Du getrennte "agent"- und "referer"-Protokolldateien haben möchtest, entferne Dir die Kommentarzeichen vor den folgenden Anweisungen.

CustomLog logs/referer.log referer

CustomLog logs/agent.log agent

Wenn eine einzige Protokolldatei gewünscht wird mit allen Informationen (kombiniertes Protokolldateiformat), kann die folgende Anweisung genutzt werden.

CustomLog logs/access.log combined

In serverseitig generierte Dokumente (Fehlermeldungen, FTP-Verzeichnislisten usw., jedoch keine von CGI-Scripts erstellten Dokumente) kannst Du eine Zeile einfügen, die die Server-Version und den virtuellen Hostnamen enthält. Wenn Du hier "EMail" angibst, erscheint außerdem ein Link mit der eMail-Adresse des Server-Administrators. Angegeben werden kann einer dieser Werte: On | Off | EMail

ServerSignature On

Apache durchsucht grundsätzlich sämtliche CGI-Scripts nach einer "shebang"-Zeile. Diese Kommentarzeile, die immer die erste Zeile des Scripts sein muß, besteht aus der Raute (#) und einem Ausrufezeichen (!), gefolgt von der Pfadangabe für das Programm, das dieses spezielle CGI-Script ausführen soll.

Beispiel:

Du möchtest ein Perl-Script als CGI-Script ausführen lassen. Der Pfad zu Deiner "perl.exe" lautet "C:\Perl\bin". Dann muß die "shebang"-Zeile folgendes Aussehen haben:
#!C:/Perl/bin/perl.exe

Auf gar keinen Fall darf die aktuelle "shebang"-Zeile eingerückt werden, und es muß auch immer die erste Zeile in Deinem Script sein. Selbstverständlich mußt Du auch die Ausführung von CGI-Programmen durch Deinen Server zulassen, indem Du entsprechende Script-Aliase vergibst oder die Anweisung "Options ExecCGI" für die infrage kommenden Dateien bzw. Verzeichnisse aktivierst.

Apache kann unter WINDOWS entweder das eben beschriebene UNIX-Verhalten befolgen oder die registry nach Dateinamen-Erweiterungen durchsuchen. Der Befehl zum Ausführen einer Datei, deren Typ in der registry beschrieben ist, wird übernommen und mit derselben Methode abgearbeitet, mit der der WINDOWS- Explorer diesen Dateityp bei einem Doppelklick mit der Maustaste behandeln würde. Du kannst unter WINDOWS selbst bestimmen, welches Programm für die Abarbeitung von bestimmten Dateinamen-Erweiterungen zuständig sein soll. Dazu öffne im WINDOWS-Explorer unter "Ansicht" die "Ordneroptionen", klicke dort auf die Registerkarte "Dateitypen" und lege in dem sich öffnenden Dialogfenster fest, welches Programm Apache beim Aufruf des ausgewählten Dateityps ausführen soll. Schlägt eine Programmausführung fehl, wird Apache versuchen, den "shebang"- Befehl zu befolgen. Dieses Verhalten soll in der Apache-Version 2.0 anders gestaltet werden.

Jeder Mechanismus hat seine eigenen Sicherheitsrisiken, wenn Du ein Programm von einer Webseite aus direkt aufrufst, aber nicht immer erst eine Sicherheitsabfrage beantworten möchtest. Die Suche nach geeigneten risikolosen Methoden ist die Grundlage einer unendlichen Debatte.

Um nun dieses spezielle WINDOWS-Verhalten zu erlauben (und den äquivalenten UNIX-Stil zu unterdrücken) entferne das Kommentarzeichen (#) vor der nächsten Zeile.

ScriptInterpreterSource registry

Die eben gegebene Anweisung kann in einen individuellen <Directory>-Block oder in eine ".htaccess"-Datei geschrieben werden, entweder mit der Option 'registry' (Windows-Verhalten) oder 'script' (Unix-Stil), womit dann die Standardoption des Servers überschrieben würde.

Aliases: Du kannst hier so viele Aliasnamen angeben wie Du willst (Anzahl ist nicht beschränkt).

<IfModule mod_alias.c>

Beachte: Wenn Du den Aliasnamen mit einem Slash ("/") beendest, verlangt Apache diesen Schrägstrich auch in einer URL. So würde hier "/icons" nicht als Alias akzeptiert, nur "/icons/" ist ein gültiger Aliasname.

Alias /icons/ "C:/Apache/icons/"

<Directory "C:/apache/icons">

Options Indexes MultiViews
AllowOverride None
Order allow,deny
Allow from all
</Directory>

ScriptAlias: Damit wird festgelegt, welche Verzeichnisse Server-Scripts enthalten. ScriptAliases sind nahezu dasselbe wie Aliases, mit dem Unterschied, daß Dokumente des mit dem Alias benannten Verzeichnisses als Applikationen behandelt werden und immer dann über den Server laufen, wenn sie angefordert werden. Es gelten für den nachgestellten Schrägstrich dieselben Bestimmungen wie oben für Aliases angegeben.

ScriptAlias /cgi-bin/ "C:/apache/cgi-bin/"
ScriptAlias /Php/ "C:/Php/"

"C:/Apache/cgi-bin" ist das Standard-Verzeichnis für CGI-Scripts. ändere den Pfad entsprechend Deinen Festlegungen für ScriptAliases; Du kannst mehrere Verzeichnisse angeben.

<Directory "C:/Apache/cgi-bin">
Options Indexes FollowSymLinks ExecCGI
AllowOverride None
Order allow,deny
Allow from all
</Directory>
</IfModule>

"Redirect" (Weiterleitung) gibt Dir die Möglichkeit, Clients bekanntzugeben, welche Dokumente lediglich im Namensraum des Servers angewendet werden dürfen und sonst nichts weiter bewirken können. Damit kann Dein Client erfahren, wo er nach einem weitergeleiteten Dokument suchen kann. Format: Redirect (alt)URI (neu)URL

Jetzt folgen Anweisungen, die über die Art und Weise der Bildschirmausgabe von servergenerierten Verzeichnisanzeigen entscheiden.

<IfModule mod_autoindex.c>

"FancyIndexing" bestimmt, ob Du eine Standardausgabe sehen oder eine grafisch etwas aufgelockerte Anzeige auf Deinem Monitor.

IndexOptions FancyIndexing

"AddIcon*" teilt dem Server mit, welches Icon für welche Datei-Erweiterung verwendet werden soll. Diese Icons siehst Du natürlich nur, wenn oben "FancyIndexing" angegeben wurde. Du kannst die Liste beliebig bearbeiten.

AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrm .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

"DefaultIcon" ist das Icon, das immer dann angezeigt wird, wenn für den Dateityp in der Liste oben kein eigenes Icon definiert wurde.

DefaultIcon /icons/unknown.gif

"AddDescription" erlaubt Dir, eine Kurzbeschreibung der verschiedenen Dateitypen anzeigen zu lassen. Diese Kurzbeschreibungen werden ebenfalls nur dann angezeigt, wenn oben "FancyIndexing" gewählt wurde.

Format: AddDescription "Beschreibungstext" Dateityp (Extension)

AddDescription "GZIP-Archiv" .gz
AddDescription "tar-Archiv" .tar
AddDescription "TGZ-Archiv" .tgz
AddDescription "ZIP-Archiv" .zip
*AddDescription "HTML-Dokument" *.htm**

"HeaderName" ist der Name einer Datei, die einem Verzeichnis-Index vorangestellt werden soll und individuelle Informationen über dieses Verzeichnis enthält.

"ReadmeName" ist der Name der README-Datei, in der der Server im Normalfall nach Zusatzinformationen sucht, die an Verzeichnislisten angehängt werden sollen.

Wenn Du in den Optionen "MultiViews" vorgegeben haben, wird der Server zuerst nach einer Datei "name.html" suchen und sie einbinden, falls sie existiert. Wenn es keine solche Datei gibt, sucht er nach "name.txt" und bindet sie als einfache Textdatei ein.

HeaderName HEADER
ReadmeName README

"IndexIgnore" ist ein Satz an Dateinamen, die bei der Auflistung von Verzeichnisinhalten ignoriert und nicht angezeigt werden sollen. Wildcards sind zulässig

```
# IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

```
</IfModule>
```

Dokument-Typen

```
<IfModule mod_mime.c>
```

"AddEncoding" ist eine Anweisung, auf deren Grundlage manche Browser (Mosaic/X 2.1+) Dateiinhalte "on the fly" aus komprimierten Archiven verarbeiten können. Beachte aber, daß nicht alle Browser dieses Feature zur Verfügung stellen. Trotz der Namensähnlichkeit haben die folgenden Add*-Anweisungen nichts mit dem oben angegebenen "FancyIndexing" zu tun.

```
AddEncoding x-compress Z
```

```
AddEncoding x-gzip gz tgz
```

"AddLanguage": damit kannst Du die Sprache eines Dokuments festlegen. Bei Server-Abfragen können dann Dateiinhalte vereinbart werden, so daß ein abfragender Browser Dateien in der Sprache erhält, die er am besten versteht.

Anmerkung 1: Das Suffix muß nicht dasselbe sein wie das in Dokumenten bzw. META-tags verwendete Schlüsselwort für die Sprache. Beispielsweise können polnische Dokumente, deren Standard-Code im WWW "pl" ist, "AddLanguage pl .po" verlangen - um Konflikte mit dem für Perl-Scripts gebräuchlichen Suffix zu vermeiden.

Anmerkung 2: Die Beispiele weiter unten zeigen, daß in sehr vielen Fällen die beiden Buchstaben für "Language" nicht mit den beiden Buchstaben für das zugehörige Land übereinstimmen - siehe z.B. "Dänemark/dk" und "Dänisch/da". Dagegen gibt es bei "Deutschland/de" und "Deutsch/de" durchaus eine Übereinstimmung.

Anmerkung 3: Bei "ltz" haben die Apache-Entwickler bewußt gegen die RFC verstoßen und ein aus drei Buchstaben bestehendes Kürzel definiert. Aber da Apache insgesamt ein "work in progress" ist, wird dieser Verstoß mit einer der nächsten Versionen korrigiert.

```
# AddLanguage da .dk
```

```
# AddLanguage nl .nl
```

```
AddLanguage en .en
```

```
# AddLanguage et .ee
```

```
# AddLanguage fr .fr
```

```
AddLanguage de .de
```

```
# AddLanguage el .el
```

```
# AddLanguage he .he
```

```
# AddCharset ISO-8859-8 .iso8859-8
```

```
# AddLanguage it .it
```

```
# AddLanguage ja .ja
```

```
# AddCharset ISO-2022-JP .jis
```

```
# AddLanguage kr .kr
```

```
# AddCharset ISO-2022-KR .iso-kr
```

```
# AddLanguage no .no
# AddLanguage pl .po
# AddCharset ISO-8859-2 .iso-pl
# AddLanguage pt .pt
# AddLanguage pt-br .pt-br
# AddLanguage ltz .lu
# AddLanguage ca .ca
# AddLanguage es .es
# AddLanguage sv .se
# AddLanguage cz .cz
# AddLanguage ru .ru
# AddLanguage tw .tw
# AddCharset Big5 .Big5 .big5
# AddCharset WINDOWS-1251 .cp-1251
# AddCharset CP866 .cp866
# AddCharset ISO-8859-5 .iso-ru
# AddCharset KOI8-R .koi8-r
# AddCharset UCS-2 .ucs2
# AddCharset UCS-4 .ucs4
# AddCharset UTF-8 .utf8
```

"LanguagePriority": damit kannst Du Prioritäten setzen, wenn Server und anfragender Browser über Dateiinhalte zu verhandeln beginnen. Gebe die Sprachen in der Reihenfolge der für Sie gültigen Prioritäten an.

```
<IfModule mod_negotiation.c>
LanguagePriority de en
</IfModule>
```

"AddType" gibt Dir die Möglichkeit, die Informationen der Datei "mime.types" einzuarbeiten, ohne diese Informationsdatei ausgeben oder bestimmten Dateien bestimmte Typen zuordnen zu müssen.

Beispielsweise verwendet das PHP3.X-Modul (das kein Teil der Apache- Distribution ist - siehe <http://www.php.net>) die folgenden charakteristischen Zeilen:

```
# AddType application/x-httpd-php .php
# AddType application/x-httpd-php .php3
# AddType application/x-httpd-php .php4
# AddType application/x-httpd-php3-source .phps
```

Diese beiden Angaben werden in späteren Apache-Versionen möglicherweise "deprecated", da sich für PHP-Scripts seit PHP 4.X die Extension "php" durchzusetzen beginnt (statt "php3"). Wenn Du PHP 4.X installiert hast, notieren Dir bitte (wobei Du alle Dateinamen-Erweiterungen angeben solltest, die ausführbaren PHP-Code enthalten dürfen):

```
# AddType application/x-httpd-php .php .php3 .php4 .phtml .phpt .htm .html .shtml
# AddType application/x-httpd-php-source .phps
```

```
AddType application/x-tar .tgz
```

"AddHandler" erlaubt es, ausgewählte Dateitypen an "Handler" zu übergeben, unabhängig vom Dateityp selbst. Diese "Handler" (Behandlungsroutinen) können entweder im Server enthalten sein oder mit dem Befehl "action" eingebunden werden (siehe unten).

Wenn Du SSI (Server Side Includes) oder CGI außerhalb der mit ScriptAlias festgelegten Verzeichnisse zulassen möchtest, entferne die Kommentarzeichen vor den nachfolgenden Angaben.

für CGI-Scripts:

```
# AddHandler cgi-script .cgi
```

für HTML-Dateien, die auf dem Server "geparst" werden.

```
# AddType text/html .shtml  
# AddType text/html .html  
AddHandler server-parsed .shtml
```

Entferne das Kommentarzeichen, wenn Du Apache's "send-asis" HTTP-Datei zulassen möchtest.

```
# AddHandler send-as-is asis
```

Wenn Du Image-Maps zulassen möchtest:

```
# AddHandler imap-file map
```

Um type-maps einzusetzen, kannst Du den folgenden Eintrag nutzen

```
# AddHandler type-map var
```

```
</IfModule>
```

"Action" definiert einen Medien(Datei)-Typ, der immer dann zugeordnet wird, wenn eine entsprechende Datei aufgerufen wird. Dadurch entfällt der Zwang, Pfadnamen immer wieder angeben zu müssen.

Format: Action media/type /cgi-script/location

Format: Action handler-name /cgi-script/location

```
# Action application/x-httpd-php "/Php/php.exe"
```

"MetaDir" ist der Name eines Verzeichnisses, in dem Apache Dateien mit "META-Informationen" finden kann. Diese Dateien enthalten zusätzliche HTTP-Header, die der Server grundsätzlich jedem auszuliefernden Dokument mitgeben soll.

```
# MetaDir .web
```

"MetaSuffix" bestimmt den Dateityp (das zugehörige Suffix) für die Datei, die diese META-Informationen enthält.

```
# MetaSuffix .meta
```

Konfigurierbare Fehlermeldungen (Apache-Stil) das kann auf drei verschiedene Arten geschehen: 1) plain text

```
ErrorDocument 500 "Der Server sagt nur ??????"
```

Nachbemerkung: Die vorangestellten Anführungszeichen (") markieren die Meldung selbst als Ausgabertext, der sonst keine weiteren Ausgaben erzeugt.

2) lokale Weiterleitung

Damit wird sofort zu einer lokalen URL `"/missing.html"` weitergeleitet.

```
# ErrorDocument 404 /missing.html
```

Du kannst natürlich auch zu einem Script oder einem Dokument mit SSI (server-side-`includes`) weiterleiten.

```
# ErrorDocument 404 /cgi-bin/missing_handler.pl
```

3) externe Weiterleitung

```
# ErrorDocument 402 http://www.error.com/fehler.html
```

Nachbemerkung: Viele mit der originalen Anfrage verbundene Umgebungs- Variablen werden einem auf diese Weise aufgerufenen Script nicht zur Verfügung stehen.

Anpassung des Browserverhaltens

```
<IfModule mod_setenvif.c>
```

Die folgenden Anweisungen modifizieren das normale Verhalten bei Reaktionen auf HTTP-Anfragen. Die erste Anweisung verhindert persistente Verbindungen für Netscape 2.x und andere Browser, die solche Verbindungen nur vortäuschen. Es gibt ein paar bekannte Probleme bei den Implementationen dieser Browser. Die zweite Anweisung gilt für Microsoft Internet Explorer 4.0b2, in den HTTP/1.1 nicht korrekt implementiert wurde und der deshalb "keepalive" nicht richtig unterstützt, sobald über persistente Verbindungen 301- oder 302 (Weiterleitungs)-Reaktionen des Servers angefordert werden.

```
BrowserMatch "Mozilla/2" nokeepalive
```

```
BrowserMatch "MSIE 4.0b2;" nokeepalive downgrade-1.0 force-response-1.0
```

Die folgende Anweisung verhindert HTTP/1.1-Ausgaben an Browser, die gegen die HTTP/1.0- Spezifikationen verstoßen und dadurch nicht fähig sind, eine Standard-Reaktion auf HTTP/1.1 zu erzeugen.

```
BrowserMatch "RealPlayer 4.0" force-response-1.0
```

```
BrowserMatch "Java/1.0" force-response-1.0
```

```
BrowserMatch "JDK/1.0" force-response-1.0
```

```
</IfModule>
```

Wenn Du dir Informationen über den Serverstatus mit der URL `http://servername/server-status` ausgeben lassen möchtest, gebe hier statt `".deine_domain.com"` den passenden Domainnamen an.

```
<Location /server-status>
```

```
SetHandler server-status  
Order deny,allow  
Allow from all  
# Allow from .deine_domain.com  
</Location>
```

Wenn Du dir mit der URL <http://servername/server-info> die Konfiguration von "Remote"-Servern anzeigen lassen willst (das setzt voraus, daß das Modul `mod_info.c` geladen ist), verwende die folgenden Zeilen. Gebe statt `".deine_domain.com"` die gewünschte Domain an.

```
<Location /server-info>  
SetHandler server-info  
Order deny,allow  
Allow from all  
# Allow from .deine_domain.com  
</Location>
```

Es gab Hinweise von Leuten, die versucht haben, einen alten Fehler aus den Tagen der Version "pre-1.1" auszunutzen. Dieser Fehler betraf die Ausgabe von CGI-Scripts auf eine Weise, als seien sie Bestandteile von Apache. Wenn Du die folgenden Zeilen auskommentierst, werden diese Zugriffe zu einem Protokollscript auf phf.apache.org umgeleitet. Du kannst sie auch zunächst selbst aufzeichnen, indem Du ein Script `"/support/phf_abuse_log.cgi"` benutzt.

```
#<Location /cgi-bin/phf*>  
# Deny from all  
# ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi  
# </Location>
```

Anweisungen zur Konfiguration eines Proxy-Servers. Entferne die Kommentarzeichen, wenn Du Proxies einrichten möchtest.

```
# <IfModule mod_proxy.c>  
# ProxyRequests On  
  
# <Directory proxy:*>  
# Order deny,allow  
# Deny from all  
# Allow from .your_domain.com  
# </Directory>
```

Zulassen/Verbieten der Behandlung von HTTP/1.1 "Via:"-Headern. ("Full" schreibt die Serverversion dazu, "Block" entfernt alle "Via:"-Header) Mögliche Angaben: Off | On | Full | Block

```
# ProxyVia On
```

Wenn Du möchtest, daß ein Proxy, Dokumente in seinem Cache zwischenspeichert, entferne die Kommentarzeichen vor den folgenden Zeilen (ohne "CacheRoot" gibt es keine Zwischenspeicherung).

```
# CacheRoot "d:/apache/proxy"  
# CacheSize 5  
# CacheGcInterval 4  
# CacheMaxExpire 24  
# CacheLastModifiedFactor 0.1  
# CacheDefaultExpire 1  
# NoCache a_domain.com another_domain.edu joes.garage_sale.com  
# </IfModule>
```

3. Abschnitt: Virtuelle Hosts

"VirtualHost": Wenn mit mehreren Domain- bzw. Host-Namen auf diesem Server gearbeitet werden soll, können virtuelle "Container" für jeden einzelnen Hostnamen festgelegt werden. Bitte unbedingt die Dokumentation auf <http://www.apache.org/docs/vhosts/> nachlesen bzw. das mit der Distribution ausgelieferte Manual. Um die Konfiguration der virtuellen Hosts zu überprüfen, steht der Befehlszeilen-Parameter "-S" zur Verfügung.

Konfiguration für DNS-Namen-gestützte virtuelle Hosts

```
# NameVirtualHost *
```

Beispiel für einen virtuellen Host:

Fast jede Apache-Anweisung kann in einem "VirtualHost"- Container verwendet werden. Der erste Container in der (beliebig langen) Liste wird dabei immer genutzt, wenn eine Webadresse ohne bekannten Servernamen verlangt wird.

```
# <VirtualHost *>  
# ServerAdmin webmaster@dummy-host.example.com  
# DocumentRoot /www/docs/dummy-host.example.com  
# ServerName dummy-host.example.com  
# ErrorLog logs/dummy-host.example.com-error_log  
# CustomLog logs/dummy-host.example.com-access_log common  
#</VirtualHost>
```

4. Abschnitt - Zusatzmodule

```
LoadModule php4_module C:/php4/sapi/php4apache.dll
```