

Kompilieren und Installieren

Dieses Dokument umfaßt nur die Kompilierung und Installation des Apache auf Unix und Unix-ähnlichen Systemen. Für die Kompilierung und Installation unter Windows lesen Sie bitte *Den Apache unter Microsoft Windows betreiben*¹. Für andere Plattformen lesen Sie bitte die *Dokumentation Plattformen*².

Die Konfigurations- und Installationsumgebung des Apache 2.0 hat sich seit dem Apache 1.3 komplett verändert. Der Apache 1.3 benutzt einen speziellen Satz von Skripten, um eine einfache Installation zu ermöglichen. Der Apache 2.0 dagegen verwendet nun `libtool` und `autoconf`, um eine Umgebung zu schaffen, die der vieler anderer Open Source Projekte ähnlich sieht.

Themen

Überblick für die Ungeduldigen.....	1
Anforderungen	1
Download.....	2
Auspacken.....	3
Den Codebaum konfigurieren.....	3
Erstellen.....	7
Installieren.....	7
Anpassen.....	7
Testen.....	7
URI-Referenzen	8

Siehe auch

- *Apache starten*³
- *Beenden und Neustarten*⁴

Überblick für die Ungeduldigen

Download	\$ <code>lynx http://www.apache.org/dist/httpd/httpd-2_1_NN.tar.gz</code>
Auspacken	\$ <code>gzip -d httpd-2_1_NN.tar.gz</code> \$ <code>tar xvf httpd-2_1_NN.tar</code>
Konfigurieren	\$ <code>./configure --prefix=PREFIX</code>
Kompilieren	\$ <code>make</code>
Installieren	\$ <code>make install</code>
Anpassen	\$ <code>vi PREFIX/conf/httpd.conf</code>
Testen	\$ <code>PREFIX/bin/apachectl start</code>

NN muss durch die Nummer der Unterversion ersetzt werden, und *PREFIX* durch den Verzeichnispfad, in dem der Server installiert werden soll. Wenn *PREFIX* nicht angegeben ist, wird die Voreinstellung `/usr/local/apache2` verwendet.

Beginnend mit den Anforderungen für die Kompilierung und Installation des Apache HTTPD ist weiter unten jeder Abschnitt des Kompilierungs- und Installationsvorganges genauer beschrieben.

Anforderungen

Folgende Anforderungen gelten für die Erstellung des Apache:

Plattenplatz

Stellen Sie sicher, dass Sie kurzzeitig wenigstens 50 MB freien Festplattenplatz zur Verfügung

Kompilieren und Installieren

haben. Nach der Installation belegt der Apache ungefähr 10 MB Plattenplatz. Der tatsächliche Platzbedarf variiert in Abhängigkeit von den gewählten Konfigurationseinstellungen und Modulen von Drittanbietern.

ANSI-C-Compiler und Generierungswerkzeuge

Stellen Sie sicher, dass Sie einen ANSI-C Compiler installiert haben. Der GNU C Compiler (GCC)⁵ der Free Software Foundation (FSF)⁶ ist empfehlenswert (Version 2.7.2 ist gut). Wenn Sie den GCC nicht besitzen, stellen Sie zumindest sicher, dass der Compiler Ihres Anbieters ANSI-kompatibel ist. Außerdem muss Ihr PATH wesentliche Generierungswerkzeuge wie make enthalten.

Zeitgenauigkeit bewahren

Elemente des HTTP-Protokolls werden in Form einer Tageszeit ausgedrückt. Darum sollten Sie jetzt prüfen, ob Ihr System die Fähigkeit zur Zeitsynchronisation besitzt, und diese gegebenenfalls installieren. Üblicherweise werden hierfür die Programme ntpdate oder xntpd verwendet, die auf dem Network Time Protocol (NTP) basieren. Nähere Informationen über NTP Software und öffentliche Zeitserver finden Sie in der Usenet Newsgroup comp.protocols.time.ntp⁷ und auf der NTP Homepage⁸.

Perl 5⁹ [OPTIONAL]

Für einige Pflegeprogramme wie apxs¹⁰ oder dbmmanage¹¹ (die in Perl geschrieben sind) wird der Perl 5 Interpreter benötigt (die Versionen ab 5.003 sind ausreichend). Wenn kein derartiger Interpreter vom 'configure'-Skript gefunden werden kann, macht das jedoch nichts. Selbstverständlich können Sie den Apache 2.0 auch so generieren und installieren. Lediglich diese Pflegeskripte können dann nicht verwendet werden. Wenn Sie mehrere Perl Interpreter installiert haben (vielleicht Perl 4 durch Ihren Händler und Perl 5 durch Sie selbst), dann ist die Verwendung der --with-perl Option (siehe unten) empfehlenswert, um sicherzustellen, dass der richtige Interpreter von ./configure ausgewählt wird.

Download

Der Apache kann von der Apache HTTP Server Downloadseite¹² heruntergeladen werden, auf der verschiedene Spiegelserver angegeben sind. Sie finden hier die jeweils aktuelle stabile Version.

Nach dem Download, insbesondere, wenn ein Mirror benutzt wurde, ist es wichtig, zu prüfen, dass Sie eine vollständige, nicht modifizierte Version des Apache HTTP Servers erhalten haben. Dies kann durch Prüfen des heruntergeladenen Tarballs gegen die PGP-Signatur erreicht werden. Dies ist wiederum ein Zweischrittverfahren. Zuerst müssen Sie die KEYS¹³-Datei ebenfalls von der Apache Distributionsseite beziehen. (Um sicherzustellen, dass die KEYS-Datei selbst nicht verändert wurde, ist es sinnvoll, eine Datei aus einer früheren Distribution des Apache zu verwenden, oder den Schlüssel von einem öffentlichen Schlüsselserver zu importieren.) Die Schlüssel werden mit einem der folgenden Befehle (abhängig von Ihrer PGP Version) zu Ihrem persönlichen Schlüsselring hinzugefügt:

```
$ gpg < KEYS
```

oder

```
$ gpg --import KEYS
```

Der nächste Schritt ist das Testen des Tarballs gegen die PGP-Signatur, die immer von der Apache Originalseite¹² bezogen werden sollte. Den Link zur Signaturdatei finden Sie hinter dem jeweiligen Download-Link oder im entsprechenden Verzeichnis auf der Apache-Distributionsseite¹⁴. Sie hat den gleichen Dateinamen wie der Tarball, aber zusätzlich die Endung .asc. Nun können Sie die

Distribution mit einem der folgenden Befehle (wiederum abhängig von Ihrer PGP Version) prüfen:

```
$ pgp httpd-2_1_NN.tar.gz.asc
```

oder

```
$ gpg --verify httpd-2_1_NN.tar.gz.asc
```

Sie sollten eine Meldung wie

```
Good signature from user "Martin Kraemer <martin@apache.org>".
```

erhalten.

Abhängig von den Vertrauensbeziehungen in Ihrem Schlüsselring können Sie auch eine Meldung erhalten, die besagt, dass die Beziehungen zwischen dem Schlüssel und dem Unterzeichner des Schlüssels nicht verifiziert werden können. Solange Sie der Authentizität der KEYS-Datei vertrauen, ist dies kein Problem.

Auspacken

Das Auspacken des Quellcodes aus dem Apache HTTPD Tarball besteht aus einem simplen Dekomprimieren und danach "Ent-tarren":

```
$ gzip -d httpd-2_1_NN.tar.gz  
$ tar xvf httpd-2_1_NN.tar
```

Dies erstellt unterhalb des aktuellen Verzeichnisses ein neues Verzeichnis, das den Quellcode für die Distribution enthält. Sie sollten mit `cd` in dieses Verzeichnis wechseln, bevor Sie mit der Kompilierung des Servers weitermachen.

Den Codebaum konfigurieren

Der nächste Schritt ist die Konfiguration des Apache-Codebaumes für Ihre spezielle Plattform und Ihre persönlichen Bedürfnisse. Dies wird mit dem Skript `configure` durchgeführt, das im Wurzelverzeichnis der Distribution enthalten ist. (Entwickler, welche die CVS Version des Apache-Codebaumes herunterladen, müssen `autoconf` und `libtool` installiert haben und müssen `buildconf` ausführen, bevor sie mit den nächsten Schritten fortfahren können. Dies wird bei offiziellen Releases nicht notwendig sein.)

Um den Codebaum mit den Standardeinstellungen zu konfigurieren, geben Sie einfach `./configure` ein. Zur Änderung dieser Voreinstellungen akzeptiert `configure` eine Reihe von Variablen und Kommandozeilenoptionen. Umgebungsvariablen werden üblicherweise vor den `./configure`-Befehl gesetzt, während andere Optionen danach gesetzt werden. Die wichtigste Option hierbei ist der vorangestellte Ablageort, unter dem der Apache später installiert wird, denn der Apache muss auf diesen Ablageort eingestellt werden, um korrekt zu arbeiten. Es gibt jedoch noch eine Reihe weiterer Optionen für Ihre persönlichen Wünsche.

Um einen kurzen Eindruck zu gewinnen, welche Möglichkeiten Sie haben, folgt hier ein typisches Beispiel, das den Apache mit einem speziellen Compiler und Compilerflags für das Installationsverzeichnis `/sk/pkg/apache` kompiliert, sowie die beiden zusätzlichen Module

`mod_rewrite` und `mod_speling` für späteres Laden durch den DSO-Mechanismus:

```
$ CC="pgcc" CFLAGS="-O2" \  
./configure --prefix=/sw/pkg/apache \  
--enable-rewrite=shared \  
--enable-speling=shared
```

Wenn `configure` startet, benötigt es mehrere Minuten, um die Verfügbarkeit von Features auf Ihrem System zu prüfen und ein Makefile zu generieren, das später zur Kompilierung des Servers verwendet wird.

Die einfachste Art und Weise, alle Konfigurationsflags für den Apache herauszufinden, ist, `./configure --help` aufzurufen. Hier folgt eine kurze Beschreibung der meisten Argumente und Umgebungsvariablen.

Umgebungsvariablen

Der `autoconf`-Generierungsvorgang verwendet verschiedene Umgebungsvariablen für die Konfiguration der Generierungsumgebung. Im allgemeinen verändern diese Variablen die Methode, die benutzt wird, um den Apache zu erstellen, nicht aber die letztendlichen Features des Servers. Diese Variablen können vor dem Aufruf von `configure` im Environment gesetzt werden, doch es ist üblicherweise einfacher, sie in der `configure`-Befehlszeile, wie im obigen Beispiel gezeigt, anzugeben.

CC=...

Der Name des C-Compiler-Kommandos.

CPPFLAGS=...

Verschiedene C-Präprozessor- und Compileroptionen.

CFLAGS=...

Debug- und Optimierungsoptionen für den C-Compiler.

LDFLAGS=...

Verschiedene Optionen, die an den Linker durchgereicht werden.

LIBS=...

Informationen über den Ablageort von Libraries ("`-L`" und "`-l`" Optionen), die an den Linker durchgereicht werden.

INCLUDES=...

Suchverzeichnisse für Header-Dateien ("`-I`*Verzeichnis*").

TARGET=... [Default: apache]

Name der lauffähigen Datei, die erstellt wird.

NOTEST_CPPFLAGS=...

NOTEST_CFLAGS=...

NOTEST_LDFLAGS=...

NOTEST_LIBS=...

Diese Variablen haben die gleiche Funktion wie ihre Namensvettern ohne `NOTEST`. Diese Variablen werden jedoch nur auf den Generierungsprozess angewendet, nachdem `autoconf` seine Featuretests durchgeführt hat. Dies erlaubt die Einbeziehung von Flags, die während des Featuretests Probleme bereiten würden, aber für die endgültige Kompilierung verwendet werden müssen.

SHLIB_PATH=...

Eine Option, die den Pfad zu Libraries angibt, welche sich Compiler und Linker teilen.

autoconf Ausgabeoptionen

--help

Gibt die Nutzungsinformationen einschließlich aller verfügbaren Optionen aus, ohne jedoch tatsächlich etwas zu konfigurieren.

--quiet

Verhindert die Ausgabe der üblichen "checking..." Nachrichten.

--verbose

Gibt während des Konfigurationsprozesses weitergehende Informationen aus, einschließlich der Namen aller untersuchter Dateien.

Pfadangaben

Gegenwärtig existieren zwei Möglichkeiten, die Pfade zu konfigurieren, in denen der Apache seine Dateien installiert. Bei der ersten können Sie ein Verzeichnis angeben und den Apache veranlassen, sich selbst unterhalb dieses Verzeichnisses in seinen voreingestellten Ablageorten zu installieren.

--prefix=PREFIX [Default: /usr/local/apache2]

Gibt das Verzeichnis an, in dem die Apache-Dateien installiert werden.

Es ist möglich, anzugeben, dass Architektur-abhängige Dateien in einem anderen Verzeichnis abgelegt werden sollen.

--exec-prefix=EPREFIX [Default: PREFIX]

Gibt das Verzeichnis an, in dem Architektur-abhängige Dateien abgelegt werden.

Die zweite und flexiblere Weise, den Installationspfad für den Apache einzustellen, ist die Verwendung der Datei `config.layout`. Wenn Sie diese Methode benutzen, können Sie den Ablageort für jeden Dateityp innerhalb der Apache-Installation separat angeben. Die Datei `config.layout` enthält verschiedene Konfigurationsbeispiele. Sie können auch anhand der Beispiele Ihre eigene individuelle Konfiguration erstellen. Die verschiedenen Layouts in dieser Datei sind gruppiert in `<Layout F00>...</Layout>`-Abschnitte und werden mit Namen wie `F00` bezeichnet.

--enable-layout=LAYOUT

Verwendet das genannte Layout aus der Datei `config.layout`, um die Installationspfade genau anzugeben.

Module

Der Apache ist ein modularer Server. Nur die grundlegendste Funktionalität ist im Kernserver enthalten. Erweiternde Features stehen in verschiedenen Modulen bereit. Während des Konfigurationsprozesses müssen Sie auswählen, welche Module für die Verwendung mit Ihrem Server kompiliert werden sollen. Eine Liste von Modulen¹⁵ ist in dieser Dokumentation enthalten. Module mit dem Status¹⁶ "Base" sind standardmäßig enthalten und müssen ausdrücklich deaktiviert werden (z.B. `mod_userdir`), wenn Sie sie nicht verwenden möchten. Module mit einem anderen Status (z.B. `mod_expvars`) müssen ausdrücklich aktiviert werden, wenn Sie sie verwenden möchten.

Es gibt zwei Möglichkeiten, Module zu kompilieren und mit dem Apache zu benutzen. Module können *statisch kompiliert* werden, was bedeutet, dass sie permanent in dem Apache Binary eingebunden sind. Wenn Ihr Betriebssystem Dynamic Shared Objects (DSOs; Anm. d. Übersetzers: dynamisch eingebundene Bibliotheken) unterstützt und autoconf diese Unterstützung erkennen kann, dann können Module alternativ *dynamisch kompiliert* werden. DSO-Module werden von der Apache Binary getrennt gespeichert und können vom Server unter Verwendung der Direktiven zur Laufzeitkonfiguration, die von `mod_so` bereitgestellt werden, ein- oder ausgeschlossen werden. `mod_so` wird automatisch dem Server hinzugefügt, sobald ein dynamisches Modul in die Kompilierung einbezogen wird. Wenn Sie Ihren Server für das Laden von DSOs tauglich machen

wollen, ohne jedoch ein dynamisches Modul zu kompilieren, können Sie dies explizit mit `--enable-so` angeben.

--enable-MODULE[=shared]

Kompiliert das Modul *MODUL* und bindet es ein. Das Kennzeichen *MODUL* ist der Modul-Bezeichner¹⁷ aus der Dokumentation zum jeweiligen Modul, jedoch ohne den Namensteil `"_module"`. Um ein Modul als DSO zu kompilieren, fügen Sie die Option `=shared` hinzu.

--disable-MODULE

Entfernt das Modul *MODUL*, das andernfalls kompiliert und eingebunden werden würde.

--enable-modules=MODUL-LISTE

Kompiliert die durch Leerzeichen getrennte Liste *MODUL-LISTE* von Modulen und bindet sie ein.

--enable-mods-shared=MODUL-LISTE

Kompiliert die durch Leerzeichen getrennte Liste *MODUL-LISTE* von Modulen als dynamisch ladbare (DSO) Module und bindet sie ein.

Die *MODUL-LISTE* in den Optionen `--enable-modules` und `--enable-mods-shared` ist üblicherweise eine durch Leerzeichen getrennte Liste von Modul-Bezeichnern. Zum Beispiel können zum Aktivieren von `mod_dav` und `mod_info` beide benutzt werden, sowohl

```
./configure --enable-dav --enable-info
```

oder äquivalent

```
./configure --enable-modules="dav info"
```

Zusätzlich können die speziellen Schlüsselwörter `all` oder `most` benutzt werden, um alle oder die meisten der Module in einem Schritt hinzuzufügen. Sie können Module, die Sie nicht benutzen möchten, mit der Option `--disable-MODULE` entfernen. Um beispielsweise alle Module mit Ausnahme von `mod_info` als DSOs einzubinden, können Sie

```
./configure --enable-mods-shared=all --disable-info
```

verwenden.

Zusätzlich zum Standard-Set von Modulen fügt der Apache 2.0 auch eine Auswahl von Multi-Processing Modulen¹⁸ (MPMs) ein. Ein, und nur ein einziges, MPM muss in den Kompilierungsprozess eingebunden werden. Das voreingestellte MPM jeder Plattform ist in der MPM Dokumentation¹⁸ aufgeführt, kann jedoch mit der `configure`-Kommandozeile überschrieben werden.

--with-mpm=NAME

Wählt das MPM *NAME*.

DBM

Verschiedene Apache-Features, darunter `mod_authn_dbm` und DBM `RewriteMap` aus `mod_rewrite`, verwenden einfache Schlüssel/Wert-Datenbanken für eine schnelle Suche nach Informationen. Der Apache-Quellcode enthält SDBM, so dass diese Datenbank immer verfügbar ist. Wenn Sie einen anderen Datenbanktyp benutzen möchten, können Sie die folgenden `configure`-Optionen dazu verwenden:

--with-gdbm[=Pfad]

--with-ndbm[=*Pfad*]

--with-berkeley-db[=*Pfad*]

Wenn kein *Pfad* angegeben ist, sucht der Apache in den normalen Suchpfaden nach den Include- und Bibliothek-Dateien. Eine explizite *Pfad*-Angabe veranlasst den Apache, in *Pfad/lib* und *Pfad/include* nach relevanten Dateien zu suchen. Letztlich kann *Pfad* auch die speziellen Include- und Bibliothek-Pfade durch Doppelpunkt getrennt auflisten.

Suexec

Der Apache enthält ein Hilfsprogramm namens suexec¹⁹, das zur Isolation benutzereigener CGI-Programme verwendet werden kann. Wenn suexec jedoch unsachgemäß konfiguriert wurde, kann es schwere Sicherheitsprobleme verursachen. Deshalb sollten Sie die suexec Dokumentation¹⁹ sorgfältig lesen und beachten, bevor Sie dieses Feature implementieren.

Erstellen

Nun können Sie die verschiedenen Teile, die das Apache-Paket bilden, einfach durch Ausführen des folgenden Befehls erstellen:

```
$ make
```

Seien Sie hierbei bitte geduldig, denn eine Basiskonfiguration benötigt ungefähr 3 Minuten auf einem Pentium III/Linux 2.2. System. Dies kann aber abhängig von Ihrer Hardware und der Anzahl der Module, die Sie aktiviert haben, sehr stark variieren.

Installieren

Nun endlich installieren Sie das Package unter dem konfigurierten Installations-*PREFIX* (siehe oben: Option `--prefix` durch Aufrufen von:

```
$ make install
```

Wenn Sie upgraden, wird die Installation Ihre Konfigurationsdateien oder Dokumente nicht überschrieben.

Anpassen

Als nächstes können Sie Ihren Apache HTTP Server anpassen, indem Sie die Konfigurationsdateien²⁰ unterhalb von *PREFIX/conf/* editieren.

```
$ vi PREFIX/conf/httpd.conf
```

Werfen Sie auch einen Blick in das Apache-Handbuch unter docs/manual/²¹. Die aktuellste Version dieses Handbuchs sowie eine komplette Referenz der verfügbaren Konfigurationsanweisungen²² finden Sie unter <http://httpd.apache.org/docs-2.1/>.

Testen

Sie können nun Ihren Apache HTTP Server starten³, indem Sie einfach

```
$ PREFIX/bin/apachectl start
```

ausführen.

Danach sollten Sie Ihr erstes Dokument unter dem URL `http://localhost/` anfordern können. Die Webseite, die Sie sehen, ist im `DocumentRoot` abgelegt, welches üblicherweise `PREFIX/htdocs/` ist. Den Server stoppen⁴ Sie wieder durch Ausführen von:

```
$ PREFIX/bin/apachectl stop
```

URI-Referenzen

- [1] <http://httpd.apache.org/docs-2.1/platform/windows.html>
- [2] <http://httpd.apache.org/docs-2.1/platform/>
- [3] <http://httpd.apache.org/docs-2.1/invoking.html>
- [4] <http://httpd.apache.org/docs-2.1/stopping.html>
- [5] <http://www.gnu.org/software/gcc/gcc.html>
- [6] <http://www.gnu.org/>
- [7] news:comp.protocols.time.ntp
- [8] <http://www.eecis.udel.edu/~ntp/>
- [9] <http://www.perl.org/>
- [10] <http://httpd.apache.org/docs-2.1/programs/apxs.html>
- [11] <http://httpd.apache.org/docs-2.1/programs/dbmmanage.html>
- [12] <http://httpd.apache.org/download.cgi>
- [13] <http://www.apache.org/dist/httpd/KEYS>
- [14] <http://www.apache.org/dist/httpd/>
- [15] <http://httpd.apache.org/docs-2.1/mod/>
- [16] <http://httpd.apache.org/docs-2.1/mod/module-dict.html#Status>
- [17] <http://httpd.apache.org/docs-2.1/mod/module-dict.html#ModuleIdentifier>
- [18] <http://httpd.apache.org/docs-2.1/mpm.html>
- [19] <http://httpd.apache.org/docs-2.1/suexec.html>
- [20] <http://httpd.apache.org/docs-2.1/configuring.html>
- [21] <http://httpd.apache.org/docs-2.1/>
- [22] <http://httpd.apache.org/docs-2.1/mod/directives.html>