

Sascha Kersken

# Apache 2

Galileo Computing 

# Auf einen Blick

Vorwort .....	13
1 IP-Netzwerke, Internet und WWW .....	19
2 Funktionsweise von Webservern .....	51
3 Apache 2 im Überblick .....	111
4 Apache kompilieren und installieren .....	153
5 Apache in Betrieb nehmen .....	203
6 Grundkonfiguration .....	237
7 Header und MIME-Einstellungen .....	305
8 Weiterleitungen und Indizes .....	353
9 Authentifizierung .....	411
10 Gesicherte Verbindungen .....	477
11 Logging .....	517
12 Skalierung und Performance-Tuning .....	555
13 Proxy- und Cache-Funktionen .....	583
14 CGI .....	617
15 Technologien zur Webprogrammierung .....	667
16 SSI und Filter .....	727
17 Apache erweitern .....	765
18 Sicherheit .....	797
A Besonderheiten von Apache 1.3 .....	815
B Kurzreferenz der Konfigurationsdirektiven .....	823
C Sonstige Tabellen .....	843
D Die Apache-Lizenz 2.0 .....	891
E Reguläre Ausdrücke .....	897
F VMware Workstation .....	899
G Rechtliche Aspekte .....	905
H Literaturverzeichnis .....	909
Index .....	911

# Inhalt

<b>Vorwort</b>	<b>13</b>
<b>1 IP-Netzwerke, Internet und WWW</b>	<b>19</b>
1.1 TCP/IP .....	21
1.1.1 Das Internet-Schichtenmodell .....	21
1.1.2 Das Internet Protocol (IP) .....	24
1.1.3 Transportprotokolle .....	30
1.2 Das Domain Name System (DNS) .....	32
1.2.1 Das DNS-Konzept .....	33
1.2.2 Der DNS-Server BIND .....	36
1.3 TCP/IP-Diagnose und -Fehlersuche .....	42
1.3.1 ping .....	42
1.3.2 traceroute .....	43
1.3.3 netstat .....	44
1.3.4 nslookup .....	45
1.3.5 telnet .....	47
1.4 Zusammenfassung .....	48
<b>2 Funktionsweise von Webservern</b>	<b>51</b>
2.1 Das HTTP .....	53
2.1.1 Die HTTP-Client-Anfrage .....	55
2.1.2 HTTP-Statuscodes .....	67
2.1.3 HTTP-Header .....	72
2.2 Einstieg für Programmierer: Ein selbst geschriebener Webserver .....	92
2.2.1 Projektanforderungen .....	92
2.2.2 Implementierungsdetails .....	93
2.2.3 Der komplette Quellcode .....	102
2.2.4 Benutzerdokumentation .....	108
2.3 Zusammenfassung .....	110
<b>3 Apache 2 im Überblick</b>	<b>111</b>
3.1 Einführung .....	113
3.1.1 Entstehungsgeschichte des Apache-Webservers .....	113
3.1.2 Die Apache Software Foundation .....	116

3.1.3	Die Apache-Softwarelizenz .....	119
3.1.4	Sonstige Webserver .....	120
<b>3.2</b>	<b>Funktionen von Apache 2 .....</b>	<b>124</b>
3.2.1	Technischer Überblick .....	127
3.2.2	Apache-Module .....	141
<b>3.3</b>	<b>Zusammenfassung .....</b>	<b>152</b>

## **4 Apache kompilieren und installieren 153**

<b>4.1</b>	<b>Apache 2 kompilieren .....</b>	<b>156</b>
4.1.1	Den Quellcode besorgen und auspacken .....	156
4.1.2	Apache 2 unter UNIX kompilieren .....	158
4.1.3	Apache 2 unter Windows kompilieren .....	188
<b>4.2</b>	<b>Die binäre Apache-Distribution für Windows installieren .....</b>	<b>195</b>
<b>4.3</b>	<b>Module nachträglich installieren .....</b>	<b>200</b>
<b>4.4</b>	<b>Zusammenfassung .....</b>	<b>202</b>

## **5 Apache in Betrieb nehmen 203**

<b>5.1</b>	<b>Apache 2 starten und beenden .....</b>	<b>205</b>
5.1.1	Apache unter UNIX steuern .....	205
5.1.2	Apache unter Windows steuern .....	217
5.1.3	Apache-Hilfsprogramme .....	227
<b>5.2</b>	<b>Apache testen .....</b>	<b>228</b>
5.2.1	Die automatische Startseite .....	228
5.2.2	Die erste Website .....	229
<b>5.3</b>	<b>Zusammenfassung .....</b>	<b>235</b>

## **6 Grundkonfiguration 237**

<b>6.1</b>	<b>Aufbau der Apache-Konfigurationsdateien .....</b>	<b>239</b>
6.1.1	Namen, Pfad und Aufgaben der Konfigurationsdateien .....	240
6.1.2	Grundlegendes zur Syntax .....	242
6.1.3	Syntaxschema .....	246
<b>6.2</b>	<b>Kontexte und Container .....</b>	<b>247</b>
6.2.1	Der Server-Kontext .....	247
6.2.2	Virtuelle Hosts .....	247
6.2.3	Verzeichnis- und Datei-Container .....	249
6.2.4	Spezialcontainer .....	254
6.2.5	.htaccess-Dateien .....	258
6.2.6	Einfügen externer Konfigurationsdateien .....	260
<b>6.3</b>	<b>Allgemeine Konfigurationsdirektiven .....</b>	<b>261</b>
6.3.1	Einrichten der Server-Umgebung .....	261
6.3.2	Plattformspezifische Server-Einstellungen .....	268

6.3.3	Konfiguration des »Hauptservers« .....	285
6.3.4	Wichtige Verzeichniseinstellungen .....	293
6.4	<b>Zusammenfassung</b> .....	302

## **7 Header und MIME-Einstellungen 305**

7.1	<b>HTTP-Header manipulieren</b> .....	307
7.1.1	MD5-Digest und ETag .....	307
7.1.2	mod_headers .....	309
7.1.3	mod_expires .....	315
7.1.4	mod_asis .....	318
7.1.5	mod_cern_meta .....	319
7.2	<b>MIME-Konfiguration</b> .....	321
7.2.1	MIME-Type-Einstellungen .....	323
7.2.2	Zeichensatzeinstellungen .....	329
7.2.3	MIME-Codierung .....	332
7.2.4	Spracheinstellungen .....	334
7.2.5	Handler festlegen .....	336
7.3	<b>Content-Negotiation</b> .....	339
7.3.1	Servergesteuerte Content-Negotiation .....	340
7.3.2	Transparente Content-Negotiation .....	347
7.3.3	Konfigurationseinstellungen für Content-Negotiation .....	349
7.4	<b>Zusammenfassung</b> .....	352

## **8 Weiterleitungen und Indizes 353**

8.1	<b>Aliase und Weiterleitungen</b> .....	355
8.1.1	mod_alias .....	356
8.1.2	mod_rewrite .....	363
8.1.3	Benutzerverzeichnisse veröffentlichen .....	383
8.1.4	Fehlerbehandlung .....	386
8.1.5	Rechtschreibkorrektur in URLs mit mod_speling .....	388
8.1.6	Status- und Konfigurationsinformationen über den Server .....	389
8.2	<b>Indizes</b> .....	392
8.2.1	mod_autoindex .....	393
8.2.2	Serverseitige Image-Maps mit mod_imagemap .....	404
8.3	<b>Zusammenfassung</b> .....	409

## **9 Authentifizierung 411**

9.1	<b>Grundlagen der Authentifizierung</b> .....	413
9.1.1	Die Organisation der Authentifizierung in Apache 2.0 .....	414
9.1.2	Die Neuordnung der Authentifizierungsmodule in Apache 2.2 ..	415
9.1.3	Ein erstes Beispiel .....	417
9.1.4	Core-Direktiven zur Authentifizierung .....	420

<b>9.2</b>	<b>Basic-Authentifizierung</b> .....	<b>424</b>
9.2.1	Das Programm htpasswd .....	424
9.2.2	Direktiven zur Textdatei-basierten Basic-Authentifizierung .....	426
<b>9.3</b>	<b>Digest-Authentifizierung</b> .....	<b>431</b>
9.3.1	Das Tool htdigest .....	432
9.3.2	Direktiven zur Digest-Authentifizierung .....	434
<b>9.4</b>	<b>Benutzer- und Passwortverwaltung in DBM-Dateien</b> .....	<b>440</b>
9.4.1	Das Tool dbmmanage .....	441
9.4.2	Das Programm httdbm .....	444
9.4.3	DBM-Direktiven .....	445
<b>9.5</b>	<b>LDAP-Authentifizierung</b> .....	<b>449</b>
9.5.1	LDAP-Authentifizierungs-Direktiven .....	450
9.5.2	LDAP-Performanceverbesserung mit mod_ldap .....	458
<b>9.6</b>	<b>Anonymous-Authentifizierung</b> .....	<b>463</b>
<b>9.7</b>	<b>Datenbankbasierte Authentifizierung mit mod_authn_dbd</b> .....	<b>466</b>
9.7.1	Datenbankverbindungen mit mod_dbd .....	467
9.7.2	mod_authn_dbd-Direktiven .....	471
<b>9.8</b>	<b>Sonstige Erweiterungen in Apache 2.2</b> .....	<b>473</b>
9.8.1	mod_authn_alias .....	473
9.8.2	mod_authz_owner .....	474
9.8.3	mod_authn_default und mod_authz_default .....	475
<b>9.9</b>	<b>Zusammenfassung</b> .....	<b>476</b>

## **10 Gesicherte Verbindungen 477**

<b>10.1</b>	<b>SSL-Grundlagen</b> .....	<b>480</b>
10.1.1	SSL einrichten .....	482
10.1.2	SSL-Grundkonfiguration .....	487
10.1.3	mod_ssl-Umgebungsvariablen .....	489
<b>10.2</b>	<b>mod_ssl-Direktiven</b> .....	<b>491</b>
10.2.1	Standard-Direktiven .....	491
10.2.2	mod_ssl-Proxy-Direktiven .....	510
10.2.3	mod_nw_ssl für NetWare .....	514
<b>10.3</b>	<b>Zusammenfassung</b> .....	<b>515</b>

## **11 Logging 517**

<b>11.1</b>	<b>Logging-Direktiven und -Module</b> .....	<b>520</b>
11.1.1	core-Direktiven .....	520
11.1.2	mod_log_config .....	525
11.1.3	mod_log_forensic .....	533
11.1.4	mod_dumpio .....	534
11.1.5	mod_usertrack .....	535
11.1.6	Logging-Direktiven in mod_rewrite .....	538

11.2	Auswertung von Logdateien .....	539
11.2.1	Apache-Hilfsprogramme .....	539
11.2.2	Logdatei-Auswertung durch eigene Skripte .....	541
11.2.3	Externe Tools .....	553
11.3	Zusammenfassung .....	554

## **12 Skalierung und Performance-Tuning 555**

12.1	Virtuelle Hosts .....	557
12.1.1	Konfigurationsbeispiele .....	558
12.1.2	Core-Direktiven für virtuelle Hosts .....	562
12.1.3	mod_vhost_alias .....	566
12.2	Performance-Tuning .....	569
12.2.1	Allgemeines .....	570
12.2.2	Benchmarks mit ab .....	571
12.2.3	Performance-bezogene Core-Direktiven .....	574
12.2.4	mod_file_cache: Häufig genutzte Dateien vorausladen .....	575
12.3	Load-Balancing .....	577
12.3.1	Load-Balancing mit mod_rewrite .....	579
12.3.2	Open-Source-Lösungen für Load-Balancing .....	580
12.4	Zusammenfassung .....	581

## **13 Proxy- und Cache-Funktionen 583**

13.1	Apache als Proxy-Server .....	585
13.1.1	Proxy-Grundkonfiguration .....	587
13.1.2	Referenz der Proxy-Direktiven .....	589
13.2	Cache-Funktionen .....	603
13.2.1	Cache-Grundkonfiguration .....	603
13.2.2	Cache-Direktiven .....	605
13.2.3	htcacheclean .....	615
13.3	Zusammenfassung .....	616

## **14 CGI 617**

14.1	Die CGI-Schnittstelle .....	619
14.2	Apache für CGI-Skripte konfigurieren .....	621
14.2.1	CGI-Verzeichnisse .....	622
14.2.2	CGI in normalen Verzeichnissen aktivieren .....	625
14.2.3	Konfigurationsanweisungen für mod_cgi und mod_cgid .....	627
14.2.4	Plattformspezifische Einstellungen .....	630
14.2.5	Das Modul mod_actions .....	632

<b>14.3</b>	<b>Umgebungsvariablen</b> .....	<b>634</b>
14.3.1	Die Umgebungsvariablen im Überblick .....	635
14.3.2	Umgebungsvariablen in der Apache-Konfiguration setzen .....	637
<b>14.4</b>	<b>Grundlagen der CGI-Programmierung</b> .....	<b>643</b>
14.4.1	Das erste Beispiel .....	644
14.4.2	Formulardaten einlesen .....	645
<b>14.5</b>	<b>Das Perl-Modul CGI.pm</b> .....	<b>647</b>
14.5.1	CGI.pm im Überblick .....	647
14.5.2	Beispiel: Ein kleiner Taschenrechner .....	654
14.5.3	CGI.pm-Kurzreferenz .....	658
<b>14.6</b>	<b>Zusammenfassung</b> .....	<b>666</b>

## **15 Technologien zur Webprogrammierung 667**

<b>15.1</b>	<b>PHP</b> .....	<b>670</b>
15.1.1	MySQL installieren .....	670
15.1.2	PHP installieren .....	677
15.1.3	Die PHP-Konfigurationsdatei php.ini .....	683
15.1.4	phpMyAdmin einrichten .....	686
15.1.5	PHP-Programmierung .....	688
<b>15.2</b>	<b>mod_perl</b> .....	<b>699</b>
15.2.1	Installation von mod_perl .....	699
15.2.2	Perl-Zugriff auf MySQL-Datenbanken .....	706
15.2.3	Perl in der Apache-Konfigurationsdatei .....	707
<b>15.3</b>	<b>Tomcat</b> .....	<b>709</b>
15.3.1	Tomcat installieren .....	709
15.3.2	Tomcat per Proxy einbinden .....	715
15.3.3	Java-Webprogrammierung .....	716
<b>15.4</b>	<b>Weitere Programmierschnittstellen</b> .....	<b>721</b>
15.4.1	ISAPI-Anwendungen mit mod_isapi .....	721
15.4.2	Sonstige Technologien .....	724
<b>15.5</b>	<b>Zusammenfassung</b> .....	<b>725</b>

## **16 SSI und Filter 727**

<b>16.1</b>	<b>Server Side Includes (SSI)</b> .....	<b>729</b>
16.1.1	SSI aktivieren .....	729
16.1.2	SSI-Elemente .....	730
16.1.3	mod_include-Direktiven .....	737
<b>16.2</b>	<b>Filterkonfiguration</b> .....	<b>740</b>
16.2.1	Grundlegende Filter-Direktiven .....	740
16.2.2	Freie Modifikation der Filter Chain mit mod_filter .....	745
16.2.3	Der Komprimierungsfiler mod_deflate .....	750
16.2.4	mod_charset_lite .....	754

16.3	Externe Filter programmieren .....	756
16.3.1	mod_ext_filter .....	756
16.3.2	Beispiele für externe Filter .....	759
16.4	Zusammenfassung .....	764

## **17 Apache erweitern 765**

17.1	WebDAV .....	767
17.1.1	Konfigurationsbeispiel .....	768
17.1.2	DAV-Direktiven .....	768
17.2	Weitere Module .....	771
17.2.1	Multiprotokoll-Unterstützung .....	771
17.2.2	Weitere Drittanbieter-Module .....	773
17.3	Programmierung eigener Module .....	774
17.3.1	mod_example – Erforschen der Modul-API .....	775
17.3.2	Arbeitsweise von Modulen .....	776
17.3.3	Die Modulentwicklung .....	777
17.3.4	mod_daytime – ein Beispiel zur Multiprotokoll-Unterstützung ..	790
17.4	Zusammenfassung .....	794

## **18 Sicherheit 797**

18.1	Sicherheit der Server-Umgebung .....	799
18.2	Apache-Sicherheit .....	801
18.2.1	Allgemeine Sicherheitshinweise .....	801
18.2.2	Sicherheitsrelevante Direktiven .....	803
18.2.3	SuEXEC .....	808
18.3	mod_security .....	811
18.4	Zusammenfassung .....	812

## **A Besonderheiten von Apache 1.3 815**

A.1	Apache 1.3 kompilieren und installieren .....	815
A.2	Wichtige Änderungen bei Direktiven .....	816
A.2.1	Exklusive 1.3-Direktiven .....	816
A.2.2	Nicht vorhandene Core-Direktiven .....	821

## **B Kurzreferenz der Konfigurationsdirektiven 823**

<b>C</b>	<b>Sonstige Tabellen</b>	<b>843</b>
C.1	MIME-Types .....	843
C.2	Sprachcodes nach ISO .....	866
C.3	Zeichensätze .....	871
C.4	Top-Level-Domains .....	881
C.4.1	Generische Top-Level-Domains .....	881
C.4.2	Länder-Top-Level-Domains .....	881
<b>D</b>	<b>Die Apache-Lizenz 2.0</b>	<b>891</b>
<b>E</b>	<b>Reguläre Ausdrücke</b>	<b>897</b>
<b>F</b>	<b>VMware Workstation</b>	<b>899</b>
F.1	Einrichtung einer virtuellen Maschine .....	899
F.2	Die virtuelle Maschine im Betrieb .....	901
F.3	Einstellungen der virtuellen Maschine ändern .....	902
F.4	VMware Tools installieren .....	903
<b>G</b>	<b>Rechtliche Aspekte</b>	<b>905</b>
<b>H</b>	<b>Literaturverzeichnis</b>	<b>909</b>
	<b>Index</b>	<b>911</b>

## Vorwort

*Aller Anfang ist heiter, die Schwelle ist der Platz der Erwartung.  
– Johann Wolfgang Goethe*

Der Apache HTTP Server, Apache-Webserver oder einfach »Apache« ist die populärste Webserver-Software der Welt: Knapp 70% aller Websites werden von Apache serviert. Zudem handelt es sich um eines der erfolgreichsten Open-Source-Softwareprojekte. Die Apache Software Foundation, der »Dachverband« der Apache-Entwickler, betreut Dutzende freier Softwareprojekte; neben dem HTTP Server gehören dazu so bekannte und beliebte Produkte wie Tomcat, SpamAssassin und Xalan. Diese Projekte beeinflussen und ergänzen einander und bestimmen so maßgeblich die Weiterentwicklung von Internet und World Wide Web mit.

Dieses Buch ist ein umfassendes Handbuch zum Apache-Webserver in der aktuellen Version 2; sämtliche Bestandteile, die zum »Lieferumfang« gehören, werden ausführlich beschrieben. Zurzeit wird die neue Version 2.2 vorbereitet, die sich in der späten Beta-Phase befindet; für Produktions-Server ist bisher noch 2.0 zu empfehlen. In diesem Buch werden beide Versionszweige mit ihren Eigenschaften und Unterschieden behandelt. In Kapitel 3 finden Sie unter anderem eine Übersicht über die Neuerungen in Version 2.2.

Vor eineinhalb Jahren, als die erste Version dieses Buches erschien, dominierte noch die alte Apache-Version 1.3. Inzwischen beginnt sich dies zu ändern: Neuinstallationen werden in aller Regel nur noch mit 2er-Versionen durchgeführt; bestehende Sites migrieren allmählich. Von den gesteigerten Praxiserfahrungen mit Apache 2 profitiert sowohl der Webserver selbst als auch dieses Buch.

Die Schwerpunktthemen dieses Buches sind Installation, Administration und Programmierung. Sie erfahren zunächst einmal, wie Sie den Server unter verschiedenen Betriebssystemen kompilieren und/oder installieren können. Im weiteren Verlauf des Buches geht es vor allem um die unzähligen Konfigurationsanweisungen (Direktiven), die in der Hauptkonfigurationsdatei von Apache zur Verfügung stehen. Anders als viele andere Serverprodukte ist Apache nämlich von Hause aus nicht mit einer grafischen Konfigurationsoberfläche ausgestattet. Dies macht seine Administration zwar schwieriger, bietet aber dafür die größtmögliche Flexibilität.

Apache ist für zahlreiche verschiedene Plattformen und Betriebssysteme verfügbar. In Version 2 wurde insbesondere die Unterstützung für Nicht-UNIX-

Systeme verbessert: Als Basis der eigentlichen Server-Implementierung wurde eine Bibliothek namens Apache Portable Runtime (APR) eingeführt, die statt der früher eingesetzten POSIX-Emulation die jeweiligen Stärken der einzelnen Systeme abstrahiert. Auch das Laufzeitverhalten wurde verbessert: Sie können nun aus mehreren so genannten Multiprocessing-Modulen (MPMs) das passendste für Ihre Plattform auswählen.

Ausführlich wird hier die Apache-Konfiguration für die Betriebssysteme UNIX (alle Varianten) und Windows (NT und seine Nachfolger) behandelt. Besonderheiten für andere Plattformen werden gegebenenfalls angemerkt, aber nicht weiter vertieft.

Einen großen Raum nehmen in diesem Buch die zahlreichen Module ein, die mit Apache 2 geliefert werden und für beinahe jeden Verwendungszweck eine praktische Lösung bieten. Auf diese Weise brauchen Sie bestimmte Aspekte der Funktionalität nur dann in Ihren Webserver zu integrieren, wenn Sie sie wirklich benötigen. Dies kann Ihnen helfen, den Überblick zu behalten und schont obendrein die Ressourcen des Server-Rechners.

### **Das Komplettpaket**

Die beiliegende CD-ROM enthält so gut wie alle Programme, Listings und Dokumente, die in diesem Buch angesprochen werden.<sup>1</sup> Unter anderem finden sie darauf Apache-Distributionen für die verschiedensten Betriebssysteme, externe Module, Zusatzprogramme, Skripte und RFC-Dokumente. Die Tatsache, dass die Dateien auf diese Weise weiterverbreitet werden dürfen, ist einer der großen Vorteile freier Software.

Dennoch sollten Sie vor der Installation eines bestimmten Programms von der CD überprüfen, ob nicht bereits eine aktuellere Version verfügbar ist – bei Open-Source-Produkten bedeutet die Bereitstellung einer neuen Release oft, dass wichtige Sicherheitsprobleme aus vorangegangenen Versionen behoben wurden. Eine jeweils aktuelle Liste mit Download-Links und zahlreiche Zusatzinformationen finden Sie auf der Website zum Buch. Die Adresse lautet **[buecher.lingoworld.de/apache2](http://buecher.lingoworld.de/apache2)**.

Auf derselben Website können Sie zudem einen Newsletter abonnieren, der jeden Werktag eine zufällig ausgewählte Apache-Konfigurationsdirektive vorstellt.

---

<sup>1</sup> Einige wenige Tools dürfen aus rechtlichen Gründen nicht auf der CD verbreitet werden. In diesen Fällen finden Sie im Buch entsprechende Download-Links.

## Das Buch im Überblick

In den einzelnen Kapiteln dieses Buches werden folgende Themen behandelt:

- ▶ In Kapitel 1, *IP-Netzwerke, Internet und WWW*, finden Sie eine kurze Übersicht über die Umgebung, in der Apache ausgeführt wird: In knapper Form wird die Technik der TCP/IP-Protokollfamilie erläutert. Darüber hinaus gibt es hier auch eine Einführung in die Einrichtung eines Nameservers und Informationen über einige Hilfsprogramme.
- ▶ Kapitel 2, *Funktionsweise von Webservern*, behandelt das Anwendungsprotokoll HTTP, das die Grundlage des World Wide Web bildet. Neben der Besprechung sämtlicher HTTP-Methoden, -Header und -Statuscodes wird hier zur Veranschaulichung die Programmierung eines kleinen Webservers in Perl gezeigt.
- ▶ In Kapitel 3, *Apache 2 im Überblick*, wird in allgemeiner Form der Funktionsumfang des Webservers beschrieben. Dazu gehören auch Themen wie die Geschichte von Apache, ein Vergleich mit anderen Webservern sowie eine Liste der verfügbaren Module.
- ▶ Wie Sie Apache auf Ihrem System installieren können, wird ausführlich in Kapitel 4, *Apache kompilieren und installieren*, beschrieben. Sie erhalten Anleitungen zur Kompilierung der Quellcode-Pakete unter UNIX und Windows sowie zur Installation diverser Binär-Distributionen.
- ▶ Kapitel 5, *Apache in Betrieb nehmen*, behandelt die Steuerung von Apache. Sie erfahren alles über das Starten, Stoppen und Neustarten des Servers sowie über Möglichkeiten, ihn beim Hochfahren des Systems automatisch zu starten.
- ▶ In Kapitel 6, *Grundkonfiguration*, wird zunächst der allgemeine Aufbau der Konfigurationsdatei `httpd.conf` erläutert. Anschließend werden alle Konfigurationsdirektiven besprochen, die für den Betrieb einer einfachen statischen Website wichtig sind.
- ▶ Kapitel 7, *Header und MIME-Einstellungen*, bietet weitere wichtige Informationen für die Administration von Websites: Apache 2 enthält Module und Konfigurationseinstellungen zur Manipulation von HTTP-Headern und MIME-Informationen. Dazu gehört auch das Thema Content-Negotiation, also die Belieferung von Clients mit deren jeweils bevorzugter Darstellungsform eines Dokuments.
- ▶ Das Thema von Kapitel 8, *Weiterleitung und Indizes*, sind Situationen, in denen sich unter der angeforderten URL kein Dokument befindet: URLs lassen sich auf Dokumente außerhalb des Website-Verzeichnisses oder sogar

auf externe URLs umleiten; Apache kann zudem selbstständig Verzeichnisindizes generieren.

- ▶ In Kapitel 9, *Authentifizierung*, wird die Absicherung von Websites behandelt: die persönliche Anmeldung einzelner User und Gruppen und die Speicherung der entsprechenden Anmeldedaten in Quellen wie Textdateien, Datenbanken oder LDAP-Verzeichnissen.
- ▶ Kapitel 10, *Gesicherte Verbindungen*, behandelt die Bereitstellung SSL/TLS gesicherter Verbindungen, die durch Verschlüsselung und andere Maßnahmen vor Mitlese- oder gar Manipulationsversuchen geschützt werden
- ▶ Kapitel 11, *Logging*, beschäftigt sich mit der Einrichtung und Verarbeitung von Logdateien. Diese wichtigen Helfer geben über alle Zugriffe auf Ihre Websites sowie über mögliche Fehler oder Angriffsversuche Aufschluss.
- ▶ Kapitel 12, *Skalierung und Performance-Tuning*, behandelt die wichtigsten Themen, die für den Betrieb besonders großer Websites relevant sind: Sie erfahren alles über virtuelle Hosts, Performance-Tuning und über Load-Balancing-Verfahren.
- ▶ Kapitel 13, *Proxy- und Cache-Funktionen*, beschreibt den Betrieb von Apache als Proxy-Server für verschiedene Protokolle (HTTP, FTP und andere). Nicht nur zu diesem Zweck ist der Einsatz des Web-Cachings interessant, der ebenfalls hier behandelt wird.
- ▶ In Kapitel 14, *CGI*, wird die klassische Schnittstelle zur Entwicklung von Web-Anwendungen vorgestellt, das Common Gateway Interface (CGI). Die Beispiele sind in Perl geschrieben; in diesem Zusammenhang lernen Sie das Perl-Modul `CGI.pm` kennen, das die Entwicklung von CGI-Skripten erheblich einfacher und komfortabler macht.
- ▶ Kapitel 15, *Technologien zur Webprogrammierung*, versammelt die beliebtesten Schnittstellen für die Entwicklung von Web-Anwendungen: PHP, `mod_perl` und Tomcat. Der Schwerpunkt ist die Integration der Module in Apache; darüber hinaus gibt es einige Programmierbeispiele und -tipps.
- ▶ In Kapitel 16, *SSI und Filter*, wird das interessante Konzept der Filter vorgestellt, das in Apache 2 neu eingeführt wurde: Eingehende Anfragedaten lassen sich ebenso leicht modifizieren wie die eigentlich schon fertige Antwort an die Clients. Das klassische SSI-Verfahren (Server Side Includes) wurde in das Filterkonzept integriert und wird deshalb ebenfalls in diesem Kapitel behandelt.
- ▶ Kapitel 17, *Apache erweitern*, beschreibt zunächst den Betrieb von Apache als WebDAV-Server, anschließend geht es um diverse Drittanbieter-Module. Zum Schluss gibt es ein Tutorial über die Programmierung eigener Module.

- In Kapitel 18, *Sicherheit*, werden einige wichtige Aspekte der Apache-Sicherheit behandelt: Die Absicherung der Systemumgebung und verschiedene Sicherheitsaspekte des Webservers selbst. Abschließend erhalten Sie einen Überblick über das Drittanbieter-Modul `mod_security`, das zusätzliche Sicherheitsmaßnahmen in den Server einfügt.

In den sich daran anschließenden Anhängen finden Sie zusätzliches interessantes Material: eine kurze Übersicht über die Besonderheiten der Vorgängerversion Apache 1.3, einige Tabellen mit MIME-Types, Sprachkürzeln und Zeichensätzen, eine Zusammenfassung rechtlicher Aspekte und andere Themen.

## Danksagungen

Dies ist die zweite Auflage eines Buches, die nicht erschienen wäre, wenn die vorige gar keinen Erfolg gehabt hätte. Insofern danke ich allen Käufern und Lesern der ersten Auflage dieses Buches, die mitgeholfen haben, dieses neue Projekt zu ermöglichen. Einige Leser und Abonnenten des Newsletters haben mir zudem ausführliches Feedback gegeben; viele ihrer Verbesserungsvorschläge haben ihren Weg in diese Neuauflage gefunden. Auch dafür möchte ich mich bedanken.

Wie immer auch vielen, vielen Dank an Stephan Mattescheck und den Rest des Teams von Galileo Press – für ihre grenzenlose Geduld und für die hervorragende Zusammenarbeit an mittlerweile fünf Projekten.

Weiterer Dank gebührt natürlich den Entwicklern des Apache-Webservers. Ohne die zahllosen Stunden, die diese Enthusiasten freiwillig in die Entwicklung und Verbesserung dieses großartigen Produkts gesteckt haben, gäbe es das Thema dieses Buches gar nicht. Wenn Sie sich erkenntlich zeigen möchten, sollten Sie die Apache Software Foundation mit Spenden bedenken, vor allem aber mithelfen, etwas gegen die Einführung von Softwarepatenten in der Europäischen Union zu tun. Ein guter Ausgangspunkt dafür ist die Website <http://www.ffii.org>.

Zu guter Letzt ist es meine Familie, die mich am meisten bei der Arbeit an diesem Buch unterstützt hat – die letzten Wochen müssen für sie gewesen sein. Ich danke meiner Frau Tülay und meinem Sohn Leon für all ihre Geduld und Unterstützung und hoffe, dass diese sich eines Tages so auszahlen werden, dass ich endlich mehr Zeit für sie habe.



## 5 Apache in Betrieb nehmen

*Im Grunde bewegen nur zwei Fragen die Menschheit: Wie hat alles angefangen und wie wird alles enden?*  
– Stephen Hawking

5

In diesem Kapitel wird die komplette Steuerung des Apache-HTTP-Servers behandelt: Sie erfahren zunächst, welche Optionen das ausführbare Programm bietet, um den Server zu starten, zu beenden oder neu zu starten. Anschließend erfahren Sie, wie sich der Start von Apache auf verschiedenen Plattformen beim Booten automatisieren lässt. Zudem lernen Sie einige Apache-Steuertools aus Systemdistributionen sowie von Drittanbietern kennen. Zu guter Letzt wird eine Minimalkonfiguration vorgestellt, mit der Sie die ordnungsgemäße Funktion des Servers überprüfen können.

### 5.1 Apache 2 starten und beenden

Nachdem Sie den Apache-Webserver nun (hoffentlich) auf die eine oder andere im vorigen Kapitel beschriebene Weise installiert haben, können Sie ihn starten. Je nach Betriebssystem und Plattform stehen dafür unterschiedliche Optionen zur Verfügung. Sie erfahren in diesem Abschnitt, wie sich der Server unter UNIX und Windows starten, beenden und neu starten lässt. Außerdem werden verschiedene Optionen des automatischen Starts beim Hochfahren besprochen.

#### 5.1.1 Apache unter UNIX steuern

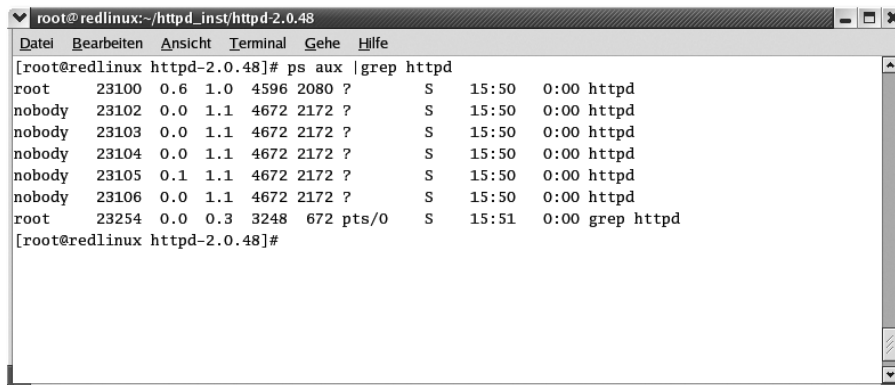
Auf UNIX-Systemen können Sie das ausführbare Programm `httpd` mit verschiedenen Parametern aufrufen, um den Webserver zu starten, neu zu starten, zu beenden oder Informationen über seinen Status zu erhalten. Die bevorzugte Variante ist allerdings die Verwendung des mitinstallierten Shell-Skripts `apachectl`, das eine genauere Kontrolle ermöglicht. Beide Varianten werden hier vorgestellt.

Ob der Webserver bereits ausgeführt wird, können Sie (als `root`) mit folgendem Kommando herausfinden:

```
# ps aux |grep httpd
```

Bei einer Standardinstallation mit dem MPM `prefork` sollte das Ergebnis etwa so aussehen wie in Abbildung 5.1, falls Apache läuft. Andernfalls erhalten Sie

höchstens eine Zeile, die besagt, dass `grep httpd` ausgeführt wird – eben der Suchfilter des Befehls, den Sie gerade eingegeben haben.



```
root@redlinux:~/httpd_inst/httpd-2.0.48
Datei Bearbeiten Ansicht Terminal Gehe Hilfe
[root@redlinux httpd-2.0.48]# ps aux |grep httpd
root      23100  0.6  1.0 4596 2080 ?        S    15:50   0:00 httpd
nobody    23102  0.0  1.1 4672 2172 ?        S    15:50   0:00 httpd
nobody    23103  0.0  1.1 4672 2172 ?        S    15:50   0:00 httpd
nobody    23104  0.0  1.1 4672 2172 ?        S    15:50   0:00 httpd
nobody    23105  0.1  1.1 4672 2172 ?        S    15:50   0:00 httpd
nobody    23106  0.0  1.1 4672 2172 ?        S    15:50   0:00 httpd
root      23254  0.0  0.3 3248   672 pts/0    S    15:51   0:00 grep httpd
[root@redlinux httpd-2.0.48]#
```

Abbildung 5.1 Anzeige der laufenden Apache-Prozesse auf einem Linux-Host

### Parameter des Programms `httpd`

Der Funktionskern des Webservers Apache 2 ist das ausführbare Programm (Binary Executable) `httpd`. (Falls Sie den Server nach der Anleitung aus dem vorigen Kapitel über die Option `--with-program-name` mit einem anderen Programmnamen kompiliert haben sollten, gilt dieser im Folgenden sinngemäß.) Je nach Installationslayout bzw. eingestelltem Verzeichnis befindet sich das Programm an unterschiedlichen Stellen in Ihrem Dateisystem – genauer gesagt, im `SBINDIR` Ihrer Installation. Haben Sie beispielsweise das Standardlayout Apache verwendet, dann finden Sie die Datei im Verzeichnis `/usr/local/apache2/bin`, beim GNU-Layout dagegen unter `/usr/local/sbin`.

Um einen sauber installierten Apache-Webserver zu starten, sollte in der Regel der folgende Befehl ausreichen:

```
$ httpd
```

(Ob Sie dem Befehl zusätzlich eine Pfadangabe voranstellen müssen, hat natürlich damit zu tun, ob das `SBINDIR` sich in Ihrem `path` befindet oder nicht.)

Wenn der Server auf diese einfache Weise nicht startet oder wenn Sie besondere Einstellungen für seinen Start vornehmen möchten, können Sie aus den folgenden Kommandozeilenparametern auswählen:

#### ► `-k` Option

Der Parameter `-k` mit einer der Optionen `shutdown`, `restart`, `graceful` oder `graceful-stop` dient dem Beenden bzw. dem Neustart des Servers.

Diese Befehle werden weiter unten in der Beschreibung des Skripts `apachectl` näher erläutert.

► **-D Name**

Dieser Parameter eröffnet Ihnen ein sehr praktisches Verfahren, in der Konfigurationsdatei `httpd.conf` optionale Direktiven unterzubringen und nach Belieben auszuführen oder auch nicht: Sie können Direktiven in einen Block nach dem Schema `<IfDefine Name>...</IfDefine>` einschließen. Sie werden nur dann ausgeführt, wenn der entsprechende `Name` beim Start von Apache mit diesem Parameter definiert wird. Die Verwendung von `<IfDefine>` in der Konfigurationsdatei wird im nächsten Kapitel erläutert.

► **-d Verzeichnis**

Mithilfe dieses Parameters lässt sich eine alternative `ServerRoot` angeben – es handelt sich um das Verzeichnis, in dem die Inhalts- und Konfigurationsdateien des Servers liegen.

► **-f Datei**

Um Ihren Server mit einer anderen Konfigurationsdatei (`ServerConfigFile`) auszuführen als mit `httpd.conf` in Ihrem `SYSCONFDIR`, können Sie diese alternative Datei mithilfe der vorliegenden Option angeben.

► **-C "Direktive"**

Diese Option ermöglicht die Angabe einer zusätzlichen Konfigurationsdirektive, die vor dem Einlesen der Konfigurationsdatei verarbeitet werden soll.

► **-c "Direktive"**

Dieser Parameter ähnelt dem vorigen, mit einem Unterschied: Eine Direktive, die Sie hier angeben, wird erst nach der Verarbeitung der Konfigurationsdatei ausgeführt – dies ermöglicht das gezielte Überschreiben einer Direktive der Server-Konfiguration.

► **-e Level**

Der Parameter gibt die Dringlichkeit beim Start auftretender Fehler an, ab der diese Fehler angezeigt werden sollen. Die möglichen Werte dieser Option sind die bekannten `syslog`-Fehlerprioritäten wie `notice`, `err` oder `alert`. Sie werden weiter unten im Zusammenhang mit der Konfigurationsdirektive `LogLevel` erläutert.

► **-E Datei**

Bestimmt, dass Startfehler nicht auf der Konsole (bzw. `stderr`) angezeigt, sondern in die angegebene Datei geschrieben werden sollen.

Die Verwendung der folgenden Optionen startet den Server nicht, sondern dient der Ausgabe verschiedener Informationen:

► **-v**

Dieser Parameter zeigt nur Versionsinformationen an.

► **-V**

Zeigt ausführliche Versionsinformationen an sowie die Einstellungen, mit denen Apache 2 kompiliert wurde. Die Ausgabe sieht unter RedHat Linux mit dem Layout GNU und der als DSOs kompilierten Modulliste `most` beispielsweise so aus:

```
Server version: Apache/2.0.55
Server built:   Oct 30 2005 15:10:15
Server's Module Magic Number: 20020903:11
Architecture:  32-bit
Server compiled with....
-D APACHE_MPM_DIR="server/mpm/prefork"
-D APR_HAS_SENDFILE
-D APR_HAS_MMAP
-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
-D APR_USE_SYSVSEM_SERIALIZE
-D APR_USE_PTHREAD_SERIALIZE
-D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
-D APR_HAS_OTHER_CHILD
-D AP_HAVE_RELIABLE_PIPED_LOGS
-D HTTPD_ROOT="/usr/local"
-D SUEXEC_BIN="/usr/local/bin/suexec"
-D DEFAULT_PIDLOG="var/apache2/run/httpd.pid"
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
-D DEFAULT_LOCKFILE="var/apache2/run/accept.lock"
-D DEFAULT_ERRORLOG="logs/error_log"
-D AP_TYPES_CONFIG_FILE="etc/apache2/mime.types"
-D SERVER_CONFIG_FILE="etc/apache2/httpd.conf"
```

Die ersten beiden Zeilen bilden übrigens die Ausgabe der Option `-v`.

Noch ein Wort zur »Module Magic Number«: Dieser Wert entspricht der Build-Nummer der vorliegenden Apache-Release. Er ist in der Quellcode-Datei `include/ap_mmn.h` durch die symbolischen `MODULE_MAGIC_NUMBER_MAJOR` und `MODULE_MAGIC_NUMBER_MINOR` definiert. Die aktuellen Werte für Apache 2.0.55 sind 20020903 und 11; Apache 2.1.9 verwendet 20051006:0. DSO-Module mit derselben Magic Number (relevant ist die

Zahl vor dem Doppelpunkt) können auf derselben Plattform im Binärformat installiert werden und brauchen nicht neu kompiliert zu werden.

- ▶ **-h**  
Zeigt eine Liste sämtlicher Kommandozeilenoptionen an (wie diese hier).
- ▶ **-l**  
Der Parameter zeigt eine Liste aller einkompilierten Module an.
- ▶ **-L**  
Mit dieser Option erhalten Sie eine Liste sämtlicher verfügbarer Konfigurationsdirektiven – welche das sind, hängt von den vorhandenen Modulen ab.
- ▶ **-t -D DUMP\_VHOSTS**  
Die Option `-t -D Name` soll allgemein bestimmte Einstellungen anzeigen, die sich aus der bereits verarbeiteten Konfiguration ergeben. Zurzeit ist nur die spezielle Variante `-t -D DUMP_VHOSTS` verfügbar, die die entsprechenden Ergebnisse für virtuelle Hosts ausgibt.
- ▶ **-S**  
Eine Kurzfassung für die Option `-t -D DUMP_VHOSTS`.
- ▶ **-t -D DUMP\_MODULES**  
Zeigt eine Liste der tatsächlich aktiven Module an, sowohl der fest einkompilierten als auch der DSO-Module, die durch eine `LoadModule`-Direktive (siehe nächstes Kapitel) geladen wurden. Diese Option steht erst ab Apache 2.1-beta zur Verfügung.
- ▶ **-M**  
Kurzschreibweise für `-t -D DUMP_MODULES`.
- ▶ **-t**  
Diese Option überprüft die gewählte Konfigurationsdatei (und eventuelle zusätzliche Direktiven) auf syntaktische Richtigkeit.

### Apache manuell beenden und neu starten

Da in Version 2 des Apache-Webservers das weiter unten behandelte Skript `apachectl` zur Verfügung steht, ist es eigentlich nicht mehr nötig, die hier beschriebenen manuellen Verfahren anzuwenden. Allerdings verdeutlichen sie die Technik, die diesem Shell-Skript zugrunde liegt. Deshalb ist ihre Kenntnis beispielsweise für eigene, angepasste Steuerskripte erforderlich.

Wie bei UNIX-Daemons üblich, reagiert der Steuerprozess – der ursprüngliche Parent-Prozess, der die Child-Prozesse zur Verarbeitung von Client-Anfragen startet – auf einige Standardsignale, die das Beenden bzw. den Neustart bewirken. Solche Signale werden durch das Hilfsprogramm `kill` versandt, das den

gleichnamigen Systemaufruf ausführt. Die allgemeine Syntax dieses Befehls lautet bekanntermaßen so:

```
# kill [-SIGNAL] PID
```

Das einzige Problem besteht darin, den Steuerprozess ausfindig zu machen. Wie Sie weiter oben in Abbildung 5.1 gesehen haben, gibt es bei einem `prefork`-Apache mehrere Prozesse mit dem Namen `httpd`. Welcher von ihnen der Steuerprozess ist, lässt sich auf zweierlei Arten ermitteln:

- ▶ Es ist derjenige Prozess, der unter der User-ID `root` ausgeführt wird. Die Child-Prozesse für die Client-Bedienung laufen unter einer anderen UID, meistens `nobody` oder `daemon` (wird durch die Direktive `User` festgelegt; siehe nächstes Kapitel).
- ▶ Die zweite Methode ist die sicherste: Apache legt beim Start eine PID-Datei an. Diese enthält die Prozess-ID des Steuerprozesses. Diese Datei wird unter anderem dazu verwendet, um bei einem Startversuch festzustellen, ob der Server bereits läuft oder ob er womöglich unsauber beendet wurde – Letzteres ist der Fall, wenn die PID-Datei zwar noch existiert, aber kein Prozess unter der Nummer läuft, die darin steht.

Wo sich die PID-Datei (`httpd.pid`) befindet, hängt wieder einmal vom verwendeten Verzeichnislayout ab. Sie liegt im `RUNTIMEDIR`: Beim Apache-Layout ist dies `/usr/local/apache2/logs`, beim GNU-Layout `/usr/local/var/apache2/run`.

Bequemerweise können Sie zur Angabe der PID im `kill`-Befehl unmittelbar den Wert aus `httpd.pid` verwenden. Setzen Sie dazu einfach den Ausgabebefehl `cat /Pfad/von/httpd.pid in `Backticks``; unter Verwendung des GNU-Layouts z.B. so:

```
`cat /usr/local/var/apache2/run/httpd.pid`
```

Nun brauchen Sie nur noch zu wissen, welche Signale verwendet werden, um die unterschiedlichen Verhaltensweisen des Servers zu bewirken:

- ▶ `TERM` beendet Apache 2 vollständig. Da `TERM` das Standardsignal für `kill` ist, brauchen Sie gar kein Signal anzugeben, um Apache zu beenden.

Hier ein Beispiel mit der nach dem ersten Verfahren ermittelten PID aus Abbildung 5.1:

```
# kill 23100
```

Natürlich können Sie das Signal `TERM` auch explizit angeben, wenn Sie möchten:

```
# kill -TERM 23100
```

- ▶ HUP bewirkt einen normalen Neustart des Webservers: Alle Child-Prozesse werden sofort beendet (wobei laufende Übertragungen natürlich abgebrochen werden); anschließend wird der Parent-Prozess neu gestartet und erzeugt wieder die vorgesehene Anfangszahl von Child-Prozessen.

Das folgende Beispiel funktioniert, wenn Ihr Apache 2 das GNU-Layout verwendet:

```
# kill -HUP `cat /usr/local/var/apache2/run/httpd.pid`
```

- ▶ USR1 sorgt für einen unterbrechungsfreien Neustart (graceful restart) des Servers: Child-Prozesse, die sich gerade um Verbindungen kümmern, werden erst beendet, wenn der aktuelle Vorgang abgeschlossen ist; untätige Child-Prozesse werden sofort beendet. Der Parent-Prozess wird schließlich neu gestartet, nachdem alle Child-Prozesse sauber abgeschlossen wurden.

Hier sehen Sie ein Beispiel für das Verzeichnislayout Apache:

```
# kill -USR1 `/usr/local/apache2/logs/httpd.pid`
```

- ▶ WINCH schließlich führt zum sauberen Beenden (graceful shutdown) von Apache: Mit dem Beenden des Hauptprozesses wird gewartet, bis alle Worker-Prozesse oder -Threads mit der Verarbeitung ihrer Verbindungen fertig sind. Optional lässt sich mithilfe der neuen Direktive `GracefulShutdownTimeout` (siehe Kapitel 6) eine maximale Wartezeit festlegen, nach der Apache auf jeden Fall beendet wird. Beachten Sie, dass dieses Feature erst seit Version 2.1 zur Verfügung steht. Hier ein Beispiel; es gilt wieder für das Standardlayout Apache:

```
# kill -WINCH `/usr/local/apache2/logs/httpd.pid`
```

### Parameter des Skripts `apachectl`

Eine bequemere Steuerung des Apache-Webservers bietet das Shell-Skript `apachectl`, das bei der Kompilierung oder Binärinstallation von Version 2 mitgeliefert wird. Es enthält zahlreiche Optionen zum Starten, Beenden, Neustarten und Überwachen des Servers.

Das Skript befindet sich im `BINDIR` Ihres Verzeichnislayouts. Dies ist beim Apache-Layout `/usr/local/apache2/bin`, beim GNU-Layout `/usr/local/bin`. Die allgemeine Syntax lautet folgendermaßen:

```
# apachectl Befehl
```

`apachectl` bietet folgende Befehle an:

- ▶ **start**  
  - k start

Der Befehl `start` bzw. `-k start` startet den Webserver mit Standardoptionen. Dies ist der Standardbefehl, wenn Sie `apachectl` ohne weitere Optionen aufrufen.
- ▶ **stop**  
  - k stop

Wenn Sie einen dieser beiden Befehle eingeben, wird der Server beendet.
- ▶ **restart**  
  - k restart

Diese Befehle senden nach dem oben beschriebenen Schema ein HUP-Signal an den HTTP-Server, um ihn ohne Rücksicht auf bestehende Verbindungen sofort neu zu starten.
- ▶ **graceful**  
  - k graceful

Diese beiden Befehle sorgen für einen unterbrechungsfreien Neustart (`graceful restart`) des Servers, der keine offenen Verbindungen fallen lässt.
- ▶ **graceful-stop**  
  - k graceful-stop

Mithilfe eines dieser Befehle können Sie den Server sauber beenden (`graceful shutdown`); er wird erst beendet, nachdem alle Worker-Prozesse oder –Threads ihre aktuellen Aufgaben abgeschlossen haben.
- ▶ **fullstatus**  

Wenn in Ihrem Webserver das Modul `mod_status` aktiv ist und der Text-Browser Lynx auf Ihrem System zur Verfügung steht, gibt dieser Befehl eine Statusmeldung des laufenden Servers inklusive aller zurzeit bedienten Client-Verbindungen aus.
- ▶ **status**  

Auch dieser Befehl funktioniert nur, wenn `mod_status` eingeschaltet und Lynx verfügbar ist. Er gibt eine einfache Statusmeldung ohne Verbindungsinformationen aus.
- ▶ **configtest**  

Dieser Befehl entspricht der Option `-t` des Programms `httpd`: Er testet die syntaktische Richtigkeit der aktuellen Konfigurationsdatei.

Neben den hier beschriebenen Befehlen können Sie auch für `apachectl` alle oben beschriebenen `httpd`-Optionen verwenden; diese werden entsprechend weitergereicht.

## Den Start automatisieren

Jedes moderne UNIX-System verfügt über eine Möglichkeit, beim Booten beliebige Programme – insbesondere Daemons wie den Apache-Webserver – zu starten. Ein kleines Problem besteht nur darin, dass dieses Verfahren in den verschiedenen Systemvarianten unterschiedlich realisiert wurde. Historisch betrachtet lässt sich der Unterschied auf die beiden UNIX-Grundströmungen System V und BSD zurückführen, inzwischen zieht er sich aber – und dann auch noch mit einigen Variationen bezüglich der Verzeichnisauswahl – quer durch die Systeme und Distributionen (zumal es immer schwieriger wird, zu unterscheiden, welche aktuellen Systeme von System V abstammen und welche von der BSD).

Hier wird zunächst jedes der beiden grundsätzlichen Verfahren kurz vorgestellt; anschließend erhalten Sie Informationen darüber, wie sich der Server unter einem entsprechend beschaffenen System automatisch starten lässt.

### ► System V Init

Diese Boot-Methode wird von immer mehr Betriebssystemen der UNIX-Familie verwendet, unter anderem auch von Linux. Systeme, die System V Init einsetzen, arbeiten mit unterschiedlichen **Runlevels**. Ein Runlevel ist ein Systemzustand, in dem jeweils nur bestimmte Prozesse laufen. Beim Wechsel des Runlevels über den Befehl `init LEVELNR.` werden bestimmte Skripte aufgerufen, die manche Programme starten und andere beenden. Einige Runlevel haben eine spezielle Bedeutung:

- 0: Heruntergefahrener Zustand
- S: (manchmal auch 1): Single-User-Modus (für Wartungsarbeiten)
- 1: (bei vielen Systemen): Multi-User-Modus ohne Netzwerk
- 2: Multi-User-Modus mit Netzwerk; nur Konsole
- 3: Multi-User-Modus mit Netzwerk und GUI (klassisch)
- 5: Multi-User-Modus mit Netzwerk und GUI (Linux)
- 6: Systemneustart (Reboot)

Betriebssysteme dieser Bauart besitzen für jedes Runlevel ein spezielles Init-Verzeichnis. Diese Verzeichnisse heißen `/etc/rcLEVELNR.d`, also etwa `/etc/rc1.d` für Runlevel 1 oder `/etc/rc5.d` für Runlevel 5. Die Shell-Skripte in diesen Verzeichnissen werden bei Erreichen des entsprechenden Levels automatisch ausgeführt, und zwar in alphabetischer Reihenfolge. Deshalb verwendet die übliche Konvention Namen, die mit **K** beginnen, für Kill-Skripte (die einen Prozess beenden) und solche mit **S** für Start-Skripte. Darüber hinaus bauen viele Daemons aufeinander auf. Deshalb wird hinter dem Anfangsbuchstaben eine zweistellige Zahl verwendet, die für eine bestimmte Reihenfolge sorgt.

In aller Regel sind die Einträge in diesen Verzeichnissen lediglich Symlinks auf Skripte, die sich eigentlich in einem anderen Verzeichnis befinden; meist in `/etc/init.d` oder `/sbin/init.d`. Für den Start und das Beenden des jeweiligen Prozesses wird normalerweise dasselbe Skript verwendet: Ein `S`-Symlink ruft es automatisch mit der Kommandozeilenoption `start` auf, ein `K` mit `stop`. Wie Sie weiter oben bereits erfahren haben, erfüllt das Shell-Skript `apachectl` demzufolge die Voraussetzungen für diese Aufgabe.

Aus diesem Grund brauchen Sie lediglich für Ihr Standard-Runlevel (3 oder 5) einen `S`-Symlink auf dieses Skript zu erzeugen. Für die Runlevel 0 und 6 (Herunterfahren bzw. Neustart) können Sie entsprechend einen `K`-Link anlegen. Da von Apache in der Regel keine anderen Dienste abhängen, können Sie ihn recht spät starten (wählen Sie einen Symlink-Namen wie `S95apache`) und ziemlich früh beenden (`K15apache` dürfte in Ordnung gehen).

Begeben Sie sich also in das jeweilige Runlevel-Init-Verzeichnis und erstellen Sie die nötigen symbolischen Links. Angenommen, Sie haben Apache mit dem GNU-Layout installiert und verwenden ein Linux-System mit dem Standard-Runlevel 5. Dann müssen Sie den folgenden Befehl für den Startskript-Link eingeben:

```
# ln -s /usr/local/bin/apachectl /etc/rc5.d/S95apache
```

Als Nächstes werden die Stopp-Links für die Runlevel 0 und 6 erzeugt. Wenn Sie das Layout Apache verwenden, sehen die beiden Befehle so aus:

```
# ln -s /usr/local/apache2/bin/apachectl /etc/rc0.d/K15apache
# ln -s /usr/local/apache2/bin/apachectl /etc/rc6.d/K15apache
```

In vielen Linux-Distributionen, beispielsweise SUSE und Fedora Core bzw. Red Hat, können Sie das Kommando `chkconfig` verwenden, um Apache 2 für den automatischen Start in den Runlevels 3 und 5 einzurichten. Erstellen Sie dazu als Erstes im Verzeichnis `/etc/init.d` einen symbolischen Link auf `apachectl`. Beispiel:

```
# ln -s /usr/local/apache2/bin/apachectl /etc/init.d/apache2
```

Nun lässt sich der Symlink mithilfe von `chkconfig -a` aktivieren:

```
# chkconfig -a apache2
apache2 0:off 1:off 2:off
        3:on 4:off 5:on 6:off
```

#### ► BSD-Startskript

BSD-basierte UNIX-Systeme verwenden im Gegensatz zu der oben beschriebenen System-V-Methode einige zentrale Startskripte. Sie befinden sich in

Verzeichnissen wie `/etc` oder `/etc/rc.d` und heißen `rc.boot`, `rc.local` usw. Interessant ist in diesem Zusammenhang das Skript `rc.local`, das Sie nach Belieben um weitere Startbefehle erweitern können.

Unter einem solchen Betriebssystem brauchen Sie `rc.local` also nur mit einem Texteditor zu öffnen und können dann den Aufruf von `apachectl` mit dem Parameter `start` hinzufügen. In diesem Skript wird normalerweise mit einer Fallentscheidung nach dem Schema `if [-x PFAD]` überprüft, ob das aufzurufende Programm oder Skript überhaupt existiert. An diese Konvention sollten Sie sich halten. Falls Sie also das GNU-Layout verwenden, können Sie folgende Zeilen an `rc.local` anfügen:

```
# httpd starten
if [ -x /usr/local/bin/apachectl ]; then
    echo "Starting Apache httpd..."
    /usr/local/bin/apachectl start
fi
```

Es gibt hier keine Lösung, die »besser« ist – beide funktionieren in der Praxis, und bei jedem System ist es eine von beiden.

Noch leichter haben Sie es natürlich, wenn Sie den Webserver über ein Paket Ihres Systemdistributors installiert haben: Diese Installer-Befehle kümmern sich normalerweise selbst darum, Apache für den automatischen Start zu konfigurieren. Diese Einstellung können Sie bei manchen Systemen auch auf der grafischen Benutzeroberfläche vornehmen; zu diesem Zweck bieten einige Distributionen spezielle Verwaltungsprogramme an. Hier nur ein paar Beispiele:

#### ► SUSE Linux

Ein gutes Argument für den Einsatz von SuSE gegenüber anderen Linux-Distributionen ist schon seit langem das komfortable Installations- und Konfigurationsprogramm `yast` (Yet Another Setup Tool). Die aktuelle Version Open SUSE 10.0 bietet eine Rubrik für den automatischen Start und die komfortable grafische Konfiguration von Apache. Starten Sie dazu über ein Terminalfenster oder aus dem KDE-Menü das Programm `yast`. Hier finden Sie in der Rubrik **Netzwerkdienste** das entsprechende Icon **HTTP-Server**.

Dieses Tool dient darüber hinaus dem Aktivieren oder Deaktivieren von Apache-Modulen. Dies muss bei der eingebauten Apache-Version von SUSE an dieser Stelle geschehen; entsprechende Änderungen in den Konfigurationsdateien werden durch `SUSEConfig` nachträglich wieder überschrieben. Außerdem können Sie an dieser Stelle auch Konfigurationsdirektiven einstellen, wie sie ab dem nächsten Kapitel ausführlich erläutert werden.

Bereits in früheren Versionen von SuSE Linux stand eine andere Möglichkeit zur Verfügung, die noch immer eingesetzt werden kann – vornehmlich, wenn Sie den Server selbst kompiliert haben, denn dann steht die `yast`-Methode nicht zur Verfügung: Über den **Runlevel-Editor** können Sie einstellen, welche Programme bzw. Daemons in welchem Runlevel ausgeführt werden sollen. Starten Sie dazu wiederum `yast` und wählen Sie **Runlevel-Editor** in der Kategorie **System**. Sie sollten Apache 2 in den Runlevels 3 und 5 aktivieren (siehe Abbildung 5.2).

► **RedHat Linux**

RedHat enthält ebenfalls ein eigenständiges Konfigurationsprogramm für den HTTP-Server. Sie erreichen es über das RedHat-Menü in der grafischen Oberfläche unter **Servereinstellungen · HTTP-Server**. Hier können Sie zahlreiche Einstellungen für Apache vornehmen, die sich normalerweise nur manuell über die Konfigurationsdatei erreichen lassen.

► **Mac OS X**

Unter Mac OS X wird Apache normalerweise automatisch mit dem System installiert. Über **Systemeinstellungen · Netzwerk** können Sie ihn für den automatischen Start aktivieren bzw. deaktivieren.

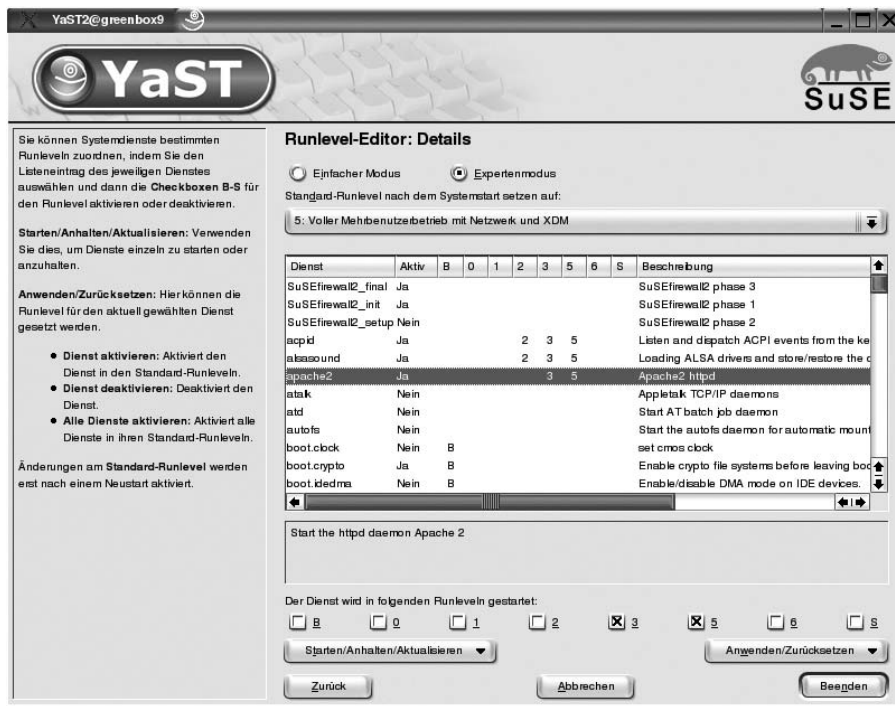


Abbildung 5.2 Aktivieren des automatischen Apache-Starts im Runlevel-Editor von SUSE Linux

### 5.1.2 Apache unter Windows steuern

Unter Windows funktioniert die Steuerung des Webservers aufgrund der Plattformunterschiede ein wenig anders als auf UNIX-Systemen. Im Prinzip lassen sich drei verschiedene Konfigurationsarten unterscheiden, deren Funktionalität sich allerdings zum Teil überschneidet:

- ▶ Sie können das ausführbare Programm `apache.exe` mit diversen Optionen aufrufen, die zum Teil dem Programm `httpd` und dem Skript `apachectl` unter UNIX entsprechen.
- ▶ Sie können Apache 2 als Dienst installieren. Ein Dienst ist die Windows-Entsprechung eines Daemons. Wenn Apache als Dienst ausgeführt wird, können Sie ihn über das Applet **Dienste** der Systemsteuerung bzw. der Microsoft Management Console steuern.
- ▶ Im Verzeichnis `bin` des Servers finden Sie ein kleines Steuerprogramm namens `ApacheMonitor.exe`. Dieses Programm installiert ein kleines Steuer-Icon in den SysTray und stellt diverse Optionen zur Verfügung.

#### Optionen des Programms Apache.exe

Unter Windows heißt das ausführbare Webserver-Programm `Apache.exe`. Es befindet sich im Verzeichnis `bin` im Ihrer `ServerRoot` (z.B. in `C:\Programme\Apache Group\Apache2\bin`). Im Folgenden sind alle Optionen dieses Programms aufgeführt. Sofern sie den weiter oben besprochenen Parametern von `httpd` bzw. `apachectl` unter UNIX entsprechen, fällt die Beschreibung recht kurz aus:

- ▶ **-D Name**  
Definiert einen Namen für die Konfigurationsdirektive `<IfDefine Name>`.
- ▶ **-d Verzeichnis**  
Gibt eine alternative `ServerRoot` an.
- ▶ **-f Datei**  
Ermöglicht die Angabe einer alternativen Konfigurationsdatei.
- ▶ **-C "Direktive"**  
Führt vor dem Einlesen der Konfigurationsdatei die angegebene Konfigurationsdirektive aus.
- ▶ **-c "Direktive"**  
Führt die angegebene Direktive nach dem Verarbeiten der Konfigurationsdatei aus, ermöglicht also das nachträgliche Überschreiben vorhandener Direktiven.

- ▶ **-k start**  
Diese Option startet Apache. Wenn er als Dienst installiert ist, wird dieser gestartet; andernfalls startet der Server als Konsolenprogramm.
- ▶ **-k runservice**  
Mit diesem Befehl wird explizit ein bereits installierter Apache-Dienst gestartet.
- ▶ **-k restart**  
Diese Option führt einen unterbrechungsfreien Neustart des Servers durch. Wenn er als Konsolenprogramm läuft, müssen Sie ein weiteres Eingabeaufforderungsfenster öffnen, um den Befehl einzugeben.
- ▶ **-k stop**  
**-k shutdown**  
Jeder dieser beiden Befehle beendet Apache. Läuft er als Konsolenprogramm, dann können Sie den Befehl von einem anderen Fenster aus eingeben.
- ▶ **-k install**  
Installiert Apache als Dienst.
- ▶ **-k config**  
Diese Option kann verwendet werden, um zusammen mit anderen Befehlen die Konfiguration des Apache-Dienstes zu ändern.
- ▶ **-k uninstall**  
Deinstalliert den Apache-Dienst.
- ▶ **-n Name**  
Mithilfe dieser Option können Sie einen alternativen Namen angeben, unter dem der Apache-Dienst installiert werden soll (der Standardname ist `Apache2`). Hat er bereits einen anderen Namen, dann dient dieselbe Option dazu, den Dienst später für die Deinstallation oder für Konfigurationsänderungen anzusprechen.
- ▶ **-w**  
Wenn diese Option angegeben wird, bleibt das Konsolenfenster bei einem Fehler geöffnet. Dies ist vor allem dann nützlich, wenn Sie Apache aus einer Batchdatei heraus starten, die per Doppelklick oder automatisch ausgeführt wird.
- ▶ **-e level**  
Dringlichkeitsstufe, ab der Fehler beim Start des Servers angezeigt werden sollen.
- ▶ **-E Datei**  
Schreibt Startfehlermeldungen in die angegebene Datei.
- ▶ **-v**  
Ausgabe von Versionsinformationen

- ▶ **-V**  
Ausgabe der Versionsinformationen und der Einstellungen, mit denen der Server kompiliert wurde
- ▶ **-h**  
Ausgabe einer Liste aller Kommandozeilenoptionen
- ▶ **-l**  
Ausgabe einer Liste der einkompilierten Module
- ▶ **-L**  
Auflisten der verfügbaren Konfigurationsdirektiven
- ▶ **-t -D DUMP\_VHOSTS**  
Ausgabe der verarbeiteten Einstellungen für virtuelle Hosts
- ▶ **-S**  
Kurzfassung von `-t -D DUMP_VHOSTS`
- ▶ **-t**  
Syntax der Konfigurationsdatei überprüfen

### Apache als Dienst betreiben

Bereits seit der Vorgängerversion 1.3 lässt sich Apache unter NT basierten Windows-Betriebssystemen als Dienst installieren. Der Vorteil ist, dass der Server in dieser Konstellation unabhängig von einem angemeldeten Benutzer ausgeführt wird. Auch die Performance ist bei einem Dienst besser, als wenn Apache 2 als Konsolenanwendung ausgeführt wird.

Wenn Sie das im vorigen Kapitel beschriebene Windows-MSI-Paket installieren, können Sie die automatische Einrichtung von Apache als Dienst wählen; sie ist sogar standardmäßig vorgegeben. Haben Sie den Server dagegen selbst kompiliert oder die Installation als Dienst abgelehnt, müssen Sie den folgenden Befehl ausführen, um den Dienst nachträglich zu installieren:

```
> apache -k install
```

Der Dienst wird dadurch ein für allemal in die Liste der Systemdienste aufgenommen. Der Standardname des Dienstes ist `Apache2`. Um einen anderen Namen festzulegen, können Sie den Befehl mit der zusätzlichen Option `-n Name` verwenden, beispielsweise so:

```
> apache -k install -n Winnetou
```

Wenn Sie ihn später wieder entfernen möchten, können Sie dies mit diesem Befehl erledigen:

```
> apache -k uninstall
```

Falls Sie einen alternativen Namen festgelegt haben, müssen Sie diesen auch bei der Deinstallation angeben:

```
> apache -k uninstall -n Winnetou
```

Einen installierten Apache-Dienst können Sie auf der Kommandozeile auch mithilfe der Befehle `net start` und `net stop` steuern. Dazu müssen Sie in jedem Fall den Dienstenamen angeben, nicht den Namen des ausführbaren Programms. Beispiele:

```
> net start Apache2
```

```
> net stop Winnetou
```

Zur Verwaltung von Diensten wie dem Apache-Dienst bietet Windows das Verwaltungs-Applet **Dienste**. In Windows XP finden Sie es unter **Start · Verwaltung · Dienste**, unter Windows 2000 in der Systemsteuerung unter **Verwaltung** und bei Windows NT 4.0 unter **Start · Einstellungen · Systemsteuerung · Dienste**. Abbildung 5.3 zeigt das Applet unter Windows XP.

Änderungen an der Konfiguration des Dienstes können Sie über die Schaltfläche **Eigenschaften** vornehmen; es handelt sich um die Schaltfläche mit dem »Gepäckanhänger«, den Sie in der Symbolleiste in Abbildung 5.3 sehen können. Dieselbe Symbolleiste enthält im Übrigen Schaltflächen zum schnellen Starten, Beenden und Neustarten des gerade ausgewählten Dienstes.

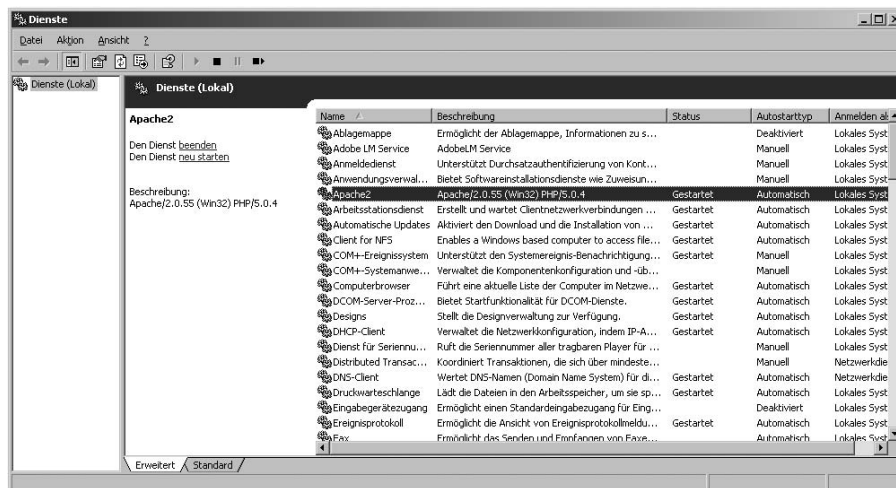


Abbildung 5.3 Das Verwaltungs-Applet »Dienste« mit ausgewähltem Apache-Dienst unter Windows XP

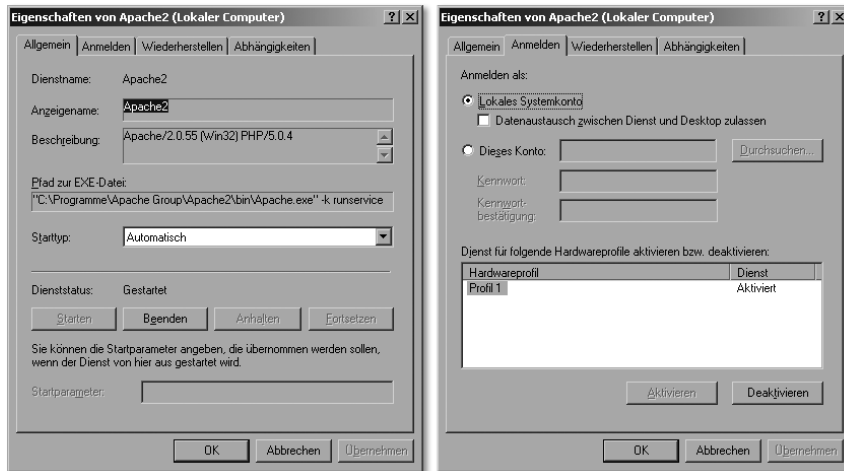


Abbildung 5.4 Die beiden ersten (und wichtigsten) Registerkarten des Eigenschaften-Dialogs für den Apache-Dienst unter Windows XP

In Abbildung 5.4 werden nebeneinander die Inhalte der beiden ersten Registerkarten des Eigenschaften-Dialogs für den Apache-Dienst gezeigt: **Allgemein** und **Anmelden**. Die vier Registerkarten des Dialogs bieten die folgenden Einstellungsmöglichkeiten:

► **Allgemein**

Hier werden die Grundeinstellungen für den Dienst vorgenommen, insbesondere ermöglicht der **Starttyp** die Einstellung, ob er automatisch gestartet werden soll. Im Einzelnen finden Sie hier folgende Felder und Schaltflächen:

- **Dienstname.** Offizielle Bezeichnung des Dienstes; nicht änderbar
- **Anzeigename.** Änderbarer Name des Dienstes, wie er in der Liste angezeigt wird
- **Beschreibung.** Nähere Information über den Dienst
- **Pfad zur EXE-Datei.** Das ausführbare Programm, das dieser Dienst ausführt, mitsamt Kommandozeilenparametern
- **Starttyp.** Legt fest, wie dieser Dienst gestartet werden soll: **Automatisch** startet ihn automatisch, **Manuell** nur auf ausdrückliche Anforderung und **Deaktiviert** gar nicht.
- **Dienststatus.** Zeigt an, ob der Dienst zurzeit läuft, beendet wurde oder deaktiviert ist.
- **Starten.** Hier können Sie den Apache-Dienst starten, wenn Sie die Startart auf **Manuell** gesetzt oder ihn zuvor beendet haben.
- **Beenden.** Diese Schaltfläche beendet den Apache-Dienst.

- ▶ **Anhalten.** Einige Windows-Server-Dienste – meist von Microsoft selbst – können über diese Schaltfläche in eine Art »Administrationsmodus« versetzt werden: Der Dienst ist dann nur noch für Benutzer mit Administratorrechten erreichbar. Bei Apache wurde dieses Feature leider noch nicht eingebaut; hier müssen Sie sich mit einer Umkonfiguration der Authentifizierung (siehe Kapitel 9, *Authentifizierung*) behelfen. Daher ist die Schaltfläche beim Apache-Dienst deaktiviert.
- ▶ **Fortsetzen.** Das Gegenstück zur Schaltfläche **Anhalten**: Ein Dienst soll aus dem eingeschränkten Wartungsmodus wieder in den Normalbetrieb zurückversetzt werden. Für Apache ebenfalls nicht verfügbar.

▶ **Anmelden**

Auf dieser Registerkarte wird festgelegt, unter welcher Benutzerkennung der Dienst `Apache2` ausgeführt wird. Die Standardeinstellung ist **Lokales Systemkonto**. Dies ist in den meisten Fällen in Ordnung, mit einer wichtigen Ausnahme:

Aus Sicherheitsgründen sollte das Benutzerkonto des Apache-Dienstes niemals Berechtigungen für Netzwerkanwendungen erhalten. Falls der »Benutzer« **Lokales Systemkonto** diese für eine andere Anwendung benötigt, sollten Sie Apache unbedingt unter einer anderen Benutzerkennung betreiben. Dazu müssen Sie einen neuen Benutzer einrichten; dieser sollte keine Administratorrechte haben, sondern ein normaler Benutzer sein. Er benötigt aber die zusätzlichen Rechte »Anmelden als Dienst« und »Als Teil des Betriebssystems handeln«.

Unter Windows XP legen Sie einen neuen Benutzer über das Applet **Benutzerkonten** in der Systemsteuerung an, unter Windows 2000 unter **Verwaltung**. Die erforderlichen Rechte können Sie über Gruppenrichtlinien oder über die lokalen Sicherheitseinstellungen in der Microsoft Management Console vergeben. Bei Windows NT 4.0 erledigen Sie beides mithilfe des Programms unter **Start · Programme · Verwaltung (Allgemein) · Benutzer-Manager**; die Rechte finden Sie dort unter **Richtlinien · Benutzerrechte**.

Wählen Sie nach dem Erstellen des neuen Benutzerkontos die Option **Dieser Benutzer**. Geben Sie den Namen dieses Benutzers und zweimal sein Passwort ein.

Unter der Einstellung des Benutzerkontos können Sie noch festlegen, in welchen **Hardwareprofilen** Apache aktiviert werden soll. Da Sie beim Booten ein bestimmtes Hardwareprofil auswählen können, bietet dieses Verfahren eine einfache Möglichkeit, Betriebssystemkonfigurationen mit und ohne aktivierten Apache-Webserver einzurichten. Eingerichtet werden Hardwareprofile übrigens in den **Systemeigenschaften (Systemsteuerung · System** oder rechte Maustaste auf das Symbol **Arbeitsplatz** und **Eigenschaften** auswählen).

► **Wiederherstellen**

Auf dieser Registerkarte können Sie detailliert festlegen, was geschehen soll, wenn Apache ausfällt, das heißt beim Systemstart nicht ausgeführt werden kann oder unerwartet beendet wird. Dies ist natürlich besonders dann wichtig, wenn Sie den betreffenden Rechner hauptsächlich als Webserver für ein Intranet oder das Internet verwenden.

Unter den Kategorien **Erster Fehlschlag**, **Zweiter Fehlschlag** und **Weitere Fehlschläge** können Sie je eine der folgenden Optionen auswählen:

- **Keinen Vorgang durchführen.** Es soll nichts Besonderes veranlasst werden. Wenn Sie Apache nur zu Test- oder Entwicklungszwecken installiert haben, ist dies die passende Auswahl.
- **Dienst neu starten.** Es soll versucht werden, Apache neu zu starten. Für den ersten Fehlschlag bietet sich diese Möglichkeit an.
- **Datei ausführen.** Wenn Sie diese Option auswählen, können Sie weiter unten eine Datei bestimmen, die ausgeführt werden soll. Dies kann ein Programm sein oder eine beliebige Datei, deren Dateityp Windows einer Anwendung zuordnen kann. Zusätzlich besteht die Möglichkeit, dieser Datei die Fehlschlagnummer als Kommandozeilenparameter in der Form `/fail=%1%` zu übergeben. Ein entsprechend präpariertes eigenes Programm kann diesen Parameter auswerten.

► **Computer neu starten**

Diese extreme Option ergibt eigentlich nur dann einen Sinn, wenn Sie einen Windows-Rechner in Ihrem Netzwerk ausschließlich als Webserver einsetzen. Unter **Informationen über Neustart** können Sie in diesem Fall eine Meldung eintragen; diese wird allen (Windows-)Benutzern im Netzwerk angezeigt, die zur Zeit des Neustarts mit diesem Host verbunden sind.

► **Abhängigkeiten**

Auf dieser Registerkarte werden die Dienste angezeigt, von deren Funktionen der aktuelle Dienst abhängt und umgekehrt. Von Apache hängen in der Regel keine anderen Dienste ab. Er selbst benötigt natürlich funktionierendes TCP/IP-Networking, was bei neueren Windows-Versionen auch als Voraussetzung angezeigt wird.

Beachten Sie, dass dieser Dialog unter Windows NT 4.0 nur aus einer einzigen Seite besteht. Auf dieser können Sie lediglich den Starttyp und das Benutzerkonto einstellen und natürlich den Dienst beenden und neu starten.

Windows 95, 98 und ME bieten von Hause aus gar keine Dienste an. Aber obwohl der Einsatz von Apache unter diesen Systemen ohnehin nicht zu empfehlen ist, wurde der als Nächstes beschriebene Apache-Monitor so geschrie-

ben, dass er auch mit einer Art »Dienstemulation« zusammenarbeitet, die auf diesen Systemen läuft.

### Der Apache-Monitor

Wenn Sie Apache über den MSI-Installer als Dienst installiert haben, wurde der Apache-Monitor automatisch eingerichtet und wird bei jedem Windows-Bootvorgang mitgestartet. Bei anderen Installationsarten können Sie ihn selbst starten oder ebenfalls für den automatischen Start konfigurieren.

Das Programm trägt den Namen `ApacheMonitor.exe` und befindet sich im Verzeichnis `bin` des Apache-Installationsordners. Wenn Sie es per Doppelklick oder über die Konsole starten, bleibt es nur für die aktuelle Systemsitzung aktiv. Um es für den automatischen Start einzurichten, gibt es zwei Möglichkeiten:

- ▶ Sie können eine Verknüpfung zu dem Programm im Ordner **Autostart** des Startmenüs anlegen. Dazu genügt es, das Icon des Programms mit der rechten Maustaste in **Start • Alle Programme • Autostart** zu ziehen und beim Loslassen die Option **Verknüpfung hier erstellen** aus dem Kontextmenü zu wählen. Diese Variante verwendet übrigens auch der MSI-Installer automatisch.
- ▶ Die Alternative besteht darin, einen Eintrag für den Start des Monitors in der Registry zu erstellen. Wählen Sie dazu **Start • Ausführen** und geben Sie `regedit` ein. Im linken Fensterbereich müssen Sie sich durch die Hierarchie zum Schlüssel **HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run** vorarbeiten. Wenn Sie den Ordner **Run** angeklickt haben, können Sie mit der rechten Maustaste in den rechten Fensterbereich klicken und **Neu • Zeichenfolge** aus dem Kontextmenü wählen. Geben Sie einen beliebigen Namen ein (in diesem Fall natürlich am besten **ApacheMonitor**). Doppelklicken Sie zu guter Letzt auf das Icon der neuen Zeichenfolge und geben Sie als Wert den Pfad des Programms `ApacheMonitor.exe` ein. In Abbildung 5.5 sehen Sie, wie es gemacht wird.

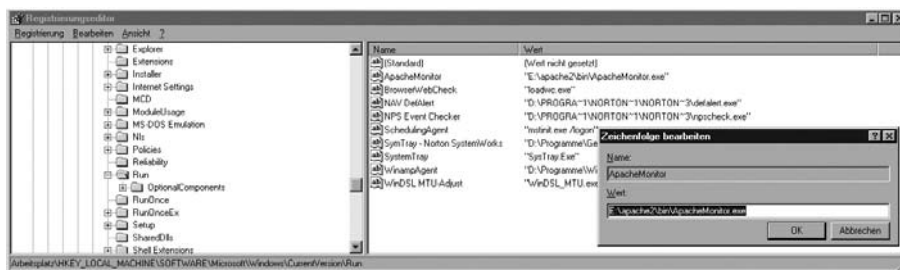


Abbildung 5.5 Den automatischen Start von `ApacheMonitor.exe` in der Windows-Registry einrichten

Wenn der Apache-Monitor ausgeführt wird, macht er sich als kleines Apache-Icon im SysTray bemerkbar – dies ist der Bereich links in der Taskleiste, neben der Uhrzeit. Wenn Sie das Icon mit der linken Maustaste anklicken, stehen Ihnen die Optionen **Start** (den Apache-Dienst starten), **Stop** (zum Beenden des Dienstes) und **Restart** (Neustart, z.B. nach einer Konfigurationsänderung) zur Verfügung. Ein Klick mit der rechten Maustaste ermöglicht dagegen das Öffnen des eigentlichen Monitor-Fensters, das in Abbildung 5.6 zu sehen ist.

Auch hier sind zunächst einmal wieder Schaltflächen zum Starten, Beenden und Neustarten zu erkennen. Darüber hinaus können Sie über **Connect** eine Verbindung zu einem anderen Windows-Rechner in Ihrem LAN herstellen, der Apache 2 ausführt, und dessen Apache-Dienst fernsteuern. Dazu benötigen Sie allerdings Administratorrechte, die auch für den entfernten Host gelten – entweder über ein entsprechendes Domänen-Benutzerkonto oder dadurch, dass Ihr aktueller Benutzername mit demselben Passwort und identischen Rechten auf dem anderen Computer existiert.

Die Schaltfläche **Services** schließlich öffnet unter Windows 2000 und XP das weiter oben besprochene Betriebssystem-Applet **Dienste**; unter Windows NT 4.0 funktioniert dies leider nicht.

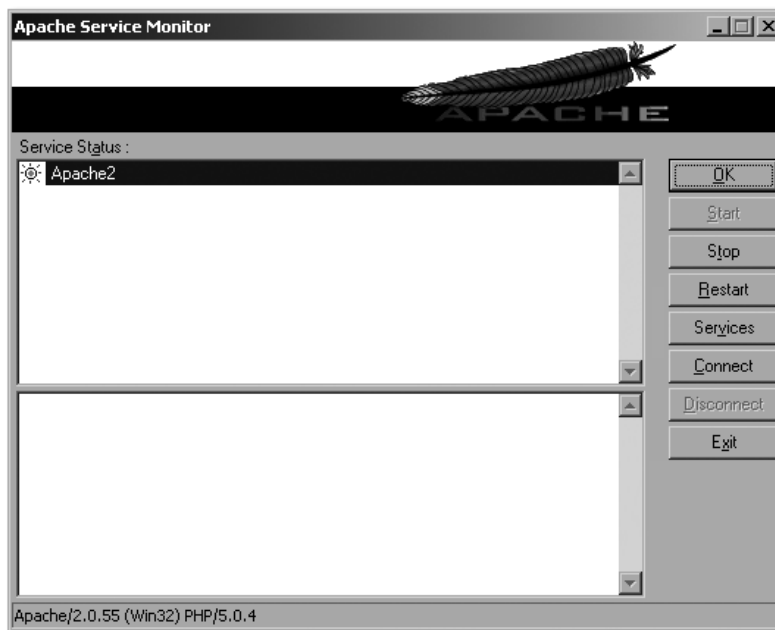


Abbildung 5.6 Das Hauptfenster des Apache-Monitors

### Eine Windows-Alternative: ApacheMon

Auf der CD-ROM zum Buch finden Sie ein interessantes Drittanbieter-Tool, das mehr Funktionen bietet als der offizielle Apache-Monitor und diesen so mehr als ersetzen kann: **ApacheMon** von **Jorge Schrauwen**. Das Programm ist für die nichtkommerzielle Nutzung kostenlos; wenn Sie es kommerziell einsetzen möchten, müssen Sie den Autor kontaktieren. Die jeweils neueste Version und weitere Informationen finden Sie auf seiner Website unter **<http://www.blackdot.be/?inc=apachemon.htm>**.

Führen Sie zur Installation einfach das Programm `ApacheMon_Installer.exe` aus und folgen Sie den Anweisungen. Im Wesentlichen geht es darum, ein Installationsverzeichnis und die zu installierenden Komponenten auszuwählen.

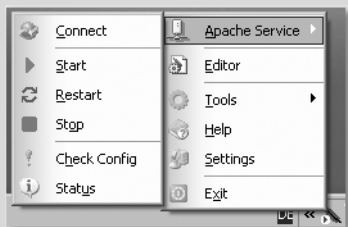


Abbildung 5.7 Das Hauptmenü von ApacheMon

Wenn Sie das Programm zum ersten Mal über das Startmenü in Betrieb nehmen, erscheint im Systray ein ähnliches Icon wie dasjenige des Apache-Monitors. Die Optionen des Programms erreichen Sie mit der rechten Maustaste (siehe Abbildung 5.7). Im Einzelnen stehen folgende Befehle zur Verfügung:

#### ► Apache Service

Dieser Menüpunkt bietet im Wesentlichen dieselben Möglichkeiten wie das Hilfsprogramm der Apache Group: Hier können Sie Apache starten, beenden oder neu starten. Mithilfe von **Connect** können Sie eine Verbindung zum Apache-Dienst auf einem anderen Windows-Rechner in Ihrem LAN herstellen; die Bedingungen, unter denen dies möglich ist, entsprechen denjenigen des Standard-Monitors.

#### ► Editor

Dies ist ein besonders nettes Werkzeug: ein Texteditor mit Syntax-Highlighting für Apache-Konfigurationsdateien.

► **Tools**

Hier finden Sie weitere praktische Zusatzwerkzeuge: Der **Password Manager** verwaltet `htpasswd`-Textdateien mit Anmeldedaten (siehe Kapitel 9, *Authentifizierung*). Mit dem **Authentication Wizard** können Sie per Dialog `.htaccess`-Dateien mit Authentifizierungseinstellungen anlegen. Der Log Resolver löst die Client-IP-Adressen in Apache-Logdateien in DNS-Namen auf, soweit diese verfügbar sind – dies ist ein grafisches Frontend für das in Kapitel 11 vorgestellte Tool `logresolve`. **Apache Benchmark** schließlich steuert das Hilfsprogramm `ab`, um die Performance eines Apache-Servers zu testen; die Kommandozeilenversion dieses Tools wird in Kapitel 12, *Skalierung und Performance-Tuning*, vorgestellt.

► **Help**

Diese Option liefert ein recht ausführliches und hilfreiches ApacheMon-Handbuch.

► **Settings**

Hier können Sie einige Voreinstellungen für das Programm vornehmen. Am wichtigsten ist **Apache's Installation Folder**; hier müssen Sie das Verzeichnis `bin` Ihrer Apache-Installation angeben, damit ApacheMon überhaupt funktioniert. **Load On Startup** richtet das Tool für den automatischen Start beim Booten ein.

### 5.1.3 Apache-Hilfsprogramme

Das Programm `httpd` (UNIX) bzw. `apache.exe` (Windows) ist nicht das einzige ausführbare Programm, das mit dem Webserver installiert wird. Es gibt zusätzlich einige nützliche Kommandozeilentools. Die meisten von ihnen sind vor allem für Aufgaben nützlich, auf die erst in späteren Kapiteln näher eingegangen wird. Aus diesem Grund finden Sie hier nur eine kurze Übersicht über die Programme mit dem Hinweis, in welchem Kapitel sie jeweils behandelt werden.

Im Einzelnen stehen neben dem ausführbaren Server-Programm und dem Skript `apachectl` folgende Programme zur Verfügung:

- `ab` – das Apache-Benchmark-Programm. Siehe Kapitel 12, *Skalierung und Performance-Tuning*.
- `apxs` – Hilfsprogramm zum nachträglichen Kompilieren von Modulen. Dieses Tool wurde bereits in Kapitel 4, *Apache kompilieren und installieren*, angesprochen.

- ▶ `dbmmanage` – Perl-Skript zur Verwaltung von Authentifizierungsdaten im DBM-Format. Siehe Kapitel 9, *Authentifizierung*.
- ▶ `htcacheclean` – ein neues Tool zur Bereinigung des festplattenbasierten Webcaches. Dieses Programm wird in Kapitel 13, *Proxy- und Cache-Funktionen*, beschrieben.
- ▶ `htdbm` – Ein binäres Verwaltungsprogramm für DBM-Authentifizierungsdaten. Siehe Kapitel 9.
- ▶ `htdigest` – Verwaltungsprogramm für Digest-Authentifizierungsdaten. Näheres in Kapitel 9.
- ▶ `htpasswd` – Verwaltungsprogramm für Basic-Authentifizierungsdaten. Auch dieses Hilfsprogramm wird in Kapitel 9 behandelt.
- ▶ `htt2dbm` – Einfaches Programm zur Umwandlung von Text- in DBM-Dateien als Nachschlagetabellen für Rewrite-Maps (URL-Umwandlung); siehe Kapitel 8, *Weiterleitungen und Indizes*.
- ▶ `logresolve` – Tool zur Ermittlung von Hostnamen zu den IP-Adressen in Logdateien. Siehe Kapitel 11, *Logging*.
- ▶ `rotatelogs` – Programm zum automatischen, regelmäßigen Wechsel von Logdateien. Auch dieses Programm wird in Kapitel 11 näher betrachtet.
- ▶ `suexec` – CGI-Skripte unter anderer User-ID ausführen. Siehe Kapitel 18, *Sicherheit*.
- ▶ `log_server_status` – Statusinformationen in eine Zeile packen und in eine Logdatei schreiben. Näheres in Kapitel 11.
- ▶ `split-logfile` – Logdateien anhand bestimmter Regeln in mehrere Einzeldateien zerlegen. Ebenfalls in Kapitel 11.

## 5.2 Apache testen

Nach Installation und Start sollten Sie überprüfen, ob Apache ordnungsgemäß funktioniert. In diesem kurzen Abschnitt werden zwei Methoden dafür beschrieben: das Überprüfen der mitgelieferten Startseite und das Einrichten einer eigenen Minimalkonfiguration.

### 5.2.1 Die automatische Startseite

Nachdem Sie den Webserver mit einer der hier beschriebenen Methoden gestartet haben, sollte er eigentlich funktionieren. Ob dies tatsächlich der Fall ist, können Sie mit einem Webbrowser testen: Das vorkonfigurierte Website-Verzeichnis `htdocs` enthält zu diesem Zweck eine Testseite, die im Browser angezeigt werden sollte, wenn Sie die Wurzeladresse Ihres Servers eingeben.

Öffnen Sie also einen Browser und geben Sie als URL **http://localhost** ein. Falls der Standardname `localhost` auf Ihrem System nicht unterstützt wird, müssen Sie statt dessen **http://127.0.0.1** eingeben. Abbildung 5.8 zeigt, wie die Seite bei Apache 2.0 aussehen sollte, wenn alles in Ordnung ist. Wenn Ihr Browser keinen `Accept-Language-Header` mit der Sprachpräferenz Deutsch sendet oder der Webserver nicht für Content-Negotiation konfiguriert ist, werden Sie die Seite allerdings – anders als in der Abbildung – auf Englisch zu sehen bekommen. Apache 2.2 zeigt nur noch die Hauptüberschrift »It works!« an.

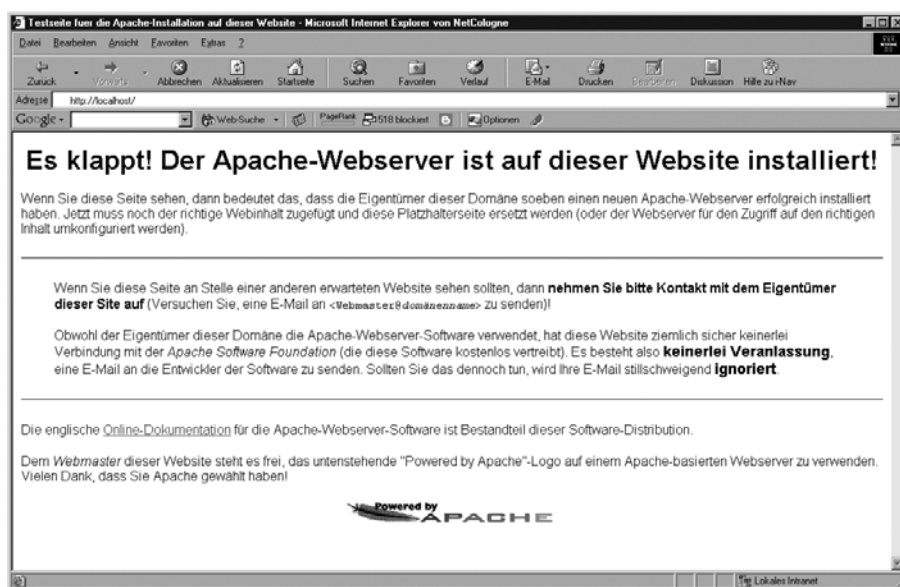


Abbildung 5.8 Die Testseite des Apache-Webservers 2.0 nach erfolgreichem Start

### 5.2.2 Die erste Website

Es genügt natürlich nicht, den Apache-Webserver einfach so zu starten. Es sind zahlreiche Anpassungen der Konfigurationsdatei `httpd.conf` erforderlich, damit er genau nach Wunsch arbeitet. Da große Teile des Inhalts von `httpd.conf` davon abhängen, welche Module im Apache-Server installiert sind, werden in diesem Unterabschnitt nur einige wenige Konfigurationsdirektiven angesprochen, die das Ausliefern statischer Dokumente ermöglichen.

Wenn Sie eine neue Apache-Installation zum ersten Mal verwenden (und vor allem, wenn Sie Apache überhaupt das erste Mal einsetzen), sollten Sie nicht einfach ohne weiteres eine Website auf die Menschheit loslassen. Es gibt Unmengen von Konfigurationsanweisungen und Anpassungsmöglichkeiten.

Aus diesem Grund empfiehlt es sich, einen neu installierten Server vor der Inbetriebnahme mit einer Test-Website zu überprüfen.

In diesem kurzen Abschnitt wird eine kleine Website mit einer minimalen Konfigurationsdatei erstellt. Einzelheiten zu den zahlreichen Optionen für die Datei `httpd.conf` finden Sie weiter unten im Buch; hier geht es erst einmal darum, überhaupt eine Website zu veröffentlichen.

Erstellen Sie als Erstes ein Verzeichnis, das den Stamm Ihrer Website bilden soll. Erstellen Sie darin eine HTML-Datei namens `index.html` mit beliebigem Inhalt (oder kopieren Sie den Inhalt des Verzeichnisses `testsite` von der CD zum Buch hinein). Als Verzeichnis für den Test können Sie beispielsweise das offizielle `htdocs`-Verzeichnis Ihres Apache-Servers oder ein neu erstelltes Unterverzeichnis davon benutzen.

Benennen Sie die vorgefertigte Konfigurationsdatei `httpd.conf` um. Erstellen Sie die folgende neue Minimaldatei (Erläuterung der austauschbaren Elemente siehe unten):

```
ServerName www.mynet.de
Listen 80
ServerRoot /usr/local/apache2

# Dieser Block wird nur bei
# DSO-basierter Modulinstallation benötigt
LoadModule dir_module modules/mod_dir.so
LoadModule autoindex_module modules/mod_autoindex.so
# Nur Apache 2.0:
LoadModule access_module modules/mod_access.so
# Nur Apache 2.1/2.2:
LoadModule authz_host_module modules/mod_authz_host.so

# An Ihr Installationslayout anpassen
DocumentRoot /usr/local/apache2/htdocs

# Absicherung durch Absperren des Wurzelverzeichnisses
<Directory />
    Options None
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>
```

```
# Freigeben der DocumentRoot
<Directory /usr/local/apache2/htdocs>
  DirectoryIndex index.html
  Options All
  AllowOverride All
  Order allow,deny
  Allow from all
</Directory>
```

Die Verzeichnisangaben für die Direktiven `ServerRoot`, `DocumentRoot` und den `<Directory>`-Container müssen Sie natürlich den Gegebenheiten Ihrer Plattform und Ihrer Apache-Installation anpassen. Das obige Beispiel entspricht dem Apache-Layout unter UNIX.

Zu beachten ist in diesem Zusammenhang, dass auch unter Windows der Slash (/) und nicht der plattformtypische Backslash (\) als Pfadtrennzeichen benutzt werden muss. Angenommen, Apache ist auf Ihrem Windows-Rechner unter `C:\Programme\Apache2` installiert, dann sieht die Datei folgendermaßen aus:

```
ServerName www.mynet.de
Listen 80
ServerRoot C:/Programme/Apache2

LoadModule dir_module modules/mod_dir.so
LoadModule autoindex_module modules/mod_autoindex.so
# Nur Apache 2.0:
LoadModule access_module modules/mod_access.so
# Nur Apache 2.1/2.2:
LoadModule authz_host_module modules/mod_authz_host.so

# An Ihr Installationsverzeichnis anpassen
DocumentRoot C:/Programme/Apache2/htdocs

# Absicherung durch Absperren des Wurzelverzeichnisses
<Directory />
  Options None
  AllowOverride None
  Order deny,allow
  Deny from all
</Directory>
```

```
# Freigeben der DocumentRoot
<Directory C:/Programme/Apache2/htdocs>
  DirectoryIndex index.html
  Options All
  AllowOverride All
  Order allow,deny
  Allow from all
</Directory>
```

Wie Sie sehen, sind die drei `LoadModule`-Direktiven unter Windows nicht optional: Bei der Standardversion, die der im vorigen Kapitel vorgestellte MSI-Installer einrichtet, werden Module immer als DSOs geladen und sind niemals statisch einkompiliert.

Im Übrigen sollten Sie **www.mynet.de** für einen lokalen Test zu Ihrer `hosts`-Datei (UNIX: `/etc/hosts`, Windows: `%Systemroot%\System32\drivers\etc\hosts`) hinzufügen. Andernfalls erhalten Sie beim Apache-Start die folgende Meldung:

```
Could not qualified the Server's full qualified domain name.
Using 127.0.0.1 for Server Name.
```

Erstellen Sie in der Datei `hosts` einen Eintrag wie diesen, um den Server auf demselben Host zu testen:

```
127.0.0.1 www.mynet.de
```

Für einen Test über das Netzwerk müssen Sie statt **www.mynet.de** die IP-Adresse der entsprechenden Netzwerkschnittstelle verwenden. Bei einem Produktions-Server muss der Name gemäß den Erläuterungen in Kapitel 1, *IP-Netzwerke, Internet und WWW*, über einen DNS-Server aufgelöst werden, damit der Server im Internet unter diesem Namen erreichbar ist.

Starten Sie Apache 2 nun neu, indem Sie der weiter oben beschriebenen Anleitung für Ihre Plattform folgen. Falls Sie die Minimalkonfigurationsdatei unter einem anderen Namen als `httpd.conf` gespeichert haben, können Sie sie dabei mit der Option `-f Dateiname` angeben. Angenommen, Sie verwenden ein UNIX-System und die Datei `/etc/httpd/httpd_minimal.conf`, dann sieht der entsprechende Befehl so aus:

```
# apachectl graceful -f /etc/httpd/httpd_minimal.conf
```

Auf einem Windows-Rechner müssen Sie dagegen diesen Befehl eingeben, falls Sie keine der bereits behandelten grafischen Methoden einsetzen und die Kon-

figurationsdatei `C:\Programme\Apache2\conf\httpd_minimal.conf` verwenden:

```
> apache -k restart
-f C:/Programme/Apache2/conf/httpd_minimal.conf
```

Nach dem Neustart können Sie einen Browser öffnen und versuchen, die neue Startseite der Website aufzurufen. Falls alles in Ordnung ist, müssten Sie nun die eben erstellte HTML-Seite sehen.

Hier noch eine kurze Übersicht über die Direktiven, die in der Minimalkonfiguration verwendet werden – sie alle werden im nächsten Kapitel genauer erläutert:

► **ServerName `www.mynet.de`**

Hier wird der Name festgelegt, unter dem der Server im Intra- oder Internet erreichbar sein soll.

► **Listen `80`**

Diese Direktive legt den TCP-Port fest, an dem der Server auf HTTP-Verbindungsanfragen lauscht. Wie Sie bereits wissen, ist 80 der Standardport für das HTTP.

► **ServerRoot `/usr/local/apache2`**

Die Konfigurationsanweisung `ServerRoot` legt das Verzeichnis fest, in dem sich die Konfigurations- und Ressourcendateien des Servers befinden, das heißt Verzeichnisse wie `conf`. Verschiedene andere Direktiven benötigen als Parameter eine Pfadangabe, die relativ zu diesem Verzeichnis angegeben werden kann.

► **LoadModule `dir_module modules/mod_dir.so`**

Diese Zeile und die beiden nächsten werden auf einem UNIX-System nur benötigt, wenn alle Module (oder die drei Module `mod_dir`, `mod_autoindex` und `mod_access` bzw. `mod_authz_host` ausdrücklich) als DSOs installiert wurden. Mithilfe von `LoadModule` wird ein DSO-Modul beim Server-Start geladen und aktiviert. Es werden jeweils ein schematischer Modulname und der Pfadname des Moduls (relativ zur `ServerRoot`) benötigt. Für den schematischen Namen müssen Sie das Präfix `mod_` vom Dateinamen entfernen und statt dessen `_module` anfügen. Aus `mod_foo` würde also beispielsweise `foo_module` mit folgender Direktive:

```
LoadModule foo_module modules/mod_foo.so
```

Unter Windows werden die `LoadModule`-Direktiven fast immer verwendet, weil Module auf dieser Plattform in aller Regel als DSOs installiert werden.

`mod_dir` ist übrigens das Modul, das die Definition des Startseitennamens ermöglicht und darüber hinaus Verzeichnis-URLs ohne abschließenden `/` verarbeitet. `mod_autoindex` generiert dagegen automatisch einen Index aller Dateien im Verzeichnis, wenn keine Startseite vorhanden ist. `mod_access` oder `mod_authz_host` schließlich erlaubt Zugriffsbeschränkungen auf Hostnamens- und IP-Adress-Ebene.

► **DocumentRoot /usr/local/apache2/htdocs**

Dies ist das Verzeichnis, in dem sich die Website befindet, die der Server veröffentlicht.

► **<Directory /> ... </Directory>**

Ein `<Directory>`-Container enthält die Konfigurationsoptionen für ein einzelnes Verzeichnis. Der Container für das Wurzelverzeichnis bestimmt die Voreinstellung für *alle* Verzeichnisse und URLs. Es gibt einige Direktiven, die nur in Verzeichniscontainern stehen dürfen.

► **Options None**

Die Direktive `Options` legt fest, welche »Dienstleistungen« in diesem Verzeichnis verfügbar sein sollen. Die Voreinstellung `None` für das Wurzelverzeichnis besagt, dass zunächst einmal sämtliche Optionen deaktiviert sind. Einzelheiten erfahren Sie im nächsten Kapitel.

► **AllowOverride None**

Es besteht die Möglichkeit, Konfigurationsdirektiven in eine Datei innerhalb eines einzelnen Verzeichnisses oder sogar Unterverzeichnisses einer Website auszulagern. Diese Datei heißt standardmäßig `.htaccess` (kann durch die Direktive `AccessFileName` geändert werden). Am bekanntesten ist dieses Verfahren zur Erstellung passwortgeschützter Verzeichnisse; neben Authentifizierungsdirektiven lassen sich aber auch andere Einstellungen durch diese Datei lokal überschreiben.

`AllowOverride` legt fest, welche Direktiven überschrieben werden dürfen. Dazu werden diese nicht einzeln angegeben, sondern es gibt einige spezielle Bezeichnungen für Funktionsgruppen, die im nächsten Kapitel ausgeführt werden. Die Voreinstellung `None` bedeutet natürlich, dass zunächst einmal keinerlei Direktiven überschrieben werden dürfen. `.htaccess`-Dateien werden in diesem Fall sogar vollständig ignoriert.

► **Order deny,allow**

Die Direktive `Order` aus dem Modul `mod_access` bestimmt, in welcher Reihenfolge die Zugriffseinstellungen `Allow` und `Deny` verarbeitet werden. `allow,deny` bedeutet, dass die Verbote die Erlaubnisse einschränken. Bei `deny,allow` ist es umgekehrt.

► **Deny from all**

Die Direktive `Deny` stammt ebenfalls aus `mod_access`. Diese rigorose Einstellung besagt, dass auf das Wurzelverzeichnis zunächst einmal *niemand* zugreifen darf.

► **<Directory /usr/local/apache2/htdocs> ... </Directory>**

Dieser Container enthält die Einstellungen für die freigegebene Website. Da das Wurzelverzeichnis sehr restriktiv abgesperrt wurde, müssen hier einige Einstellungen durch ausdrückliche Erlaubnisse überschrieben werden: `Options` und `AllowOverride` werden auf `All` gesetzt – unterhalb der `DocumentRoot` sind alle Verzeichnisoptionen verfügbar, und das Maximum an Direktiven darf in `.htaccess`-Dateien überschrieben werden. Order erhält den Wert `allow,deny` – die `Allow`-Einstellung soll Vorrang besitzen.

► **DirectoryIndex index.html**

Diese durch das Modul `mod_dir` bereitgestellte Direktive definiert die Namen eines oder mehrerer Dokumente, die der Server ausliefert, wenn die URL in einer Anfrage nur einen Verzeichnis-, aber keinen Dateinamen enthält. Falls mehrere Dateien angegeben werden, sucht Apache in der angegebenen Reihenfolge nach ihnen und liefert das erste Dokument aus, das er findet. `index.html` ist die Standardeinstellung.

► **Allow from all**

Genau wie `Deny` ist auch `Allow` in `mod_access` definiert. Diese Zeile erlaubt allen Hosts aus dem gesamten Internet den Zugriff auf die Website.

## 5.3 Zusammenfassung

Das ausführbare Programm, das den Kern des Apache-Webservers bildet, lässt sich – typisch für Software aus der UNIX-Welt – mit zahlreichen Parametern und Optionen aufrufen. Sie können den Server damit nicht nur einfach starten, sondern über die bestehende Konfigurationsdatei hinaus anpassen oder auch wichtige Informationen über den Status des HTTP-Servers erhalten. Dies gilt sowohl für Apache auf der UNIX-Plattform als auch auf Windows-Systemen.

Daneben gibt es zahlreiche Hilfsmittel zur Steuerung des Webservers. Auf UNIX-Systemen ist er mit dem Shell-Skript `apachectl` ausgestattet, das nicht nur den angepassten Start, sondern auch das Beenden und den Neustart von Apache 2 ermöglicht. Unter Windows wurden einige dieser Optionen zum ausführbaren Programm (`apache.exe`) hinzugefügt.

Eine gewisse Herausforderung besteht in dem Problem, Apache beim Hochfahren des Systems automatisch zu starten. Auf einem UNIX-Rechner müssen Sie herausfinden, ob Ihr System System V Init oder die BSD-Boot-Methode ver-

wendet, um die entsprechenden Startbefehle hinzuzufügen. Unter Windows empfiehlt sich zu diesem Zweck der Betrieb des Webservers als Dienst.

Nachdem Sie Ihren Server gestartet haben, sollten Sie mit einer Minimalkonfiguration wie der hier beschriebenen überprüfen, ob er auch ordnungsgemäß seine Arbeit erledigt, bevor Sie ihn praktisch im Internet oder Intranet einsetzen.

## 18 Sicherheit

*Sicherheit erreicht man nicht, indem man Zäune errichtet, sondern indem man Türen öffnet.*  
– Urho Kekkonen

Dieses kurze Kapitel versammelt die wichtigsten Informationen, die Sie zur Absicherung Ihres Apache-Webserver benötigen. Damit ist der Schutz vor Crackerattacken und sonstigen unberechtigten Zugriffen gemeint. Die Beschränkung Ihrer Websites auf bestimmte User wird hier dagegen nicht behandelt; dieses Thema wurde bereits ausführlich in Kapitel 9, *Authentifizierung*, behandelt.

### 18.1 Sicherheit der Server-Umgebung

Sicherheitskonzepte, die nichts mit dem Betrieb von Apache selbst zu tun haben, würden den Rahmen dieses Buches sprengen. Selbstverständlich sollten Sie aber folgende zusätzliche Sicherheitsmaßnahmen in Erwägung ziehen:

#### ► Eine Firewall

Natürlich können Sie einen Webserver, dessen Aufgabe die Veröffentlichung von Informationen für den allgemeinen, anonymen Zugriff ist, nicht vollständig hinter einer Firewall verbergen. Vielmehr sollten Sie über eine intelligente Verteilung der betroffenen Rechner im Netzwerk nachdenken. Möglicherweise können Sie Proxy-Dienste verwenden, um eine Firewall zwischen dem eigentlichen Webserver und Backend-Servern, etwa für Anwendungen oder Datenbanken, zu überbrücken.

Übrigens sollten Sie neben einer netzwerkweiten Firewall auch auf dem Webserver-Host selbst eine Software einrichten, die Zugriffe auf bestimmte Ports unterbindet oder spezielle Pakete zurückweist. Wenn Sie Linux verwenden, ist die Firewall in Form von `iptables` bereits im Kernel enthalten. Für Windows können Sie beispielsweise die **Kerio Personal Firewall** installieren; Sie finden sie auf der beiliegenden CD-ROM. Durch die grafische Konfigurationsoberfläche ist die Definition von Regeln bei dieser Firewall sehr einfach; zudem verfügt sie über einen »Lernmodus«, in dem Sie für jeden Vorfall ad hoc eine Verhaltensweise wählen oder als Regel festlegen können.

#### ► Ein Intrusion Detection System (IDS)

Angriffe oder Angriffsversuche sind nicht immer leicht zu erkennen. Am besten ist es, Sie installieren eine Software, die unerwünschte Änderungen erkennt und Alarm schlägt. Eine gute Wahl für Linux ist beispielsweise

tripwire. Sie finden dieses Open-Source-Programm auf der CD zum Buch. Weiter unten wird zudem `mod_security` vorgestellt – es handelt sich um die Verwirklichung eines IDS als Apache-Modul.

► **chroot-Umgebung**

Auf UNIX-Systemen gibt es noch eine weitere interessante Sicherheitsmaßnahme: Sie können ein Programm in einen `chroot`-»Käfig« sperren. Das bedeutet, dass der Software vorgegaukelt wird, ihr lokales Unterverzeichnis sei das Wurzelverzeichnis des Dateisystems. Beachten Sie allerdings, dass auch dies keine absolute Sicherheit gewährleistet – es gibt immer wieder Sicherheitslücken in den eingesperrten Programmen oder in den `chroot`-Implementierungen, die es einem Angreifer ermöglichen, aus der `chroot`-Umgebung zu entkommen.

Die Einrichtung einer solchen Umgebung ist vor allem deshalb nicht ganz einfach, weil Sie herausfinden müssen, welche Verzeichnisse und Dateien Apache außer seinem eigenen Installationsumfang benötigt, um ausgeführt werden zu können. Die genaue Konfiguration hängt stark von Ihrem konkreten System ab. Auf der Website zum Buch finden Sie Links zu Informationen für diverse Linux- und UNIX-Versionen.

► **»Crackertools«**

Im ersten Moment hört es sich vielleicht merkwürdig an, aber eine der besten Schutzmaßnahmen vor Angriffen besteht darin, die Werkzeuge der potentiellen Angreifer zu benutzen, um Sicherheitslücken zu erkennen und nach Möglichkeit zu schließen. Eines der wertvollsten Tools in dieser Sparte ist **Nessus**.<sup>1</sup> Dieses freie Programm funktioniert unter zahlreichen UNIX-Varianten; für Windows ist ein Port namens **NeWT** verfügbar. Sie finden beide Versionen auf der CD.

► **Geschulte Mitarbeiter!**

Die beste technische Sicherheitsmaßnahme nützt nichts, wenn die User nicht darüber Bescheid wissen. Menschliches Versagen ist immer die größte Sicherheitslücke; einige der erfolgreichsten Cracker haben nicht nur durch ausgeklügelte technische Maßnahmen Zugriff auf geschützte Systeme erlangt, sondern vor allem durch »Social Engineering« – siehe beispielsweise [MITN 2003]. Besonders der vernünftige Umgang mit Passwörtern ist weithin unbekannt. Weisen Sie Ihre Mitarbeiter deshalb besonders auf folgende Punkte hin:

- Passwörter dürfen niemals sinnvolle Wörter sein, sondern sollten möglichst wirre Kombinationen aus Groß- und Kleinbuchstaben, Ziffern und Sonderzeichen sein.

---

<sup>1</sup> Damit will ich nicht gesagt haben, die Intention der Nessus-Entwickler sei die Unterstützung von Crackern!

- ▶ Passwörter müssen regelmäßig gewechselt werden. Wenn Ihr Betriebssystem eine Möglichkeit bietet, die Benutzer nach einer bestimmten Zeit zur Änderung zu zwingen, sollten Sie diese nutzen.
- ▶ Ein Passwort wird niemals aufgeschrieben! Ein sicheres Passwort, das man sich dennoch leicht merken kann, ergibt sich am einfachsten aus den Anfangsbuchstaben eines möglichst sinnlosen Satzes; einige Zeichen sollten allerdings keine Buchstaben sein.
- ▶ Administratoren, Mitarbeiter von Internet-Providern, Polizisten oder Staatsbeamte fragen *niemals* nach dem Passwort. Wer auch immer Sie also danach fragt, ist ein Betrüger.
- ▶ Passwörter sind erst der Anfang – die vorgestellten Maßnahmen zur Passwortsicherheit sind wichtig. Für sich allein sind sie aber noch lange keine ausreichende Absicherung; das sollten sowohl Sie als auch die restlichen Benutzer wissen.

## 18.2 Apache-Sicherheit

Dieser Abschnitt stellt zunächst einige allgemeine Tipps zusammen, die Sie über das Buch verteilt bereits erhalten haben; anschließend werden einige eingebaute Direktiven zur Sicherheit vorgestellt.

### 18.2.1 Allgemeine Sicherheitshinweise

In diesem Buch finden Sie die wichtigsten Hinweise zur Sicherheit jeweils an Ort und Stelle. Dennoch ist das Thema so bedeutend, dass die wichtigsten Sicherheitstipps hier noch einmal im Überblick zusammengefasst werden.

#### ▶ Im Allgemeinen schützen, im Besonderen freigeben

Sie sollten es sich zur Regel machen, sämtliche Direktiven, die mit der Aktivierung von Zugriffsrechten oder der Freischaltung von Optionen zu tun haben, grundsätzlich so restriktiv wie möglich einzustellen und dann nur in Einzelfällen zu lockern. Dazu gehört insbesondere die grundsätzliche Voreinstellung zum Sperren aller Verzeichnisse:

```
<Directory />
  Order deny,allow
  Deny from all
  Options None
  AllowOverride None
</Directory>
```

► **ScriptAlias-Verzeichnisse für CGI verwenden**

Es ist ein relativ großes Sicherheitsrisiko, Verzeichnisse unterhalb der `DocumentRoot` für CGI-Anwendungen freizuschalten. Sie sollten deshalb nach Möglichkeit `ScriptAlias`-Verzeichnisse verwenden. In manchen Situationen lässt es sich allerdings nicht verhindern, CGI innerhalb normaler Site-Verzeichnisse zu aktivieren – das ist beispielsweise der Fall, wenn Sie Hosting-Dienstleistungen mit virtuellen Hosts anbieten. Näheres zu diesem Thema finden Sie in Kapitel 14, *CGI*.

► **Eine separate User- und Group-ID für Apache benutzen**

Auf UNIX-Systemen muss Apache unter der User-ID `root` gestartet werden, beispielsweise weil der Zugriff auf die TCP-Ports 0 bis 1023 anders nicht möglich ist. Die Child-Prozesse, die Anfragen bearbeiten, werden allerdings unter den IDs ausgeführt, die Sie mithilfe der Direktiven `User` und `Group` festgelegt haben. Achten Sie darauf, dass dieser Benutzer und diese Gruppe speziell für Apache angelegt werden. Das Benutzerkonto sollte keine Möglichkeit zur persönlichen Anmeldung besitzen; sogar das Passwort sollten Sie nach der Eingabe vergessen.

Übrigens besteht auch unter Windows die Möglichkeit, einen separaten Benutzer für den Betrieb von Apache einzurichten. Das Verfahren wurde in Kapitel 5, *Apache in Betrieb nehmen*, ausführlich erläutert.

► **Regelmäßige Updates durchführen**

Open-Source-Software besitzt zahlreiche Vorteile gegenüber kommerziellen Programmen. Einer der wichtigsten ist die schnellere Erkennung von Sicherheitslücken und die meist ebenso prompte Bereitstellung von Patches. Wichtig ist allerdings, dass Sie diese Patches auch einspielen; zahlreiche Angriffe sind nur deshalb erfolgreich, weil die zuständigen Admins dies monatelang versäumt haben. Sie sollten zumindest stets die neueste Release von Apache installieren, weil darin jeweils alle bisher bekannten Sicherheitslücken geschlossen werden. Die Devise »Never change a running system« gilt hier nicht!

Allein der Wechsel von 2.0.54 zu 2.0.55 hat mehrere Sicherheitslücken geschlossen, wenn auch für unterschiedliche Plattformen. Eine vollständige Liste der Änderungen in allen Versionen von Apache 2.0, einschließlich Sicherheitslücken und Gegenmaßnahmen dazu, finden Sie in der Datei **[www.apache.org/dist/httpd/CHANGES\\_2.0](http://www.apache.org/dist/httpd/CHANGES_2.0)**. Diese Datei ist übrigens auch Teil jeder Apache-Distribution.

Andererseits sollten Sie Apache 2.1/2.2 noch nicht für ein wichtiges Produktsystem einsetzen, solange die Software sich offiziell im Betastadium befin-

det. Bei Erscheinen dieses Buches wird es aber wahrscheinlich bereits die erste Stable Release von Apache 2.2 geben.

► **Logdateien auswerten**

Unregelmäßigkeiten in Ihren Logdateien deuten oft auf Angriffsversuche hin. Sie sollten diese sorgfältig untersuchen und dann gegebenenfalls Schutzmaßnahmen gegen eine bestimmte Art von Attacken einleiten. Alles Wichtige über Logdateien steht in Kapitel 11, *Logging*.

► **Sicherheits-Mailinglisten abonnieren**

Als verantwortungsvoller Administrator sollten Sie Mailinglisten zum Thema Sicherheit lesen, in denen regelmäßig über Sicherheitslücken und eventuelle Patches berichtet wird. Die bekannteste derartige Liste ist Bugtraq (mittlerweile aufgeteilt in zahlreiche Spezialgebiete). Sie finden sie auf der Site [www.securityfocus.com](http://www.securityfocus.com).

Eine recht brauchbare wöchentliche Übersicht von Sicherheitsthemen bietet darüber hinaus der Heise Security Newsletter. Sie können ihn unter [www.heise.de](http://www.heise.de) abonnieren.

### 18.2.2 Sicherheitsrelevante Direktiven

Es gibt einige Apache-Konfigurationsdirektiven, die indirekt die Sicherheit betreffen. Sie schützen insbesondere vor Denial-of-Service-Attacken, indem sie verschiedene Aspekte der HTTP-Anfrage sowie der Ressourcen-Nutzung einschränken.

#### **LimitInternalRecursion**

Maximale Anzahl interner Weiterleitungen

<b>Modul</b>	core
<b>Kontext</b>	Server, <VirtualHost>
<b>Syntax</b>	LimitInternalRecursion Anzahl [Anzahl]
<b>Standardwert</b>	10

Diese Direktive verhindert, dass Weiterleitungen bzw. Unteranfragen in eine Endlosschleife geraten; nach der angegebenen Höchstzahl von Umleitungen wird die Anfrage verworfen. Der erste Wert gilt für Weiterleitungen. Wenn Sie einen zweiten Wert angeben, wird dieser für verschachtelte Unteranfragen verwendet; ansonsten gilt der erste Wert für beide.

## LimitRequestBody

Zulässige Größe für den Body der HTTP-Anfrage

<b>Modul</b>	core
<b>Kontext</b>	Server, <VirtualHost>, <Directory>, <Location>, <Files>, .htaccess (All)
<b>Syntax</b>	LimitRequestBody Bytes
<b>Standardwert</b>	0 (unbegrenzt)

Mithilfe dieser Direktive können Sie die Größe einschränken, die der Body von Anfragen mit den HTTP-Methoden `POST` oder `PUT` besitzen darf. Der Wert kann zwischen 0 (unbegrenzt; Voreinstellung) und 2.147.483.647 (2 Gigabyte) liegen. Das folgende Beispiel begrenzt die Größe auf 500 Kilobyte:

```
LimitRequestBody 512000
```

## LimitRequestFields

Zulässige Höchstzahl von HTTP-Anfrage-Headern

<b>Modul</b>	core
<b>Kontext</b>	Server
<b>Syntax</b>	LimitRequestFields Anzahl
<b>Standardwert</b>	100

Diese Direktive limitiert die Anzahl der HTTP-Header-Zeilen, die von einem Client akzeptiert werden. Der Wert darf zwischen 0 (unbegrenzt) und 32767 liegen. Im Apache-Quellcode existiert die Konstante `DEFAULT_LIMIT_REQUEST_FIELDS`, die den Vorgabewert (normalerweise 100) bestimmt. Mehr als 100 Header-Felder sind für normale Anfragen übrigens so gut wie ausgeschlossen; eine solche Anfrage weist eher auf einen Angriffsversuch hin. Normalerweise können Sie den Wert sogar noch einschränken. Beispiel:

```
LimitRequestFields 30
```

## LimitRequestFieldSize

Zulässige Maximalgröße des HTTP-Anfrage-Headers

<b>Modul</b>	core
<b>Kontext</b>	Server
<b>Syntax</b>	LimitRequestFieldSize Bytes
<b>Standardwert</b>	8190

Diese Direktive beschränkt die Gesamtlänge aller HTTP-Anfrage-Header, die ein Client senden darf. Der Vorgabewert wird durch die Konstante `DEFAULT_LIMIT_REQUEST_FIELDSIZE` bestimmt und beträgt zunächst 8190. Das ist für gewöhnlich mehr als ausreichend.

### LimitRequestLine

Zulässige Höchstlänge einer HTTP-Anfragezeile

<b>Modul</b>	core
<b>Kontext</b>	Server
<b>Syntax</b>	LimitRequestLine Bytes
<b>Standardwert</b>	8190

Mit dieser Direktive wird die Länge der ersten Zeile einer HTTP-Anfrage begrenzt. Diese Zeile besitzt normalerweise einen Aufbau wie im folgenden Beispiel (Näheres siehe Kapitel 2, *Funktionsweise von Webservern*):

```
GET / HTTP/1.1
```

Da Pfadnamen selten eine solche Länge erreichen, geht es vor allem um eine Beschränkung für die Länge von Query-Strings, die bekanntlich Teil der angeforderten URL sind. Der Standardwert von 8190 genügt in aller Regel. Er wird durch die Konstante `DEFAULT_LIMIT_REQUEST_LINE` bestimmt.

### LimitXMLRequestBody

Zulässige Maximalgröße eines HTTP-Anfrage-Bodys im XML-Format

<b>Modul</b>	core
<b>Kontext</b>	Server, <VirtualHost>, <Directory>, <Location>, <Files>, .htaccess (All)
<b>Syntax</b>	LimitXMLRequestBody Bytes
<b>Standardwert</b>	1000000

Diese Direktive schränkt die maximal erlaubte Größe eines HTTP-Anfrage-Bodys im XML-Format ein. Dies ist besonders im Zusammenhang mit WebDAV (siehe voriges Kapitel) wichtig. Wenn Sie den Wert 0 verwenden, wird diese Prüfung deaktiviert.

### **RLimitCPU**

Begrenzt die CPU-Nutzung durch Prozesse, die von Apache-Child-Prozessen gestartet wurden.

<b>Modul</b>	core
<b>Kontext</b>	Server, <VirtualHost>, <Directory>, <Location>, <Files>, .htaccess (All)
<b>Syntax</b>	RLimitCPU Sekunden max
<b>Standardwert</b>	unbestimmt

Mit »Prozessen, die von Apache-Child-Prozessen gestartet wurden«, sind CGI-Skripte und andere serverseitige Anwendungen gemeint. Die `RLimit*`-Direktiven schützen nicht so sehr vor externen Angriffsversuchen, sondern viel stärker vor schlechter Programmierung.

`RLimitCPU` limitiert die CPU-Nutzung durch untergeordnete Prozesse. Sie können einen oder zwei Werte angeben. Die erste Angabe ist eine weiche Begrenzung (es wird kein neuer Prozess gestartet, wenn der Wert überschritten wurde); die zweite ist die absolute Obergrenze. Die Angabe erfolgt in Sekunden pro Prozess. Der spezielle Wert `max` bezeichnet die Höchstgrenze des Betriebssystems.

### **RLimitMEM**

Begrenzt die Speichernutzung durch Prozesse, die von Apache-Child-Prozessen gestartet wurden.

<b>Modul</b>	core
<b>Kontext</b>	Server, <VirtualHost>, <Directory>, <Location>, <Files>, .htaccess (All)
<b>Syntax</b>	RLimitMEM Bytes max
<b>Standardwert</b>	unbestimmt

Diese Direktive bestimmt, wie viel Speicher abhängige Unterprozesse wie CGI oder SSI nutzen dürfen. Sie können entweder eine bestimmte Byte-Anzahl angeben oder den Wert `max` verwenden, der den Grenzwert des Systems darstellt.

### **RLimitNPROC**

Begrenzt die Erzeugung neuer Prozesse durch Prozesse, die wiederum von Apache-Child-Prozessen gestartet wurden

<b>Modul</b>	<code>core</code>
<b>Kontext</b>	<code>Server</code> , <code>&lt;VirtualHost&gt;</code> , <code>&lt;Directory&gt;</code> , <code>&lt;Location&gt;</code> , <code>&lt;Files&gt;</code> , <code>.htaccess</code> (All)
<b>Syntax</b>	<code>RLimitNPROC Anzahl max [Anzahl max]</code>
<b>Standardwert</b>	unbestimmt

Mit dieser Direktive können Sie festlegen, wie viele Child-Prozesse ein abhängiger Prozess starten darf. Mit anderen Worten: Es geht um spezielle Webanwendungen, die ihrerseits `fork()` aufrufen, um Child-Prozesse zu erzeugen. Genau wie bei `RLimitCPU` ist der erste Wert eine weiche Grenze; der optionale zweite Wert ist das absolute Maximum. `max` ist wiederum die Voreinstellung des Betriebssystems.

### **EnableExceptionHook**

Aktiviert einen zusätzlichen Hook für die Ausnahmebehandlung.

<b>Seit Version</b>	2.0.49
<b>Modul</b>	<code>prefork</code> , <code>worker</code> , <code>threadpool</code> , <code>leader</code> , <code>perchild</code>
<b>Kontext</b>	<code>Server</code>
<b>Syntax</b>	<code>EnableExceptionHook On Off</code>
<b>Standardwert</b>	<code>Off</code>

Diese Direktive steht nur zur Verfügung, wenn Apache mit der Option `--enable-exception-hook` kompiliert wird (siehe Kapitel 4, *Apache kompilieren und installieren*). Wenn Sie den Wert `On` setzen, wird ein zusätzlicher Hook aktiviert, der beim Absturz eines Worker-Prozesses eine Funktion aufrufen kann. Der Umgang mit Hooks in der Modulprogrammierung wurde im vorigen Kapitel erläutert; es gibt bereits zwei Zusatzmodule, die von diesem Hook

Gebrauch machen: `mod_whatkilledus` und `mod_backtrace`. Beide dienen dazu, die Umstände des Absturzes aufzuklären. Näheres über den Exception-Hook und diese Module erfahren Sie auf den Seiten von Jeff Trawick unter [http://www.apache.org/~trawick/exception\\_hook.html](http://www.apache.org/~trawick/exception_hook.html).

### 18.2.3 SuEXEC

Mithilfe von SuEXEC haben Sie die Möglichkeit, CGI-Skripte und Server Side Includes unter einer anderen User- und Group-ID auszuführen als Apache selbst. Einem Angreifer, dem der Einbruch über ein schlecht abgesichertes Skript gelingt, wird auf diese Weise der Zugriff auf den eigentlichen Webserver erschwert. Apache ruft zu diesem Zweck das Hilfsprogramm `suexec` auf, das im Wesentlichen ein `setuid`- und `setgid`-Wrapper für eigene Skripte ist.

Damit `suexec` benutzt werden kann, muss Apache 2 zunächst einmal mit den passenden Einstellungen kompiliert worden sein. Diese `configure`-Optionen wurden bereits in Kapitel 4, *Apache kompilieren und installieren*, erwähnt. Hier sehen Sie sie noch einmal in etwas ausführlicherer Beschreibung:

▶ **--enable-suexec[=shared]**

Damit wird das Modul `mod_suexec` grundsätzlich aktiviert; die Variante mit `shared` müssen Sie natürlich eingeben, um es als DSO-Modul zu kompilieren.

▶ **--with-suexec-bin=/Pfad/von/suexec**

Aus Sicherheitsgründen muss der Pfad des Programms `suexec` fest einkompiliert werden. Die Voreinstellung variiert je nach Installationslayout.

▶ **--with-suexec-caller=UID**

Dies ist die User-ID des Benutzers, der `suexec` aufrufen soll. Es handelt sich um die UID, unter der auch Apache selbst ausgeführt wird.

▶ **--with-suexec-userdir=Verzeichnis**

Mit dieser Option wird der Name des Verzeichnisses unterhalb der Home-Verzeichnisse von Benutzern angegeben, in dem `suexec`-CGI-Skripte verwendet werden können. Der Standardwert ist `public_html`.

▶ **--with-suexec-docroot=Verzeichnis**

Hier wird die `DocumentRoot` von Apache angegeben. Diese bildet das allgemeine Verzeichnis, in dem `suexec`-Skripte gestattet sind.

▶ **--with-suexec-uidmin=UID**

Dies ist der niedrigste numerische Wert für die User-ID, unter der `suexec`-CGIs ausgeführt werden dürfen. Der Standardwert ist 100, was meist in Ordnung ist.

- ▶ **--with-suexec-gidmin=GID**  
Dieser Wert ist die niedrigste numerische Group-ID, der die Ausführung von `suexec`-Skripten erlaubt wird. Auch diese Option besitzt die Voreinstellung 100.
- ▶ **--with-suexec-logfile=/Pfad/zur/Datei**  
`suexec` führt eine eigene Logdatei, in der sämtliche Aktionen und Fehler protokolliert werden. Der Standardwert ist `suexec_log` im Standardverzeichnis für Logdateien (`--logfiledir`).
- ▶ **--with-suexec-safepath=Verzeichnis**  
Diese Option bestimmt den Inhalt der Umgebungsvariablen `PATH`, der an `suexec`-CGIs übermittelt wird.

Nachdem Sie `suexec` auf diese Weise eingerichtet haben, werden CGI-Skripte und Server Side Includes unter einer alternativen User- und Group-ID ausgeführt, wenn eine der beiden folgenden Voraussetzungen erfüllt ist:

- ▶ Sie haben für den Server oder den jeweiligen virtuellen Host die Direktive `SuexecUserGroup` gesetzt.
- ▶ Das Skript befindet sich in der Website unter einem mittels `mod_userdir` eingebundenen Home-Verzeichnis (siehe Kapitel 8, *Weiterleitungen und Indizes*). In diesem Fall wird der Prozess unter der User- und Group-ID des jeweiligen Benutzers selbst ausgeführt.

`suexec` führt nacheinander folgende Sicherheitsüberprüfungen durch, bevor es ein Skript tatsächlich ausführt:

1. User- und Group-ID müssen gültig sein.
2. Das Wrapper-Programm `suexec` muss korrekt aufgerufen worden sein (darum kümmert sich Apache hinter den Kulissen; das Programm ist nicht für den Kommandozeilenaufruf gedacht)<sup>2</sup>.
3. Der User muss die Berechtigung besitzen, den SuEXEC-Wrapper aufzurufen; dies darf nur die User-ID, unter der Apache läuft (Option `--with-suexec-caller`).
4. Der Pfad des auszuführenden Skripts darf keine unsichere URL-Referenz sein, das heißt, sein Pfad darf nicht mit `/` oder `../` beginnen. Das Skript muss unterhalb der `DocumentRoot` oder in einem der festgelegten User-Verzeichnisse liegen.
5. Der `suexec`-User muss existieren.

---

<sup>2</sup> Es besitzt eine einzige dokumentierte Kommandozeilenoption: `-V` gibt für `root` die Optionen aus, mit denen es kompiliert wurde.

6. Auch die `suexec`-Group muss vorhanden sein.
7. Der `suexec`-User darf nicht `root` sein (dies würde das ganze Konzept ad absurdum führen).
8. Die numerische User-ID muss über dem einkompilierten Limit (`--with-suexec-uidmin`) liegen.
9. Auch die `suexec`-Group darf nicht `root` sein.
10. Die numerische Group-ID muss oberhalb des Wertes liegen, der mithilfe der Option `--with-suexec-gidmin` konfiguriert wurde.
11. Der Wechsel der User- und Group-ID für den `suexec`-Wrapper muss möglich sein.
12. Das angegebene Verzeichnis, in dem sich das auszuführende Skript befindet, muss vorhanden sein.
13. Das Verzeichnis des `suexec`-Skripts muss unter der Apache-DocumentRoot oder in einem gültigen Website-Verzeichnis unterhalb eines Home-Verzeichnisses liegen.
14. Das User-Verzeichnis darf nur Schreibrechte für den Eigentümer besitzen.
15. Das aufgerufene Skript selbst muss existieren.
16. Nur der Eigentümer darf über Schreibrechte an dem CGI/SSI-Skript verfügen.
17. Bei dem Skript selbst dürfen die `setuid`- bzw. `setgid`-Bits nicht gesetzt sein.
18. `suexec`-Benutzer und -Gruppe müssen Eigentümer des Skripts sein.
19. `PATH` und andere Umgebungsvariablen müssen sich vor dem Start des Skripts auf sichere Werte setzen lassen (siehe `--with-suexec-safepath`).
20. Das Skript muss sich tatsächlich ausführen lassen.

### **SuexecUserGroup**

Benutzer- und Gruppen-ID für SuEXEC-CGI-Skripte

<b>Seit Version</b>	2.0
<b>Modul</b>	<code>mod_suexec</code>
<b>Kontext</b>	Server, <VirtualHost>
<b>Syntax</b>	<code>SuexecUserGroup Benutzer Gruppe</code>
<b>Standardwert</b>	nicht gesetzt

Diese Direktive gibt die User- und Group-ID an, unter der `suexec`-CGI-Skripte ausgeführt werden sollen. Wenn Sie möchten, können Sie für jeden virtuellen Host einen anderen Benutzer und eine andere Gruppe festlegen. Die gewöhnlichen Apache-Child-Prozesse, die Anfragen beantworten, werden allerdings weiterhin unter der User- und Group-ID ausgeführt, die Sie mithilfe der Direktiven `User` und `Group` festgelegt haben (siehe Kapitel 6, *Grundkonfiguration*). Wenn Sie dies ändern möchten, können Sie das Multiprocessing-Modul `perchild` verwenden, das allerdings noch experimentell ist. Hier ein Verwendungsbeispiel:

```
SuexecUserGroup suser sgroup
```

### 18.3 mod\_security

Eine praktische Möglichkeit, die Sicherheit von Apache 2 weiter zu verbessern, ist die Installation des Moduls `mod_security` von **Ivan Ristic**. Dieser hat seine Erfahrungen mit der Apache-Sicherheit inzwischen zu dem empfehlenswerten Buch [RIST 2005] verarbeitet. Sie können das Modul auf der Website [www.mod-security.org](http://www.mod-security.org) herunterladen; auf der CD zum Buch finden Sie die zurzeit aktuelle stabile Version 1.9. Das Modul ist ein frei konfigurierbares Intrusion-Detection- und Filter-System für den Webserver.

Die Installation als DSO-Modul unter UNIX beschränkt sich auf eine einzige Zeile:

```
# /usr/local/apache2/bin/apxs -cia mod_security.c
```

Für Windows gibt es ein vorkompiliertes Binary, das Sie einfach nur in Ihr `modules`-Verzeichnis kopieren müssen.

In jedem Fall muss das DSO-Modul nun natürlich in der Konfigurationsdatei aktiviert werden. Das funktioniert nach dem üblichen Schema:

```
LoadModule security_module modules/mod_security.so
```

Das Modul definiert eine Reihe neuer Direktiven zur Überwachung und Abwehr von Angriffsversuchen. Eine vollständige Referenz würde zu weit führen; auf der CD zum Buch oder auf der `mod_security`-Website finden Sie die Originaldokumentation im PDF-Format.

Hier einige kommentierte Beispiele:

```
<IfModule mod_security.c>
  # Filtermechanismus einschalten
  SecFilterEngine On
```

```

# URL überprüfen (mehrere / oder ../ entfernen)
SecFilterCheckURLEncoding On

# Nur Bytes mit den angegebenen Werten erlauben
# Hier alle Bytes:
SecFilterForceByteRange 0 255
# Paranoia - nur ASCII; keine Steuerzeichen:
# SecFilterForceByteRange 32 127

# Nur verdächtige Anfragen protokollieren
SecAuditEngine RelevantOnly

# mod_security-Logdatei:
SecAuditLog logs/audit_log

# Body von POST-Anfragen überprüfen
SecFilterScanPOST On

# Verdächtige Anfragen protokollieren und
# mit Status 500 Internal Script Error verbieten
SecFilterDefaultAction "deny,log,status:500"
</IfModule>

```

Die nähere Beschäftigung mit `mod_security` lohnt sich. Vor allem die Log-Datei, die das Modul anlegt, sollten Sie regelmäßig auswerten.

## 18.4 Zusammenfassung

Dieses Kapitel kann nur einen ersten Überblick über das ernste und wichtige Thema Sicherheit geben; im ganzen Buch verteilt finden Sie weitere wichtige Hinweise, die die Sicherheit der einzelnen Features betreffen.

Ein gut abgesicherter Apache-Webserver nützt wenig, wenn er in einer schlecht konfigurierten Systemumgebung arbeiten muss. Beachten Sie daher bitte die Hinweise zur Systemsicherheit im ersten Abschnitt.

Anschließend sollten Sie sich um die Sicherheit von Apache selbst kümmern und Ihre Konfiguration noch einmal kritisch anhand der Hinweise im zweiten Abschnitt überprüfen. Zusätzlich existieren diverse Konfigurationsdirektiven, die den möglichen Spielraum für Angriffsversuche einschränken, indem sie den Ressourcenzugriff durch Apache begrenzen.

Schließlich lohnt sich auch die Beschäftigung mit dem externen Modul `mod_security`; es fügt ein eigenes Sicherheitskonzept mit integriertem Intrusion Detection System zum Apache-Webserver hinzu.

Damit ist nicht nur das Kapitel, sondern auch das Buch abgeschlossen – es folgen noch einige Anhänge mit nützlichen Informationen zum Nachschlagen. Ich hoffe, dass das Buch Ihnen nicht nur Nutzen für Ihre tägliche Administrations- und Entwicklungsarbeit mit Apache 2 gebracht, sondern auch ein wenig Spaß gemacht hat. Ich freue mich stets über Anregungen, Fragen und (konstruktive) Kritik; zögern Sie nicht, die Site zum Buch unter <http://buecher.lingoworld.de/apache2> zu besuchen.

## Index

- \$/, Perl-Spezialvariable 95
  - +/-Zeichen
    - bei *Installationslayouts* 164
  - .htaccess
    - Benutzerverzeichnisse* 383
    - DocumentRoot-Einstellungen* 302
    - Verzeichnisvoreinstellung* 301
  - .htaccess-Dateien 258
    - Erlaubte Direktiven festlegen* 296
    - Namen ändern* 293
    - Sinnvoller Einsatz* 260
  - .NET 725
  - /etc/hosts, Datei 32
  - :lange Zeilen in httpd.conf 243
  - <AuthnProviderAlias>, Container 473
  - <Directory>, Container 249
  - <DirectoryMatch>, Container 251
  - <Files>, Container 253
  - <FilesMatch>, Container 253
  - <IfDefine>, Container 254
  - <IfModule>, Container 255
  - <IfVersion>, Container 256
  - <Limit>, Container 255
  - <LimitExcept>, Container 256
  - <Location>, Container 252
  - <LocationMatch>, Container 252
  - <Perl>, Container 707
  - <Proxy>, Container 590
  - <ProxyMatch>, Container 591
  - <VirtualHost>, Container 248
- A**
- A Patchy Web Server 114
  - ab, Hilfsprogramm 227, 571
  - Accept, HTTP-Header 75
  - Accept-Charset, HTTP-Header 76
  - Accept-Encoding, HTTP-Header 76
  - AcceptFilter, Direktive 266
  - AcceptFilter, Direktive (1.3) 816
  - Accept-Language, HTTP-Header 77
  - AcceptMutex, Direktive 272
  - AcceptPathInfo, Direktive 642
  - Accept-Ranges, HTTP-Header 77
  - access.conf 239
  - AccessConfig, Direktive (1.3) 817
  - AccessFileName, Direktive 293
  - Action, Direktive 633
  - Active Server Pages (ASP) 121
  - ActivePerl 704
  - AddAlt, Direktive 401
  - AddAltByEncoding, Direktive 401
  - AddAltByType, Direktive 402
  - AddCharset, Direktive 331
  - AddDefaultCharset, Direktive 330
  - AddDescription, Direktive 402
  - AddEncoding, Direktive 333
  - AddHandler, Direktive 338
  - AddIcon, Direktive 398
  - AddIconByEncoding, Direktive 399
  - AddIconByType, Direktive 400
  - AddInputFilter, Direktive 744
  - AddLanguage, Direktive 335
  - AddModule, Direktive (1.3) 817
  - AddModuleInfo, Direktive 391
  - AddOutputFilter, Direktive 743
  - AddOutputFilterByType, Direktive 742
  - AddType, Direktive 324
  - Age, HTTP-Header 77
  - AJP-Proxy
    - Tomcat* 716
  - Alexandria 117
  - Alias 355
  - Alias, Direktive 356
  - AliasMatch, Direktive 357
  - Allow, Direktive 298
  - Allow, HTTP-Header 77
  - AllowCONNECT, Direktive 603
  - AllowEncodedSlashes, Direktive 643
  - AllowOverride, Direktive 296
  - Alternates, HTTP-Header 348
  - Anmeldung → Authentifizierung 413
  - Anonymous, Direktive 464
  - Anonymous\_Authoritative, Direktive 465
  - Anonymous\_LogEmail, Direktive 466
  - Anonymous\_NoUserID, Direktive 464
  - Anonymous\_VerifyEmail, Direktive 465
  - Ant 116
  - Apache
    - ab, Hilfsprogramm* 227
    - als Dienst (Windows)* 219
    - Apache Portable Runtime* 134
    - Apache.exe (Windows-Binary)* 217
    - Apache-Monitor (Windows)* 224
    - apxs, Hilfsprogramm* 227
    - Automatische Indizes* 392
    - Autostart (UNIX)* 213
    - beenden (UNIX)* 212

*beenden und neu starten* 209  
*CGI-Konfiguration* 621  
*Content Negotiation* 131  
*Dateiendungen* 322  
*dbmmange, Hilfsprogramm* 228  
*Demo-Website* 228  
*Dienst starten (Windows)* 220  
*dynamische Inhalte* 131  
*Fehlerbehandlung* 386  
*Filter* 130  
*Funktionen* 124  
*Geschichte* 113  
*Header-Manipulation* 132  
*Hilfsprogramme* 227  
*htccacheclean, Hilfsprogramm* 228  
*htdbm, Hilfsprogramm* 228  
*htdigest, Hilfsprogramm* 228  
*htpasswd, Hilfsprogramm* 228  
*httpd, Binary* 206  
*httpd.conf, Konfigurationsdatei* 239  
*httpd.conf-Notwendigkeiten* 230  
*httxtzdbm, Hilfsprogramm* 228, 375  
*Image-Maps* 404  
*Index, automatischer* 392  
*Installationsarten* 155  
*Installationsverzeichnisse* 161  
*Kompilieren* 156  
*Konfigurationsinformationen* 389  
*Laufzeitmodelle* 136  
*log\_server\_status, Hilfsprogramm* 228  
*logresolve, Hilfsprogramm* 228  
*Makefile* 159  
*manuell beenden (Windows)* 218  
*manuell neu starten (Windows)* 218  
*manuell starten (Windows)* 218  
*MIME-Konfiguration* 321  
*MIME-Type-Ermittlung* 131  
*MIME-Types* 323  
*mit apachectl steuern* 211  
*mit Perl installieren (Windows)* 199  
*Module* 141  
*Module installieren* 200  
*Multiprocessing-Module (MPM)* 136  
*neu starten (UNIX)* 212  
*Performance* 127  
*rotatelog, Hilfsprogramm* 228  
*Sicherheit* 128  
*Skalierbarkeit* 128  
*Softwarelizenz* 119  
*split\_logfile, Hilfsprogramm* 228  
*SSL* 132  
*Stabilität* 128  
*starten (UNIX)* 212  
*Statusinformationen* 389  
*steuern (UNIX)* 205  
*steuern (Windows)* 217  
*steuern mit kill* 210  
*suexec, Hilfsprogramm* 228  
*technische Details* 127  
*Test mit einfacher Site* 229  
*testen* 228  
*testen mit ps* 205  
*unterstützte Betriebssysteme* 125  
*URL-Manipulation* 130  
*User- und Group-ID* 269  
*Verbreitung* 115, 127  
*Versionierung* 128  
*Versionsgeschichte* 115  
*virtuelle Hosts* 132  
*Zeichensatzeinstellungen* 329  
*Zugriffsbeschränkung* 131  
Apache 1.3 815  
Apache 2.2, Neuerungen 133  
  *Event-MPM* 140  
Apache Benchmark → *ab*, Hilfsprogramm 571  
Apache Group  
  *Mitwirkende* 114  
Apache Portable Runtime (APR) 116, 134  
  *Einsatzgebiete (außer HTTPD)* 136  
  *Funktionsumfang* 135  
Apache Software Foundation (ASF) 116  
  *Incubator* 117  
Apache, Installationslayout 164  
Apache.exe, Windows-Binary 217  
apachectl, Steuerskript 211  
ApacheMon, Steuer-Tool 226  
Apache-Monitor (Windows) 224  
  *automatisch starten* 224  
Apache-Negotiation-Algorithmus 345  
Apache-Softwarelizenz  
  *Version 2.0* 120  
APR → Apache Portable Runtime (APR) 116, 134  
apxs, Hilfsprogramm 200, 227  
A-Record (DNS) 40  
ARPANet 21  
ASP → Active Server Pages (ASP) 121  
ASP.NET 725  
ASP.NET einbinden 725  
AssignUserID, Direktive 565  
Asymmetrische Verschlüsselung 480  
AuthAuthoritative, Direktive 429

AuthBasicAuthoritative, Direktive 430  
 AuthBasicProvider, Direktive 426  
 AuthDBDUserPWQuery, Direktive 472  
 AuthDBDUserRealmQuery, Direktive 472  
 AuthDBMAuthoritative, Direktive 448  
 AuthDBMGroupFile, Direktive 446  
 AuthDBMType, Direktive 447  
 AuthDBMUserFile, Direktive 445  
 AuthDefaultAuthoritative, Direktive 475  
 AuthDigestAlgorithm, Direktive 436  
 AuthDigestDomain, Direktive 437  
 AuthDigestEnableQueryStringHack, Variable 432  
 AuthDigestFile, Direktive 435  
 AuthDigestGroupFile, Direktive 436  
 AuthDigestNcCheck, Direktive 439  
 AuthDigestNonceFormat, Direktive 439  
 AuthDigestNonceLifetime, Direktive 438  
 AuthDigestProvider, Direktive 434  
 AuthDigestQop, Direktive 439  
 AuthDigestShmemSize, Direktive 438  
 Authentifizierung  
   *anonyme* 463  
   *auf Benutzerseite* 419  
   *Basic-Authentifizierung* 424  
   *Core-Direktiven* 420  
   *DBM-Dateien* 440  
   *Digest-Authentifizierung* 431  
   *Grundbegriffe* 413  
   *htpasswd* 424  
   *in Apache 2.0* 414  
   *in Apache 2.2* 415  
   *Konfigurationsbeispiel* 417  
   *mod\_auth* 424  
   *mod\_auth\_anon* 463  
   *mod\_auth\_digest* 431  
   *mod\_auth\_ldap* 449  
   *Reihenfolgenproblem* 429  
   *SQL-Datenbank* 471  
   *Vergleich zur Adresskontrolle* 413  
 AuthGroupFile, Direktive 428  
 AuthLDAPAuthoritative, Direktive 456–457  
 AuthLDAPBindDN, Direktive 452  
 AuthLDAPBindPassword, Direktive 453  
 AuthLDAPCharsetConfig, Direktive 453  
 AuthLDAPCompareDNOnServer, Direktive 454  
 AuthLDAPDereferenceAliases, Direktive 454  
 AuthLDAPEnabled, Direktive 457  
 AuthLDAPFrontPageHack, Direktive 457  
 AuthLDAPGroupAttribute, Direktive 455  
 AuthLDAPGroupAttributesDN, Direktive 455  
 AuthLDAPRemoteUserIsDN, Direktive 456  
 AuthLDAPUrl, Direktive 450  
 AuthName, Direktive 421  
 Authorization, HTTP-Header 77  
 AuthType, Direktive 421  
 AuthUserFile, Direktive 427  
 AuthzDBMAuthoritative, Direktive 448  
 AuthzDBMType, Direktive 447  
 AuthzDefaultAuthoritative, Direktive 475  
 AuthzGroupFileAuthoritative, Direktive 431  
 AuthzOwnerAuthoritative, Direktive 474  
 AuthzUserAuthoritative, Direktive 430  
 automatisch starten  
   *chkconfig* 214  
   *System V Init* 213  
   *UNIX* 213  
 awk 190

**B**

BalancerMember, Direktive 600  
 Banana Ware 129  
 Behlendorf, Brian 114  
 Benutzerverzeichnisse  
   *veröffentlichen* 383  
 BeOS, Installationslayout 168  
 beos, MPM 140  
 Berners-Lee, Tim 113  
 Betriebssysteme  
   *Apache-Unterstützung* 125  
 BindAddress, Direktive (1.3) 818  
 BIND-Nameserver 36  
   *A-Record* 40  
   *CNAME-Record* 41  
   *Installation* 36  
   *Konfiguration* 37  
   *MX-Record* 42

- NS-Record* 41
- PTR-Record* 40
- Reverse-Lookup-Zone* 38
- SOA-Record* 40
- Zonendaten-Dateien* 39
- Zonendefinition* 37
- Booten
  - BSD-Startskripte* 214
  - System V Init* 213
- BrowserMatch, Direktive 640
- BrowserMatchNoCase, Direktive 642
- BS2000Account, Direktive 273
- BSDI, Installationslayout 169
- BSD-Softwarelizenz 120
- BSD-Startskripte 214
- BufferedLogs, Direktive 528
- C**
- Cache 603
  - aufräumen* 615
  - htcacheclean, Hilfsprogramm* 615
  - Konfigurationsbeispiele* 604
- Cache-Control, HTTP-Header 78
- CacheDefaultExpire, Direktive 606
- CacheDirLength, Direktive 611
- CacheDirLevels, Direktive 612
- CacheDisable, Direktive 606
- CacheEnable, Direktive 605
- CacheFile, Direktive 577
- CacheIgnoreCacheControl, Direktive 607
- CacheIgnoreNoLastMod, Direktive 607
- CacheLastModifiedFactor, Direktive 607
- CacheMaxExpire, Direktive 608
- CacheMaxFileSize, Direktive 611
- CacheMinFileSize, Direktive 611
- CacheNegotiatedDocs, Direktive 350
- CacheRoot, Direktive 610
- CacheSize, Direktive 610
- CacheStoreNoStore, Direktive 609
- CacheStorePrivate, Direktive 609
- ccTLDs, Tabelle → Länder-Top-Level-Domains 881
- CERN httpd 114
- CERN-Meta-Dateien 319
- CGI → Common Gateway Interface (CGI)
  - Non-parsed Header (NPH)* 319
  - SuEXEC* 808
- CGI.pm, Perl-Modul 647
- CGI-Methoden 660
  - checkbox\_group(), Methode* 665
  - cookie(), Methode* 651, 661
  - end\_form(), Methode* 664
  - end\_html(), Methode* 652, 663
  - filefield(), Methode* 664
  - Formularmethoden* 663
  - Formularpraxis* 655
  - Header einstellen* 650
  - header(), Methode* 660
  - HTML-Erzeugung* 650
  - Import-Referenz* 658
  - param(), Methode* 648
  - password\_field(), Methode* 664
  - popup\_menu(), Methode* 664
  - Pragmata* 659
  - prozedural* 647
  - radio\_group(), Methode* 665
  - redirect(), Methode* 661
  - reset(), Methode* 666
  - self\_url(), Methode* 663
  - start\_form(), Methode* 663
  - start\_html(), Methode* 651, 661
  - submit(), Methode* 665
  - textfield(), Methode* 664
  - url(), Methode* 663
- CGI::Carp, Perl-Modul 647
- CGI::Pretty, Perl-Modul 648
- cgi-bin, Verzeichnis 622
- CGICommandArgs, Direktive (1.3) 818
- CGIMapExtension, Direktive 632
- cgi-script, Handler 336, 627
- CGI-Skripte
  - Erzwingen bei GET* 61
  - vs. CGI-Programme* 620
- CharsetDefault, Direktive 755
- CharsetOptions, Direktive 755
- CharsetSourceEnc, Direktive 754
- checkbox\_group(), CGI.pm-Methode 665
- CheckSpelling, Direktive 389
- child\_init, Hook 777
- ChildPerUserID, Direktive 283
- chkconfig (für Apache-Autostart) 214
- chmod
  - CGI ausführbar machen* 623
- chroot-Umgebung 800
- CIDR 26
- ClearModuleList, Direktive (1.3) 819
- CLF → Common Log Format 519
- Client-Anfrage (HTTP) 55
- CNAME-Record (DNS) 41

- Cocoon 116
- ColdFusion MX 725
- Combined Log Format 519
- Common Gateway Interface (CGI) 619
  - CGI.pm, Perl-Modul 647
  - CGI-Skript/-Programm 620
  - Entwicklung 619
  - Formulardaten manuell einlesen 645
  - Grundlagen 620
  - in DocumentRoot-Verzeichnissen 625
  - Konfiguration in Apache 621
  - objektorientiert 648
  - plattformspezifische Einstellung 630
  - Programmierung 643
  - Shebang 623
  - Skripte ausführbar machen 623
  - Umgebungsvariablen 634
  - Verzeichnisse für 622
  - Zuordnung unter Windows 623
- Common Log Format 519
- Concurrency
  - Nebenläufigkeit 137
- config, SSI-Element 731
- config.layout 172
- configure
  - PREFIX 161
- configure, Build-Einstellungen 159
  - Optionen 160
- configure, Build-Optionen
  - disable-mods-shared 178
  - disable-modules 177
  - enable-mods-shared 177
  - enable-modules 177
  - enable-so 176
  - Umgebungsvariablen 187
  - weitere Einstellungen 184
- CONNECT, HTTP-Methode 66
- Connection, HTTP-Header 79
- Container
  - <AuthnProviderAlias> 473
  - <Directory> 249
  - <DirectoryMatch> 251
  - <Files> 253
  - <FilesMatch> 253
  - <IfDefine> 254
  - <IfModule> 255
  - <IfVersion> 256
  - <Limit> 255
  - <LimitExcept> 256
  - <Location> 252
  - <LocationMatch> 252
  - <Perl> 707
  - <Proxy> 590
  - <ProxyMatch> 591
  - <VirtualHost> 248
    - für Direktiven 247
    - Verschachtelung 258
- Content Negotiation 131
- ContentDigest, Direktive 307
- Content-Encoding, HTTP-Header 79
- Content-Language, HTTP-Header 80
- Content-Length, HTTP-Header 80
- Content-Location, HTTP-Header 80
- Content-MD5, HTTP-Header 80
  - mit Apache setzen 307
- Content-Negotiation 339
  - Apache-Negotiation-Algorithmus 345
  - Direktiven 349
  - MultiViews 344
  - servergesteuerte 340
  - transparente 347
  - Type-Maps 341
- Content-Range, HTTP-Header 80
- Content-Type, HTTP-Header 81
- cookie(), CGI.pm-Methode 651, 661
- Cookie, HTTP-Header 81
- CookieDomain, Direktive 536
- CookieExpires, Direktive 536
- CookieLog, Direktive 529
- CookieName, Direktive 537
- Cookies
  - in PHP 693
  - mit CGI.pm 661
- CookieStyle, Direktive 537
- CookieTracking, Direktive 538
- CoreDumpDirectory, Direktive 274
- Crackertools 800
- cronolog 553
- CustomLog, Direktive 529

**D**

- Darwin, Installationslayout 166
- Date, HTTP-Header 81
- Datei-Container 249
- Dateien
  - .htaccess 258
- Dateiendungen 322
- Datenbankverbindungen (Apache 2.2) 467
- Datum und Uhrzeit
  - strftime() 527
- DAV → WebDAV
- Dav, Direktive 768
- DavDepthInfinity, Direktive 769

- DavGenericLockDB, Direktive 770
- DavLockDB, Direktive 770
- DavMinTimeout, Direktive 769
- DB 116
- DBDExpTime, Direktive 470
- DBDKeep, Direktive 470
- DBDMax, Direktive 470
- DBDMin, Direktive 469
- DBDPersist, Direktive 468
- DBDPrepareSQL, Direktive 469
- DBDriver, Direktive 467
- DBI, Perl-Modul 706
- DBM-Dateien
  - für RewriteMaps 374
  - zur Authentifizierung 440
- dbmmanage, Hilfsprogramm 228, 441
- DBParams, Direktive 468
- DDN-Schichtenmodell 22
- Debian, Installationslayout 171
- default-handler 336
- DefaultIcon, Direktive 400
- DefaultLanguage, Direktive 334
- DefaultType, Direktive 323
- DEFLATE, Filter 750
- DeflateBufferSize, Direktive 751
- DeflateCompressionLevel, Direktive 751
- DeflateFilterNote, Direktive 752
- DeflateMemLevel, Direktive 753
- DeflateWindowSize, Direktive 753
- Deinstallation (Windows) 199
- DELETE, HTTP-Methode 64
- Deny, Direktive 300
- Dienst
  - Apache als 219
  - deinstallieren 220
  - installieren 219
  - starten 220
- dig, Dienstprogramm 46
- Digest-Authentifizierung 431
  - Internet Explorer 432
- DirectoryIndex, Direktive 290
- DirectorySlash, Direktive 292
- Direktiven
  - <AuthnProviderAlias>, Container 473
  - <Directory>, Container 249
  - <DirectoryMatch>, Container 251
  - <Files>, Container 253
  - <FilesMatch>, Container 253
  - <IfDefine>, Container 254
  - <IfModule>, Container 255
  - <IfVersion>, Container 256
  - <Limit>, Container 255
  - <LimitExcept>, Container 256
  - <Location>, Container 252
  - <LocationMatch>, Container 252
  - <Perl>, Container 707
  - <Proxy>, Container 590
  - <ProxyMatch>, Container 591
  - <VirtualHost>, Container 248
  - AcceptFilter 266
  - AcceptFilter (1.3) 816
  - AcceptMutex 272
  - AcceptPathInfo 642
  - AccessConfig (1.3) 817
  - AccessFileName 293
  - Action 633
  - AddAlt 401
  - AddAltByEncoding 401
  - AddAltByType 402
  - AddCharset 331
  - AddDefaultCharset 330
  - AddDescription 402
  - AddEncoding 333
  - AddHandler 338
  - AddIcon 398
  - AddIconByEncoding 399
  - AddIconByType 400
  - AddInputFilter 744
  - AddLanguage 335
  - AddModule (1.3) 817
  - AddModuleInfo 391
  - AddOutputFilter 743
  - AddOutputFilterByType 742
  - AddType 324
  - Alias 356
  - AliasMatch 357
  - AllowCONNECT 603
  - AllowEncodedSlashes 643
  - AllowOverride 296
  - Anonymous 464
  - Anonymous\_Authoritative 465
  - Anonymous\_LogEmail 466
  - Anonymous\_MustGiveEmail 465
  - Anonymous\_NoUserID 464
  - Anonymous\_VerifyEmail 465
  - Apache 1.3 816
  - AssignUserID 565
  - AuthAuthoritative 429
  - AuthBasicAuthoritative 430
  - AuthBasicProvider 426
  - AuthDBDUserPWQuery 472
  - AuthDBDUserRealmQuery 472
  - AuthDBMAuthoritative 448

*AuthDBMGroupFile* 446  
*AuthDBMType* 447  
*AuthDBMUserFile* 445  
*AuthDefaultAuthoritative* 475  
*AuthDigestAlgorithm* 436  
*AuthDigestDomain* 437  
*AuthDigestFile* 435  
*AuthDigestGroupFile* 436  
*AuthDigestNcCheck* 439  
*AuthDigestNonceFormat* 439  
*AuthDigestNonceLifetime* 438  
*AuthDigestProvider* 434  
*AuthDigestQop* 439  
*AuthDigestShmemSize* 438  
*AuthGroupFile* 428  
*AuthLDAPAuthoritative* 456–457  
*AuthLDAPBindDN* 452  
*AuthLDAPBindPassword* 453  
*AuthLDAPCharsetConfig* 453  
*AuthLDAPCompareDNOnServer* 454  
*AuthLDAPDereferenceAliases* 454  
*AuthLDAPEnabled* 457  
*AuthLDAPFrontPageHack* 457  
*AuthLDAPGroupAttribute* 455  
*AuthLDAPGroupAttributeIsDN* 455  
*AuthLDAPRemoteUserIsDN* 456  
*AuthLDAPUrl* 450  
*AuthName* 421  
*AuthType* 421  
*AuthUserFile* 427  
*AuthzDBMAuthoritative* 448  
*AuthzDBMType* 447  
*AuthzDefaultAuthoritative* 475  
*AuthzGroupFileAuthoritative* 431  
*AuthzOwnerAuthoritative* 474  
*AuthzUserAuthoritative* 430  
*BalancerMember* 600  
*BindAddress* (1.3) 818  
*BrowserMatch* 640  
*BrowserMatchNoCase* 642  
*BS2000Account* 273  
*BufferedLogs* 528  
*CacheDefaultExpire* 606  
*CacheDirLength* 611  
*CacheDirLevels* 612  
*CacheDisable* 606  
*CacheEnable* 605  
*CacheFile* 577  
*CacheIgnoreCacheControl* 607  
*CacheIgnoreNoLastMod* 607  
*CacheLastModifiedFactor* 607  
*CacheMaxExpire* 608  
*CacheMaxFileSize* 611  
*CacheMinFileSize* 611  
*CacheNegotiatedDocs* 350  
*CacheRoot* 610  
*CacheSize* 610  
*CacheStoreNoStore* 609  
*CacheStorePrivate* 609  
*CGICommandArgs* (1.3) 818  
*CGIMapExtension* 632  
*CharsetDefault* 755  
*CharsetOptions* 755  
*CharsetSourceEnc* 754  
*CheckSpelling* 389  
*ChildPerUserID* 283  
*ClearModuleList* (1.3) 819  
*Container* 247  
*ContentDigest* 307  
*CookieDomain* 536  
*CookieExpires* 536  
*CookieLog* 529  
*CookieName* 537  
*CookieStyle* 537  
*CookieTracking* 538  
*CoreDumpDirectory* 274  
*CustomLog* 529  
*Dateendungen als Wert* 244  
*Dav* 768  
*DavDepthInfinity* 769  
*DavGenericLockDB* 770  
*DavLockDB* 770  
*DavMinTimeout* 769  
*DBDExpTime* 470  
*DBDKeep* 470  
*DBDMax* 470  
*DBDMin* 469  
*DBDPersist* 468  
*DBDPrepareSQL* 469  
*DBDriver* 467  
*DBParams* 468  
*DefaultIcon* 400  
*DefaultLanguage* 334  
*DefaultType* 323  
*DeflateBufferSize* 751  
*DeflateCompressionLevel* 751  
*DeflateFilterNote* 752  
*DeflateMemLevel* 753  
*DeflateWindowSize* 753  
*Deny* 300  
*DirectoryIndex* 290  
*DirectorySlash* 292  
*DocumentRoot* 290  
*DumpIOInput* 535

*DumpIOOutput* 535  
*EnableExceptionHook* 807  
*EnableMMAP* 574  
*EnableSendfile* 575  
*ErrorDocument* 387  
*ErrorLog* 521  
*Example* 775  
*ExpiresActive* 315  
*ExpiresByType* 317  
*ExpiresDefault* 316  
*ExtendedStatus* 391  
*ExtFilterDefine* 756  
*ExtFilterOptions* 758  
*feste Werte* 245  
*FileETag* 308  
*FilterChain* 749  
*FilterDeclare* 746  
*FilterProtocol* 748  
*FilterProvider* 747  
*FilterTrace* 750  
*ForceLanguagePriority* 351  
*ForceType* 324  
*ForensicLog* 534  
*für Content-Negotiation* 349  
*für virtuelle Hosts* 247  
*GracefulShutdownTimeout* 268  
*Group* 271  
*Header* 309  
*HeaderName* 403  
*HolidayMonth* 777  
*HolidayURL* 777  
*HostnameLookups* 523  
*IASPIFakeAsync* 723  
*IdentityCheck* 524  
*ImapBase* 407  
*ImapDefault* 408  
*ImapMenu* 408  
*in 1.3 nicht vorhandene* 821  
*Include* 260  
*IndexIgnore* 397  
*IndexOptions* 394  
*IndexOrderDefault* 397  
*ISAPIAppendLogToErrors* 722  
*ISAPIAppendLogToQuery* 723  
*ISAPICacheFile* 723  
*ISAPILogNotSupported* 724  
*ISAPIReadAheadBuffer* 724  
*KeepAlive* 263  
*KeepAliveTimeout* 264  
*Kontextangabe* 246  
*Kontexte* 247  
*LanguagePriority* 351  
*LDAPCacheEntries* 459  
*LDAPCacheTTL* 460  
*LDAPOpCacheEntries* 460  
*LDAPOpCacheTTL* 461  
*LDAPSharedCacheFile* 461  
*LDAPSharedCacheSize* 462  
*LDAPTrustedCA* 462  
*LDAPTrustedCAType* 462  
*LimitInternalRecursion* 803  
*LimitRequestBody* 804  
*LimitRequestFields* 804  
*LimitRequestFieldSize* 804  
*LimitRequestLine* 805  
*LimitXMLRequestBody* 805  
*Listen* 265  
*ListenBackLog* 271  
*LoadFile* 267  
*LoadModule* 267  
*LockFile* 274  
*LogFormat* 531  
*LogLevel* 522  
*MaxClients* 275  
*MaxKeepAliveRequests* 264  
*MaxMemFree* 275  
*MaxRequestsPerChild* 276  
*MaxRequestsPerThread* 282  
*MaxSpareServers* 281  
*MaxSpareThreads* 276  
*MaxThreads* 282  
*MaxThreadsPerChild* 284  
*MCacheMaxObjectCount* 613  
*MCacheMaxObjectSize* 613  
*MCacheMaxStreamingBuffer* 614  
*MCacheMinObjectSize* 613  
*MCacheRemovalAlgorithm* 614  
*MCacheSize* 612  
*MetaDir* 320  
*MetaFiles* 320  
*MetaSuffix* 320  
*MimeMagicFile* 327  
*MinSpareServers* 281  
*MinSpareThreads* 277  
*MMapFile* 576  
*ModMimeUsePathInfo* 327  
*Modulangabe* 246  
*mögliche Werte* 243  
*MultiViewsMatch* 349  
*NameVirtualHost* 563  
*NoProxy* 592  
*numerische Werte* 244  
*NumServers* 284  
*NWSSLTrustedCerts* 514

*ohne Wert* 245  
*On|Off* 243  
*Options* 294  
*Order* 297  
*PerlModule* 701  
*Pfadangaben* 244  
*PidFile* 269  
*plattformspezifische* 268  
*Port (1.3)* 819  
*ProtocolDaytime* 790  
*ProtocolEcho* 772  
*ProtocolReqCheck (1.3)* 820  
*ProxyBadHeader* 593  
*ProxyBlock* 594  
*ProxyDomain* 594  
*ProxyErrorDomain* 595  
*ProxyIOBufferSize* 595  
*ProxyMaxForwards* 596  
*ProxyPass* 596  
*ProxyPassReverse* 598  
*ProxyPassReverseCookieDomain* 599  
*ProxyPassReverseCookiePath* 600  
*ProxyPreserveHost* 601  
*ProxyReceiveBufferSize* 601  
*ProxyRemote* 592  
*ProxyRemoteMatch* 592  
*ProxyRequests* 591  
*ProxyTimeout* 602  
*ProxyVia* 602  
*ReadmeName* 404  
*Redirect* 358  
*RedirectMatch* 360  
*RedirectPermanent* 362  
*RedirectTemp* 363  
*Reguläre Ausdrücke* 245  
*RemoveCharset* 332  
*RemoveEncoding* 334  
*RemoveHandler* 339  
*RemoveInputFilter* 745  
*RemoveLanguage* 335  
*RemoveOutputFilter* 744  
*RemoveType* 325  
*RequestHeader* 314  
*Require* 422  
*ResourceConfig (1.3)* 820  
*RewriteBase* 370  
*RewriteCond* 369  
*RewriteEngine* 364  
*RewriteLock* 379  
*RewriteLog* 538  
*RewriteLogLevel* 539  
*RewriteMap* 371  
*RewriteOptions* 379  
*RewriteRule* 365  
*RLimitCPU* 806  
*RLimitMEM* 806  
*RLimitNPROC* 807  
*Satisfy* 423  
*ScoreBoardFile* 277  
*Script* 634  
*ScriptAlias* 622  
*ScriptAliasMatch* 624  
*ScriptInterpreterSource* 630  
*ScriptLog* 628  
*ScriptLogBuffer* 628  
*ScriptLogSize* 629  
*ScriptSockSize* 629  
*SecureListen* 515  
*SendBufferSize* 278  
*ServerAdmin* 285  
*ServerAlias* 563  
*Server-Kontext* 247  
*ServerLimit* 278  
*ServerName* 286  
*ServerPath* 564  
*ServerRoot* 262  
*ServerSignature* 289  
*ServerTokens* 287  
*ServerType (1.3)* 820  
*SetEnvIf* 639  
*SetEnvIfNoCase* 640  
*SetHandler* 337  
*SetInputFilter* 743  
*SetOutputFilter* 741  
*SSIEndTag* 737  
*SSIErrorMsg* 738  
*SSIStartTag* 737  
*SSITimeFormat* 738  
*SSIUndefinedEcho* 739  
*SSLCACertificateChainFile* 494  
*SSLCACertificateFile* 491  
*SSLCACertificatePath* 492  
*SSLCADNRequestFile* 492  
*SSLCADNRequestPath* 493  
*SSLCARevocationFile* 493  
*SSLCARevocationPath* 494  
*SSLCertificateFile* 495  
*SSLCertificateKeyFile* 495  
*SSLCipherSuite* 496  
*SSLCryptoDevice* 499  
*SSLEngine* 500  
*SSLHonorCipherOrder* 501  
*SSLMutex* 501  
*SSLOptions* 502

SSLPassPhraseDialog 503  
 SSLProtocol 504  
 SSLProxyCACertificateFile 510  
 SSLProxyCACertificatePath 511  
 SSLProxyCARevocationFile 511  
 SSLProxyCARevocationPath 511  
 SSLProxyCipherSuite 512  
 SSLProxyEngine 512  
 SSLProxyMachineCertificateFile 512  
 SSLProxyMachineCertificatePath 513  
 SSLProxyProtocol 513  
 SSLProxyVerify 513  
 SSLProxyVerifyDepth 514  
 SSLRandomSeed 505  
 SSLRequire 506  
 SSLRequireSSL 508  
 SSLSessionCache 508  
 SSLSessionCacheTimeout 509  
 SSLVerifyClient 509  
 SSLVerifyOpen 510  
 Standardwertangabe 247  
 StartServers 279  
 StartThreads 279  
 String-Werte 244  
 SuexecUserGroup 810  
 Syntaxangabe 247  
 Syntaxschema (in diesem Buch) 246  
 ThreadLimit 280  
 ThreadsPerChild 280  
 ThreadStackSize 283  
 TransferLog 533  
 TypesConfig 326  
 UnsetEnv 638  
 UseCanonicalName 286  
 User 269  
 UserDir 384  
 Versionsangabe 246  
 Verzeichniscontainer 249  
 VirtualDocumentRoot 567  
 VirtualDocumentRootIP 568  
 VirtualScriptAlias 568  
 VirtualScriptAliasIP 569  
 wichtigste im Überblick 233  
 Win32DisableAcceptEx 284  
 XBitHack 739  
 zur Grundkonfiguration 261  
 Direktiven, Allow 298  
 Direktiven, TimeOut 263  
 --disable-mods-shared, configure-  
 Option 178  
 --disable-modules, configure-Option  
 177  
 DNS → Domain Name System (DNS)  
 33  
 DNS-Server 35  
 DocumentRoot, Direktive 290  
 Domain Name System (DNS) 33  
   *BIND-Nameserver* 36  
   *Funktionsweise* 33  
   *Nameserver* 35  
   *Round-Robin-Verfahren* 41  
 Doxygen 778  
 DSO → Dynamic Shared Objects  
 (DSO) 176  
 DumpIOInput, Direktive 535  
 DumpIOOutput, Direktive 535  
 Dynamic Shared Objects (DSO) 176  
**E**  
 echo, SSI-Element 732  
 Eddie, Load-Balancer 581  
 Eigener Webserver (Perl)  
   *Accept-Schleife* 96  
   *Benutzerdokumentation* 108  
   *Client-Anfrage* 96  
   *Dateigröße ermitteln* 100  
   *Datumsformate* 97  
   *Implementierungsdetails* 93  
   *Kommandozeilenparameter* 95  
   *Logging* 101  
   *MIME-Type ermitteln* 100  
   *MIME-Types* 94  
   *Projektanforderungen* 92  
   *Quellcode* 102  
   *Server-Antwort* 99  
   *Socket-Erzeugung* 95  
   *Startseite* 97  
 eigener Webserver (Perl) 92  
 Einwegverschlüsselung 480  
 EnableExceptionHook, Direktive 807  
 EnableMMAP, Direktive 574  
 --enable-mods-shared, configure-  
 Option 177  
 --enable-modules, configure-Option  
 177  
 EnableSendfile, Direktive 575  
 --enable-so, configure-Option 176  
 Encoding → MIME-Codierung 332  
 end\_form(), CGI.pm-Methode 664  
 end\_html(), CGI.pm-Methode 652,  
 663  
 Engelschall, Ralf S. 363, 479  
 ErrorDocument, Direktive 387  
 ErrorLog, Direktive 521

- ETag, HTTP-Header 82
  - mit Apache setzen* 308
- event, MPM 140
- Example, Direktive 775
- example-handler, Handler 775
- exec, SSI-Element 733
- Expect, HTTP-Header 82
- Expires, HTTP-Header 83
  - mit Apache setzen* 315
- ExpiresActive, Direktive 315
- ExpiresByType, Direktive 317
- ExpiresDefault, Direktive 316
- ExtendedStatus, Direktive 391
- ExtFilterDefine, Direktive 756
- ExtFilterOptions, Direktive 758

**F**

- Fancy-Index 393
- Fehlerbehandlung 386
- Fielding, Roy T. 114
- FileETag, Direktive 308
- filefield(), CGI.pm-Methode 664
- Filter 130, 740
  - colors, eigenes Beispiel* 763
  - DEFLATE* 750
  - Direktiven für* 740
  - externe* 756
  - INCLUDES* 730
  - Perl-Beispiele* 759
  - sourceview, eigenes Beispiel* 762
  - txt2html, eigenes Beispiel* 761
  - x4u, eigenes Beispiel* 760
- Filter Chain 741, 745
- FilterChain, Direktive 749
- FilterDeclare, Direktive 746
- FilterProtocol, Direktive 748
- FilterProvider, Direktive 747
- FilterTrace, Direktive 750
- Firewalls 799
- fixups, Hook 777
- flastmod, SSI-Element 734
- ForceLanguagePriority, Direktive 351
- ForceType, Direktive 324
- ForensicLog, Direktive 534
- forensische Logdateien 533
- Forking-Server 137
- Forward-Proxy 585
  - Konfigurationsbeispiele* 587
  - mit Cache* 604
- FreeBSD, Installationslayout 171
- freie Software 113
  - Versionierung* 129

- From, HTTP-Header 83
- fsize, SSI-Element 734

**G**

- Generische Top-Level-Domains,
  - Tabelle 881
- GET, HTTP-Methode 56
  - CGI-Ausführung erzwingen* 61
  - Formularversand* 59
- Gleichzeitigkeit → Nebenläufigkeit 137
- GNU General Public License (GPL) 119
- GNU, Installationslayout 165
- GPL → GNU General Public License (GPL) 119
- GracefulShutdownTimeout, Direktive 268
- Group, Direktive 271
- gTLDs, Tabelle → generische Top-Level-Domain 881
- Gültigkeitsdauer (Dokumente) 315

**H**

- Hagberg, Eric 114
- Handler 336
  - cgi-script* 336, 627
  - default-handler* 336
  - example-handler* 775
  - imap-file* 336, 405
  - ldap-status* 459
  - send-as-is* 318, 336
  - server-info* 336, 390
  - server-parsed* 336
  - server-status* 337, 390
  - type-map* 337, 341
- handler, Hook 777
- Hartill, Rob 114
- HEAD, HTTP-Methode 59
- header(), CGI.pm-Methode 660
- Header, Direktive 309
- header\_parser, Hook 777
- HeaderName, Direktive 403
- Hilfsprogramme
  - dbmmanage* 441
  - htcacheclean* 615
  - htdbm* 444
  - htdigest* 432
  - htpasswd* 424
- Hilfsprogramme (mit Apache geliefert) 227
- HolidayMonth, Direktive 777
- HolidayURL, Direktive 777
- Home-Verzeichnisse → Benutzerverzeichnisse 383

- Hooks 776
  - child\_init* 777
  - fixups* 777
  - handler* 777
  - header\_parser* 777
  - insert\_filter* 777
  - map\_to\_storage* 777
  - open\_logs* 777
  - post\_config* 777
  - pre\_config* 777
  - translate\_name* 777
  - type\_checker* 777
- Host, HTTP-Header 83
- HostnameLookups, Direktive 523
- htcacheclean, Hilfsprogramm 228, 615
- htdbm, Hilfsprogramm 228, 444
- htdigest, Hilfsprogramm 228, 432
- htpasswd, Hilfsprogramm 228, 424
- HTTP-Anfrage 55
- httpd, Binary 206
  - Kommandozeilenoptionen* 207
- httpd.conf
  - Abschnitte* 243
  - Aufbau* 239
  - bedingte Anweisungen* 254
  - CGI-Einstellungen* 621
  - Container* 247
  - Container für virtuelle Hosts* 247
  - Container-Verschachtelung* 258
  - Direktiven-Syntaxschema* 246
  - Direktivenwerte* 243
  - Entwicklung* 239
  - Erstellen* 239
  - Handler* 336
  - Konfigurationsdatei-Import* 260
  - Kontexte* 247
  - lange Zeilen trennen* 243
  - Notwendiges im Überblick* 230
  - plattformsspezifische Einstellungen* 268
  - Server-Kontext* 247
  - Syntax* 242
  - Verzeichniscontainer* 249
  - Verzeichnisoptionen* 294
  - wichtigste Direktiven* 233
- HTTP-Header
  - Accept* 75
  - Accept-Charset* 76
  - Accept-Encoding* 76
  - Accept-Language* 77
  - Accept-Ranges* 77
  - Age* 77
  - Allow* 77
  - Alternates* 348
  - Authorization* 77
  - Cache-Control* 78
  - Connection* 79
  - Content-Encoding* 79
  - Content-Language* 80
  - Content-Length* 80
  - Content-Location* 80
  - Content-MD5* 80, 307
  - Content-Range* 80
  - Content-Type* 81
  - Cookie* 81
  - Date* 81
  - ETag* 82, 308
  - Expect* 82
  - Expires* 83, 315
  - From* 83
  - Host* 83
  - If-Match* 84
  - If-Modified-Since* 84
  - If-None-Match* 84
  - If-Range* 85
  - If-Unmodified-Since* 85
  - Last-Modified* 85
  - Location* 85
  - Manipulation mit Apache* 307
  - Max-Forwards* 86
  - mit Apache modifizieren* 309
  - mit CGI.pm setzen* 650
  - Negotiate* 86, 347
  - Pragma* 86
  - Proxy-Authenticate* 87
  - Proxy-Authorization* 87
  - Range* 87
  - Referer* 87
  - Retry-After* 88
  - Server* 88
  - Set-Cookie* 88
  - TCN* 348
  - TE* 89
  - Trailer* 89
  - Transfer-Encoding* 89
  - Übersicht* 72
  - Upgrade* 90
  - User-Agent* 90
  - Vary* 90
  - Via* 91
  - Warning* 91
  - WWW-Authenticate* 91
- HTTP-Methoden
  - CONNECT* 66

- DELETE 64
- GET 56
- HEAD 59
- Idempotenz 60
- OPTIONS 66
- POST 60
- PUT 62
- TRACE 65
- HTTP-Protokoll → Hypertext Transfer Protocol 53
- HTTPS-Verbindungen 481
- httxtzdbm, Hilfsprogramm 228, 375
- Hypertext Transfer Protocol (HTTP) 53
  - Anfrage 55
  - CONNECT-Methode 66
  - DELETE-Methode 64
  - GET-Formularversand 59
  - GET-Methode 56
  - Header, Übersicht 72
  - Header-Manipulation mit Apa 307
  - Header-Manipulation mit Apache 132
  - HEAD-Methode 59
  - idempotente Methoden 60
  - Kommunikationsablauf 53
  - OPTIONS-Methode 66
  - POST-Formularversand 60
  - POST-Methode 60
  - PROFIND-Methode 773
  - PUT-Methode 62
  - Query-String 59
  - Statuscodes 67
  - TRACE-Methode 65
- I**
- IANA (Internet Assigned Numbers Authority) 26
- IASPIFakeAsync, Direktive 723
- Idempotenz (HTTP) 60
- IdentityCheck, Direktive 524
- if/elif/else/endif, SSI-Elemente 735
- If-Match, HTTP-Header 84
- If-Modified-Since, HTTP-Header 84
- If-None-Match, HTTP-Header 84
- If-Range, HTTP-Header 85
- If-Unmodified-Since, HTTP-Header 85
- IIS → Internet Information Server 121
- Image-Maps 404
  - Syntax 405
- ImapBase, Direktive 407
- ImapDefault, Direktive 408
- imap-file, Handler 336, 405
- ImapMenu, Direktive 408
- Include, Direktive 260
- include, SSI-Element 734
- INCLUDES, Filter 730
- inconv, Programm 754
- Incubator 117
- Index
  - aktivieren 393
  - automatisch generierter 392
  - Fancy-Index 393
- IndexIgnore, Direktive 397
- IndexOptions, Direktive 394
- IndexOrderDefault, Direktive 397
- inetd 820
- insert\_filter, Hook 777
- Installation
  - Windows 195
  - Windows, Apache/Perl 199
- Installationsarten 155
- Installationslayouts 164
  - +Zeichen 164
  - Apache 164
  - BeOS 168
  - BSDI 169
  - Darwin 166
  - Debian 171
  - eigene 173
  - FreeBSD 171
  - GNU 165
  - Mac OS X 165
  - OpenBSD 170
  - opt 167
  - Originalsyntax 172
  - RedHat 166
  - Solaris 170
  - SuSE 168
- Installieren
  - Module 200
- Integritätsprüfung
  - des Apache-Quellcodes 156
- Internet Information Server 121
  - .NET-Integration 122
  - Active Server Pages (ASP) 121
  - ISAPI 121
- Internet Protocol (IP) 24
  - IP-Adressen 24
  - IPv4 und IPv6 24
  - Routing 29
  - TTL 30
- Internet-Schichtenmodell 22
- Intrusion Detection Systems 799
- IP-Adressen
  - CIDR 26

- IP-basierte virtuelle Hosts* 558
  - IPv4* 24
  - IPv6* 28
  - Klassen* 25
  - private* 26
  - spezielle* 26
  - Subnet Mask* 27
  - VLSM* 27
  - IP-Protokoll → Internet Protocol (IP) 24
  - IP-Routing 29
  - ISAPI 121, 721
  - ISAPIAppendLogToErrors, Direktive 722
  - ISAPIAppendLogToQuery, Direktive 723
  - ISAPICacheFile, Direktive 723
  - ISAPILogNotSupported, Direktive 724
  - ISAPIReadAheadBuffer, Direktive 724
  - ISO-Sprachkürzel, Tabelle 866
- J**
- Jakarta 117
  - James (Java-Mailserver) 117
  - Java
    - Tomcat-Server* 709
  - JavaServer Pages (JSP) 709
    - Beispiel* 718
    - MySQL-Zugriff* 718
  - JavaServlets 709
    - Beispiel* 717
    - MySQL-Zugriff* 718
  - JDBC-Schnittstelle 718
  - JSP → JavaServer Pages (JSP) 709
- K**
- KeepAlive, Direktive 263
  - KeepAliveTimeout, Direktive 264
  - Kew, Nick 466
  - kill
    - Apache steuern mit* 210
  - Kompilieren
    - Apache 1.3* 815
    - configure* 159
    - Einführung* 156
    - Installationslayouts* 164
    - Module* 174, 200
    - Module wählen* 178
    - Quellcode herunterladen* 156
    - UNIX, Überblick* 158
    - Verzeichniswahl* 161
    - weitere Optionen* 184
    - Windows* 188
  - Windows, IDE* 193
  - Windows, Kommandozeile* 190
  - Konfiguration
    - .htaccess* 258
    - Dateien importieren* 260
    - Notwendiges im Überblick* 230
  - Konfigurationsdatei → httpd.conf 239
  - Konfigurationsdateien
    - Apache 2.0* 240
    - Apache 2.2* 240
    - SUSE Linux* 241
  - Konfigurationsdirektiven → Direktiven 233
  - Konfigurationsinformationen 389
  - Kontexte
    - für Direktiven* 247
  - Kryptografie → Verschlüsselung 479
- L**
- Länder-Top-Level-Domains, Tabelle 881
  - LanguagePriority, Direktive 351
  - Last-Modified, HTTP-Header 85
  - Laufzeitmodelle für Server 136
  - Laurie, Ben 479, 533
  - Layouts → Installationslayouts 164
  - lbnamed, DNS-Load-Balancer 581
  - LDAP → Lightweight Directory Access Protocol 449
  - LDAPCacheEntries, Direktive 459
  - LDAPCacheTTL, Direktive 460
  - LDAPOpCacheEntries, Direktive 460
  - LDAPOpCacheTTL, Direktive 461
  - LDAPSharedCacheFile, Direktive 461
  - LDAPSharedCacheSize, Direktive 462
  - ldap-status, Handler 459
  - LDAPTrustedCA, Direktive 462
  - LDAPTrustedCAType, Direktive 462
  - leader, MPM 139
  - Lerdorf, Rasmus 670
  - Lightweight Directory Access Protocol (LDAP) 449
    - Connection-Pooling* 458
    - Schema* 449
  - LimitInternalRecursion, Direktive 803
  - LimitRequestBody, Direktive 804
  - LimitRequestFields, Direktive 804
  - LimitRequestFieldSize, Direktive 804
  - LimitRequestLine, Direktive 805
  - LimitXMLRequestBody, Direktive 805
  - Linux
    - Apache kompilieren* 158

- Apache steuern 205
  - apachectl, Steuerskript* 211
  - Apache-Verwaltung (RedHat) 216
  - Apache-Verwaltung (SuSE) 215
  - mod\_perl-Installation* 699
  - MySQL-Installation 671
  - Perl-Installation 700
  - PHP-Installation 678
  - Runlevel 213
  - Runlevel-Editor (SuSE) 216
  - System V Init 213
  - Tomcat-Installation 710
  - Listen, Direktive 265
  - ListenBackLog, Direktive 271
  - Lizenz
    - Apache 119
    - BSD 120
    - GPL 119
  - Load-Balancing 577
    - mit *mod\_proxy* 589
    - mit *mod\_rewrite* 579
    - Round-Robin-DNS 578
    - spezielle Lösungen 580
  - LoadFile, Direktive 267
  - LoadModule, Direktive 267
  - Location, HTTP-Header 85
  - LockFile, Direktive 274
  - log\_server\_status, Hilfsprogramm 228, 541
  - log4j 117
  - Logdateien 519
    - Combined Log Format 519
    - Common Log Format 519
    - cronolog 553
    - Datums- und Uhrzeitformate 527
    - forensische 533
    - Formatdefinitionen 525
    - logresolve, Hilfsprogramm 540
    - Mescalero 553
    - rotatelog, Hilfsprogramm 539
    - split-logfiles, Hilfsprogramm 541
    - Webalizer 553
    - Wusage 553
  - LogFormat, Direktive 531
  - Logging 519
    - Analyse-Tools 553
    - Apache-Direktiven 520
    - CGI-Skripte 628
    - eigene Perl-Skripte 541
    - formatierte Ausgabe (Perl) 542
    - Hilfsprogramme 539
    - in Datenbank schreiben (Perl) 548
    - Statistik (Perl) 545
  - Logging, ASF-Projekt 117
  - LogLevel, Direktive 522
  - logresolve, Hilfsprogramm 228, 540
- M**
- Mac OS X
    - Apache-Verwaltung 216
  - Mac OS X, Installationslayout 165
  - Macromedia ColdFusion MX 725
  - Makefile 159
  - map\_to\_storage, Hook 777
  - Matsumoto, Yukihiko 725
  - Matz → Matsumoto, Yukihiko 725
  - Maven 118
  - MaxClients, Direktive 275
  - Max-Forwards, HTTP-Header 86
  - MaxKeepAliveRequests, Direktive 264
  - MaxMemFree, Direktive 275
  - MaxRequestsPerChild, Direktive 276
  - MaxRequestsPerThread, Direktive 282
  - MaxSpareServers, Direktive 281
  - MaxSpareThreads, Direktive 276
  - MaxThreads, Direktive 282
  - MaxThreadsPerChild, Direktive 284
  - MCacheMaxObjectCount, Direktive 613
  - MCacheMaxObjectSize, Direktive 613
  - MCacheMaxStreamingBuffer, Direktive 614
  - MCacheMinObjectSize, Direktive 613
  - MCacheRemovalAlgorithm, Direktive 614
  - MCacheSize, Direktive 612
  - McCool, Rob 114
  - McEachern, Doug 699
  - Meritocracy 116
  - Mescalero 553
  - Meta-Dateien 319
  - MetaDir, Direktive 320
  - MetaFiles, Direktive 320
  - MetaSuffix, Direktive 320
  - Microsoft .NET 122, 725
  - Microsoft Internet Information Server 121
  - MIME-Codierung 332
  - MIME-Konfiguration 321
  - MIME-Magic-Datei 328
  - MimeMagicFile, Direktive 327
  - MIME-Spracheinstellungen 334
  - MIME-Type
    - Einstellung in Apache 323

für Webformulare 60  
 MIME-Types, Tabelle 843  
 MinSpareServers, Direktive 281  
 MinSpareThreads, Direktive 277  
 MMapFile, Direktive 576  
 mod\_access, Modul 297  
 mod\_actions, Modul 632  
 mod\_alias, Modul 356, 622  
 mod\_asis, Modul 318  
 mod\_aspdotnet, Modul 725  
 mod\_auth, Modul 424  
 mod\_auth\_anon, Modul 463  
 mod\_auth\_basic, Modul 424  
 mod\_auth\_dbm, Modul 440  
 mod\_auth\_digest, Modul 431  
   *und Internet Explorer* 432  
 mod\_auth\_ldap, Modul 449  
 mod\_authn\_alias, Modul 473  
 mod\_authn\_anon, Modul 463  
 mod\_authn\_dbd, Modul 471  
 mod\_authn\_dbm, Modul 440  
 mod\_authn\_default, Modul 475  
 mod\_authn\_file, Modul 424  
 mod\_authnz\_ldap, Modul 449  
 mod\_authz\_dbm, Modul 440  
 mod\_authz\_default, Modul 475  
 mod\_authz\_groupfile 424  
 mod\_authz\_host, Modul 297  
 mod\_authz\_owner, Modul 474  
 mod\_authz\_user, Modul 424  
 mod\_autoindex, Modul 393  
 mod\_backhand, Modul (Apache 1.3)  
   580  
 mod\_cache, Modul 605  
 mod\_cern\_meta, Modul 319  
 mod\_cgi, Modul 621  
 mod\_cgid, Modul 622  
 mod\_charset\_lite, Modul 754  
 mod\_dav, Modul 767  
   *Konfiguration* 768  
 mod\_dav\_fs, Modul 767  
 mod\_dav\_lock, Modul 770  
 mod\_daytime, Modul (eigenes) 790  
 mod\_dbd, Modul 467  
 mod\_deflate, Modul 750  
 mod\_dir, Modul 290  
 mod\_disk\_cache, Modul 610  
 mod\_dumpio, Modul 534  
 mod\_echo, Modul 772  
 mod\_env, Modul 637  
 mod\_example, Modul 775  
 mod\_expires, Modul 315  
 mod\_ext\_filter, Modul 756  
 mod\_file\_cache, Modul 575  
 mod\_filter, Modul 745  
 mod\_ftpd, Modul 771  
 mod\_headers, Modul 309  
 mod\_holiday, Modul (eigenes) 777  
 mod\_ident, Modul 524  
 mod\_imagemap, Modul 404  
 mod\_imap, Modul 404  
 mod\_include, Modul 737  
 mod\_info, Modul 390  
 mod\_isapi, Modul 721  
 mod\_jk2, Modul+ 710, 715  
 mod\_ldap, Modul 458  
 mod\_log\_config, Modul 525  
 mod\_log\_forensic, Modul 533  
 mod\_mem\_cache, Modul 612  
 mod\_mime, Modul 321  
 mod\_mime\_magic, Modul 327  
 mod\_mono, Modul 725  
 mod\_negotiation, Modul 339  
 mod\_nw\_ssl, Modul 514  
 mod\_perl 118, 699  
   *Apache-Komplettpaket (Windows)*  
     199  
   *Installation, UNIX* 699  
   *Installation, Windows* 704  
   *MySQL-Zugriff* 706  
   *PerlModule, Direktive* 701  
   *Startup-Datei* 703  
 mod\_php 118  
 mod\_pop3, Modul 771  
 mod\_proxy, Modul 586  
 mod\_proxy\_ajp, Modul 586  
 mod\_proxy\_balancer, Modul 586  
 mod\_proxy\_connect, Modul 586  
 mod\_proxy\_ftp, Modul 586  
 mod\_proxy\_http, Modul 586  
 mod\_python, Modul 724  
 mod\_rewrite, Modul 363  
   *Beispiele* 380  
   *für Load-Balancing* 579  
   *für Session-Tracking* 382  
   *Logging-Direktiven* 538  
 mod\_ruby, Modul 725  
 mod\_security, Modul 811  
 mod\_setenvif, Modul 637  
 mod\_speling, Modul 388  
 mod\_ssl, Modul  
   *Grundkonfiguration* 487  
   *im Proxy-Betrieb* 510  
   *Umgebungsvariablen* 489

mod\_status, Modul 390  
 mod\_unique\_id, Modul 533  
 mod\_userdir, Modul 383  
 mod\_usertrack, Modul 535  
 mod\_vhost\_alias, Modul 566  
     Formatkürzel 566  
 ModMimeUsePathInfo, Direktive 327  
 Module  
     apxs, Hilfsprogramm 200  
     Arbeitsablauf 776  
     Dynamic Shared Objects (DSO) 176  
     externe 151  
     Kompilieren 174  
     Liste 142  
     mod\_access 297  
     mod\_actions 632  
     mod\_alias 356, 622  
     mod\_asis 318  
     mod\_aspdotnet 725  
     mod\_auth 424  
     mod\_auth\_anon 463  
     mod\_auth\_basic 424  
     mod\_auth\_dbm 440  
     mod\_auth\_digest 431  
     mod\_auth\_ldap 449  
     mod\_authn\_alias 473  
     mod\_authn\_anon 463  
     mod\_authn\_dbd 471  
     mod\_authn\_dbm 440  
     mod\_authn\_default 475  
     mod\_authn\_file 424  
     mod\_authnz\_ldap 449  
     mod\_authz\_dbm 440  
     mod\_authz\_default 475  
     mod\_authz\_groupfile 424  
     mod\_authz\_host 297  
     mod\_authz\_owner 474  
     mod\_authz\_user 424  
     mod\_autoindex 393  
     mod\_backhand (Apache 1.3) 580  
     mod\_cache 605  
     mod\_cern\_meta 319  
     mod\_cgi 621  
     mod\_cgid 622  
     mod\_charset\_lite 754  
     mod\_dav 767  
     mod\_dav\_fs 767  
     mod\_dav\_lock 770  
     mod\_daytime (eigenes) 790  
     mod\_dbd 467  
     mod\_deflate 750  
     mod\_dir 290  
     mod\_disk\_cache 610  
     mod\_dumpio 534  
     mod\_echo 772  
     mod\_env 637  
     mod\_example 775  
     mod\_expires 315  
     mod\_ext\_filter 756  
     mod\_file\_cache 575  
     mod\_filter 745  
     mod\_ftpd 771  
     mod\_headers 309  
     mod\_holiday (eigenes) 777  
     mod\_ident 524  
     mod\_imagemap 404  
     mod\_imap 404  
     mod\_include 737  
     mod\_info 390  
     mod\_isapi 721  
     mod\_jk2 710, 715  
     mod\_ldap 458  
     mod\_log\_config 525  
     mod\_log\_forensic 533  
     mod\_mem\_cache 612  
     mod\_mime 321  
     mod\_mime\_magic 327  
     mod\_mono 725  
     mod\_negotiation 339  
     mod\_nw\_ssl 514  
     mod\_perl 699  
     mod\_pop3 771  
     mod\_proxy 586  
     mod\_proxy\_ajp 586  
     mod\_proxy\_balancer 586  
     mod\_proxy\_connect 586  
     mod\_proxy\_ftp 586  
     mod\_proxy\_http 586  
     mod\_python 724  
     mod\_rewrite 363  
     mod\_ruby 725  
     mod\_security 811  
     mod\_setenvif 637  
     mod\_speling 388  
     mod\_status 390  
     mod\_suexec 808  
     mod\_unique\_id 533  
     mod\_userdir 383  
     mod\_usertrack 535  
     mod\_vhost\_alias 566  
     Modulstruktur 779  
     nachinstallieren 200  
     selbst programmieren 774  
     statisch 175

- Typen 141
- Überblick 141
- weitere von Drittanbietern 773
- zum Kompilieren auswählen 178
- Module Magic Number 208
- Modulprogrammierung 774
  - Anfragen verarbeiten 784
  - Direktiven registrieren 781
  - Direktiven-Funktionen 783
  - Grundkonfiguration erzeugen 780
  - Header-Dateien 777
  - Hooks 776
  - Hooks registrieren 784
  - Modulstruktur 779
  - Multiprotokoll-Unterstützung 790
  - request\_rec, Struktur 784
- Modulstruktur 779
- Mono 725
- MPM → Multiprocessing-Module (MPM) 136
- mpm\_netware 140
- mpm\_winnt 140
- mpmt\_os2 140
- Multiprocessing-Module (MPM) 136
  - beos 140
  - Direktiven 268
  - event 140
  - leader 139
  - mpm\_netware 140
  - mpm\_winnt 140
  - mpmt\_os2 140
  - perchild 140
  - prefork 139
  - threadpool 140
  - worker 139
- Multiprotokoll-Unterstützung 771
  - Programmierbeispiel 790
- MultiViews 344
- MultiViewsMatch, Direktive 349
- MX-Record
  - BIND-Nameserver 42
- MySQL
  - Installation, UNIX 671
  - Installation, Windows 673
  - mysql, PHP-Schnittstelle 697
  - phpMyAdmin 686
  - Testdatenbank 675
  - Zugriff über Java 718
  - Zugriff über Perl 706
  - Zugriff über PHP 694
- mysql-Client 673
- mysql-Schnittstelle 697

**N**

- namensbasierte virtuelle Hosts 560
- Nameserver 35
  - BIND 36
- NameVirtualHost, Direktive 563
- NCSA HTTPd 114
- Nebenläufigkeit 137
  - Forking-Server 137
  - Preforking-Server 137
  - select()-Server 138
  - Threading-Server 138
- Negotiate, HTTP-Header 86, 347
- Nessus 800
- netstat, Dienstprogramm 44
- NetWare
  - SSL-Einsatz 514
- Netware, MPM 140
- Non-parsed Header (NPH) 319
- NoProxy, Direktive 592
- NPH → Non-parsed Header (NPH) 319
- nslookup, Dienstprogramm 45
- NS-Record (DNS) 41
- NumServers, Direktive 284
- NWSSLTrustedCerts, Direktive 514

**O**

- ObjectRelationalBridge 116
- open\_logs, Hook 777
- OpenBSD, Installationslayout 170
- Open-Source-Software → freie Software 113
- OpenSSL 479
  - Installation, Windows 482
  - Zertifikat erzeugen 483
- opt, Installationslayout 167
- Options, Direktive 294
- OPTIONS, HTTP-Methode 66
- Order, Direktive 297
- OS/2, MPM 140
- OSI-Referenzmodell 22

**P**

- param(), CGI.pm-Methode 648
- PassEnv, Umgebungsvariable 638
- password\_field(), CGI.pm-Methode 664
- Passwörter 800
- Paulsen, Brian 648
- perchild, MPM 140
- Performance-Tuning 569
  - allgemeine Hinweise 570
  - Datei-Caching 575
  - Direktiven für 574

Perl  
   *\$/*, Variable 95  
   CGI.pm, Modul 647  
   CGI::Carp, Modul 647  
   CGI::Pretty, Modul 648  
   DBI, Modul 706  
   eigener Webserver 92  
   HTTP::Date 94  
   in httpd.conf 707  
   Input Record Separator 95  
   Installation, UNIX 700  
   Installation, Windows 704  
   IO::Socket 94  
   Logdateien auswerten 541  
   mod\_perl 118, 699  
   MySQL-Zugriff 706  
   POSIX 94  
   Zeilenumbrüche 94  
 PerlModule, Direktive 701  
 Peters, Frank 114  
 Pfadangaben  
   in Direktiven 244  
 PGP → Pretty Good Privacy (PGP) 157  
 PHP 118, 670  
   Cookies 693  
   Datei-Uploads 691  
   Formulardaten auslesen 689  
   Installation, UNIX 678  
   Installation, Windows 681  
   mysql-Schnittstelle 697  
   mysql-Schnittstelle 694  
   php.ini, Konfigurationsdatei 683  
   Programmiertipps 688  
   Sessions 692  
   Zend Engine II 670  
 php.ini, Konfigurationsdatei 683  
 phpMyAdmin 686  
   Sicherheit 687  
 PidFile, Direktive 269  
 ping, Dienstprogramm 42  
 Pioch, Nicolas 114  
 popup\_menu(), CGI.pm-Methode 664  
 Port, Direktive (1.3) 819  
 Port-basierte virtuelle Hosts 561  
 POST, HTTP-Methode 60  
   Formularversand 60  
 post\_config, Hook 777  
 Postel, Jon 82  
 Pound, Load-Balancer 580  
 Pragma, HTTP-Header 86  
 pre\_config, Hook 777  
 PREFIX 161  
  
 prefork, MPM 139  
 Preforking-Server 137  
 Pretty Good Privacy (PGP)  
   Quellcode-Integritätsprüfung 157  
 printenv, SSI-Element 735  
 Programmierung von Modulen 774  
 PROPFIND, HTTP-Methode 769  
 ProtocolDaytime, Direktive 790  
 ProtocolEcho, Direktive 772  
 ProtocolReqCheck, Direktive (1.3) 820  
 Protokolldateien → Logdateien 519  
 Proxy  
   Tomcat einbinden 715  
 Proxy-Authenticate, HTTP-Header 87  
 Proxy-Authorization, HTTP-Header 87  
 ProxyBadHeader, Direktive 593  
 ProxyBlock, Direktive 594  
 ProxyDomain, Direktive 594  
 ProxyErrorDomain, Direktive 595  
 ProxyIOBufferSize, Direktive 595  
 ProxyMaxForwards, Direktive 596  
 ProxyPass, Direktive 596  
 ProxyPassReverse, Direktive 598  
 ProxyPassReverseCookieDomain,  
   Direktive 599  
 ProxyPassReverseCookiePath,  
   Direktive 600  
 ProxyPreserveHost, Direktive 601  
 ProxyReceiveBufferSize, Direktive 601  
 ProxyRemote, Direktive 592  
 ProxyRemoteMatch, Direktive 592  
 ProxyRequests, Direktive 591  
 Proxy-Server 585  
   Aufgaben 586  
   Forward-Proxy 585  
   Grundkonfiguration 587  
   mit Cache 604  
   Reverse-Proxy 585  
   umgehen 592  
 ProxyTimeout, Direktive 602  
 ProxyVia, Direktive 602  
 ps, Apache-Test mit 205  
 PTR-Record (DNS) 40  
 Public-Key-Verschlüsselung 480  
 PUT, HTTP-Methode 62  
 Python 724  
  
**Q**  
 Quellcode  
   herunterladen 156  
   Integritätsprüfung 156  
   PGP-Integritätsprüfung 157

- Windows 189
- Query-String 59
- R**
- radio\_group(), CGI.pm-Methode 665
- Range, HTTP-Header 87
- ReadmeName, Direktive 404
- Rechtschreibkorrektur
  - in URLs 388
- RedHat Linux
  - Apache-Verwaltung 216
- RedHat, Installationslayout 166
- Redirect → Weiterleitung 355
- redirect(), CGI.pm-Methode 661
- Redirect, Direktive 358
  - häufige Fehlerquelle 360
- RedirectMatch, Direktive 360
- RedirectPermanent, Direktive 362
- RedirectTemp, Direktive 363
- Referer, HTTP-Header 87
- RegExp → Reguläre Ausdrücke (RegExp) 245
- Reguläre Ausdrücke (RegExp)
  - als Direktivenwerte 245
- Remote Variant Selection Algorithm (RVSA) 347
- RemoveCharset, Direktive 332
- RemoveEncoding, Direktive 334
- RemoveHandler, Direktive 339
- RemoveInputFilter, Direktive 745
- RemoveLanguage, Direktive 335
- RemoveOutputFilter, Direktive 744
- RemoveType, Direktive 325
- request\_rec, Datenstruktur 784
- RequestHeader, Direktive 314
- Require, Direktive 422
- reset(), CGI.pm-Methode 666
- ResourceConfig, Direktiven (1.3) 820
- Retry-After, HTTP-Header 88
- Reverse-Proxy 585
  - Konfigurationsbeispiele 589
- RewriteBase, Direktive 370
- Rewrite-Beispiele 380
- RewriteCond, Direktive 369
- RewriteEngine, Direktive 364
- RewriteLock, Direktive 379
- RewriteLog, Direktive 538
- RewriteLogLeve, Direktive 539
- RewriteMap, Direktive 371
- RewriteOptions, Direktive 379
- RewriteRule, Direktive 365
- RFC 24
  - 1035 (DNS) 33
  - 1123 (Internet-Host-Anforderungen) 82
  - 1413 (identd) 524
  - 2295 (TCN) 347
  - 2296 (RVSA) 347
  - 2460 (IPv6) 24
  - 2518 (WebDAV) 767
  - 2616 (HTTP/1.1) 53
  - 2822 (Textnachricht, aktuell) 55
  - 768 (UDP) 32
  - 791 (IPv4) 24
  - 793 (TCP) 31
  - 822 (Textnachricht) 55
- Ristic, Ivan 811
- RLimitCPU, Direktive 806
- RLimitMEM, Direktive 806
- RLimitNPROC, Direktive 807
- Robinson, David 114
- Robustheitsprinzip → Robustness Principle 82
- Robustness Principle 82
- Rossum, Guido van 724
- rotatelog, Hilfsprogramm 228, 539
- Round-Robin-DNS 41
- Routing 29
- Ruby 725
- Runlevel 213
- Runlevel-Editor (SuSE) 216
- RVSA → Remote Variant Selection Algorithm (R) 347
- S**
- Satisfy, Direktive 423
- Schichtenmodell 21
  - Alltagsbeispiel 22
  - TCP/IP 22
- Schrauwen, Jorge 189, 226
- ScoreBoardFile, Direktive 277
- Script, Direktive 634
- ScriptAlias, Direktive 622
- ScriptAliasMatch, Direktive 624
- ScriptInterpreterSource, Direktive 630
- ScriptLog, Direktive 628
- ScriptLogBuffer, Direktive 628
- ScriptLogSize, Direktive 629
- ScriptSockSize, Direktive 629
- Secure Sockets Layer (SSL) 479
  - Apache-Konfiguration für HTTPS-Verbindungen 487
  - OpenSSL 479
  - OpenSSL-Einrichtung 482

- Überblick 480
- Umgebungsvariablen 489
- Zertifikate 483
- Zertifizierungspfad 485
- SecureListen, Direktive 515
- select()-Server 138
- self\_url(), CGI.pm-Methode 663
- send-as-is, Handler 318, 336
- SendBufferSize, Direktive 278
- Server Side Includes (SSI) 729
  - aktivieren 729
  - config, Element 731
  - echo, Element 732
  - Elemente 730
  - exec, Element 733
  - flastmod, Element 734
  - fsize, Element 734
  - if/elif/else/endif, Elemente 735
  - include, Element 734
  - printenv, Element 735
  - Programme ausführen 733
  - set, Element 735
  - Umgebungsvariablen 732
  - Variablen definieren 735
- Server, HTTP-Header 88
- ServerAdmin, Direktive 285
- ServerAlias, Direktive 563
- server-info, Handler 336, 390
- Server-Kontext 247
- ServerLimit, Direktive 278
- ServerName, Direktive 286
- server-parsed, Handler 336
- ServerPath, Direktive 564
- ServerRoot, Direktive 262
- ServerSignature, Direktive 289
- server-status, Handler 337, 390
- ServerTokens, Direktive 287
- ServerType, Direktive (1.3) 820
- Service → Dienst 219
- Servlets → Java Servlets 709
- Session-Tracking
  - mit mod\_rewrite 382
- set, SSI-Element 735
- Set-Cookie, HTTP-Header 88
- SetEnv, Direktive 639
- SetEnv, Umgebungsvariable 637
- SetEnvIfNoCase, Direktive 640
- SetHandler, Direktive 337
- SetInputFilter, Direktive 743
- SetOutputFilter, Direktive 741
- Shebang 623
- Sicherheit 799
  - chroot-Umgebung 800
  - Crackertools 800
  - Direktiven für 803
  - Firewalls 799
  - Intrusion Detection Systems 799
  - menschliches Versagen 800
  - Passwörter 800
  - SuEXEC 808
  - Überblick 801
- Skolnick, David 114
- SOA-Record (DNS) 40
- Softwarelizenz
  - Apache 119
  - BSD 120
  - GPL 119
- Solaris, Installationslayout 170
- Sourcecode → Quellcode 156
- split\_logfile, Hilfsprogramm 228
- split-logfiles, Hilfsprogramm 541
- Spracheinstellungen 334
- Sprachkürzel, Tabelle 866
- srm.conf 239
- SSI → Server Side Includes (SSI) 729
- SSIEndTag, Direktive 737
- SSLErrorMsg, Direktive 738
- SSIStartTag, Direktive 737
- SSITimeFormat, Direktive 738
- SSIUndefinedEcho, Direktive 739
- SSL → Secure Sockets Layer (SSL) 479
- SSLCACertificateFile, Direktive 491
- SSLCACertificatePath, Direktive 492
- SSLCADNRequestFile, Direktive 492
- SSLCADNRequestPath, Direktive 493
- SSLCARevocationFile, Direktive 493
- SSLCARevocationPath, Direktive 494
- SSLCertificateChainFile, Direktive 494
- SSLCertificateFile, Direktive 495
- SSLCertificateKeyFile, Direktive 495
- SSLCipherSuite, Direktive 496
- SSLCryptoDevice, Direktive 499
- SSLEngine, Direktive 500
- SSLHonorCipherOrder, Direktive 501
- SSLMutex, Direktive 501
- SSLOptions, Direktive 502
- SSLPassPhraseDialog, Direktive 503
- SSLProtocol, Direktive 504
- SSLProxyCACertificateFile, Direktive 510
- SSLProxyCACertificatePath, Direktive 511
- SSLProxyCARevocationFile, Direktive 511

- SSLProxyCARevocationPath, Direktive 511
  - SSLProxyCipherSuite, Direktive 512
  - SSLProxyEngine, Direktive 512
  - SSLProxyMachineCertificateFile, Direktive 512
  - SSLProxyMachineCertificatePath, Direktive 513
  - SSLProxyProtocol, Direktive 513
  - SSLProxyVerify, Direktive 513
  - SSLProxyVerifyDepth, Direktive 514
  - SSLRandomSeed, Direktive 505
  - SSLRequire, Direktive 506
  - SSLRequireSSL, Direktive 508
  - SSLSessionCache, Direktive 508
  - SSLSessionCacheTimeout, Direktive 509
  - SSLVerifyClient, Direktive 509
  - SSLVerifyOpen, Direktive 510
  - start\_form(), CGI.pm-Methode 663
  - start\_html(), CGI.pm-Methode 651, 661
  - StartServers, Direktive 279
  - StartThreads, Direktive 279
  - Statuscodes (HTTP) 67
    - 100 Continue 68
    - 101 Switching Protocols 68
    - 1xx (Informationen) 68
    - 200 OK 69
    - 201 Created 69
    - 2xx (Erfolg) 68
    - 301 Moved Permanently 69
    - 302 Found 69
    - 303 See Other 69
    - 304 Not Modified 70
    - 307 Temporary Redirect 69
    - 3xx (Umleitung) 69
    - 401 Unauthorized 71
    - 403 Forbidden 71
    - 404 Not Found 71
    - 4xx (Client-Fehler) 70
    - 500 Internal Server Error 72
    - 5xx (Server-Fehler) 72
    - Typen 67
  - Statusinformationen 389
  - Stein, Lincoln D. 647
  - strftime() 527
  - Struts 117
  - submit(), CGI.pm-Methode 665
  - Subnet Mask 27
  - SuEXEC 808
    - Compiler-Optionen für 808
    - Sicherheitsüberprüfungen 809
  - suexec, Hilfsprogramm 228
  - suexec, Modul 808
  - SuexecUserGroup, Direktive 810
  - SuSE Linux
    - Apache-Konfigurationsdateien 241
    - Apache-Verwaltung 215
    - Installationslayout 168
    - Runlevel-Editor 216
  - symmetrische Verschlüsselung 480
  - System V Init 213
- T**
- TCL, ASF-Projekt 118
  - TCN → Transparente Content-Negotiation 347
  - TCN, HTTP-Header 348
  - TCP/IP
    - Diagnose und Fehlersuche 42
    - dig 46
    - HTTP 53
    - IP-Protokoll 24
    - Kommunikationsverfahren 23
    - netstat 44
    - nslookup 45
    - ping 42
    - TCP 31
    - telnet 47
    - traceroute 43
    - Transportprotokolle 30
    - UDP 32
  - TCP/IP-Schichtenmodell 22
  - TCP-Protokoll → Transmission Control Protocol 31
  - TE, HTTP-Header 89
  - Teilnetzmaske 27
  - telnet, Dienstprogramm 47
  - Terbush, Randy 114
  - textfield(), CGI.pm-Methode 664
  - Thau, Robert S. 114
  - Threading-Server 138
  - ThreadLimit, Direktive 280
  - threadpool, MPM 140
  - ThreadsPerChild, Direktive 280
  - ThreadStackSize, Direktive 283
  - TimeOut, Direktive 263
  - TLS → Transport Layer Security (TLS) 479
  - Tomcat 117, 124, 709
    - Installation, UNIX 710
    - Installation, Windows 714
    - Konfiguration 712

- mod\_jk2* 710, 715
- Programmierbeispiele* 716
- Proxy-Anbindung* 715
- Top-Level-Domains, Tabelle 881
- TRACE, HTTP-Methode 65
- traceroute, Dienstprogramm 43
- Trailer, HTTP-Header 89
- Transfer-Encoding, HTTP-Header 89
- TransferLog, Direktive 533
- translate\_name, Hook 777
- Transmission Control Protocol (TCP) 31
- Transparente Content-Negotiation 347
- Transport Layer Security (TLS) 479
- tripwire 799
- TTL (Time To Live) 30
- type\_checker, Hook 777
- type-map, Handler 337, 341
- Type-Maps 341
  - Syntax* 342
- TypesConfig, Direktive 326

**U**

- UDP-Protokoll → User Datagram Protocol (UDP) 32
- Umgebungsvariablen 432, 634
  - configure* 187
  - in verschiedenen Sprachen* 635
  - mit Apache setzen* 637
  - PassEnv* 638
  - SetEnv* 637
  - SSI* 732
  - SSL* 489
- uname() 94
- UNIX
  - Apache beenden* 212
  - Apache kompilieren* 158
  - Apache neu starten* 212
  - Apache starten* 212
  - Apache steuern* 205
  - apachectl, Steuerskript* 211
  - Apache-Unterstützung* 125
  - BSD-Startskripte* 214
  - mod\_perl-Installation* 699
  - MySQL-Installation* 671
  - Perl-Installation* 700
  - PHP-Installation* 678
  - Runlevel* 213
  - Tomcat-Installation* 710
  - User- und Group-ID für Apache* 269
- UNIX System V Init 213
- UnsetEnv, Direktive 638
- Upgrade, HTTP-Header 90
- url(), CGI.pm-Methode 663
- UseCanonicalName, Direktive 286
- User Datagram Protocol (UDP) 32
- User, Direktive 269
- User-Agent, HTTP-Header 90
- UserDir, Direktive 384
- Usertracking → Session-Tracking 382

**V**

- van Rossum, Guido 724
- Variablen
  - AuthDigestEnableQueryStringHack* 432
- Vary, HTTP-Header 90
- Verfallsdatum (Dokumente) 315
- Verschlüsselung
  - asymmetrische* 480
  - Einwegverschlüsselung* 480
  - Grundbegriffe* 479
  - symmetrische* 480
- Versionierung
  - bei Apache* 128
  - bei freier Software* 129
- Verzeichniscontainer 249
- Verzeichnisdienste 449
- Verzeichnisse
  - cgi-bin* 622
  - für CGI* 622
- Via, HTTP-Header 91
- VirtualDocumentRoot, Direktive 567
- VirtualDocumentRootIP, Direktive 568
- VirtualScriptAlias, Direktive 568
- VirtualScriptAliasIP, Direktive 569
- Virtuelle Hosts 132, 557
  - als httpd.conf-Kontext* 247
  - Direktiven für* 562
  - IP-basierte* 558
  - Konfigurationsbeispiele* 558
  - namensbasierte* 560
  - Port-basierte* 561
- VLSM 27

**W**

- Warning, HTTP-Header 91
- Web Services
  - ASF-Projekt* 118
- Webalizer 553
- Web-Cache 603
  - Konfigurationsbeispiele* 604
- WebDAV 767
  - Konfiguration* 768
  - PROPFIND-Methode* 769
- Web-Formulare

- GET-Versand 59
- Webformulare
  - manuell in CGIs einlesen 645
  - MIME-Types 60
  - mit CGI.pm 655
  - mit CGI.pm, Referenz 663
  - mit PHP auslesen 689
  - POST-Versand 60
- Webserver
  - Apache-Konkurrenz 120
  - CERN httpd 114
  - Entwicklung 113
  - Microsoft IIS 121
  - NCSA HTTPd 114
  - Tomcat 124
  - Zeus 123
- Website
  - einfache zum Test 229
- Weiterleitung 355
- Wilson, Andrew 114
- Win32DisableAcceptEx, Direktive 284
- Windows
  - Apache kompilieren 188
  - Apache manuell beenden 218
  - Apache manuell neu starten 218
  - Apache manuell starten 218
  - Apache steuern 217
  - Apache.exe, Binary 217
  - Apache/Perl-Paket 199
  - Apache-Monitor 224
  - ASP.NET mit Apache 725
  - awk 190
  - Binärinstallation 195
  - CGI-Zuordnung 623
  - Deinstallation 199
  - ISAPI 721
  - mod\_perl-Installation 704
  - MySQL-Installation 673
  - OpenSSL-Installation 482
  - Perl-Installation 704
  - Pfadangaben in httpd.conf 244
  - PHP-Installation 681
  - Tomcat-Installation 714
- Windows\_NT, MPM 140
- worker, MPM 139
- Wusage 553
- WWW-Authenticate, HTTP-Header 91

**X**

- XBitHack, Direktive 739
- XML
  - ASF-Projekt 119

**Y**

- YaST
  - Apache-Verwaltung 215

**Z**

- Zeichensätze
  - Einstellung in Apache 329
  - Konvertierung 754
- Zeichensätze, Tabelle 871
- Zeilenumbrüche
  - in verschiedenen Systemen 94
- Zend Engine II 670
- Zertifikat (OpenSSL)
  - erstellen 483
  - Zertifizierungspfad 485
- Zeus, Webserver 123
- Zugriffsbeschränkung
  - nach IP-Adresse 297