Apache Compile HOWTO (Linux edition)

Luc de Louw

luc at delouw.ch

Revision History

Revision 1.9.12	2002−04−22
Added mod_gzip and mod_gunzip, Corrected some typos, updates of software mentioned in the HOWTO, separated the additional modules into an own section.	
Revision 1.9.11	2002−04−07
Corrected lots of typos (non−technical), updates of software mentioned in the HOWTO	
Revision 1.9.11−pre1	2002−03−15
Corrected some grammar, updates of software mentioned in the HOWTO	
Revision 1.9.10	2002−03−09
Corrected some grammar, updates of software mentioned in the HOWTO	
Revision 1.9.9	2002−02−11
Fixed a major bug in openssl config, restructured the document, added sources for further informations	
Revision 1.9.8	2002−02−08
Updates of software mentioned in the HOWTO, and fixed some bugs	
Revision 1.9.7	2001−12−26
Updates of software mentioned in the HOWTO, tested the HOWTO procedures on Linux running on IBM S/390 (zSeries) Machines (See "platforms" for more info), Added some basic support for Tomcat (Binaries only)	
Revision 1.9.6	2001−10−27
Updates of software mentioned in the HOWTO, and fixed some bugs	
Revision 1.9.5	2001−08−27
Yet another rewrite in DocBook 3.1	
Revision 1.9.4	2001−08−26
Updated the Software−Versions mentioned in the document, corrected some typos	
Revision 1.9.3	2001−06−23
Current Version 2.0.0−pre3 in Linux DocBook format	
Revision 1.0.0	2000−08−05
First publication of the html−based document	

This document describes howto compile the Apache Webserver with the most important modules like mod_perl, mod_dav, mod_auth_ldap, mod_dynvhost, mod_roaming, mod_jserv, and mod_php

Table of Contents

Table of Contents

Table of Contents

1. Introduction

1.1. Contributors and Contacts

First I would thank all those people who send questions and suggestions that made a further development of this document possible. It shows me, sharing knowledge is the right way. I would encourage you to send me more suggestion, just write me an email `<luc at delouw.ch>`

1.2. Why I wrote this document

All Linux−distributions I tested had a non−optimal default setup of Apache. Additionally all major distributions don't have current versions of Apache.

Finally no commercial Unix are delivered with pre−installed Apache

Since I am installing a lot of customized webservers on different Un*xes therefor I wrote a plaintext document and placed it on my website so I can access it at work. Later a friend posted the URL to a mailinglist, and the first questions arrived. So I decided to put more information on the page.

After a lot of people requested the document as an "official" HOWTO, I decided to prepare it to be one

1.3. What this document is supposed to be

Compiling all the items described below needs a lot of configure−options that nobody can memorize. This is supposed to be a "copy−paste−ready" text to compile apache and friends

Also, people should learn how to build a full−featured Apache−webserver by themself to be independent from any Linux distributors.

1.4. What this document doesn't do for you

It is just a Document, not a script that makes the work for you. You have to do all the steps by yourself.

1.5. Platforms

The original document was for all major Un*x platforms. Now the HOWTOs are separated for each platform. You will find the same document adapted for:

- Linux (This Document)
- IBM AIX 4.3 and 5.1L
- Sun Solaris 6/7/8
- Hewlett−Packard HP−UX 11
- {Free|Net|Open}−BSD

Important Notice for users running Linux on IBM S/390 (zSeries): PostgreSQL and Jserv wont compile on that system. All other programs and modules mentioned in the HOWTO are working perfectly

Other Un*x−platforms: Feel free to create a guest−account for me on your Un*x platform, so I can have a look at the differences.

Windows−Users: I'm sorry, I'm too young for a heart−attack, You will need to upgrade your machine to a "real" operating system ;−)

1.6. Copyright Information

1.7. Disclaimer

No liability for the contents of this documents can be accepted. Use the concepts, examples and other content at your own risk. As this is a new edition of this document, there may be errors and inaccuracies, that may of course be damaging to your system. Proceed with caution, and although this is highly unlikely, the author(s) do not take any responsibility for that.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals.

1.8. New Versions

This is the 10th Revision

New revisions of this document will be announced at http://freshmeat.net/apacompile

The latest version of this document is to be found at http://www.delouw.ch/linux

- HTML.
- Postscript (ISO A4 format).
- Acrobat PDF.
- SGML Source.
- HTML gzipped tarball.

1.9. Credits

I would thank all the nice people at `< discuss (at) linuxdoc.org>` for supporting me in writing HOWTOs

---

1.10. Feedback

Feedback is most certainly welcome for this document. Without your submissions and input, this document wouldn't exist. Please send your additions, comments and critics to the following email address : `<luc at delouw.ch>`.

1.11. Translations

At the moment there are translations available for:

- German
- French

Translations to other languages are always welcome. If you translated this document, please let me know, so I can set a link here

---

1.12. About the author

Luc (in english Luke) is 29 years old, playing around with computers since 20years. Currently he is working as Unix System Engineer for an IT–corporation located in Kloten (Zurich), Switzerland. Main–focus is developing all flavors of innovative Systems running on Linux (and other Un*xes) . Further, for all major Un*x platforms all the "impossible" tasks will end up on his desk (yes, its funny and he loves it!)

---

2. Prerequisites

2.1. General

- flex 2.54
- bison 1.28
- autoconf 2.52
- automake 1.4
- libtool 1.4
- yacc−91.7.30
- freetype2−devel
- re2c (Only if PHP to be built from CVS tree)

To be continued

All major distributions will include this general prerequisites

---

2.2. Distribution specific

2.2.1. SuSE Linux

2.2.1.1. SuSE Linux 7.1 (maybe applicable to other distributions)

SuSE 7.1 and maybe other distributions have problems to build apache 1.3.19 (1.3.20 and later are okay) with mod_rewrite

To correct this misbehavior create a symlink:

```
cd /usr/include/db1
ln -s ../ndbm.h ndbm.h
```

In SuSE 7.2 and later this issue is fixed

2.3. OpenSSL

2.3.1. What is OpenSSL

Quoting www.openssl.org

The OpenSSL Project is a collaborative effort to develop a robust, commercial−grade, full−featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full−strength general purpose cryptography library. The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan, and develop the OpenSSL toolkit and its related documentation.

OpenSSL is based on the excellent SSLeay library developed by Eric A. Young and Tim J. Hudson. The OpenSSL toolkit is licensed under an Apache−style license, which basically means that you are free to get and use it for commercial and non−commercial purposes subject to some simple license conditions.

From authors points of view, its the basic to build a secure Unix−Server with Opensource Software, its needed for all major products like mod_ssl, OpenSSH and lot of other stuff that provides encrypted Data−processing

OpenSSL provides the libraries and include−files needed be the products mentioned above and also provides a Application to build Server and client−Certificates.

2.3.2. Download the source

Origin−Site http://www.openssl.org

2.3.3. Building and installing

```
cd /usr/local
tar -xvzf openssl-0.9.6c.tar.gz

cd openssl-0.9.6c

./config shared

make
make test
make install

echo "/usr/local/ssl/lib" >> /etc/ld.so.conf
ldconfig
```

2.4. MySQL

2.4.1. What is MySQL

Mysql is a very fast, powerful and very nice to handle Database.

Especially for Webapplication where most access is read and few write, MySQL is the first choice. The newest Version is also transaction−capable. If you plan a Webapplication, that writes a lot of Data into the DB, maybe PostgreSQL is better suited for your project

You need the C−API from Mysql for compiling php if you wish MySQL−Support in php. It is also needed if you want to use mod_authmysql, See Section 4.3 for more information

2.4.2. Download

Origin−Site: http://www.mysql.com/downloads/

2.4.3. Building and installing

```
cd /usr/local
tar −xvzf mysql−3.23.49a.tar.gz
cd mysql−3.23.49a

./configure \
−−prefix=/usr/local/mysql \
−−enable−assembler \
−−with−innodb

make
make install

/usr/local/mysql/bin/mysql_install_db
echo /usr/local/mysql/lib/mysql >> /etc/ld.so.conf
ldconfig
```

For security−improvement add a mysql−user on your system i.e. "mysql", then

```
chown −R mysql /usr/local/mysql/var
```

and change the line user=root to user=mysql in the file /usr/local/mysql/bin/safe_mysqld

you may wish to start mysql automatically at boottime, copy `/usr/local/mysql/share/mysql/mysql.server` to `/etc/init.d/` (or wherever your rc−script are located) and create the corresponding symbolic link in the runlevel directories.

```
cp /usr/local/mysql/share/mysql/mysql.server /etc/init.d/

ln −s /etc/init.d/mysql.server /etc/init.d/rc3.d/S20mysql
ln −s /etc/init.d/mysql.server /etc/init.d/rc3.d/K20mysql
```

2.5. Building mm

2.5.1. What is mm

Quoting www.engelschall.com

The MM library is a 2−layer abstraction library which simplifies the usage of shared memory between forked (and this way strongly related) processes under Unix platforms. On the first layer it hides all platform dependent implementation details (allocation and locking) when dealing with shared memory segments and on the second layer it provides a high−level malloc(3)−style API for a convenient and well known way to work with data−structures inside those shared memory segments.

From the authors point of view:

It is a common lib that enables Unix−Programmers to simplify shm accesses. It is used by many products, i.e. PHP and mod_ssl

Since the author is not a programmer, he is unable to explain the exact usage of that lib

2.5.2. Download

Origin Site: http://www.engelschall.com/sw/mm/mm−1.1.3.tar.gz

2.5.3. Building and installing

```
cd /usr/local

tar -xvzf mm-1.1.3.tar.gz

cd mm-1.1.3

./configure
make
make test
make install

ldconfig
```

3. Getting, build and install Apache with its basic modules

3.1. Get and untar the Apache Source

3.1.1. What is Apache

Quoting www.apache.org

The Apache Project is a collaborative software development effort aimed at creating a robust, commercial−grade, featureful, and freely−available source code implementation of an HTTP (Web) server. The project is jointly managed by a group of volunteers located around the world, using the Internet and the Web to communicate, plan, and develop the server and its related documentation. These volunteers are known as the Apache Group. In addition, hundreds of users have contributed ideas, code, and documentation to the project. This file is intended to briefly describe the history of the Apache Group and recognize the many contributors.

From the authors point of view:

It is simply the best Webserver−Software, it is very flexible to configure to match your needs, and its E−X−T−R−E−M−E stable! I personally never experienced a crash in a productive (=non−experimental stuff) environment

---

3.1.2. Download the source

Origin−Site http://www.apache.org/dist/httpd/

```
cd /usr/local/

tar −xvzf apache_1.3.24.tar.gz
```

3.1.3. Patch for large−scale sites

If your webserver should answer very much requests at the same time, and your machine is strong enough to serve such an amount of requests, you can change the limit of maximum running processes

Download the patch from: http://www.delouw.ch/linux/apache−patch_HARD_SERVER_LIMIT.txt

```
--- httpd.h     Thu Mar 21 18:07:34 2002
+++ httpd.h-new Sun Apr  7 13:34:11 2002
@@ -320,7 +320,7 @@
 #elif defined(NETWARE)
 #define HARD_SERVER_LIMIT 2048
 #else
-#define HARD_SERVER_LIMIT 256
+#define HARD_SERVER_LIMIT 512
 #endif
```

```
#endif
```

This patch does increase the maximum concurrent accessing clients to 512. Feel free to increase it further, if you hacked your kernel and edited your /etc/security/limits.conf (this is ONLY for experienced users! With wrong settings this could end as a "self−denial−of−service−attack"!! Be sure you have enough processes left for root)

Apply the patch using:

```
cd /usr/local/apache_1.3.24/src/include

patch -p0 < apache-patch_HARD_SERVER_LIMIT.txt
```

3.2. mod_ssl

3.2.1. What is mod_ssl

Quoting www.modssl.org

This module provides strong cryptography for the Apache 1.3 webserver via the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols by the help of the Open Source SSL/TLS toolkit OpenSSL, which is based on SSLeay from Eric A. Young and Tim J. Hudson.

From the authors point of view:

This module is needed to enable Apache for SSL−Requests (https). It applies a patch to the Apache source−code and extends its API (EAPI)

Make sure any module for your Apache−Server is compiled with the compiler−flag −DEAPI, or your Webserver might crash or can not be started

Almost all modules I know adds the −DEAPI flag by themself except mod_jserv and mod_jk

3.2.2. Download the source

Origin−Site:http://www.modssl.org

3.2.3. Applying the patch to the apache source

```
cd /usr/local/

tar -xvzf mod_ssl-2.8.8-1.3.24.tar.gz
cd mod_ssl-2.8.8-1.3.24/

./configure --with-apache=../apache_1.3.24
```

3.3. mod_perl

3.3.1. What is mod_perl

Quoting perl.apache.org

With mod_perl it is possible to write Apache modules entirely in Perl. In addition, the persistent interpreter embedded in the server avoids the overhead of starting an external interpreter and the penalty of Perl start−up time.

From the authors point of view:

mod_perl is a kind of substitute for cgi−bin's. cgi's typically forks a new process for each request, and produces overhead. With mod_perl the perl−interpreter is loaded persistent in the apache−server and does not need to fork processes for each request

---

3.3.2. Download the source

Origin−Site: http://www.apache.org/dist/perl

3.3.3. Building and installing

```
cd /usr/local

tar -xvzf mod_perl-1.26.tar.gz

cd mod_perl-1.26

perl Makefile.PL \
EVERYTHING=1 \
APACHE_SRC=../apache_1.3.24/src \
USE_APACI=1 \
PREP_HTTPD=1 \
DO_HTTPD=1

make
make install
```

Notice: Do not compile mod_perl as dso (Dynamic Shared Object)! According to various sources, apache will crash (I never tried)

3.4. Configure and build Apache

Now the two static modules mod_ssl and mod_perl are configured and the Apache Source has been patched, and we can proceed with building Apache.

3.4.1. Building and installing

```
EAPI_MM="/usr/local/mm-1.1.3" SSL_BASE="/usr/local/ssl" \
./configure \
--enable-module=unique_id \
--enable-module=rewrite \
--enable-module=speling \
--enable-module=expires \
--enable-module=info \
--enable-module=log_agent \
--enable-module=log_referer \
--enable-module=usertrack \
--enable-module=proxy \
--enable-module=userdir \
--enable-module=so \
--enable-shared=ssl \
--enable-module=ssl \
--activate-module=src/modules/perl/libperl.a \
--enable-module=perl

make
make install
```

3.4.2. Create self−signed SSL−certificate

```
cd /usr/local/ssl/bin

./openssl req -new > new.cert.csr
./openssl rsa -in privkey.pem -out new.cert.key
./openssl x509 -in new.cert.csr -out new.cert.cert \
-req -signkey new.cert.key -days 999

cp new.cert.key /usr/local/apache/conf/ssl.key/server.key
cp new.cert.cert /usr/local/apache/conf/ssl.crt/server.crt
```

Notice: OpenSSL asks for different things. A common error is to enter a wrong "common name". This should be the FQHN (Fully Qualified HostName) of your Server, i.e www.foo.org

---

4. Additional modules

4.1. mod_dav

4.1.1. What is mod_dav

Quoting www.webdav.org

mod_dav is an Apache module to provide DAV capabilities (RFC 2518) for your Apache web server. It is an Open Source module, provided under an Apache−style license.

mod_dav is maintained by gstein at lyra.org

From the authors point of view:

DAV means: "Distributed authoring and Versioning". It allows you to manage your Website similar to a filesystem. It is meant to replace ftp−uploads to your Webserver.

DAV is supported by all major Web−development−Tools (newer versions) and is going to be a widely accepted standard for Web−publishing

---

4.1.2. Download the source

Origin−Site: http://www.webdav.org/mod_dav/

4.1.3. Building and installing

```
cd /usr/local

tar -xvzf mod_dav-1.0.3-1.3.6.tar.gz
cd mod_dav-1.0.3-1.3.6

./configure --with-apxs=/usr/local/apache/bin/apxs

make
make install
```

Notice: the filename mod_dav−1.0.3−1.3.6 suggests that it will only run with apache 1.3.6 but it actually will run with all apaches >= 1.3.6

4.2. auth_ldap

4.2.1. What is auth_ldap

Quoting www.rudedog.org

auth_ldap is an LDAP authentication module for Apache, the world's most popular web server. auth_ldap has excellent performance, and supports Apache on both Unix and Windows NT. It also has support for LDAP over SSL, and a mode that lets MicrosÜ1 Frontpage clients manage their web permissions while still using LDAP for authentication.

From the authors point of view:

If you like to consolidate your login−facilities to a common user/passwd base, LDAP is the right way. LDAP is an open standard and widely supported.

Login−facilities for LDAP:

Unix−Logins for Linux, Solaris (others?) FTP−Logins (some ftp−daemons) http Basic Authentication Tarantella Authentication and Role−Management Samba Authentication (2.2.x should support this) LDAP is role based. That means, i.e. you can define a role "manager" assign a user as member and that user can login wherever a manager is allowed to login

4.2.2. Download the source

Origin−Site: http://www.rudedog.org/auth_ldap/

4.2.3. Building and installing

```
cd /usr/local

tar -xvzf auth_ldap-1.6.0.tar.gz

cd auth_ldap-1.6.0

./configure --with-apxs=/usr/local/apache/bin/apxs \
--with-sdk=openldap

make
make install
```

4.3. mod_auth_mysql

4.3.1. What is mod_auth_mysql

It is a http−Basic Authentication Module. It allows to maintain your user comfortable in a MySQL−Database

4.3.2. Download the source

Origin−Site: ftp://ftp.kciLink.com/pub/mod_auth_mysql.c.gz

4.3.3. Building and installing

```
gunzip mod_auth_mysql.c.gz

/usr/local/apache/bin/apxs \
-c -I/usr/local/mysql/include \
-L/usr/local/mysql/lib/mysql \
-lmysqlclient -lm mod_auth_mysql.c

cp mod_auth_mysql.so /usr/local/apache/libexec/
```

Add this line in your httpd.conf:

```
LoadModule mysql_auth_module libexec/mod_auth_mysql.so
```

And where the other modules are added:

```
AddModule mod_auth_mysql.c
```

Take care that the path of Mysql libs and includes are correct!

Notice: Be sure that /usr/local/mysql/lib/mysql is in /etc/ld.so.conf before compiling

Use AuthMySQLCryptedPasswords Off or it does not work! (under investigation why not)

4.3.4. Sample configuration

4.3.4.1. /usr/local/apache/conf/httpd.conf

```
<location /manual/>
  AuthType Basic
  AuthUserfile /dev/null
  AuthName Testing
  AuthGroupFile /dev/null
  AuthMySQLHost localhost
  AuthMySQLCryptedPasswords Off
  AuthMySQLUser root
  AuthMySQLDB users
  AuthMySQLUserTable user_info
  <Limit GET POST>
    require valid-user
  </limit>
</location>
```

4.3.4.2. Script for creating the MySQL−Database

just type:

```
mysql < authmysql.sql
```

The File authmysql.sql contents:

```
  create database http_users;
  connect http_users;

  CREATE TABLE user_info (
  user_name CHAR(30) NOT NULL,
  user_passwd CHAR(20) NOT NULL,
  user_group CHAR(10),
  PRIMARY KEY (user_name);
```

4.4. mod_dynvhost

4.4.1. What is mod_dynvhost

It is a module that allows to define new Virtual Host "on−the−fly". Just create a new Directory in your vhost−path, thats it. It is not need to restart your Webserver

It is a good solution for Mass−Virtual−hosting for ISP's

4.4.2. Download the source

Origin−Site: http://funkcity.com/0101/projects/dynvhost/mod_dynvhost.tar.gz

4.4.3. Building and installing

```
cd /usr/local

tar -xvzf mod_dynvhost.tar.gz

cd dynvhost/

/usr/local/apache/bin/apxs -i -a -c mod_dynvhost.c
```

Notice: Take a look at httpd.conf if mod_dynvhost.so is loaded at startup:

```
LoadModule dynvhost_module libexec/mod_dynvhost.so
```

4.4.4. Sample configuration

4.4.4.1. /usr/local/apache/conf/httpd.conf

```
<DynamicVirtualHost /usr/local/apache/htdocs/vhosts/>
  HomeDir /
</DynamicVirtualHost>
```

Now create a Directory for each virtualhost in /usr/local/apache/htdocs/vhosts/

i.e.

/usr/local/apache/htdocs/vhosts/foo.bar.org

You don't need to restart your Webserver

4.5. mod_roaming

4.5.1. What is mod_roaming

Quoting www.klomp.org/mod_roaming/

With mod_roaming you can use your Apache webserver as a Netscape Roaming Access server. This allows you to store your Netscape Communicator 4.5 preferences, bookmarks, address books, cookies etc. on the server so that you can use (and update) the same settings from any Netscape Communicator 4.5 that can access the server.

From the authors point of view:

This is really cool stuff! Unfortunately it does not work over proxy−connection. You can keep your Netscape 4.x bookmarks etc. synchronized on different machines. It is not supported by any other browsers, including Mozilla and Netscape 6.x

4.5.2. Download the source

Origin−Site: http://www.klomp.org/mod_roaming/

4.5.3. Building and installing

```
cd /usr/local

tar −xvzf mod_roaming−1.0.2.tar.gz

cd mod_roaming−1.0.2

/usr/local/apache/bin/apxs −i −a −c mod_roaming.c
```

Notice: Check httpd.conf if mod_roaming is loaded at startup:

```
LoadModule roaming_module libexec/mod_roaming.so
```

Should be installed automatically

4.5.4. Sample configuration

4.5.4.1. /usr/local/apache/conf/httpd.conf

```
RoamingAlias /roaming /usr/local/apache/roaming
<Directory /usr/local/apache/roaming>
  AuthUserFile /usr/local/apache/conf/roaming-htpasswd
  AuthType Basic
  AuthName "Roaming Access"
  <Limit GET PUT MOVE DELETE>
    require valid-user
  </Limit>
</Directory>
```

5. Compressed delivery

There are basically two modules available for output compression: mod_gzip and mod_gunzip. They are using different approaches to reach the the goal of bandwidth reduction.

mod_gunzip expects compressed file on the filesystem, and uncompress them if the browser cannot handle compressed data. The benefit is a low cpu−usage, because most browsers are capable to handle gzipped content. On the oder side, most of today's content is served dynamically i.e. PHP, and this content will be delivered uncompressed.

mod_gzip does not need compressed files on the system, all defined content will be compressed before delivery. The benefit is to have the dynamically generated content also compressed, the other side is a higher cpu−usage, because every request has to be compressed on−the−fly. Mod_gzip can handle already compressed data i.e. index.html.gz and send it as−is.

The conclusion: You carefully have to make a decision which of the two modules makes more sense for you. If you have to pay for every GB delivered and CPU−power does not matter, then mod_gzip is the choice for you. If response time matters (delay between request and delivery), and your bandwidth is cheap or unlimited, mod_gunzip matches your needs better.

A good page that helps you to make this decision is Martin Kiff's document about mod_gunzip http://www.innerjoin.org/apache−compression/howto.html

5.1. mod_gzip

5.1.1. Download the source

Origin−Site: http://www.ehyperspace.com/htmlonly/products/mod_gzip.html

5.1.2. Building and installing

```
/usr/local/apache/bin/apxs -i -a -c -lz mod_gzip.c
```

5.1.3. Sample configuration

Put the following in your `/usr/local/apache/conf/httpd.conf`:

```
mod_gzip_on                  Yes
mod_gzip_can_negotiate       Yes
mod_gzip_dechunk             Yes
mod_gzip_minimum_file_size   600
mod_gzip_maximum_file_size   0
mod_gzip_maximum_inmem_size  100000
mod_gzip_keep_workfiles      No
mod_gzip_temp_dir            /usr/local/apache/gzip
mod_gzip_item_include        file \.html$
```

```
mod_gzip_item_include       file \.txt$
mod_gzip_item_include       file \.jsp$
mod_gzip_item_include       file \.php$
mod_gzip_item_include       file \.pl$
mod_gzip_item_include       mime ^text/.*
mod_gzip_item_include       mime ^application/x-httpd-php
mod_gzip_item_include       mime ^httpd/unix-directory$
mod_gzip_item_include       handler ^perl-script$
mod_gzip_item_include       handler ^server-status$
mod_gzip_item_include       handler ^server-info$
mod_gzip_item_exclude       file \.css$
mod_gzip_item_exclude       file \.js$
mod_gzip_item_exclude       mime ^image/.*
```

5.2. mod_gunzip

5.2.1. Download the source

Origin−Site: http://www.oldach.net/mod_gunzip.tar.gz

5.2.2. Building and installing

```
tar -xvzf mod_gunzip.tar.gz
cd mod_gunzip-2

/usr/local/apache/bin/apxs -i -a -c -lz mod_gunzip.c
```

5.2.3. Sample configuration

Put the following in your `/usr/local/apache/conf/httpd.conf`:

```
AddType text/html .htmz
AddHandler send-gunzipped .htmz
```

Now you can gzip your html files and rename them to i.e:

```
gzip index.html
mv index.html.gz index.htmz
```

Of course you have to change all links to htmz, i.e. Some page

6. mod_php and its prerequisites

6.1. What is mod_php

Quoting www.php.net

PHP is a server−side, cross−platform, HTML embedded scripting language.

In the beginning it was just a simple guestbook−processor, and it was growing and growing Since Version 3 it is really powerful Webdevelopment−language

From the authors point of view:

Since Version 4 PHP capable and robust enough for enterprise webapplications. It is powerful, supports almost all important Databases natively, and other through ODBC. It a few times faster than ASP on M$−Systems on the same Hardware.

There are other extensions available like apc which speed up processing about 50−400% (depends on the php−code you wrote)

6.2. Prerequisites

Depending on your needs there are some software to install first. One already installed Software according this document is MySQL, because its needed by mod_auth_mysql.

6.2.1. IMAP client

6.2.1.1. What is IMAP client

IMAP means "Internet Mail Application Protocol" and is a substitute for the POP protocol. It allows to keep all Mails in different folders on the server, which (should) be backed up − Never again lose important email, because your local harddrive crashed

---

6.2.2. Download the source

Origin−Site:

Origin−Site: http://www.washington.edu/imap/

6.2.3. Building and installing

```
cd /usr/local

tar −xvfz imap.tar.Z
```

```
cd imap

make slx SSLTYPE=unix
```

Notice:

imap.tar.Z is usually a symlink to the latest release, today its linked to imap−2001a.tar.Z

If you don't need SSL−Support, remove the "SSLTYPE=unix" behind the make command

6.2.4. PostgreSQL

6.2.4.1. What is PostgreSQL

PostgreSQL is a very Powerful and fast Database

Like MySQL wonderful for Webapplications. From my Point of view, not as comfortable to handle as MySQL. If your Webapplication performs mostly writes, or you need proofed transaction−capabilities, PostgreSQL is your friend

6.2.4.2. Download the source

Origin−Site: http://www.postgresql.org (Select a mirror close to you)

6.2.4.3. Building and installing

```
cd /usr/local

tar -xvzf postgresql-7.2.1.tar.gz

cd postgresql-7.2.1

./configure \
--with-perl \
--enable-odbc \
--with-unixodbc  \
--with-pam \
--with-openssl \


make
make install

echo /usr/local/pgsql/lib >> /etc/ld.so.conf

ldconfig
```

6.2.5. pdflib

6.2.5.1. What is pdflib

Quoting www.pdflib.com

PDFlib: A library for generating PDF on the fly PDFlib is the premier software component if you want to generate PDF on your server, convert text and graphics, or implement PDF output in your own products.

From the authors point of view:

This is a commercial Product. Read the license carefully to see if you need a commercial license or not

---

6.2.5.2. Download the source

Origin−Site: http://www.pdflib.com/pdflib/download/pdflib−4.0.2.tar.gz

---

6.2.5.3. Building and installing

```
cd /usr/local/
tar -xvzf pdflib-4.0.2.tar.gz

cd pdflib-4.0.2

./configure --enable-shared-pdflib --enable-cxx

make
make install

ldconfig
```

6.2.6. gettext

6.2.6.1. What is gettext

gettext is a library for i18n (Internationalization, "I", 18 chars and "n") of software, and needed by php

---

6.2.6.2. Download the source

Origin−Site: ftp://ftp.gnu.org/gnu/gettext (select a mirror close to you)

6.2.6.3. Building and installing

```
cd /usr/local

tar -xvzf gettext-0.11.1.tar.gz
```

```
cd gettext-0.11.1

./configure

make
make check
make install

ldconfig
```

6.2.7. zlib

6.2.7.1. What is zlib

zlib is a lossless data−compression library for use on virtually any computer hardware and operating system

6.2.7.2. Download the source

Origin−Site: ftp://ftp.info−zip.org/pub/infozip/zlib/zlib−1.1.4.tar.gz (select a mirror close to you)

6.2.7.3. Building and installing

```
cd /usr/local

tar -xvzf zlib-1.1.4.tar.gz

cd zlib-1.1.4/

./configure

make
make test
make install

ldconfig
```

6.3. Building and installing PHP4

```
cd /usr/local

tar -xvzf php-4.1.2.tar.gz

cd php-4.1.2


./configure \
--with-apxs=/usr/local/apache/bin/apxs \
--with-mysql=/usr/local/mysql \
--with-pgsql=/usr/local/pgsql \
--enable-track-vars \
--with-openssl=/usr/local/ssl \
```

```
--with-imap=/usr/local/imap-2001a \
--with-gd --with-ldap \
--enable-ftp \
--enable-sysvsem \
--enable-sysvshm \
--enable-sockets \
--with-pdflib=/usr/local \
--with-gettext \
--with-mm=/usr/local/mm-1.1.3 \
--with-jpeg-dir=/usr/lib \
--with-zlib-dir=/usr/local \

make
make install
```

After installing your httpd.conf is modified by axps. It should now look as follows:

```
<IfDefine SSL>
LoadModule ssl_module         libexec/libssl.so
LoadModule php4_module        libexec/libphp4.so
</IfDefine>
```

If you compiled Apache with mod_ssl then the php−module will only be loaded when staring Apache with ssl (apachectl startssl). If you will start Apache without ssl support (but compiled like described in this document) you need to change this:

```
<IfDefine SSL>
LoadModule ssl_module         libexec/libssl.so
</IfDefine>
LoadModule php4_module        libexec/libphp4.so
```

Copy the sample php.ini to /usr/local/lib/php.ini

```
cp /usr/local/php-4.1.2/php.ini-dist /usr/local/lib/php.ini
```

uncomment (remove the # at begin of line) the following lines in /usr/local/apache/conf/httpd.conf

Apache 1.3.24 default httpd.conf does lack of this entries. You have to add them instead of uncommenting

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

you may wish to add the following:

```
AddType application/x-httpd-php .phtml
AddType application/x-httpd-php .php3
```

Restart apache by issuing the following command:

```
/usr/local/apache/bin/apachectl restart
```

7. PHP extensions

There are many different extensions available for php, which can be added in your php.ini

7.1. APC (Alternative PHP−cache)

7.1.1. What is APC

Quoting www.apc.communityconnect.com/

APC is the Alternative PHP Cache. It was conceived of to provide a free, open, and robust framework for compiling and caching php scripts. APC was conceived of to provide a way of boosting the performance of PHP on heavily loaded sites by providing a way for scripts to be cached in a compiled state, so that the overhead of parsing and compiling can be almost completely eliminated. There are commercial products which provide this functionality, but they are neither open−source nor free. Our goal was to level the playing field by providing an implementation that allows greater flexibility and is universally accessible. We also wanted the cache to provide visibility into it's own workings and those of PHP, so time was invested in providing internal diagnostic tools which allow for cache diagnostics and maintenance. Thus arrived APC. Since we were committed to developing a product which can easily grow with new version of PHP, we implemented it as a zend extension, allowing it to either be compiled into PHP or added post facto as a drop in module. As with PHP, it is available completely free for commercial and non−commercial use, under the same terms as PHP itself. APC has been tested under PHP 4.0.3, 4.0.3pl1 and 4.0.4. It currently compiles under Linux and FreeBSD. Patches for ports to other OSs/ PHP versions are welcome.

From the authors point of view:

The author made some performance−Tests with apc and it was real surprise. A PHP−Webpage with mysql−queries in a loop (total 10 queries) was more than 50% faster

Contra APC: If you have other users on the system coding php they maybe are not comfortable with APC, because the changes are all ignored unless you reset the cache or restart Apache. The other way, namely that APC checks the php−script for a newer version before every run costs speed.

7.1.2. Download the source

Origin−Site: http://apc.communityconnect.com/sources/apc−cvs.tar.gz

7.1.3. Building and installing

```
cd /usr/local

tar −xvzf apc−cvs.tar.gz

cd apc

./configure −−enable−apc −−with−php−config=/usr/local/bin/php−config
```

```
make
make install

cp modules/php_apc.so /usr/local/lib/php/extensions

echo 'zend_extension="/usr/local/lib/php/extensions/php_apc.so"' \
>> /usr/local/lib/php.ini
echo "apc.mode = shm" >> \
apc.mode = shm
```

Restart your Apache−Webserver. Try it out, create a php−file with the following content:

```
<?php
apcinfo();
?>
```

7.2. Zend−Optimizer (Do _NOT_ combine with APC−Cache!)

7.2.1. What is Zend−optimizer

Quoting www.zend.com

The Zend Optimizer goes over the intermediate code generated by the standard Zend run−time compiler located in the Zend Engine, and then optimizes it for faster execution.

From the authors point of view:

Zend−Optimizer is a freeware closed source product. On the same testcode used for the APC−test, there was speed−decrease of about 5% compared to PHP without APC

You have to make your own test, to see, if you have some improvements with your own code. Be sure not to NOT use Zend Optimizer together with APC, or your whole setup will not work

7.2.2. Download the binary

Origin−Site: https://www.zend.com/store/free_download.php?pid=13

You have to register yourself at zend.com to get access to the download−page

7.2.3. Installing

There is noting to build, this product is closed−source and so only available as binary for different platforms. The filename varies according your platform, the sample is for Linux on IA32.

```
cd /usr/local

tar −xvzf ZendOptimizer−1.2.0−PHP_4.1.0−Linux_glibc21−i386.tar.gz
```

```
cd ZendOptimizer-1.2.0-PHP_4.1.0-Linux_glibc21-i386.tar.gz

cp ZendOptimizer.so /usr/local/lib/php/extensions

echo 'zend_optimizer.optimization_level=15' \
>> /usr/local/lib/php.ini
echo'zend_extension="/usr/local/lib/php/extensions/ZendOptimizer.so"' \
>> /usr/local/lib/php.ini
```

8. Server Side Java

8.1. mod_jserv

8.1.1. What is mod_jserv

Quoting java.apache.org

Apache JServ is a 100% pure Java servlet engine fully compliant with the JavaSoft Java Servlet APIs 2.0 specification. Since we believe in the great value complete portability between different servlet platforms, together with portability of Java binary code, we worked very close with JavaSoft and other JVM implementers to provide correct interpretations or Java specifications both for servlets and for the Java language itself.

From the authors point of view:

mod_jserv is very comfortable to maintain

Unfortunately mod_jserv is no longer actively developed, only bugfixes if bugs where found. Tomcat is the successor of mod_jserv As soon as the author gets enough time and experience with Tomcat he will extend this HOWTO.

JSDK2.0 is out of Date too. It is needed for mod_jserv only.

8.1.2. Prerequisites

8.1.2.1. JSDK 2.0 (mod_jserv req.)

8.1.2.1.1. What is JSDK

It is the API from Sun Microsystems for developing Servlets

8.1.2.1.2. Download

Origin−Site: http://java.sun.com/products/servlet/archive.html

---

8.1.2.1.3. Building and installing

Since JSDK is closed source there is nothing to build.

Notice: The filename "jsdk20−solaris2−sparc.tar.Z" suggests to be a binary for sparc platform −> ignore it, it will work on all platforms

```
cd /usr/local/

uncompress jsdk20−solaris2−sparc.tar.Z
```

```
tar -xvf jsdk20-solaris2-sparc.tar
```

8.1.3. Download the source

Origin−Site: http://java.apache.org/jserv/dist/

8.1.4. Building and installing

```
cd /usr/local

tar -xvzf ApacheJServ-1.1.2.tar.gz
cd ApacheJServ-1.1.2

./configure \
--prefix=/usr/local/jserv \
--with-apxs=/usr/local/apache/bin/apxs \
--with-JSDK=/usr/local/JSDK2.0 \
--enable-EAPI \

make
make install
```

It is important to use flag −−enable−EAPI if apache is compiled with mod_ssl!

8.2. jakarta−tomcat

8.2.1. What is Tomcat

Quoting jakarta.apache.org

Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed by Sun under the Java Community Process.

From the authors point of view:

Tomcat is the successor of jserv which is no longer developed. Tomcat supports the latest jsp and servlet−APIs defined by sun. Unfortunately Tomcat is very difficult to build from source, because it is using its own building−system called "ant". There is also a very long list of prerequisites if you want to build from source. See http://jakarta.apache.org/tomcat/tomcat−4.0−doc/BUILDING.txt for more details − Good luck, and give some feedback to the author.

In the meantime the HOWTO is providing some basic support for Tomcat installed from binaries.

The Author is searching for some volunteers who tries to build Tomcat from source and tells what steps are required

8.2.2. Prerequisites

8.2.2.1. Java2

8.2.2.1.1. What is Java2>

Quoting java.sun.com

Too much for this HOWTO, please see http://java.sun.com/j2se/1.3/docs/relnotes/features.html

---

8.2.2.2. Download the binaries

Go to http://java.sun.com/j2se/1.3/ ,choose your platform and follow the steps on the site.

8.2.2.3. Installing the binaries

Execute the binary:

```
chmod +x j2sdk-1_3_1_02-linux-i386.bin

./2sdk-1_3_1_02-linux-i386.bin
```

Notice: There is a newer version of java available, if you do not care about possible instability you can downloads version 1.4.0

After accepting the license, unpack the stuff and move the resulting directory to "/usr/lib"

8.2.3. Download the binaries

Origin−Site:
http://jakarta.apache.org/builds/jakarta−tomcat−4.0/release/v4.0.1/bin/jakarta−tomcat−4.0.1.tar.gz/

8.2.4. Installing the binaries

```
cd /usr/local

tar -xvzf jakarta-tomcat-4.0.3.tar.gz

cd jakarta-tomcat-4.0.3

cd bin

rm *.bat

echo export JAVA_HOME=/usr/lib/java/ >> /etc/profile
. /etc/profile
```

Now you should be able to startup tomcat:

```
/usr/local/apache/jakarta-tomcat-4.0.3/bin/startup.sh
```

You should now be able to connect to: http://localhost:8080/index.html

For the further steps like installing your servlets and jsp−files, you are responsible by yourself...

Notice: Since the author is NOT a Java−Programmer he will not be able to help you with Java−Problems! The author is an experienced Sysadmin and could you give some hints: Be sure your CLASSPATH Variable is set right. This is the most common error done. Have fun, and as mentioned above, the author welcomes some feedback to be able to provide more information to the community.

---

9. Further Information

Here are some other resources available on the internet

9.1. News groups

Some of the most interesting news groups are:

- alt.apache.configuration
- comp.infosystems.www.servers.unix
- alt.comp.lang.php
- alt.php
- comp.databases

Also check out your country newsgroups e.g ch.comp.os.linux

Most newsgroups have their own FAQ that are designed to answer most of your questions, as the name Frequently Asked Questions indicate. Fresh versions should be posted regularly to the relevant newsgroups. If you cannot find it in your news spool you could go directly to the FAQ main archive FTP site. The WWW versions can be browsed at the FAQ main archive WWW site.

9.2. Mailing Lists

9.2.1. `<users@httpd.apache.org>`

Send an empty email to `<users-subscribe@httpd.apache.org>`

Before writing to the list, check out the archive: http://marc.theaimsgroup.com/?l=apache−httpd−users

9.2.2. `<modperl@apache.org>`

Send an mail to `<modperl@apache.org>` with the content (not subject):

```
subscribe modperl
```

Before writing to the list, check out the archive: http://outside.organic.com/mail−archives/modperl/

9.2.3. `<openssl-users@openssl.org>`

Send an mail to `<majordomo@openssl.org>` with the content (not subject):

```
subscribe openssl-users
```

Before writing to the list, check out the archive: http://www.mail–archive.com/openssl–users@openssl.org/

9.2.4. <modssl–users@modssl.org>

Send an mail to <majordomo@modssl.org> with the content (not subject):

```
subscribe modssl–users
```

Before writing to the list, check out the archive: http://www.mail–archive.com/modssl–users@modssl.org/

9.2.5. <mysql@lists.mysql.com>

Send an empty mail to <mysql–subscribe@lists.mysql.com>

Before writing to the list, check out the archive: http://lists.mysql.com/cgi–ez/ezmlm–cgi/

9.2.6. <pgsql–general@postgres.org>

Fill out the subscription form at http://developer.postgresql.org/mailsub.php

Before writing to the list, check out the archive: http://archives.postgresql.org/pgsql–general/

9.2.7. <pgsql–general@postgres.org>

Fill out the subscription form at http://www.php.net/mailing–lists.php

There are several php related mailinglist to subscribe, some of them are also available on php.net's newsserver

Before writing to the list, check out the archive that are linked also on the subscription–page

9.2.8. <apc–cache@lists.communityconnect.com>

Send an mail to <apc–cache–request@lists.communityconnect.com> with the content (not subject):

```
subscribe
```

9.3. HOWTO

These are intended as the primary starting points to get the background information. They also show you how to solve a specific problem. Some relevant HOWTOs are

`Apache-Overview-HOWTO`, `Apache-WebDAV-LDAP-HOWTO`, `LDAP-HOWTO`, `LDAP-Implementation-HOWTO` and the `PHP-HOWTO`

The main site for these is the LDP archive

9.4. Local Resources

Usually distributions install some documentation on your system. Usually they are located in `/usr/share/doc/packages` or `/usr/local/share/doc`

The software products mentioned here provide a lot of documentation in their source−directories. Apache does install its documentation in the default DocumentRoot `/usr/local/apache/htdocs/manual`

9.5. Web Sites

There are a large number of informative web sites available. By their very nature they change quickly, so do not be surprised if these links become outdated very fast.

A good starting point is of course the Linux Documentation Project home page, a central information repository for documentation, project pages and much more.

To get more information about the Software mentioned in this document, then the following sites are good starting points.

- http://httpd.apache.org
- http://www.openssl.org
- http://www.modssl.org
- http://perl.apache.org/
- http://www.webdav.org
- http://www.mysql.com
- http://www.postgresql.org
- http://www.pdflib.com
- http://www.php.net
- http://www.phpbuilder.com

Please let me know if you have any other leads that can be of interest.

10. Questions and Answers

Here I answer the questions which I got from users. If you don't find an answer feel free to contact me

- Q: Why you don't add a description howto compile and setup mod_xyz?

  A: Because nobody requested it yet and I either did not know about a mod_xyz, or I did not found it useful. Feel free to write me some suggestions what to add to the HOWTO. If there is more than one request, and it makes sense, it will maybe added in further releases.

- Q: If my clients are connecting to https://myserver.org an errormessage similar to this appears "Certificate not valid"

  A: The certificate produced like described in this HOWTO is just a self−signed certificate. This means the CA (Certification Authority) is you. Your CA is not recognized as a valid CA by your users browser. You can either install the certificate on your users machines (Makes sense in small Intranet environments) or buy a certificate from a CA that is recognized by all major browsers. An example of such a CA is Verisign http://www.verisign.com. Such a certificate cost approx. 300 USD a year, depending on the strength of the key (56 or 128 Bits)

- Q: When I request a php file, the browser want to download it. Whats wrong?

  A: You forgot to tell apache what to do with the php files. So the php files are not processed by the php engine. To do so, add the application type like described in Section 6.3

- Q: Does this HOWTO also work on other platforms?

  A: Not sure, Solaris should work, AIX and HP−UX do not. I did not got the time to try FreeBSD yet. My goal is to provide a version of the HOWTO for all major Un*x platforms.