63

Apache

Web Server

Table of Contents

Table of Contents

Apache Web Server

Note: Exercises belonging to some themes are shown in a grey area. These exercises are only good for the SuSE Distribution and the Apache that is provided with it.

1. Introduction of http protocol history

- Document server need with basic formatting and links
- First Web Browsers 'Mosaic': Graphic Oriented
- First Web Server programmed by Tim Berners-Lee at CERN
  CERN= Centre Europeen de Recherche Nucleaire, Switzerland
  2nd Web Server was made in USA by US. Gov. at NCSA
  NCSA= Nastioanl Center for Supercomputing Applications
- Apache was built on collection of code and ideas of most popular HTTP servers ..... A-Patch!
- First Apache 1994-1995
- Runs on: - Linux(process copies, from Version.2.xx will have threads)
  - NT (threaded Daemon, not so secure)
  - Windows 98(less stable threads, run from command line)
  - Mac OS(from version 1.3.6 on)
  - Solaris, AIX, OS/2, 680x0, PowerPC-based Mac, BeOS
- Set-up through Configuration file and its directives
- Modules: Core is small but can contain or load modules
  - From version 1.3: dynamic loading of modules
    Disadvantage is bigger memory need and slower
  - 3rd party modules are available: *mod_fastcgi*, *mod_perl*, etc.
- More Memory the better the performance

2. How to install it

- Via YaST
  - 'n' series 'Apache' software
  - 'modify config file' START_HTTPD=yes

- Via a downloaded file (`http://www.apache.org`)
  - Uncompress
  - Compile with needed features

3. First try of Apache

Use one of the Browsers:

Text Browsers:	**`lynx`** and **`w3m`**
Graphic Browsers:	**Netscape, Mozilla, Opera, Arena, Konqueror, Browsex Galeon** and others

- http://localhost

- Help on this page (Bottom right)

- Edit the page title a bit and reload the page:

  - `/usr/local/httpd/htdocs/index.html`
    'Willkommen bei SuSE Linux'
    change to **'Willkommen bei 'Mario' Linux'**

- Connect to the other participant's modified pages.

4. HTTP Protocol

4.1 - HTTP Format

Method | URI(Uniform Resource Identifier) | version | headers
Note: Headers can modify the behaviour of the request (the 'what to do')

4.2 - Try a HTTP request by hand:

- use `ethereal` to capture `lo` device port 80

In xterm:

```
telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.0 <Enter> <Enter>
```

```
HTTP/1.1 200 OK
Date: Fri, 02 Jun 2000 15:53:28 GMT
Server: Apache/1.3.12 (Unix)  (SuSE/Linux) DAV/0.9.14 mod_perl/1.21 mod_ssl/2.6.2
OpenSSL/0.9.5
Connection: close
Content-Type: text/html       <-----IMPORTANT This line describes the MIME type
<HTML>
<HEAD>
<TITLE>Apache HTTP Server - Beispielseite</TITLE>
</HEAD>
<BODY bgcolor=#ffffff>
<H1> Der Apache WWW Server </H1> <BR>
Diese Seite soll nur als Beispiel dienen.
Die <A HREF="./manual/">Dokumentation zum Apache-Server</A> finden Sie hier.
.........
```

4.3 - Watch a Netscape generated HTTP request

In Netscape **`http://localhost <enter>`**

In ethereal:(capture **lo** device)
- Stop the capture after Netscape showed response
- Click on a captured Packet from http protocol
- in Menu **Tools--->Follow TCP Stream**

```
GET / HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.72 [en] (X11; I; Linux 2.2.14 i586)
Host: localhost
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: en, de
Accept-Charset: iso-8859-1,*,utf-8
```

4.4 - List of http methods: (See also section 14.5 for <Limit *method* > Directive)

----- HTTP/0.9 -------- (normally never used)

GET Get a header and resource from the server.

POST Send information<data> to the server
(response can contain confirmation)

------ HTTP/1.0 --------

HEAD Get a header only without resource.

------ HTTP/1.1 --------

OPTIONS Return the list of methods allowed by the server.

TRACE Trace a request to see what the server sees.

DELETE Deletes a resource on the server.
(normally not allowed)

PUT Create or change a file on the server.

CONNECT Enables Proxys to switch to a tunnel mode. For SSL
Use the AllowCONNECT directive to enable it.

Extra Apache methods:
PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, and UNLOC

Exercise: Methods : Try different methods via telnet

```
telnet localhost 80
........

HEAD / Http/1.1         + 2 times <Enter> key

OPTIONS / Http/1.1      + 2 times <Enter> key
```

```
TRACE / Http/1.1
Host: This Host here    + 2 times <Enter> key
```

4.5 - HTTP Clients: (Browsers)

`lynx` and **`w3m`** (ASCII Only)
`Netscape , Mozilla, Opera, konqueror, Nautilus` (Graphic)

5. What is URL and URI

Uniform **R**esource **L**ocator
Uniform **R**esource **I**dentifier
String identifying a resource by name and possibly including location.
example of URL: **`http://`** **`www.elop.de`** **`/bilder/kopf1.jpg`**

```
                 1          2              3
```

1: Protocol
2: ServerAddress
3: Location and Resource(**URI**)

6. Where is what ?

6.1 - Server---- /usr/sbin/httpd

- Server loader script:---------- **`/etc/init.d/apache`**
- Manual loading link :---------- **`/usr/sbin/rcapache`**
- Run levels links to `/sbin/init.d/apache`
---- **`/etc/init.d/rc3.d`** and **`rc5.d`**
- '**`rcapache`**' parameters:

`start`|**`stop`** : Load / Unload httpd Daemon

`restart`: Does a **`start`** then a **`stop`**

`reload`: Keeps httpd running but re-reads **`httpd.conf`**

`status`: Short status eg. (results)
Checking for service httpd: OK

`full-status`: Long server status
(same info as `http://localhost/server-status`)
Note: The server-status must be turned on for localhost to get a result.

6.2 - Configuration files and their order of reading:

- SuSE Distribution
- /etc/httpd/httpd.conf
- /etc/httpd/srm.conf
- /etc/httpd/access.conf

Note: New with Apache 1.3.13 is a feature where if any configuration file is actually a directory, Apache will enter that directory and parse any files (and subdirectories) found there as configuration files. One possible use for this would be to add VirtualHosts by creating small configuration files for each host, and placing them in such a configuration directory. Thus, you can add or remove VirtualHosts without editing any files at all, simply adding or deleting them. This makes automating such processes much easier.

6.3 - Apache Modules

- **/usr/lib/apache/xxxxxxxx.so**

6.4 - Default Log files (settings in httpd.conf)

- /var/log/httpd/access_log
- /var/log/httpd/referer_log
- /var/log/httpd/error_log
- /var/log/httpd/agent_log

6.5 - Documents and Help files:

Apache Help - `/usr/local/httpd/htdocs/manual/index.html`
(apache-doc in serie 'n')
PHP3-Test/Settings/Status - `/usr/local/httpd/htdocs/test.php3`
CGI-Test/mini settings/Status - `/usr/local/httpd/cgi-bin/test.pl`

6.6 - Apache Process ID:

- Running Process ID `/var/run/httpd.pid`
- Killing the httpd process `kill 'cat /var/run/httpd.pid'`
or `killall httpd`

6.7 - Landing zone of httpd (web) clients(DocumentRoot)

- /usr/local/httpd/htdocs

7- Apache options (on command line) for all versions of Apache(Linux,Win,etc.)

7.1 - General Options (see man httpd)

Syntax: **/usr/sbin/httpd -options**

Options:

-D *name*	Defines a name for use in <IfDefine *name*> directives <IfDefine name> is used to define different server global settings and chose which one will be read at start-up of Apache.
-d *ServerRootDir*	Specifies an alternate initial ServerRoot directory.
-f *ConfigFile*	Specifies an alternate configuration file.(ServerConfigFile)
-C *Directive*	Processes this directive before reading config files
-c *Directive*	Processes this directive after reading config files
-v	Display Apache's version number
-h	List valid command line options
-l (small L)	List compiled-in modules
-L	List core configuration directives
-S	Show virtual hosts settings
-t	Run syntax test for configuration files only.

7.2 - For Linux Only:

-X	Single process foreground debugging mode
-R	specify an alternate location for loadable modules

7.3 - For Window95/98 only:

-k restart or shutdown	Start and stop the Apache Server program.

7.4 - WindowNT only:

-i	register a service
-u	deregister a service
-s	do not register a service

8 - Apache Server status and information

8.1 - Server-Status:

- **Use:** Allows to display the server status on remote browsers. It needs the module: *mod_status* to be loaded and installed.

  Important Note: In **SuSE 8.0** and upwards the Module must be enabled in:
  /etc/sysconfig/apache
  HTTPD_SEC_ACCESS_SERVERINFO=yes
- **Configuration Directives involved:**

  ExtendedStatus On (SuSE 7.1 Around line **433**)
  in /etc/httpd/httpd.conf)

  The SetHandler already triggers the server-status in the module *mod_status* when the Location /server-status is requested.

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from localhost
</Location>
```

- **How to access:**

  From allowed host browser as URL:

http://*localhost*/server-status	Full status page
http://*localhost*/server-status/?notables	Full status page without tables for text browsers
http://*localhost*/server-status/?refresh	Send current status every second to browser.
http://*localhost*/server-status/?refresh=10	Send current status every 10 second to browser
http://*localhost*/server-status/?auto	Gives short general statistics of server's activities.

- **Combination of options:**

eg1.	http://*localhost*/server-status/?auto&refresh=10	Gives the statistics every 10 sec.
eg2.	http://*localhost*/server-status/?notables&refresh=10	Gives the server status (without tables) each10 sec.

8.2 - Server Info:

- **Use:** Gives server's internal structure and module list. Needs the *mod_info* to be loaded.
- **Configuration Directives involved:**

  The SetHandler triggers the server-info in the module *mod_info* when the Location /server-info is requested. It should be inserted in a <Location> as follows:

```
<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from localhost
</Location>
```

- **How to access:** From allowed host browser as URL:

http://*localhost*/server-info	Gives a full detailled information page

- **Server Information through PHP3 Page:**

http://*localhost*/test.php3	Gives a very good full long formatted server info.

8.3 - Pearl Info:

- **Use:** Gives perl module environment status. Needs the *mod_perl* to be installed (series 'n').

Mod_Perl is a full perl interpreter in integrated a module

- **Configuration Directives involved:**

(SuSE 7.1 Around line **1261**)

- The SetHandler triggers the perl-script
- The Apache::Status is the internal perl routine used to deliver the status when the Location /perl-status is requested.

```
<IfModule mod_perl.c>
  <Location /perl-status>
          SetHandler perl-script
     PerlHandler Apache::Status
     order deny,allow
          deny from all
          allow from localhost
  </Location>
</IfModule>
```

- **How to access:** From allowed host browser as URL:
  `http://`*`localhost`*`/perl-status` Gives a full detailled information page

9 - Configuration files:

httpd.conf	Standard config file
access.conf	Name set by **AccessConfig** Directive in httpd.conf
srm.conf	Name set by **ResourceConfig** Directive in httpd.conf

Include <*Configfile*>

This directive allows to include extra config files.
Can be repeated at will in httpd.conf
eg.

```
Include conf/virtualhosts_1
Include conf/virtualhosts_2
Include ..........
```

Advantage is some program can be written to generate these included files.

9.1 - Conditional configurations:

Usefullness: - Set temporary testing directives
- Turning ON the *mod_status* debugging tool
- Switching ON the secure server SSL

Command line conditions:

```
httpd -D <configname_1> -D <configname_2>

<IfDefine  configname_1>
      specific configuration directives
      ..............
</IfDefine>
```

Module loading condition:

If a module **is** loaded then do the enclosed directives

```
<IfModule modulename.c>
      directives ......
</IfModule>
```

If a module is **NOT** loaded

```
<IfModule !modulename.c>
      directives ......
</IfModule>
```

9.2 - Configuration files structure:

- If Apache sees an unrecognisable directive, **Apache will refuse to start.**
- Comments start with #
- Directives and comments can have spaces or tabs before them
- The configurations are separated into 3 sections each one overriding the one above it:
    1. **Server Level** (they MUST be outside any container to apply globally)
        - Server only directives
        - Global defaults
    2. **Container level** (selective for each controlled item: dir. files. URL's and Methods)
    3. **Per directory level** (.htaccess files)

10 - Containers

10.1 - Definition:

- Containers allow to limit the scope of the directives enclosed within them.
- Containers Guidelines:
    - All paths that are not having the leading / are assumed to be from the ServerRootDir
    - Reading order of directive blocks (Containers) is as follows:
        - <Directory>
        - .htaccess
        - <DirectoryMatch>
        - <Files> and <FilesMatch> as per config file order
        - <Location> <LocationMatch> as per config file order
        - <VirtualHost>

10.2 - Access control containers:

<**Directory** */dir* > Directory and its subdirectories access directives container
./dir must be an absolute Path

<**DirectoryMatch** "*regex*" >........... Directory and its subdirectories access directives container with regular expressions. *regex* must refer to an absolute path

<**Files** [*path*]*file(s)* >...................... File access directives container.
File(s) without leading '/' in path are relative to DocumentRoot

<**FilesMatch** "*regex*" >................... File access directives container with regular matching expressions.

<**Location** *URI* >........................... URI access directives container.If dir. then it must be absolute path
- Behaves similarly as <Directory> is not limited to the file system.
- It also does not recognize the Options FollowSymLinks.
- The location (URI) given is relative to the DocumentRoot
- The URI always starts with leading / eg. /docs

<**LocationMatch** "*regex*" >........... URI access directives container with regular matching expressions.

<**Limit** *METHOD(s)* >...................... HTTP Methods Directive container. Normally used inside other containers to limit the type of access the client has.
Best use is with authentication.

<**LimitExcept** *METHOD(s)* >........... HTTP Methods Directive container for undefined Methods

.htaccess file .... Per-Directory access directives stored in the directory affected by the directives it contains.
Set by **AccessFileName** directive in httpd.conf

10.3 - Nesting Containers

- Containers of the same type cannot be nested.
- `<IfModule>` and `<IfDefine>` can be nested anywhere
- `<Files>` can be alone or nested inside `<Directory>` only
- `<Limit>` and `<LimitExcept>` can be nested in any other type of container.

11 - Directives

11.1 - Definition:

Keywords placed in a configuration file that affect the functionning of different parts of the Server.

11.2 - Guidelines

1. The directives are either **core** directives or **module** directives:
   1. Command `httpd -L | less` displays all inbuilt core directives compiled with Apache.
   2. **`file:///usr/share/doc/packages/apache/manual/mod/index.html`**
2. Shows each module and their directives.
3. The last directive read overrides all previously parsed ones in the configuration file.
4. Directives can exist alone in the configuration file or .htaccess or within a container.
5. Location of Directives:
   1. Not in a container — Main server and Global Defaults
   2. In a container — Overrides Golbal defaults for the container only.
   3. in **`.htaccess`** files — Per directory directives (see AllowOverrride directive)

11.3 - Basic Server Directives:

ServerName Name of the local server where Apache runs.
This name must be a recognizable FQDN by a DNS.

Port Default port number for the main server.

Timeout Time between the TCP connection buildup and the first HTTP request allowed before the TCP connection is closed.

MaxClients Max number of simulteaneous active servers serving requests.

MaxRequestsPerChild Max number of requests a server will serve before dying.

KeepAlive on/off If on child servers will wait to serve the client for more requests .

StartServers Number of servers to start at startup(before the first request)

MaxSpareServers Maximum spare servers as they are becoming idle.

MinSpareServers Minimum spare servers to start as the load increase.

KeepAliveTimeout Timeout between last sent response and the next request before the TCP connection is closed.

ServerRoot Defines the base (default) location for : logs, Config files etc. SuSE has redefined these locations so now the ServerRoot has very little meaning. It can be used as a relative path to declare other config files without giving the path.

DocumentRoot Defines the Landing Zone for all main server http requests.
In SuSE **`DocumentRoot`** is defined as
`/usr/local/httpd/htdocs`
Take a look via MC.

User & **Group** Sets the user,and group name which identifies the Apache Child servers within the system for ALL http requests.
Run the following command: **`ps -fC httpd`**
See single **root** process and others belonging to **wwwrun**

DirectoryIndex List of filenames of pages that will be sent to client automatically when a directory is requested.
See in **`/etc/httpd/httpd.conf`**

• Apache Kurs Übungen vorbereitung

1. In **`/etc/httpd/httpd.conf`** ganz am Ende die volgende Zeile eintragen:
   `Include /etc/httpd/user.conf`
2. **`/etc/httpd/user.conf`** Datei erzeugen.
   Befehl: **`touch /etc/httpd/user.conf`**
3. Via YaST-1 die **`/etc/hosts`** Auto-Änderungen ausschalten
   yast ---> Administration des Systems ---> Konfigurationsdatei verändern
   Parameter : **`CHECK_ETC_HOSTS = no`**
4. **`/www`** Verzeichnis erzeugen.
   Befehl: **`mkdir /www`**
5. **`/mnt/public7`** und **`/mnt/public8`** Verzeichnisse erzeugen.
   Befehle:
   ```
   mkdir /mnt/public7
   mkdir /mnt/public8
   ```
6. In **`/etc/fstab`** Datei die volgende Eintrage schreiben:
   ```
   192.168.xx.yy:/public/public7 /mnt/public7  nfs  noauto,user  0  0
   192.168.xx.yy:/public/public8 /mnt/public8  nfs  noauto,user  0  0
   ```
   Bemerkung: `192.168.xx.yy` ist die Dozent Rechner Addresse.
7. **nedit** Program von CD installieren.
8. **`/mnt/public7/.nedit`** Datei kopieren nach **`/root/`** Verzeichnis.
9. Anwendungen Icons auf Desktop erzeugen:
   - Title: **USER.CONF** (Desktop **1**)
     Befehl: **`nedit /etc/httpd/user.conf`**
     or **`kwrite -caption USER.CONF /etc/httpd/user.conf`**
   - Title: **RELOAD** (Desktop **1**)
     Befehl: **`xterm -geometry 60x5 -T RELOAD`**
   - Title: **NETSCAPE** (Desktop **2**)
     Befehl: **`netscape`**
   - Title: **Dozent VNC** (Desktop **3**)
     Befehl: **`vncviewv 192.168.`*`xx`*`.`*`yy`*`:1`**
   - Title: **ERROR_LOG** (Desktop **4**)
     Befehl:
     ```
     xterm -geometry 110x20 -fn 9x15 -T "ERROR_LOG" -e tail \
           -n20 -f /var/log/httpd/error_log
     ```

11.3 - Alias:

- Sets a correspondence (shortcut) from anywhere in the file system to a directory relative to DocumentRoot
- It enbles to access resources that are not related to the DocumentRoot
- Advantages over symbolic links:
  - Alias are limited to Apache server they are not accessible from other programs within the system.
- Syntax: **`Alias`** ***`Fakename`*** ***`RealPathName`***
  - e.g. **`/etc/httpd/susehelp.conf`** has a lot of **`alias`** for suse help

Exercise: Set alias to system **`/www/`** directory

- in **`user.conf`** enter:

```
alias /www/  /www/
```

- In Browser:

`http://localhost/www/` You get an Index of **`/www`**

12 - Options:

Note: The use of + or - leading an option simply adds or subtract the option from the already existing ones (e.g. default). Without any sign the options defined are the only ones set.

`All` (Default) Almost all options enabled except Multiviews. Same as :
`Options ExecCGI Includes FollowSymLinks Indexes`

`None` **No options** are set.

`FollowSymLinks` Allows to follow symbolic links. Overrides **`SymLinksIfOwnerMatch`**

Exercise: FollowSymLinks: Link from System **`DocumentRoot`** to **`/www`**

- Create a Symlink **`/usr/local/httpd/htdocs/www2`** pointing to **`/www`**

```
ln -s /www /usr/local/httpd/htdocs/www2
```

- Try **`http://localhost/www2/`**..........NOT ALLOWED
- Add the following entries in **`user.conf`**

```
<Directory /usr/local/httpd/htdocs>
      options +FollowSymlinks
</Directory>
```

- Try **`http://localhost/www2/`**..........ALLOWED. Index of **`/www`** is shown
- Change the System Access rights and disallow **`/www`** to **`wwwrun`** ('**o**ther' access rights) **`chmod 750 /www`**
- Try **`http://localhost/www2/`**..........NOT ALLOWED again
- Allow the system access rights to **`wwwrun`** for **`/www`** back to normal. **`chmod 755 /www`**

`SymLinksIfOwnerMatch` Follows symbolic links only if destination of link is same owner as link.

`Includes` Allows Server-Side Includes(SSI) in html

`IncludesNOEXEC` Allows Server-Side Includes(SSI) in html but not #exec and #include SSI commands.

`Indexes` Allows indexes generation if no DirectoryIndex file set or existing in directory.

Exercise: Indexes: Enable/Disable display of Indexes of Directories

1 - Disabling Indexes for **/www** (accessed via SymLink)

- In **user.conf** enter:

```
<Directory /www>
    Options -Indexes
</Directory>
```

- Try **http://localhost/www2/** Result:Indexes are still shown
- Modify the **<Directory /www>** to
  <Directory /usr/local/httpd/htdocs/www2>
- Try **http://localhost/www2/** Result: NOT ALLOWED
- Put a # in front of **Options -Indexes** to reenable the indexes

2 - Compare Disabling Indexes for **/www/** (accessed via Alias)

- in **user.conf** enter:

```
<Directory /www>
    Options -Indexes
</Directory>
```

- In Browser:
  http://localhost/www/................Result: NOT ALLOWED
- Put a # in front of **Options -Indexes** to reenable the indexes

3 - Disabling Indexes for /www/ (accessed via Alias) using **<Location>**

- in **user.conf** enter:

```
<Location /www>
    Options -Indexes
</Location>
```

- In Browser:
  http://localhost/www/................Result: NOT ALLOWED
- Put a # in front of **Options -Indexes** to reenable the indexes

ExecCGI Allows execution of CGI programs. Almost the same as declaring **ScriptAlias** but here only the files with a recognized cgi extention will be run as CGI.
The **ScriptAlias** and **SetHandler cgi-script** are treating all files in the defined directory as CGI programs.
eg. **AddHandler cgi-script .cgi** directives can be used to define only the type of files that will be treated as CGI Programs.
(See Running CGI section for more details)

Exercise: **ExecCGI**: Set the **/www/cgitest/** Directory to run the **test2.mycgi** program.

- In Browser: **http://localhost/cgitest/test2.mycgi** Source code is shown
- In **user.conf**:

```
<Location /www/cgitest>
      AddHandler cgi-script .mycgi
</Location>
```

- In Browser: **http://localhost/cgitest/test2.mycgi** NOW it runs!
- In **user.conf**:

```
<Location /www/cgitest>
      AddHandler cgi-script .mycgi
      Options -ExecCGI
</Location>
```

- In Browser: **http://localhost/cgitest/test2.mycgi** NOT Allowed

Multiviews Content-negotiated views allowed. Guessing what the client wants when the requested URL does **not** exist. This can be based on the Content-Language value (eg.:de)sent in the http header by the browser in the http request for the page.

See **AddLanguage**, **LanguagePriority** and **DefaultLanguage**.
See Page 142 in *Professional Apache* Book.
eg.

File requested:	index.html (does not exist)
Browser Content-Language	de
First file searched to send:	index.html.de (if not existing then)
Second file searched to send:	index.html.en

(as per LanguagePriority directive)

Exercise: Multiviews: Get different pages as per Browser language setting

- Check in **httpd.conf** approx. line 560 the Options of Directory ***/***
  and note the presence of +Multiviews. It is therefore enabled! for the whole system.
- in Browser**: http://localhost/www/multi/**
  We see the main Apache page with Dancing Pinguin
- We change the name of **index.html** to **index.html.orig**
- in Browser: **http://localhost/www/multi/**
  We see an english web page (index.html.en)
- Disable the Multiviews from **/www/multi** directory

```
<Directory /www/multi>
     Options -Multiviews
</Directory>
```

  We see an index of the **/www/multi** directory.
- Enable back the Multiviews

```
<Directory /www/multi>
     Options +Multiviews
</Directory>
```

- Change the language priority in Browser to **fr, de, en**
- in Browser**: http://localhost/www/multi/**
  We see the french page

XBitHack Sets the scope HTML files will be parsed for SSI commands.

on	All **.html** or **.htm** files with execute permissions on owner is considered a **SSI** file and will be parsed for SSI commands.
off	(Default) .html and .htm files will NOT be parsed by server for SSI commands.
full	Complicated...but can be used to control the caching of proxies making the requests (See page 161 Apache Server Bible)

13 - Directives

Here are a selection of directives related to specific areas of influence in Apache operation

13.1 - Resource access control Directives........ ALLOW-DENY

for <Directory>, <Files>, <Location> and <Limit>
(See page 252 of *Apache Server Bible*)
Default is Allow from all. But **ATTENTION**: since we might set a deny from all on the / directory for basic security precautions then each requested resource must be explicitly allowed one by one (Directories or Locations or files)

Order is only necessary when both `Deny from ...`and `Allow from ...`are used.

`Order allow,deny`	deny rule scope(read last) is overriding conflicting allow ones:
`Order deny,allow`	allow rule scope(read last) is overriding conflicting deny ones:
	Note: Please no space between the , and the deny and the allow

Setting of scope:

`allow from` *xxxx*
`deny from` *yyyy*

xxxx and *yyyy* can be:

All	Apply to everybody (Default for Allow)
None	Apply to Nobody (Default for Deny)
Hostname(s)	Apply to this host only(need DNS)
IP Addr.(s)	Apply to these IP Addresses only eg. `192.168.12.30  192.168.30.12`
partial Nr.(s)	eg. `192.168`
IP Range	eg. `192.168.10.0/255.255.255.0` or `192.168.10.0/24`
NetDomaine	Apply to whole domain e.g. **`.michel.home`**
`env=`*`variable`*	Apply if environment variable matches variable Eg. For controlling access as per browser (for example for VBScript Code): see P.109 of Professional Apache

Exercise:Allow/Deny: Show different ways of access control.

1. Try **`http://localhost/www/`**........Index Appear
2. Add the following entries in **`user.conf`**
```
<Location /www>
      order allow,deny
      Allow from all
      Deny from localhost
</Location>
```
3. Try from Dozent **`http://localhost/www`** and it is NOT ALLOWED
4. Change the Allow to Dozent IP.Addr. and test again. Only dozent can
5. Change the Allow from localhost to **192.168.xx.0/29** (limiting only a part of class)
6. Check with Browser from some participants
7. Demonstrate the Read Sequence of Containers <Directory> and <Location>
## This <Directory> is to show that it has no effect since the <Location> overrides it after
```
<Directory /www/selfhtml>
      <Files selfhtml.htm>
        order allow,deny
        deny from all
      </Files>
</Directory>
<Location /www/selfhtml/selfhtml.htm>
        order deny,allow
        allow from all
</Location>
```
8. Example of limiting access to different Browsers:
```
      BrowserMatch Mozilla Netscape_Browser
      BrowserMatch MSIE    MS_Browser
```

```
        <Location /www/mozilla-test>
          order deny,allow
          deny from all
          allow from env=Netscape_Browser
        </Location>
        <Location /www/MSIE-test>
          order deny,allow
          deny from all
          allow from env=MS_Browser
        </Location>
```

13.2 - ErrorDocument Directive:

This directive allows to change the Server Generated Error pages per error type. Good for Web sites that uses languages other than english.

When using a filename for the document, the **path of the file is RELATIVE** to the DocumentRoot of the server. It is also true for a VirtualHost.

Syntax: **ErrorDocument *errorCode Text|document***

```
eg. ErrorDocument 500 http://foo.example.com/cgi-bin/tester
    ErrorDocument 404 /cgi-bin/bad_urls.pl
    ErrorDocument 401 /subscription_info.html
    ErrorDocument 403 "Sorry can't allow you access today"
```

Exercise: ErrorDocument : Change the error document for a directory in **/www/selfhtml**.

- Create a log direcetory in **/www/selfhtml**

```
     mkdir /www/selfhtml/log
```

- Create 2 error documents:
  - **/www/selfhtml/DocNotFound.html**
  - **/www/selfhtml/DirNotAllowed.html**
- In **user.conf**:

```
     <Location /www/selfhtml>
          ErrorDocument 404 /www/selfhtml/DocNotFound.html
     </Location>
     <Location /www/selfhtml/log>
          order allow,deny
          deny from all
          ErrorDocument 403 /www/selfhtml/DirNotAllowed.html
     </Location>
```

- In Browser:

http://localhost/www/selfhtml/log/	DirNOTAllowed Message
http://localhost/www/selfhtml/xxx.html	DocNOTFound Message

14 - Limiting Access to Directories/Files/URIs and Methods

14.1 - Access control Guidelines:

- The file and directories access attributes for all resources usable by Apache must be set to Read(r) for others - for files and Read(r) and Search(x) for directories. chmod 755 <file/dir.name>
- As Default, the access to resources(files,directories, programs(CGI) etc.) from the Apache is granted. The limiting is done by adding Containers and directives accordingly.
- When a directory is limited, all sub-directories are also limited the same way. To change this limitation for a child directory, a new container directive can be given for this directory. It will then apply to all of its subdirectories.

14.2 - Directories:

```
     Syntax:     <Directory abs.DirPath > ...... </Directory>
                 <DirectoryMatch abs.regex > ...... </DirectoryMatch>
```

- The processing overriding order for <Directory> is as follows:

- Narrower scopes are processed first and override wider scopes(independent of written order):
  - e.g. `<Directory /www/mydir>` directives overrides the `<Directory /www >` directives
- In non-regular expression <Directory> <Files>, wildcards like * and ? can be used
  e.g. `<Directory /www/mydirs.*>` or `<Files /html/seite*.html>`
- A good practice is to start with most restrictive Global default directives and then selectively override the restrictions one by one later in the configuration file as needed.

```
e.g.    <Directory / >          Most restrictive
            Options -FollowSymLinks +Indexes
            AllowOverride None
            order allow,deny
            deny from all
        </Directory>
        <Directory /home >      Allowing for all subdirectories in /home
            order deny,allow
            allow from all
        </Directory>
```

14.3 - Files:

Syntax: `<Files [abs.path/]filename>.....</Files>`
`<FilesMatch regex>.....</FilesMatch>`

- Files **must** be nested within <Directory> only. They cannot be placed alone or inside a <Location>
- They don't recognize the **`Options`** Directive
- They can be selected using wildcards e.g.: * and ?
- The <Directory> where it is used should not conflict with a <Location>. <Location> is read last.
- Can be used inside **`.htaccess`**

Exercise:<Files> : Limiting access of a single file.

- In Browser : **`http://localhost/gif`** Index of pictures appear
- Click on **`apache_logo.gif`** in index and iamge should be shown
- In **`user.conf`**

```
<Directory /usr/local/httpd/htdocs/gif>
    <Files apache_logo.gif>
        Order allow,deny
        deny from all
    </Files>
</Directory>
```

- Click on **`apache_logo.gif`** in index and it should be NOT allowed now

14.4 - Location (URI):

Format: `<Location <relative.URI>.....</Location>`
`<LocationMatch <relative.regex>.....</LocationMatch>`

- Function almost the same as `<Directory>` but have the following differences
  - Locations are URL paths from the browser(extra directory added to the main domain name).
  - They are relative to the `DocumentRoot` directory
  - The can refer to:
    - an existing directory. Its path is relative to the `DocumentRoot`
    - a single file. Its path is relative to `DocumentRoot`
    - an alias directory declared previously through the `Alias` Directive
      - e.g. `Alias /icons/  /usr/local/apache/icons/`
        then the browser document URL can be `http://<servername>/icons/myicon.gif`
        To control this access to this URL the Location would be:

```
<Location /icons/myicon.gif>
        directives......
</Location>
```

  - Behaves similarly as `<Directory>` but is not limited to the file system.

<Location> does not recognize the following:

- `Options FollowSymLinks` and `SymLinksIfOwnerMatch`
- `AllowOverride <overrides....>`
- Nested `<Files...>`
- `ReadmeName, HeaderName, IndexIgnore`

- The URI always starts with leading `/` eg. `/docs`
- If a Location refers to a dir. or dir.alias, `Options [+]indexes` need to be set to get an index of the directory, otherwise Apache tells that it is not permitted....which is not true.
- **Location is read AFTER Files** and therefore overrides it if pointing to the same item.

Exercise :<Location>: Re-enable the acess of a file that was denied through <Directory><Files>

- In Browser : **`http://localhost/gif`** Index of pictures appear
- Click on **`apache_logo.gif`** in index and it should be NOT allowed because of <Directory>
- **In `user.conf:`**

```
<Location /gif/apache_logo.gif>
        order deny,allow
        allow from all
</Location>
```

- Now **`apache_logo.gif`** **<u>is again</u>** Accessible because the Location was read after Directory.

14.5 - Limit (METHODS):

Format: **`<Limit METHOD>.........</Limit> and`**
`<LimitExcept METHOD>.........</LimitExcept>`

- Can be nested in any other container
- <Limit> detects the client's request METHOD defined here and decide on what to do
- <LimitExcept> detects the METHODs that are NOT the ones defined here and decide on what to do.

Exercise 1:<u><Limit></u>:limiting the access through GET method of the **`apache*.gif`** files

- In Browser: **`http://localhost/gif/`** we see the index of /gif dir.
- Click on **`apache_logo.gif`** the image is shown
- In **`user.conf:`**

```
<Location /gif/apache*.gif>
   <Limit GET>
      order allow,deny
      deny from all
   </limit>
</Location>
```

- In Browser: **http://localhost/gif/** we see the index of /gif dir.
- Click on any gif image starting by **apache....** the image is **<u>not allowed</u>**

Exercise 2:<u><LimitExcept></u>:Preventing scripts access from being called by POST method

- Try **`telnet localhost 80`**
  - **`GET /www/cgitest/test1.cgi         all ok`**
  - **`POST /www/cgitest/test1.cgi        all ok`**
- In **`user.conf:`**

```
<Location /www/cgitest/test1.cgi>
        <LimitExcept GET>
              order allow,deny
              deny from all
        </Limit>
</Location>
```

- Try **`telnet localhost 80`**
  - **`GET /www/cgitest/test1.cgi         all ok`**
  - **`POST /www/cgitest/test1.cgi        NOT ALLOWED`** and garbage!!

15 - Indexes

15.1 Sequence of events when a Directory is requested from a browser:

1- Is there a **DirectoryIndex** directive declared for this resource?
If <u>yes</u>: Is the file(s) declared in DirectoryIndex present ?
if yes: Send the first file declared in Directory Index found to Browser.

2 - Is the **Options MultiViews** turned on for this resource ?
if <u>yes</u>: Is the Browser having any preference of language ?

```
                if yes:  Is the file(s) declared in DirectoryIndex with the right extention present ?
                         if yes:  Send the first found file (eg. index.html.en)
                         if no:   Go to Question 3
                if no:   Set the language preference as per LanguagePriority directive setting.
                         Is the file(s) declared in DirectoryIndex with the right extention present ?
                         if yes:  Send the first found file (eg. index.html.en)
    3 - Is the Options Indexes turned on for the requested resource ?
           if yes:  Is the FancyIndexing turned on for this resource ?
                    if yes:  Send the Index of the resource according to FancyIndexing's options
                    if no:   Send a Plain index of the resource.
           if no:   Send ERROR page
```

DirectoryIndex File name of auto-sending file when accessing this dir. (mod_dir.so)

Tip: To force sending an Index of a page use:
`DirectoryIndex dummy` (make sure dummy is not present)

Syntax:

```
   DirectoryIndex htmlfile1 htmlfile2 ......
eg. DirectoryIndex index.htm index.html index.php index.php3
```

Exercise: `DirectoryIndex`: Assign a specific web page to be sent automatically when a Directory is accessed.

- In Browser: **`http://localhost/www/selfhtml/`** The Index is shown
- Add in **`user.conf`:**

```
<Location /www/selfhtml>
  DirectoryIndex selfhtml.htm
</Location>
```

- In Browser: **`http://localhost/www/selfhtml/`** The selfhtml.htm page is shown

AddDescription Adds a description of file(s) or Directory:

Syntax:

```
    AddDescription "Description" Full/partial_file/dir_name
eg. AddDescription "GiF Format Pictures" .gif
```

Exercise: `AddDescription`: Add description for directories and certain files

- In **`user.conf`**:

```
<Directory /www>
   AddDescription "<B>Samba Help Directory</B>" samba
   AddDescription "<B>Deutsche Linux Kurs Verzeichnis</B>" linuxkurs
   AddDescription "<B>Apache Reference Documents</B>" manual
```

- See changes at bottom of **`/www/selfhtml`** directory after entering the following lines.

```
   AddDescription "<B>MS-Word Documents</B>" .doc
   AddDescription "<B>WAVE Fromat Sound File</B>" .wav
   AddDescription "<B>Web Pages</B>" .html .htm shtml .php3 .php
   AddDescription "<B>Java Applet File</B>" .class
   </Directory >
```

- **Note:** Watch out for files having the same name as the directories
- To Change the size of the Description field to unlimited:

```
     IndexOptions DescriptionWidth=*
```

AddIcon Associate icons to files with specific extention :
Note: The ***`iconURL`*** is the DocumentRoot **relative** path of icon filename.

Syntax: **`AddIcon iconURL Full/partialFile/Dirname(s)`**

eg. **`AddIcon /icons/file1.gif .txt .text`**

Exercise: `AddIcon`: Adding Icons for the **`/www`** Directories

1. Install **image Manager** from series 'kpa'
2. Check the icons generated by Apache as default Icon for Directories. as well as the icons in **`/www/selfhtml`**
3. See line 997 of `httpd.conf`

   ```
   AddIcon /icons/folder.gif ^^DIRECTORY^^
   AddIcon /icons/blank.gif ^^BLANKICON^^
   ```
4. Add some or all of the following AddIcon directives and try the difference

```
        AddIcon /www/gif/icons/hand.right.gif     multi
        AddIcon /www/gif/icons/binhex.gif         mozilla-test
        AddIcon /www/gif/icons/binhex.gif         msie-test
        AddIcon /www/gif/icons/world1.gif         samba
        AddIcon /www/gif/icons/continued.gif      bashshell
        AddIcon /www/gif/icons/generic.gif        selfhtml
        AddIcon /www/gif/icons/box1.gif           webalizer
        AddIcon /www/gif/icons/burst.gif          gif
        AddIcon /www/gif/icons/generic.red.gif    .html .htm .php .php3 .shtml
```

5. See that the cgitest directory has retained its server default AddIcon. of **unknown.gif**

AddIconByEncoding

Assign icons as per recognized Encoding MIME type

`AddIconByEncoding /icons/zipfile.gif  x-gzip`

AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType

Assign icons by **MIME-Type**:

`AddIconByType (HTML, /icons/htmlfile.gif) text/html`

Search for mod_autoindex.c in httpd.conf, there are more examples.

The extentions for the files refered as a certain MIME type are declared in the file **`/etc/httpd/mime.types`**

DefaultIcon

Sets the default icon if file type is not recognized

Syntax: **`DefaultIcon`** ***`iconURL`***

eg. **`DefaultIcon /www/gif/icons/a.gif`**

Exercise: `DefaultIcon`: Change the default Icon for unknown files.

- Check the default icon in **`httpd.conf`** and change it there to **`DefaultIcon /icons/a.gif`**
- Check with browser in **`/www/selfhtml`** at bottom.

HeaderName Name of file that is displayed as Header in the directory index. If the file is an **`.html`** it will be formatted accordignly

Note: The Header (Index of /....) produced by Apache will be removed by this directive and replaced by the content of the file.

IMPORTANT: Only works in <Directory> or .htaccess but **NOT** in <Location>

Exercise: `HeaderName`:

Adding a header to the Index of **`/www/selfhtml` dir.**

– Create a text file called **`header.html`** in **`/www/selfhtml`** directory.

– Include some **HTML formatting** commands

– Add the following in `<Directory /www/selfhtml>`
  `HeaderName header.html`

– In Browser: **`http://localhost/www/selfhtml/`**

ReadmeName Name of file that is displayed as footer in directory index.
If the file is an **`.html`** it will be formatted accordignly
The server generated footer will be replaced by this file.

IMPORTANT: Only works in <Directory> or .htaccess but **NOT** in <Location>

Exercise: `ReadmeName` : Add a footer to the Index of **`/www/selfhtml`**

- Create a text file called `footer.html` in `/www/selfhtml` directory.
- Include some **HTML formatting** commands
- Add the following in `<Directory /www/selfhtml>`

```
ReadmeName footer.html
```

- In Browser: **http://localhost/www/selfhtml/**

IndexIgnore `file1 file2 ...` Hides certain files from the index listing:

Notes: - The subdirectories of this one will enherit from these attributes.
- If it is set for a directory, it cannot be overriden by `.htaccess`.
If not then it can be written into the `.htaccess` if Override is activated with `AllowOverride Indexes`.

IMPORTANT: Only works in <Directory> or .htaccess but **NOT** in <Location>

Exercise: IndexIgnore : Hide `header.html` and `footer.html` in `/www/selfhtml`

1. In Netscape: **http://localhost/www/selfhtml/**
   `header.html` and `footer.html` files are displayed
2. add the **IndexIgnore in Location:**

```
<Location /www/selfhtml>
  IndexIgnore header.html footer.html
......
</Location>
```

3. In Netscape: **http://localhost/www/selfhtml/** again
   `header.html` and `footer.html` files are not visible.
4. To hide the Item **Parent Directory**, add '**..**' in the **IndexIngnore** list

```
IndexIgnore header.html footer.html ..
```

5. In Netscape: **http://localhost/www/selfhtml** again
   Parent Directory item is gone.

FancyIndexing *On/Off*

No Parameters. Its presence turns it ON.
Allows to display Fancyier indexes instead of old regular ones.
NOTE: Turning this directive ON/OFF has only an effect if the FancyIndexing Option of IndexOptions (below) has been turn off with the IndexOptions -FancyIndexing

```
FancyIndexing On
```

Exercise: FancyIndexing : Turning off the fancy Indexing of `/www/selfhtml/`

1. **Disable** the **FancyIndexing** twice in Location**:**

```
<Location /www/selfhtml>
  FancyIndexing off
  IndexOptions -FancyIndexing
  ......
</Location>
```

2. Check with Browser : **http://localhost/selfhtml/** No FancyIndexing

IndexOptions

Options for Indexing.
IMPORTANT: If used, then set above `FancyIndexing off`,
Instead use the following indexing options:
Any option can be truned on or off by adding a '+' or '-' before the option.

eg. **`indexOptions +FancyIndexing -FoldersFirst -IconsAreLinks`**

`FancyIndexing` Same effect as above(`FancyIndexing on`)

`DescriptionWidth={n | *}` Sets the width in characters for the Index description field.
If * is given then the width is as long as the longest description.

`IconsAreLinks` Make icons also links

`IconHeight=`*`pixels`* Height of icons

`IconWidth=`*`pixels`* Width of icons

`FoldersFirst` Displays Folders on top of the Index before the files

`NameWidth=`*`n`* Specifies the width of the File/Directory Name.
If n=* then the width is as long as the longest name.

`ScanHTMLTitles` Scan HTML files for TITLE tags and uses the values as the file description.
Important: For this function to work it is necessary that no description is given for the .html extention via `AddDescription`

`SuppressColumnSorting` Disables the generation of sortable listings.

`SuppressDescription` Supresses the file description column

`SuppressHTMLPreamble` Apache will use the HTML header of the HeaderName file instead of it's own generated one if:

- HeaderName directive is specified
- The specified file exists
- It has a valid HTML Header

`SuppressLastModified` Suppress the last-modified date and time column

`SuppressSize` Suppress the file size column.

(See page **113** in *Professional Apache* or page **106** in *Apache Server Bible)*

Exercise : `IndexOptions`: Modify the behaviour of Fancy indexing

1. In **`User.conf:`**

```
<Location /www/selfhtml>
   FancyIndexing off
   IndexOptions +FancyIndexing +ScanHTMLTitles
+SuppressLastModified
                     +DescriptionWidth=* +NameWidth=*
    ......
</Location>
```

2. In Browser:

```
        http://localhost/www/selfhtml
```

16 - AllowOverride and .htaccess (allowed only in <Directory> container)

- Sets the set of directives that can be overridden by a per-directory access control file (.htaccess)
  The file name of this file can be changed Globaly or per Directory with the `AccessFileName` directive
  - Parameters are:
    - **All** (**Default**) Allows all directives to be overridden by `.htaccess` - **Dangerous** !!!
    - **AuthConfig** Allows use of authorization directives:

      AuthName Label displayed by browser as authorization title
      AuthType Type of authorization mechanism. Available: basic
      -Needs AuthUserFile and AuthGroupFile to work
      Warning:user and passwd are passed as clear text

AuthUserFile	Filename of list of allowed users and passwords
AuthGroupFile	Filename of list of allowed groups and passwords
AuthDBMUserFile	Filename of list of allowed users and passwords
AuthDBMGroupFile	Filename of list of allowed groups and passwords
require	Selects users/groups that can access the resource Users and groups are listed in above files (Auth...)
Satisfy	Satisfy the allow/deny or user/group or both when both access control directives apply to a resource. Values are: `any` any one of allow/deny or Auth. that is right will do to give access `all` both allow/deny and Auth. must be right to give access

- **FileInfo** Allow to use directives controlling document MIME-types: (page 116 in *Apache Server Bible*)

AddEncoding	Adds type of encoding recognized by its extention
AddLanguage	Adds a language recognized by its file extention
AddType	Adds a document type recognized by its extention
DefaultType	Selects the type of document assumed as default if the document type recognition failed.
AddHandler	Adds a module handler for a file by its extention
SetHandler	Sets a module handler for all files in the directory
ForceType	Forces a type of file for all files of the directory
ErrorDocument	Name of document that will be sent if error occurs
LanguagePriority	Sequence of language choice for Multiviews

- **Indexes** Allow directives controlling the appearance of directory indexes.

AddDescription	Adds a description of a type of file. eg.: AddDescription "Graphics file" *.gif *.jpg *.bmp
AddIcon	Assign icons to files with specific extention : eg. AddIcon /icons/picture.gif *.gif *.jpg *.bmp
AddIconByEncoding	Assign icons as per recognized Encoding type
AddIconByType	Assign icons
DefaultIcon	Sets the default icon if file type not recognized
DirectoryIndex	File name of auto-sending when accessing this dir.
FancyIndexing	No Parameters. Its presence turns it ON
HeaderName	Name of file that is displayed as Header in dir.index.
ReadmeName	Name of file that is displayed as footer in dir.index.
IndexIgnore	Hides certain files from the index listing eg.: IndexIngnore .htaccess *.conf
IndexOptions	Options for Indexing. If used the do NOT use above FancyIndexing directive. Instead use the following indexing options: - FancyIndexing Same effect as above - IconsAreLinks Make icons also links - IconHeight=*pixels* Height of icons - IconWidth=*pixels* Width of icons - etc. (See page 20 -21for more options)

Limit Allow use of directive controlling the hosts access:

`order deny,allow` (or allow,deny)

`allow from` ***`xxxx`***

`deny from` ***`yyyy`***

- **Options** Allow use of **`options`** directives in **.htaccess** for controlling indexes features:

All	All options included except for MultiViews. This is the default setting.
ExecCGI	Execution of CGI scripts is permitted.
FollowSymLinks	The server will follow symbolic links in this directory. Note: even though the server follows the symlink it does not change the pathname used to match against other

	<Directory> sections. Also this option gets ignored if set inside a <Location> section.
Includes	Server Side Includes(SSI) commands are permitted in HTML files.
IncludesNOEXEC	Server Side Includes(SSI) are permitted, but the #exec and #include commands are disabled.
Indexes	If a URL which maps to a directory is requested, and the there is no DirectoryIndex (e.g., index.html) in that directory, then the server will return a formatted listing(index) of the directory.
MultiViews	Content negotiated MultiViews are allowed. This feature is a mechanism for guessing what the client wants when the URL requested doesn't exist.
SymLinksIfOwnerMatch	The server will only follow symbolic links for which the target file or directory is owned by the same user id as the link. Note: this option gets ignored if set inside a <Location> section.

(see Section 17 - Options below and p.101 Prof. Apache)

Exercise: `AllowOverride` and `.htaccess`: Allow controlling of `/www/multi/` from **.htaccess** file.

- Using the Previous Multiviews exercise in the `user.conf` :

```
    <Directory /www/multi>
           Options +Multiviews
           AllowOverride Options Indexes
    </Directory>
```

- In Browser: `http://localhostwww/multi` we get the index.html.xx
- In `/www/multi/.htaccess` :

```
    Options -Multiviews
    AddDescription "Multiviews Document" *.html.*
    AddDescription "Powered by Apache Image" apache_pb.gif
    IndexIgnore test.php3 robots.txt date.php3
```

- In Browser: `http://localhost/www/multi` we get the Index with descriptions
- Click on `/gif` directory and see that the `apache_pb.gif` image has the same description as above directory.

17 - Virtual Hosts (IP Based and Name Based)

The next example supports 2 IP addresses(IP Based) for the same ethernet card and 2 Virtual Hosts per Address(name based). The number of Virtual Hosts per IP address is unlimited....well almost.

The default virtual host for each served IP addr. is taken from the first one read in the Virtual Hosts configurations for this IP Address.

17.1 - Set the Virtual hosts Names in /etc/hosts or in DNS(/var/named/xxx.zone):

e.g. for name based Virtual Host we would enter the following entry in DNS Table.

```
manual                IN A    192.168.10.60
```

or in `/etc/hosts:`

```
192.168.10.60   www.manual.de
```

Note: If the browser is connecting to the Apache via a Proxy server then the Proxy server will take care of the name resolution(local 'hosts' file or DNS), otherwise the computer where the browser is should resolve the name via local 'hosts' file or via DNS.

17.2 - Viewing the Virtual Host configuration for the server:

```
/usr/sbin/httpd -S
```

17.3 - The Listen Directive

The listen directive is used to tell the server to listen to more than one Interface and port.
It is **not** needed if we are using only the main Host address and port 80. But is is needed for each IPAddr:port combination to be listened to if more than one IP Number or Port are present and NOT all the interfaces in the host are listened to. The recommended syntax is:

```
Listen IPAddress:Port
```

eg.

```
Listen 192.168.10.50:80
```

So the one of the main rules for listen is:

- If we use only the main address and default port of the server then NO Listen.
- If we are using more than one IP address and want all the network cards to be supported then also NO Listen. The server should listen to all cards (physical or virtual) present in the host.
- If we want the server to listen to all the cards in the host but with other ports number than the standard 80 then we need to use the listen with each port number we want to support, including the standard port 80.
- If we want the server to support only certain network cards and not others then Listen directive is needed to specify which card and which port is listened to.

eg.	- Server Listens to all cards in system.	**NO** Listen
	- Server Listens to all cards in system. and to port 8000	**Listen 80** **Listen 8000**
	- Server Listens to only 2 cards in a 4 card system	**Listen *card1IPAddr*:80** **Listen *card2IPAddr*:80**
	- Server Listens to only 2 cards in a 4 card system but on the second card at port 8000	**Listen *card1IPAddr*:80** **Listen *card2IPAddr*:8000**

17.4 - Setting up our first Virtual Host.

Exercise: VirtualHost: Setting-up the Apache Manual as VirtualHost.

- Add the following IP Numbers to **/etc/hosts** :

```
192.168.xx.yy  manual.linux.local   manual apache.linux.local
```

Note: The **192.168.xx.yy** is your own host address.

- Enter the following VirtualHost settings in **user.conf**

```
NameVirtualHost 192.168.xx.yy
<VirtualHost 192.168.xx.yy>
    ServerName manual.linux.local
    ServerAlias manual apache.linux.local
    DocumentRoot /www/manual
    <Location />
         order deny,allow
         DirectoryIndex invoking.html
    </Location>
    TransferLog /www/manual/log/access_log
    ErrorLog /www/manual/log/error_log
```

```
        </VirtualHost>
```

- Create a **/www/manual/log directory:**

```
                mkdir /www/manual/log
```

- If a proxy is used to to to Internet then make sure in Browser Preferences:
  NoProxy for manual.linux.local

2) Exercise 2 for the students to do alone:

Virtual Host for	**www.bash.de**	**same IP Address**
Web Page Location	**/www/bashshell/**	
First Page sent to Browser	**/www/bashshell/bashref.html**	

17.5 - Set-up of Virtual interfaces for IP Based Virtual Hosts:

- To support IP Based Virtual Hosts we need to set-up extra either physical or virtual network interfaces.

- For each extra virtual Interface the manual command (which can and should be inserted in a script) looks like this:
  eg. For the extra address **192.168.20.166**
  as root in terminal: **ifconfig eth0:1 192.168.20.166**

- then in configuration file **NameVirtualHost 192.168.20.166**

17.6 - Examples of Virtual Hosts based on a different IP Address and Port:

IMPORTANT NOTE: Always use IP addresses for **NameVirtualHost** and **VirtualHost.**

- **Exercise-1:** <u>**VirtualHost**</u> **:** Setting-up virtual Host with extra IP Number.
  - in terminal **ifconfig eth0:1 192.168.20.166**
  - in /etc/hosts **192.168.20.166 www.bash.com**
  -

```
NameVirtualHost  192.168.20.166
<VirtualHost  192.168.20.166>
     ServerName www.bash.com
     DocumentRoot /www/bashshell/bourne_shell
</VirtualHost>
```

  - in Browser: **http://www.bash.com**

- **Exercise-2:** <u>**VirtualHost**</u> **:** Setting-up virtual Host with non-standard port number
  - in /etc/hosts **192.168.20.166 www.shell.de**
  - in config file

```
Listen 80
Listen 8000
NameVirtualHost  192.168.20.166:8000
<VirtualHost  192.168.20.166:8000>
  ServerName www.shell.de
  DocumentRoot /www/bashshell/shell_programming
</VirtualHost>
```

  - in Browser: **http://www.shell.de:8000**

17.7 - Automatizing Virtual Hosts settings:

Here is a primitive example of a scrip automatizing the setting-up of one virtual host with one command.

```
#! /bin/sh
# Script for creation of www clients in /www directory
# Syntax: wwwclient clientname servername localIP
#            $0         $1         $2       $3
#
# ----- To do only once by administrator ------------------
# mkdir /www
# chmod 755 /www
# mkdir /etc/dummy
# cp /etc/httpd/httpd.conf /etc/httpd/httpd.conf.orig
#
#----------- Creation of client work space ----------------------
groupadd $1
useradd -mk /etc/dummy -d /www/$1 -g $1 $1
chmod 755 /www/$1
#---- Create a log files directory -only readable from owner -----
mkdir /www/$1/log
chmod 700 /www/$1/log
chown $1.wwwgr /www/$1/log
#----------- Creation of client virtual host ---------------------
echo "#--------- $1 Virtual Host ----------" > /etc/httpd/$1.conf
echo "<VirtualHost $3>" >> /etc/httpd/$1.conf
echo "   ServerName $2" >> /etc/httpd/$1.conf
echo "   DocumentRoot /www/$1" >> /etc/httpd/$1.conf
echo "   ErrorLog /www/$1/log/fehler.log" >> /etc/httpd/$1.conf
echo "   TransferLog /www/$1/log/verbindung.log" >> /etc/httpd/$1.conf
echo "</VirtualHost>" >> /etc/httpd/$1.conf
# --------- Write the Include at the end of httpd.conf file ------
echo "Include /etc/httpd/$1.conf" >> /etc/httpd/httpd.conf
#---------- Write the new address and name into /etc/hosts ------
echo "$3 $2" >> /etc/hosts
#-------------- Asking for the password for the www client--------
passwd $1
#-------- Feedback of what we have created in client config file------
echo -------------Virtual Host Configured---------------------------
cat /etc/httpd/$1.conf
echo -------------End of httpd.conf--------------------------
tail -n2 /etc/httpd/httpd.conf
echo ---------------------------------------------------------
```

Exercise-2: `VirtualHost` : Setting-up multiple virtual Hosts.

- **Definition of exercise:**
  - Transfer and Error logs for every Virtual Hosts in `/log` directories
  - **Alias** of **`/apachehelp/`** pointing to **`/www/manual/`** who works for all
  - **`Bashshell`:** Needs - DirectoryIndex (basheref.html)
    - Other Names for server : bash
  - **`Linuxkurs`:** Needs: - Other names (alias) for server.
    `linuxkurs` and **`linuxhelp.linux.local`**
    - Force showing an Index.
    - Auto Descriptions based on HTML Titles
    - block access to `/log` Directory for all except local Host (192.168.10.60).

- **`Manual`: -** Multiple names:
  `manual apache.linux.local`
  - Descriptive Index for `/images` directory.
  - Header and footer for the `/images` index.
  Attention: use **`<Directory /www/manual/images>`** for
  HeaderName, **ReadmeName**, and **IndexIgnore**
  - Hide the Header and Footer files from Index
  - Do not allow **`windows.html`** in **`/`** to be seen by dozent
- **`Selfhtml`:** Needs settings via **`.htaccess`** file of:
  - DirectoryIndex of **`selfhtml.htm`**
  - Deny access to **`xweb.gif`** (no web image at start page)
- **`samba:`** Needs **-** Another IP Nr.
  - port 8000
  - deny access to **`inx.html`** (index of samba book)
  - ErrorDocument for not allowed documents (error 403) Use the one from selfhtml exercise.

Solutions of exercise 3:

```
NameVirtualHost 192.168.10.60
alias /manual/ /www/manual/

<VirtualHost 192.168.10.60>
     ServerName bashshell.linux.local
     ServerAlias bashshell
     DocumentRoot /www/bashshell
     <Location />
          order deny,allow
          allow from all
          DirectoryIndex bashref.html
     </Location>
     TransferLog /www/bashshell/log/access_log
     ErrorLog /www/bashshell/log/error_log
</VirtualHost>

<VirtualHost 192.168.10.60>
     ServerName linuxkurs.linux.local
     ServerAlias linuxkurs linuxhelp.linux.local
     DocumentRoot /www/linuxkurs
     <Location />
          order deny,allow
          DirectoryIndex dummy
          FancyIndexing off
          IndexOptions DescriptionWidth=*
          IndexOptions +FancyIndexing +ScanHTMLTitles
     </Location>
     <Location /log>
          order deny,allow
          deny from all
          allow from 192.168.10.60
     </Location>
     TransferLog /www/linuxkurs/log/access_log
     ErrorLog /www/linuxkurs/log/error_log
```

```
</VirtualHost>

<VirtualHost 192.168.10.60>
      ServerName manual.linux.local
      ServerAlias manual apache.linux.local
      DocumentRoot /www/manual
      <Location />
          order deny,allow
          DirectoryIndex invoking.html
      </Location>
      <Directory /www/manual/images>
          AddDescription "JPEG Format Image" .jpg
          AddDescription "GIF Format Image" .gif
          AddDescription "Unknown Text File" .fig
          HeaderName header.html
          ReadmeName footer.html
          IndexIgnore header.html footer.html
      </Directory>
```

```
<Location /windows.html>
           order allow,deny
           deny from localhost
       </Location>
       TransferLog /www/manual/log/access_log
       ErrorLog /www/manual/log/error_log
</VirtualHost>

<VirtualHost 192.168.10.60>
      ServerName selfhtml.linux.local
      ServerAlias selfhtml
      DocumentRoot /www/selfhtml
      <Directory /www/selfhtml>
           order deny,allow
           AllowOverride Indexes Limit
      </Directory>
      TransferLog /www/selfhtml/log/access_log
      ErrorLog /www/selfhtml/log/error_log
</VirtualHost>
--------------------------------------------
```

(The content of **/www/selfhtml/.htaccess** is)

```
           DirectoryIndex selfhtml.htm
           <Files xweb.gif>
                 order allow,deny
                 deny from all
           </Files>
```

----- IP: 192.168.10.80 -- Port 8000 ----------------

```
Listen 80
listen 8000
NameVirtualHost 192.168.10.80:8000

<VirtualHost 192.168.10.80:8000>
      ServerName samba.linux.local
      ServerAlias samba
      DocumentRoot /www/samba
      ErrorDocument 403 /DocNotAllowed.html
      <Location /inx.html>
           order allow,deny
           deny from all
      </Location>
      TransferLog /www/samba/log/access_log
      ErrorLog /www/samba/log/error_log
</VirtualHost>
```

17.8 - Redirection of Virtual Hosts

There is quite a number of different ways a URL can be redirected. It all depends on a few factors like where is the destination URL relative to the given URL. Here are some of the redirecting types:

Definitions:	Given_URL:	URL given by client Browser
	Redir_URL:	URL where the given URL should be redirected to.

17.8.1 - Same Server , Same IP for Given_URL and Redir_URL

Redirection Method:

ServerAlias Directive: VirtualHost has 2 names or more.

Syntax:

```
ServerName Redir_URL
ServerAlias Given_URL
```

Exercise1: **Redirection**: **www.samba.de** has alias as **www.linuxkurs.de**

- in **/etc/hosts**

```
    192.168.xx.yy    www.samba.de   www.linuxkurs.de
```

- in **user.conf**

```
    <VirtualHost 192.168.xx.yy>
         Servername www.samba.de
         Serveralias www.linuxkurs.de
         DocumentRoot /www/samba
    </VirtualHost>
```

- in Browser

```
     http://www.samba.de
     http://www.linuxkurs.de
```

17.8.2 - Same Server , different IPs for Given_URL and Redir_URL

Redirection Method:

Same **DocumentRoot** for both **www.linuxkurs.de** and **www.samba.de**

Syntax:

```
<VirtualHost ....>
     ServerName Destination_URL
     DocumentRoot Given_URL_DocumentRoot
</VirtualHost>
<VirtualHost ....>
     ServerName Given_URL
     DocumentRoot Given_URL_DocumentRoot
</VirtualHost>
```

Exercise2: **Redirection:www.linuxkurs.de** gets the same resources as **www.samba.de**

- in **/etc/hosts**

```
    192.168.xx.yy   www.samba.de
    192.168.xx.zz   www.linuxkurs.de
```

- in **user.conf**

```
<VirtualHost 192.168.222.71>
      Servername www.samba.de
      DocumentRoot /www/samba    <----same DocumentRoot
</VirtualHost>

<VirtualHost 192.168.222.171>
     Servername www.linuxkurs.de
     DocumentRoot /www/samba     <----same DocumentRoot
</VirtualHost>
```

17.8.3 - Different Server, different IP for Given_URL and Redir_URL

Redirection Method:

Redirect directive. **www.linuxkurs.de** redirects to **www.samba.de**

Syntax: **Redirect** ***DocumentDir RedirURL***

eg. **Redirect / http://www.mydocs.com**

Details:

In one server:

```
<VirtualHost ....>
     ServerName Destination_URL
     DocumentRoot Given_URL_DocumentRoot
</VirtualHost>
```

In the other server:

```
<VirtualHost ....>
     ServerName Given_URL
     DocumentRoot /empty_directory
     Redirect / Destination_URL
</VirtualHost>
```

Note: To achieve a proper redirection from a VirtualHost, make sure that there are no containers inside the Given_URL's VirtualHost refering to the same Directory, neither via <Directory> nor <Location>.

Exercise3: **Redirection**:**www.linuxkurs.de** gets the same resources as **www.samba.de**

- Create an empty directory: **/www/umleitung**
- In **/etc/hosts**

```
    192.168.xx.yy   www.samba.de
    192.168.xx.zz   www.linuxkurs.de
```

- In **users.conf**

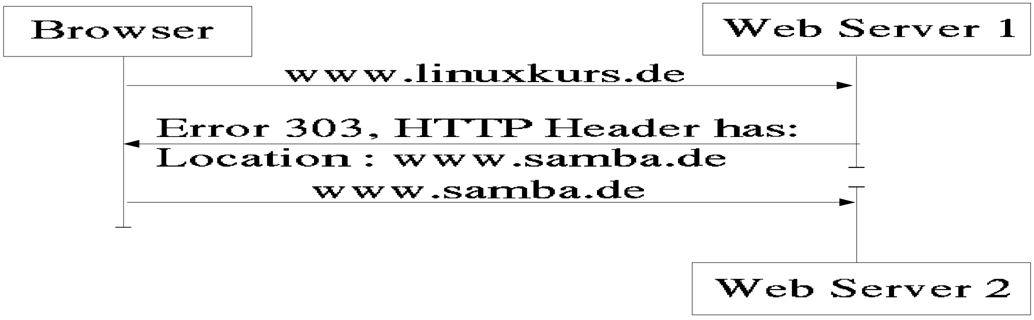
```
    <VirtualHost 192.168.222.71>
            Servername www.samba.de
            DocumentRoot /www/samba
    </VirtualHost>

    <VirtualHost 192.168.222.171>
            Servername www.linuxkurs.de
            DocumentRoot /www/umleitung
            Redirect / http://www.samba.de
    </VirtualHost>
```

- in Browser

  http://www.samba.de
  http://www.linuxkurs.de

- **Redirect** Directive effect/functionning:



18 - Running CGI Programs (Common Gateway Interface)

18.1 - Principle:

- CGIs can be of different languages as long as they observe the behavior of standard CGI definitions. The CGI can be compiled programs or interpreted scripts
- The first line of a CGI script must have the path and name of the script interpreter in the following format:
  `#!`*`/path/and/filename/of/interpreter parameters`*
  - e.g.1. `#!/bin/sh` Shell interpreter
  - e.g.2. `#!/usr/bin/pearl -w` Pearl Interpreter
  - e.g.2. `#!/usr/bin/python` Python Interpreter

18.2 - Process of running CGI (GET Method) - typical example of keyword search

- The Browser receives a form with fields to fill in.
- The Client fills in the fields presses on the Search button
- The browser sends the request to run a cgi program with the entered fields values
  e.g. `GET http://www.bestsearch.com/cgi-bin/search.cgi?books=law&author=murphy`
- The Apache sets the environment variables:
  REQUEST_METHOD = GET
  QUERY_STRING=books=law&author=murphy
- Apache runs the requested CGI program (`/cgi-bin/search.cgi`)
- The search.cgi program runs by:
  - Reading the REQUEST_METHOD and see if it is a GET method.
  - If yes then it processes the content of QUERY_STRING
  - When finished it writes the Content-Type (MIME Type) or result to STDOUT
  - Then writes the found result to STDOUT
  - The program search.cgi end its operation...dies!!
- Apache detects the exit of the cgi program
- Apache search the STDOUT to find the Content-Type and produces a HTML Header with the Content-Type
- Apache reads the STDOUT (rest of cgi result) and send it to the browser

18.3 - Process of running CGI (POST Method) - typical example is keyword search

- The Browser receives a form with fields to fill in.
- The Client fills in the fields presses on the Search button
- The browser sends the request to run a cgi program with the entered fields values
  e.g. `POST http://www.bestsearch.com/cgi-bin/search.cgi`
  `books=law&author=murphy` are encoded and sent with the request
- Apache sets the environment variables:
  REQUEST_METHOD = POST
  CONTENT_LENGTH = *Data_Length_of_Received_Fields*
- Apache decodes the encoded data and send it to the STDIN of the search.cgi program
- Apache runs the requested CGI program (/cgi-bin/search.cgi)
- The search.cgi program runs by:
  - Reading the REQUEST_METHOD and see if it is a POST method.
  - If yes then it reads the content of STDIN and processes it
  - When finished it writes the Content-Type or result to STDOUT
  - Then writes the found result to STDOUT
  - The program search.cgi end its operation...dies!!
- Apache detects the exit of the cgi program
- Apache search the STDOUT to find the Content-Type and produces a HTML Header with the Content-Type
- Apache reads the STDOUT (rest of cgi result) and send it to the browser

18.4 - Apache environment variables passed to CGI programs:

- Valuable info of the Apache environment and settings can be used by any CGI program.
- This information is passed to the CGI programs by setting environment variables for each CGI program before it runs it.
- These environment variables are:(see p.185-191 Apache Server Bible)

- **Server Variables**

  SERVER_SOFTWARE
  SERVER_ADMIN
  DOCUMENT_ROOT

Client request information variables

SERVER_NAME	HTTP_HOST	HTTP_ACCEPT
HTTP_ACCEPT_CHARSET	HTTP_ACCEPT_LANGUAGE	HTTP_USER_AGENT
HTTP_REFERER	HTTP_CONNECTION	SERVER_PORT
REMOTE_HOST	REMOTE_PORT	REMOTE_ADDR
REMOTE_USER	SERVER_PROTOCOL	REQUEST_METHOD
REQUEST_URI	REMOTE_IDENT	AUTH_TYPE
CONTENT_TYPE	CONTENT_LENGTH	SCRIPT_NAME
SCRIPT_FILENAME	QUERY_STRING	PATH_INFO
PATH_TRANSLATED		

18.5 - Running -cgi- Scripts in Virtual hosts

18.5.1 - HTML Forms format for sending data to a CGI

HTML Forms can be run using the HTTP Methods: GET or POST to pass on Data to the CGIs. **Appendix -M** shows an example of a Form that will send its data via the GET method.

- **18.5.2 - AddHandler** and **SetHandler** Directives
  - The **AddHandler** is used to associate files with specific extentions to certain handlers.
  - The **SetHandler** is used to associate the current scope (Directory or Location) with a specific Server Handler regardless of the files extentions.
  - **Handlers:**

    Here is a list of core handlers already accessible by Default:

`cgi-script`	Conternt (HTML Page) generated by a CGI script.
`default-handler`	Static web pages generation
`imap-file`	ImageMap Rule File
`perl-script`	Content generated by a mod_perl script.
`send-as-is`	File already includes HTTP Headers and is sent as is
`server-info`	Apache generated server information HTML page
`server-status`	Apache generated server status HTML page
`server-parsed`	Server-Side-Include file
`type-map`	Content selection type map.

- **18.5.3 - Mixed CGI-Scripts and HTML files in the same directory**

eg.

```
<VirtualHost 192.168.10.166>
   DocumentRoot /www/vhost1
   ServerName vhost1.michel.home
   <Location />
        AddHandler cgi-script .cgi
        (all .cgi files in this virtual Host will be run as scripts)
   </Location>
</VirtualHost>
```

- **18.5.4 - Exclusive Scripts Directories**

  Syntax: **`ScriptAlias <False_Name> <Real_System_Dir_Path>`**

e.g.

```
<VirtualHost 192.168.10.166>
   DocumentRoot /www/vhost1.michel.home
   ServerName vhost1.michel.home
   ScriptAlias  /allcgi/  /www/vhost1.michel.home/cgi-bin/
</VirtualHost>
```

Note: the **`ScriptAlias`** is sufficient to enable the cgi execution of the whole defined resource(directory or file(s)) without the need to add the **`options ExecCGI`** and **`SetHandler cgi-script`**. These last 2 directives are almost always together.

18.5.4 - Examples of Handlers settings:

--------- **`ScriptAlias`**, **`options ExecCGI, SetHandler`** --------
The Directive:

```
ScriptAlias /cgi-bin/  /www/vhost1/cgi-bin/
```

Is equivalent to:

```
<Directory /www/vhost1/cgi-bin>
     AllowOverride None
     options ExecCGI
     SetHandler cgi-script
</Directory>
```

besides being equivalent it adds an alias to the main server
(Default for all VirtualHosts)

------------- **`options ExecCGI, AddHandler`** ---------
To declare specific files types as CGI-Script::

```
<Directory /home/foo/cgifiles>
     AllowOverride none
     Options ExecCGI
     AddHandler cgi-script .mycgi .cgi
</Directory>
```

To declare multiple file types as CGI-Scripts::

```
<Directory /home/foo/cgifiles/*.cgi">
     AllowOverride none
     Options ExecCGI
     SetHandler cgi-script
</Directory>
```

Exercise-1: `ExecCGI, SetHandler and AddHandler`:

- In **user.conf**

```
  <VirtualHost 192.168.10.60>
    ServerName cgitest2.linux.local
    DocumentRoot /www/cgitest
  </VirtualHost>
```

- In Browser: **`http://cgitest2.linux.local`**
  and click on the test2.mycgi......Text only
- add the following in above VirtualHost container in **`user.conf`**:

```
    <Location />
         order deny,allow
         Options +ExecCGI
         AddHandler cgi-script .mycgi
    </Location>
```

- In Browser: **`http://cgitest2.linux.local`**
  and click on the test2.mycgi......CGI Runs

Exercise-2: **`Running CGI`**: Run our first Shell and Perl CGI

- Setup Virtual Host **`www.erstecgi.de`** in **`/www/erstecgi`**
- Set it to run **`.mycgi`** and **`.pl`** as CGI(**`AddHandler`**)

```
    <Location />
         Options +ExecCGI
         AddHandler cgi-script .mycgi .pl
    </Location>
```

- **`http://www.erstecgi.de/test1.mycgi`**
- **http://www.erstecgi.de/test1.mycgi?Name=joe&Address=Haupstr.+18&Ort=Hof**
- **`http://www.erstecgi.de/test4.pl`**

- **Exercise-3: `FORMS and CGI`**: Running a form and a cgi responding to the form.

- Create a FORM (**`anmeldung.html`**) in **`/www/erstecgi`(see Appendix M)**

- Create a **`test1.mycgi`** in **`/www/erstecgi`** to respond to the form by feeding back the values sent by the form. **(see Appendix M)**

- **`http://www.erstecgi.de/anmeldung.html`**

Exercise-4: `FORM-CGI-Visitor's Log`: Create a visitors log

- Create an empty file owned by **`wwwun`** called **`visitors.cvs`**

- Add the section of Besucher into the CGI for writing the parameters into the file and displaying the file back to Browser.

- **`http://www.erstecgi.de/anmeldung.html`**

Exercise-5: `SUDO and root commands`:

Run **`/sbin/fdisk -l`** command via a CGI using Sudo in it.

- Edit the **`/etc/sudoers`** using **`visudo`** command.

```
root    ALL=(ALL) ALL
Host_Alias THIS_HOST=hof400
Cmnd_Alias SYSTEM=/sbin/fdisk -l,/sbin/modprobe ppa
wwwrun THIS_HOST=NOPASSWD:SYSTEM
```

- Add the command in the **`/www/erstecgi/test1.mycgi`**:

```
echo "<Center><H1>Festplatteliste</H1></Center><BR>"
sudo /sbin/fdisk -l | sed -e 's/.*$/&\<BR\>/'
echo "<HR>"
```

- **`http://www.erstecgi.de/test1.mycgi`**

Tip: To prevent any Proxy to save the result of a CGI or a static HTML file then enter the following meta tag at the beginning of the file:

`<Meta http-equiv="expires" content="0">`

19 - CGI Wrapper : suEXEC (page 79 of Professional Apache book)

This feature allows Apache to run CGI scripts under a different user name and group than the one assigned to Apache's main server(wwwrun).

Note: The suEXEC feature and its settings must be enabled at compile time of Apache.

```
# ./configure --enable-suexec ........
```

-If the suEXEC is enabled correctly in Apache compilation, then the following message will appear in the main server's error log: (/var/log/httpd/error_log)

```
suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
```

- Any error occuring regarding the suEXEC? then look in the following log file for info on what caused it: **`/var/log/httpd/suexec.log`** (SuSE)

19.1 - Advantages of suEXEC:

Since all clients in Apache are working as wwwrun and nogroup or similar, all CGI's from one VirtualHost can access and change and run CGIs or change the files of other VirtualHosts. This CGI Wrapper allows CGIs from each VirtualHost that desires so to run as the user and group they that owns the VirtualHost, therefore avoiding disturbances between Virtual Hosts.

Suggestion:

The suEXEC is best combined with entries in /etc/sudoers for administration programs access restricted to the user of suEXEC.

19.2 - Using suEXEC

There are 2 ways where **suEXEC** will be triggered to run a CGI as another user then the wwwrun (SuSE).

19.2.1 - In a VirtualHost by using the directives 'User' and 'Group'.

If the suEXEC is enabled (in Apache) any CGI that is run from within the VirtualHost will be run as the defined User and Group.

Conditions for suEXEC to work in Virtual Hosts:

1 - The User and Group must be valid in the system.(root is not allowed)

2 - The DocumentRoot of the VirtualHost(s) MUST be a physical subdirectory of the Default DocumentRoot (set at compile time) (SuSE=/usr/local/httpd/htdocs) of the Main Server. No symbolic link! Changing the DocumentRoot of the main server in the `httpd.conf` does not work, because the DocumentRoot was given as being the same as the main server's default DocumentRoot at compile time and cannot be changed without a new compiling.

3 - The directory where the script resides and the script itself MUST belong to the defined User and Group and have the Write access rights for Group and Other set to NOT ALLOWED.

4 - The script MUST have NO SUID or SGID set.

5- The script must be owned by the intended user.

Suggestion for VirtualHosts DocumentRoot:

Set the VirtualHosts DocumentRoot Directories as subdirectories of:

```
/usr/local/httpd/htdocs/Virtual1
   "    "     "     "   /Virtual2   etc.
```

19.2.2 - In a User's Home directories.

If the suEXEC is enabled when Apache starts then any script that will be run from their UserDir (**`public_html`** set in main server) and subdirectories of it will be run under the user's Name and Group.

The browser must use the ~ . eg.

`http://`*`mainservername/~Username/cgiscript`*

Conditions for the suEXEC to work in user's directories.

1 - The directory where the script resides and the script itself MUST belong to the defined User and Group and have the Write access rights for Group and Other set to NOT ALLOWED.
2 - The script MUST have NO SUID or SGID set.

19.3 - Using SUDO with suEXEC for system administration commands

Since a normal user (like the one used by suEXEC) cannot execute system administration commands, we neede to configure SUDO to allow a suEXEC user to execute the ones to be allowed.

19.3.1 - Configuring SUDO

SUDO needs to be configured via editing its configuration file: **`/etc/sudoers.`** It is important to edit this file via the command: **`visudo`**
Which will run the editor set by the environment variable EDITOR and edit the file **`/etc/sudoers`**.
The configuration file syntax is as follows:
For example if we want to allow:

`cgitest` user to use the command
`fdisk -l` for a list of all storage devices
and **`modprobe`** for loading kernel modules.
`isdnctrl dial ippp0` and **`isdnctrl hangup ippp0`**
to allow to dial and hangup the internet connection
to ISP via the ISDN interface.
`ecofarm` user to use only the **`fdisk -l`** and **`lsmod`** commands.

We would enter the following entries in visudo editor:

Declare the local host name via an alias.(just the first name ..not the FQDN)

```
Host_Alias THIS_HOST=laptop
```

Declare the alias for the command(s) to allow users to run
Note: All commands MUST have the full path and the correct allowed options and arguments to be able to be run. NO Space between comma and next command.

```
Cmnd_Alias SYSTEM=/usr/sbin/modprobe ppa,/sbin/fdisk -l
Cmnd_Alias ISDNCTRL=/usr/sbin/isdnctrl dial ippp0, \
                    /usr/sbin/isdnctrl hangup ippp0
```

Declare who has the right to run which type of commands and how.

```
cgitest THIS_HOST=NOPASSWD:SYSTEM,NOPASSWD:ISDNCTRL
ecofarm  THIS_HOST=NOPASSWD:SYSTEM
```

19.3.2 - Using SUDO

To use SUDO the user just need to add the word **`sudo`** in front of the allowed command (in the CGI if the command is issued from there): eg.

```
sudo /sbin/fdisk -l
```

will run the **`/sbin/fdisk -l`** command via **`sudo`**.

Exercise-1 : <u>suEXEC</u>: Run a CGI and another user in VirtualHost.
1. Enter the IP Number of **`cgitest.linux.local`** in **`/etc/hosts`**

```
192.168.30.56    cgitest.linux.local
```

2. Create a virtual network card as **eg.** **`192.168.30.56`**
3. Create a user and group as **`'cgitest'`**

```
groupadd cgitest
useradd -g cgitest -m cgitest
```

4. in **`user.conf`** enter the following:

```
  NameVirtualHost  192.168.30.56
    <VirtualHost 192.168.30.56>
         ServerName cgitest.linux.local
         DocumentRoot /usr/local/httpd/htdocs/cgitest
         User cgitest
         Group cgitest
         <Directory /usr/local/httpd/htdocs/cgitest>
              Options +ExecCGI
              AddHandler cgi-script .cgi
         </Directory>
    </VirtualHost>
```

5. Create a VirtualHost DocumentRoot Directory owned by **cgitest** user.

```
mkdir -m 755 /usr/local/httpd/htdocs/cgitest
cp /www/cgitest/test1.cgi /usr/local/httpd/htdocs/cgitest/test3.mycgi
chown -R cgitest. /usr/local/httpd/htdocs/cgitest/
chown -R cgitest. /usr/local/httpd/htdocs/cgitest/test3.mycgi
```

6. Edit the **/etc/sudoers** file via **visudo** command to include **fdisk -l** command for **cgitest** user.

```
Host_Alias THIS_HOST=laptop
Cmnd_Alias SYSTEM=/sbin/fdisk -l
cgitest THIS_HOST=NOPASSWD:SYSTEM
```

7. In **/usr/local/httpd/htdocs/cgitest/cgitest.cgi** add the commands to get the devices listings:

```
#--- Display block devices existing  in Linux system -----
echo "<Center><H1>System Block Devices</H1></Center><BR>"
sudo /sbin/fdisk -l | sed  -e 's/.*$/&\<BR\>/'
echo "<HR>"
```

8. Enter in Browser: **http://cgitest.linux.local/test3.cgi**

Exercise-2 : suEXEC: Run a CGI and another user in users /home direcetory.

1 - Set the access rights of user's home directory to 705.

```
chmod 705 /home/cgitest
```

2 - Create a subdirectory for the cgi script. (public_html).

```
mkdir -m 755 /home/cgitest/public_html
```

3 - Make this directory be owned by the user.

```
chown cgitest. /home/cgitest/public_html
```

4 - Copy the cgi script into the directory.

```
cp -a /usr/local/httpd/htdocs/cgitest/test1.cgi /home/cgitest/public_html/
```

5- In /etc/httpd/user.conf

```
<Directory /home>
     Options +ExecCGI
     AddHandler cgi-script .cgi
</Directory>
```

6 - Enter in Browser: **http://localhost/~cgitest/test1.cgi**

20 - UNCGI : The GET and POST Parameters wrapper

20.1 - Description of 'uncgi'

Uncgi decodes all the form fields from a GET or a POST HTML Method and sticks them into environment variables for easy use by a shell script, a C program, a Perl script, or whatever you like, then executes whatever other program you specify.
The names of the environment variables that are created using **uncgi** are all starting by `WWW_`*`fieldname`* . The fieldname is the same as the <input name=xxxx> given in HTML form. So for example: from a form having the input fields as follows:

```
<input NAME="Address" TYPE=text VALUE=""> ....</Input>
```

Then uncgi would create an environment variable named **`WWW_Address`** and give it the user entered value. This goes for all from fields being sent from the HTML form to **uncgi**.

20.2 - Getting, Configuring, Compiling and Installing 'uncgi'

The documentation, along with the most recent version of the software, is available via the World-Wide Web at **`http://www.midwinter.com/~koreth/uncgi.html`**.

Unfortunately uncgi doesn't have a way of being configured by a configuration file at startup. The program must be configured for each Virtual Host in its **`Makefile`** before compiling it. Then each compiled program can be placed in the various Virtual Hosts DocumentRoot area for easy use of it. The Makefile just needs to know where will the uncgi be placed (**DESTDIR**) and where it should look for various cgi programs to run(**SCRIPT_BIN**).

After modifying these 2 values in the Makefile just compile it by:
- Change directory(cd...) to where the **`Makefile`** and **`uncgi.c`** are
- Issues the command **`make install`**
The program will compile and be installed in the proper DESTDIR directory.
Do this procedure of editing the **`Makefile`** and compiling it for each Virtual Host where you need the uncgi.
Important: Since uncgi was initially used on freeBSD system, a declaration error may occur during compiling under Linux. To fix that we need to edit the uncgi.c file and add an underscore in the definition as follows:
Before (at line 43):

```
   #ifndef __bsdi__
   extern char *sys_errlist[];
```

After:

```
   #ifndef __bsdi__
   extern char *_sys_errlist[];
```

20.3 - Using uncgi

The use of uncgi is quite simple. The HTML form sends its request to the Apache Web Server via a GET or POST method with its fields content. Apache runs uncgi which creates the extra environment variables(WWW_xxxxx). Then uncgi runs the regular CGI which can enjoy using these variables.

20.3.1 - In HTML Forms

The way to tell Apache to run the uncgi and then the regualr CGI, is done via a path that looks like this:

```
<FORM ACTION="/cgidir/uncgi.cgi/test2.mycgi" METHOD="GET">
```

Where :

- **/cgidir** is where the uncgi.cgi is located (relative to DocumentRoot)
- **uncgi.cgi** is the compiled uncgi program.
- **`test2.mycgi`** is the CGI program to run.

This might look strange since the `uncgi.cgi` is seen here as a directory. Well in fact Apache sees the `uncgi.cgi`, runs it and gives it the `test2.mycgi` as a parameter. In this case `uncgi.cgi` is located in **/cgidir** directory as well as the **test2.mycgi**. The **uncgi.cgi** was compiled with its location(DESTDIR) as being the same path as the one for CGIs to run(SCRIPT_BIN).

20.3.2 - How does the CGI uses it

After the uncgi.cgi has been run and the environment variables has been prepared, it calls the defined CGI and runs it. The defined CGI can then use the created WWW_xxx environment variables (which are all the HTML form fields and their values) to do its work. The regular CGI environment variables are still available as usual.

20.3.3 - Parsing Multiple Choice check boxes:

UnCGI puts hash marks ("#") between checkbox selections if there are several of them. How you parse that depends entirely on what language you're using. In C, use strtok(). In Python, use string.splitfields(). In Perl, use split(). In Bourne shell, do something like:

```
echo $WWW_checkboxname | tr \# \\012 | while read result; do
    echo "checkboxname has value: $result"
done
```

20.3.4 - General procedure to use uncgi

- Edit the **DESTDIR** and **SCRIPT_BIN** in Makefile
  DESTDIR is where the uncgi goes
  SCRIPT_BIN is where are the CGIs that uncgi will run
- Compile the uncgi with command **make install**
- Run the uncgi from the HTML from via the
  <FORM ACTION=/*cgidir*/uncgi.cgi/*mycgi.cgi* ....
- Use the **WWW_*fieldname*** variables in all the CGIs run by uncgi.

Exercise: <u>uncgi</u>: Run a CGI via UnCGI and display new uncgi variables

- Create a directory **/usr/local/uncgi**
- Copy the downloaded uncgi into **/usr/local/uncgi**
- Untar the uncgi : **cd /usr/local/uncgi; tar fvxz uncgi.tar.gz**
- Edit the Makefile and edit the following variables: (**cd uncgi ; mcedit Makefile**)

```
CC=gcc -g
DESTDIR=/www/forms
SCRIPT_BIN=/www/forms
EXTENSION=.cgi
```

- Edit **uncgi.c** and add the underscore'_' to prevent compile errors.

Before (at line 43):

```
    #ifndef __bsdi__
    extern char *sys_errlist[];
```

After:

```
    #ifndef __bsdi__
    extern char *_sys_errlist[];
```

- Compile and install uncgi:
  cd uncgi
  make install (**uncgi.cgi** is compiled and copied to **/www/forms** directory)
- in **/www/forms** directory, make a copy of **whoareyou.html** to **uncgitest.html**
  cp /www/forms/whoareyou.html /www/forms/uncgitest.html
- Change the ACTION in **/www/forms/uncgitest.html** to
  <FORM ACTION=./uncgi.cgi/test2.mycgi ....
- In **test2.mycgi** : add the following section:

```
#------Display only CGI Environment Variables created by 'uncgi' ------
echo "<Center><H1>uncgi generated Environment variables</H1></Center><BR>"
```

```
printenv | grep "WWW_" | sed -e 's/.*$/&\<BR\>/'
echo "<HR>"
```

- In Browser: **`http://localhost/www/forms/uncgitest.html`**
- Fill in the upper form and click on its **send** button.....
  the ***`WWW_xxx`*** variables and their contents are shown. ***`xxx`*** is each variable's name.

21 - Server-Side Includes (SSI and XSSI)

(see p.158 of Apache Server Bible)

21.1 - Definition

Server-Side Includes are imbedded commands inside a normal html page that extend the features of the HTML language. The principle is a bit like PHP3. The files are mostly having the extension .shtml
Requires:

- The module mod_include to be loaded.
- Add a new handler for SSI/XSSI HTML Pages
  `AddHandler server-parsed .shtml`
- Add a new file extension for SSI/XSSI HTML Pages
  `AddType text/html  .shtml`
- Enable SSI parsing for a directory
  `Options +include`
- Embedded SSI and XSSI commands in HTML pages

21.2 - Server-Side programming Language

- The SSI code is seen as comments from the browser (in case it is not processed by server)
- the format is:
  <!-#command argument1=value1 argument2=value2 argument3=value3 .... -->
  - The commands are:

#config errmsg="*error message*"	Defines the error message if error occurs
#config sizefmt=["bytes" | "abbrev"]	Defines the file size info format
#config timefmt= *Formatstring*	Defines the format of time display when needed The FormatString is a %x x=letter meaning a specific format.
#echo var="*VariableName*"	Prints the defined variable to client
#exec cgi="*path/to/cgi/program*"	Execute the defined CGI program
#exec cmd=" *path/to/other/program"*	Execute the defined other program. e.g. perl prgm.
#fsize file="*path/to/file*"	Prints the size of the defined file
#fsize virtual="*URL*"	Prints the size of the defined URL file
#flastmod file="*path/to/file*"	Prints the last modification date of defined file
#flastmod virtual="*URL*"	Prints the last modification date of defined URL file
#include file="*path/to/file*"	Includes an .html .htm or .shtml file
#include virtual="*URL*"	Includes an .html .htm or .shtml URL file

Examples of SSI Includes

```
<!--#exec cmd="(cat /etc/SuSE-release 2>/dev/null || echo SuSE Linux) | head -1" --><BR>
<!--#exec cmd='echo "Host: `hostname -f`, Kernel: `uname -r` (`uname -m`)"' -->
<!--#exec cmd="(cat /etc/SuSE-release 2>/dev/null || echo SuSE Linux) | head -1" --><BR>
<FONT SIZE=-1><!--#echo var="SERVER_SOFTWARE" --></FONT>
```

Execute an imbedded shell script as follows:

```
<!--#exec cmd='
   if test -f /usr/lib/apache/libphp3.so ; then
        echo "  <LI><A HREF=\"/doc/packages/mod_php/doc/manual.html">\
        PHP Handbuch</A>"
    else
       echo "  <LI>PHP is not installed"
   fi
 ' -->
```

See also: `/usr/local/httpd/htdocs/index.html` for more examples.

21.3 - Tech tip: Dynamic log files display

If you want to make a web page based on your server logs (like a who s linking to me page), there s no need to run a cron job to generate HTML. Just put the appropriate HTML tags in a CustomLog directive, and use a server-side **`include`** command to include the log on the page. It s totally real-time, too.

22 - Setting-up Apache as proxy server(s)

(see p.286 Professional Apache)

22.1 - Principle:

Apache main server can be configured to be used as proxy server (in Global Diredtives area) or one or more Virtual Host(s)can be used as proxy server(s).
It serves HTTP, FTP and HTTPS (SSL) requests.

22.2 - Setting it up:

Include the proxy server directives in a Virtual host container and set them up accordingly

Method: - **Select proxy Port number**

to which the Virtual Proxy will listen to
- Extra from the standard directive (port 80) for Web Serving Listen directives must be used:

```
Port    80
Listen  80    # needed !!
Listen  8080  # For the virtual proxy server
```

- **Set-up a Virtual Host as proxy server**

22.2.1 - Minimal Configuration

```
NameVirtualHost  192.168.10.60:8080
<VirtualHost 192.168.10.60:8080>
      <IfModule mod_proxy.c>
            ProxyRequests On|Off       Enable/Disable Proxy Services
            CacheRoot "/dir/of/cache"   Only needed if enable caching is desired
            <Directory proxy:*>        Optional:  Allow to limit the proxy services
                  Order deny,allow
                  Deny from all
                  Allow from localhost .our.domaine
            </Directory>
      </IfModule>
    TransferLog /dir/to/proxy/log/access.log  Proxy requests Access Log file
    ErrorLog /dir/to/proxy/log/error.log      Proxy requests Errors log file
</VirtualHost>
```

22.2.2 - Extra configuration directives: (for the proxy server only)

```
<IfModule mod_proxy.c>
```

Limiting proxy services by protocol

```
<Directory proxy:http:*>   # Allow to limit which hosts can use the http proxy services
      ........Access Directives for http only
</Directory>

<Directory proxy:ftp:*>    # Allow to limit which hosts can use the ftp proxy services
      ........Access Directives for ftp only
</Directory>

<Directory proxy:https:*>  # Allow to limit which hosts can use the https proxy services
      ........Access Directives for https only
</Directory>

<Directory proxy:*/www.special.site.com/*> Limits proxy services for www.special.site.com
      ........Access Directives for www.special.site.com only
</Directory>

ProxyVia On|Off|Full|Block
```

ProxyVia	Enable/disable the handling of HTTP/1.1 "Via:" headers. Possible parameters are:
Full	Adds the server version to the added Via: Header;
Block	Removes all outgoing Via: headers. Including the ones already existing.
On	Adds a conventional Via: header to signal that this doc. is served by proxy
Off	Doesn't add a Via: header but leaves the already existing ones.(default)

Blocking specific web sites from being served (security or decency filtering)

```
ProxyBlock unwanted.domain bad.domaine.com     # Blocks proxying these web sites
```

22.3 - Proxy Redirection

Note: ProxyRemote directive can be given as many time as needed

Redirection as per URL:

`ProxyRemote` ***`Requested.URL remote.proxy.URL:port`***

Redirect this request to another proxy having a specific port

e.g. **`ProxyRemote http://main.site.com http://proxy.remote.com:8080`**

or **`ProxyRemote * http://proxy.remote.com:8080`**

Redirects all Proxy requests to remote proxy

exercise : `RemoteProxy` : Redirecting all requests via squid proxy server

- Install squid and start it
- in **`proxy.linux.local`** VirtualHost in **`user.conf:`**
  `ProxyRemote * http://localhost:3128`
- in Browser: Set the apache proxy in Preferences:
  `proxy.linux.local` port **`8080`**
- in Browser: **`http://selfhtml.linux.local`**
  we get the **`selfhtml.linux.local`** Page via Apache proxy and squid
- Kill squid and retry the **`http://selfhtml.linux.local`** ERROR

NOTE: Try from another computer. Local check doesn't always work.

Redirection as per Protocol

`ProxyRemote` ***`protocol remote.proxy:port`***

Redirects all requests of this *`protocol`* to a remote proxy

Combining direct local VirtualHosts sites serving and Remote Proxy redirection.

If we want to send all requests to a remote proxy but serve the local Virtual Hosts directly:

```
ProxyRemote *  http://proxy.remote.com:8080
```

then either:

```
NoProxy 192.168 (local Virtual Hosts are served locally)
```

or

```
NoProxy Virtual.Host1.Site VHost_IP  .....
```

22.4 - Adding domain automatically to complete the full local site name

instead of using ServerAlias in Virtual Host:

```
ProxyDomain  .my.local.domain
```

This will add the **`.my.local.domain`** after the incomplete local site name

e.g.

`http://www.site1` will be translated as request to

`http://www.site1.my.local.domain`

22.5 - Caching directives

`CacheRoot "/var/cache/httpd"`	Dir. Absolutely needed to enable the caching
`CacheSize <kBytes>`	No. of kBytes used for the cache. **Default=5**..too low. Better 100MB
`CacheGcInterval <Hours>`	Interval in Hours between cache area Garbage collection. **Default=0** Fractions of hours are also allowed. e.g. 1.25 = 75 minutes
`CacheMaxExpire <Hours>`	Hours after which a document will be forced to expire. **Default=24**
`CacheLastModifiedFactor <Factor>`	If no expiration time supplied by document, then expiry time = <*time since Last modified*> x <*Factor*>
`CacheDefaultExpire <No.of Hours>`	No. of hours after which the documents that has unknown last modified time expires from the cache. **Default=1**
`NoCache a_domain.com another_domain.edu`	# No caching performed for these sites
`CacheNegociatedDocs`	If present then content-negotiated documents are cached
`CacheDirLevel No_of_subDirs`	No.of subdirs created for the cache.No need to change **default=3**

22.6 - Example of Virtual Hosts as Proxy server

Note: This following DocumentRoot and <Directory> of the proxy is not necessary but if used it is accesses via **`http://proxy.linux.local:8080`**

Exercise: **`Proxy Server:`** Setting-up a proxy server as Virtual Host

- Make sure we have a `/www/proxy/log` directory

- in **user.conf**:

```
Listen 80
Listen 8080
NameVirtualHost 192.168.10.60:8080
<VirtualHost 192.168.10.60:8080>
      ServerName proxy.linux.local
      DocumentRoot /www/proxy
      <Directory /www/proxy>
            order deny,allow
            allow from all
      </Directory>
      <IfModule mod_proxy.c>
            ProxyRequests On
            <Directory proxy:*>
                  Order deny,allow
                  Allow from all
            </Directory>
            ProxyVia On
            # CacheRoot Directory should be 755 user:wwwrun group:root
            # If not present the proxy doesn't cache
            CacheRoot "/var/cache/httpd"   (made ready by SuSE)
            CacheSize 50000
            CacheGcInterval 4
            CacheMaxExpire 24
            CacheLastModifiedFactor 0.1
            CacheDefaultExpire 1
            #NoCache a_domain.com another_domain.edu
    </IfModule>
    ErrorLog /www/proxy/log/error.log
    TransferLog /www/proxy/log/access.log
</VirtualHost>
```

- Set the Browser proxy to **`192.168.10.60`** port **`8080`**

- In Browser: **`http://selfhtml.linux.local`** we see the selfhtml page

22.7 Use `wget` with proxy server.

To use the wget program through a proxy set the environment variable in bash as follows before running the wget:

```
export http_proxy=192.168.71.9:3128
```

The `wget` has its default to **`--proxy=on`**

To turn it off:
eg.

```
      wget --proxy=off -r http://www.linux.com
```

23 - Log files format and statistics

23.1 - Definition

- Log files are written according to the Common Log Format (CLF) standard.
- The module mod_log_config.c is responsible to write log file.
- The log file name is set by the directive:**TransferLog** and **ErrorLog**

  These directives can be issued many times causing multi files

  `e.g. TransferLog </Absolute/path/to/access/log/access.log>`

23.2 - Log files CLF Format (**C**ommon **L**og **F**ile)

- The CLF format allows for one entry per line. Each item in the line is separated by spaces
- The CLF format is as follows:

```
host indent authuser date request status bytes
```

host	The fully qualified domain name of the client
indent	If the IdentityCheck directive is enabled and the client machine runs identd then this is the identity information reported be the client.
authuser	If the requested URL requested a succcessful Basic HTTP authentication, then the value of this token is the user name
date	Date and time of the request
request	The request line form the client enclosed in quotes(")
status	The 3-digit HTTP status code returned to the client (see the list on another page)
bytes	The number of bytes of the object returned to the client, excluding all HTTP headers.

date format: [day/month/year:hour:minutes:seconds zone]
e.g. [02/Jan/1998:00:22:01 -0800]

23.3 - Format Definition

- The format its log files can re-defined using the following directives.

`LogFormat <format> <Nickname>`	Sets the Nick Name for this particular log format
`LogFormat <format>`	Sets the format for the access log file
`LogFormat <Nickname>`	Sets the format for the access log file
`CustomLog <file-pipe> <format>`	Sends the log info to an external program as well
`CustomLog <file-pipe> <Nickname>`	Sends the log info to an external program as well

see p.298 Apache Server Bible for Formatting parameters list.

23.4 - Statistics:

- Many programs offer the ability to create statistics based on the access log file. Here are some:

Wusage	Commercial Program
WebTrends Professional Suite	Commercial Program
Wwwstat	Free CLF format web log analyser
Analog	
Webalizer	Free CLF format web log analyser given with SuSE

23.5 - Running Webaliser: (see reports in /webalizer of the zip drive)

- Webaliser processes a CLF formatted access log file and produces a full html/images statistics web page. The index page is called index.html
- Command format to produce an html report : `webalizer [options] [LogFileName]`
- Install it from CD 1 `webalizer` from the n series
- Start the program with the command:

  `webalizer -o <HtmlOutputDirectory> <LogFileName>`
- When started the program looks for a config file called **`webalizer.conf`** first in current dir then in `/etc` directory. Command line options overrides the configuration file settings.
- Start a netscape and load the index.html file produced by webalizer.
- Note: A suggestion would be:
  - Create a Virtual Host to host the result of the report
  - Periodically save or delete the content of DocumentRoot location and reproduce another report to be viewed via a browser requesting this Virtual Host URL.

24 - MIME Types, Content Negotiation and Language Negotiation

24.1 - Definition

MIME = **M**ultimedia **I**nternet **M**ail Extensions

24.2 - Module needed

mod_mime.c (default=present)

24.3 - Use

Allow Apache to determine the type of file from its extension

List of known file types is in /etc/httpd/mime.types.
More MIME types can be defined by editing this file or by using Directives in the httpd.conf file.

24.4 - Identification of a file type

- Multiple extensions can be used to identify a file type.
  e.g. `myfile.html.de` Is recognized as german language html file)
- Any unrecognized extension wipes out any extension meaning to its left.
  e.g. `myfile.html.xyz.de` Is recognized as a german file but nothing else (html will be ignored)

24.5 - MIME Types Directives

TypesConfig *<Filename>*
Path and filename to known mime types list
Default: conf/mime.types
Where: Global Server Config

AddType *<mime-type> <ext > <ext > <..*
Adds a mime type to correspond to one or more file extensions
Where: anywhere e.g. `AddType image/gif .gif89`

DefaultType *<mime-type>*
If the content type is not recognized then assume this one
Where: anywhere e.g. `DefaultType text/plain`

AddEncoding *<mime-enc> <ext> <ext ...*
Add a new type of encoding to the list.
When Apache gets a request for a file with a specific extension and this extension is listed as mime-encoding type, then
Apache will issue the Type Encoding Header parameter (in the HTTP protocol) as appropriate mime-encoding so that the client browser knows how to decode it before the file gets used.
Where: anywhere e.g. AddEncoding x-gzip .zip .gz .z

ForceType *<mime-type>*
Force a mime-type for all the files contained in a directory.
Where: <Directory> and .htaccess
e.g.

```
<Directory /www/mydomain/images>
        ForceType image/gif
</Directory>
```

24.6 - Content Negotiation:

Content negotiation is a mechanism that guesses the type of resource to send to a client according to the client's preferences or settings of their browsers.

- There are 2 types of Content Negotiations mechanisms:
  - Multiviews - simple and limited
  - Type maps (*.var files)* more complex and more powerful
  - **Multiviews method**
    - **Image Negotiation**
      - When a request is made to Apache the browser sends a list of acceptable formats:
        e.g. HTTP_ACCEPT=image/gif, image/x-xbitmap, image/jpeg, image/pjpeg etc.
      - Apache then tries to serve exactly what the client asked for within the capabilities of the browser
      - If the Multiviews is turned ON (Options +Multiviews) for a directory or a location, then Apache will serve the smallest file of the same mime-type as the requested resource.
        e.g. picture1.gif and picture1.jpg exist in a directory.
        Client requested picture1.gif .....Client receives the smallest of the two (probably picture.gif)

Language Negotiation

- The HTTP protocol provides for assertion of language in the request with the header:

```
.........HTTP_ACCEPT_LANGUAGE=de.......
```

- The language works similarly by adding a known suffix to the file name.
  e.g. `index.html.de` (german index)
  Before this can work it needs the Options +Multiviews turned ON as well as using the AddLanguage directive to define the extension that will match the language type (`.en` for `en` `.de` for `de`)

- `AddLanguage` *`<Mime-Lang.>`* *`<Ext>`* Adds a correspondence of a mime language to an extension
  e.g. `AddLanguage it .it`
  Adds the recognition of hallo.html.it as an italian lang. file.

- `LanguagePriority` *`<Mime-Lang.>`* *`<Mime-Lang.>`* *`<Mime-Lang.>`* `....`
  Sets the language priority for requests that **don't** specify any language.

- **Type Maps (*.var* files) method**

This method implies the use of definition files called .var files that contains the information necessary for the mechanism to make the most probable choice of resource depending on the request data.

----

25 - Authentication

25.1 - Basic Authentication:

- The authentication is the procedure of requesting the client to send its user and password to have access to be possibly granted access to the requested directory.
- To request authentication to access to a directory is done within a <Directory ....> container or in the .htaccess file. In general it is used within the realm of a Virtual Host
- To request authentication a normal 'valid user' from the client for access to directory issue the following directives:

Basic Authentication:

```
<Directory /dir/to/authenticate>
      AuthType Basic
      AuthName PrivateArea
      AuthUserFile /auth/my.do.main/.okusers
      AuthGroupFile  /auth/my.do.main/.okgroups
      #AuthDBMUserFile  /authDB/my.do.main/.ok_users
      #AuthDBMGroupFile /authDB/my.do.main/.ok_groups
      require valid-user
      # require user charlie
      # require group sales
      # require group directors
</Directory>
```

Digest Authentication:

```
<Directory /dir/to/authenticate>
      AuthType Digest
      AuthName PrivateArea
      AuthDigestDomain /dir/to/authenticate
      AuthDigestFile /auth/my.do.main/.digest_okusers
      AuthGroupFile  /auth/my.do.main/.okgroups
      require valid-user
      # require user charlie
      # require group sales
      # require group directors
</Directory>
```

25.2 - Directives explained:

AuthType ***type*** Authentication type. Can be Basic (DES) or Digest (MD5)
Digest is recognized by Opera and Konqueror browser but not by Netscape 4.77 or 6.0 or Mozilla. Maybe by Explorer

AuthName *label* Name (Realm)of the label which will be displayed by the browser as auth. title. If name has spaces then enclose it in quotes(")
e.g.:"Name-Passw"

AuthUserFile *Filename* Name of the File (For **Basic** Authentication) containing the user names and encrypted passwords.
It is recommended that the AuthUserFile and AuthGroupFile be in a directory level above the DocumentRoot for security reasons.

AuthDigestFile ***Filename*** Name of the File (For **Digest** Authentication) containing the user names and encrypted passwords.
It is recommended that the AuthDigestFile be in a directory level above the DocumentRoot for security reasons.

AuthDigestDomain *Path [Path]*
Path of the directories that will be using the same Names and passwords for **Digest** authentication. This entry must be present and at least have the same path as the one to authenticate

eg. <Directory /home/myweb>
AuthType Digest
....
AuthDigestDomain /home/myweb
....
This directive prevents Apache to ask for authentication on each request within (and below) the path(s) entered here.

`require valid-user` Start the authentication mechanism into action for a valid-user: Any user found in the password with his correct password will be granted access to the directory.

`require user` *user1 user2* Start the authentication mechanism into action for allowing access to user1 and user2 ...if authentication succeed.

`require group` *group1 group2* Start the authentication mechanism into action for allowing access to users being part of group1 and group2 ...if authentication succeed.

`Satisfy` `all|any` Used only if both allow from ... and require are used.
This is to request authentication on:
host addr. AND user/password authentication (all) or
host addr. OR user/password authentication (any)
e.g. Policy of allowing a particular host without authentication but require authentication for everybody else.

```
order deny,allow
allow from <privileged host IP#>
deny from all
require valid-user
Satisfy any
```

`Satisfy all` Client needs to satisfy the allow/deny restrictions and satisfy a valid user and password
`Satisfy any` Client needs to satisfy either the allow/deny or satisfy a valid user and password

25.3 - Creating authentication users/passwords files:

The program used to create/modify users/passwords files for **Basic** Authentication is: **`/usr/bin/htpasswd`**

Syntax: **`htpasswd2 [-c]`** ***`passwordfile username`***

option -c is for creating a new file.

e.g.
```
htpasswd2 -c /auth/my.domain/ok-users  michel
htpasswd2    /auth/my.domain/ok-users  irmgard
```

it writes 2 lines in the **`/auth/my.domain/ok-users`** looking like this:

```
michel:hSk74EsdLkid7dhr.f
irmgard:kdgftKedpTutdGbhfd
```

The program used to create/modify users/passwords files for **Digest** Authentication is: **`/usr/bin/htdigest`**

Syntax: **`htdigest [-c]`** ***`passwordfile realm username`***

option -c is for creating a new file.

e.g.
```
htdigest -c /auth/my.domain/Digest_ok-users PrivateArea  michel
htdigest    /auth/my.domain/Digest_ok-users PrivateArea irmgard
```

it writes 2 lines in the **/auth/my.domain/Digest_ok-users** looking like this:

```
michel:hSk74EsdLkid7dhr.f
irmgard:kdgftKedpTutdGbhfd
```

25.4 - Creating authentication group files:

The group file is created using a text editor. The format is as follows:

```
GroupNameA: User1 User2 User3 User.....
GroupNameB: User10 User11 User12 User.....
```

e.g. the file **/usr/auth/my.domain/ok-groups** may contain:

```
accounting: bob joe jerry louis peter
sales: matt johanne charlie pat
directors: herbert john
administrator: michel
```

exercise: <u>Authentication</u> : Authenticating users to allow to see the **/log** in **linuxkurs**

- Create the directory **/usr/local/httpd/auth** owned by root

```
mkdir /usr/local/httpd/auth
```

- Create authentication accounts for **hans**, **otto**, **mary** and **laura**

```
htpasswd -c /usr/local/httpd/auth/.okusers hans
htpasswd    /usr/local/httpd/auth/.okusers otto
htpasswd    /usr/local/httpd/auth/.okusers mary
htpasswd    /usr/local/httpd/auth/.okusers laura
```

- Create the authentication groups in **/usr/local/httpd/auth/.okgroups**

Enter the following lines in the **.okgroups** file:

```
admin: hans mary
finanz: otto laura
```

- In **manual** VirtualHost in **user.conf**:

```
    <VirtualHost 192.168.10.60>
         ServerName linuxkurs.linux.local
         .............
         .............
         <Location /log>
              order deny,allow
              deny from all
              allow from 192.168.10.60
              #------- Authentication part-----
              AuthType Basic
              AuthName Restricted_Area
              AuthUserFile /usr/local/httpd/auth/.okusers
              AuthGroupFile /usr/local/httpd/auth/.okgroups
              require valid-user
              satisfy any
         </Location>
         .............
         .............
    </VirtualHost>
```

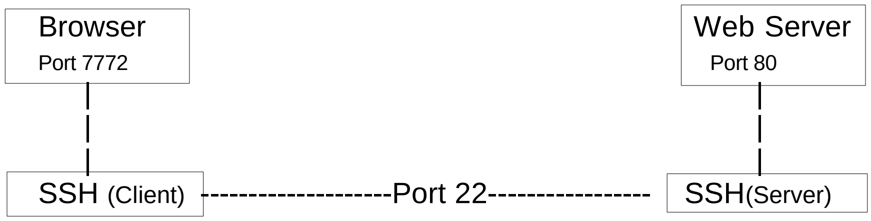
- In Browser : **`http://linuxkurs.linux.local`**
  Click on **`/log`** directory and authenticate.

- Try to change the satisfy from any to all. and play with combinations
  of allow/deny and authentication.

26. Secure HTTP

26.1 - Using SSH

1. Start a terminal and mak a connection ssh to remote web server.
   `ssh -2 remoteIP/name -L secureport:remoteIP/name:serviceport`
   eg. **`ssh -2 sun.linux.local -L 7772:sun.linux.local:80`**
   This will use the port 22 for the ssh connection and the port 7772 to tunnel the port 80 of the web server in sun.linux.local.
2. Start a web browser and give the address:
   `http://localhost:7772`
   This will use the local ssh client(port 22) as a tunnel to the remote web server.



26.2 - Using SSL (in SuSE 7.1)

26.2.1 - What is SSL

SSL stands for **S**ecure **S**ockets **L**ayer for HTTP Communication.
The new TLS (Transport Layer Security) is the future.
There are 2 types of SSL Mechanisms develloped for Apache.

- SSLeay - Proprietery SSL Function Libraries. Further development closed.
- OpenSSL - Free SSL Function Libraries. SSL 2 and 3 and TLS 1 (new)
- Apache_SSL - Free . Produced by Ben Laurie. Uses SSL Libraries.
- mod_ssl - Free . Easier to install than Apache-SSL. More functions. Uses SSL Libs.

26.2.1 - Activating the SSL as a VrtualHost in SuSE 7.1

- Uncomment or change(in bold characters) the following lines at the end of **`/etc/httpd/httpd.conf`** as follows:
  - SSLEngine **on**
  - SSLCertificateFile `/etc/httpd/ssl.crt/`**snakeoil-ca-rsa.crt**
  - SSLCertificateKeyFile /etc/httpd/ssl.key/**snakeoil-ca-rsa.key**
  - SSLCACertificateFile /etc/httpd/ssl.crt/ca-bundle.crt
- In Browser: **`https://`*`MySecureWebAddress`***

26.2.2 - What are the components of SSL communication.

- **X.509 Certificate:**
  A certificate is a signature produced by a Certificate Authority organization to ensure the Authenticity of the person(s) requesting the certificate for their Web Server.
  It is composed of:
  - ......to be continued......

27 - Web Robots

27.1 - Definition

Web Robots are programs that scan the web for indexing and mirroring web sites. Some have the purpose of only check the validity of the hyper-links.

The list of web robots is in mitp - Apache Webserver (German) boot page 571,572.

27.2 - Web Robots Control File

There is a file which is placed in the DocumentRoot of the server and dictates the the behaviour of the Web Robots.

All the web robots should take notice and follow the directives found in this file.

The file name is **`/robots.txt`**

27.3 - Format of Web Robots Control File Directives

- Lines starting with '#' are comments.
- **`User-Agent:`** ***RobotName***
- **`Allow:`** ***DirectoryAllowed***
- **`Disallow:`** ***DirectoryNOTAllowed***

- User-Agent, Allow and Disallow can be delcared as many time as needed.
- The ***DirectoryAllowed*** and ***DirectoryNOTAllowed*** are relative to the DocumentRoot of the server or VirtualHost. They MUST have a '/' at the end.

eg.

```
User-Agent: wget
Allow: /info/
Disallow: /cgi-bin/
Disallow: /daily/news.html
User-Agent: slurp
Allow: /price/
Disallow: /log/
Disallow: /pictures/
```

27.3.1 - Sequence of reading the robots.txt

The **robots.tx**t is read so that the first valid correspondence is taken as the only valid one for the requested.URL.

eg.

```
Allow: /info/
Disallow: /info/docs/
```

In this case the whole Directory of /info/ is allowed including the /info/docs/ Because the Allow: /info/ is read when a request is done for anything in this directory and the ones under it and since it is allowed then it never reads the **`Dissallow: /info/docs/`**. This is TOTALLY contrary to the way Apache functions.

The solution to get what we want here is to simply change the sequence:

```
Disallow: /info/docs/
Allow: /info/
```

27.3.2 - Special meanings of the configuration:

- **`User-Agent: *`** means ALL the User-Agents
- **`User-Agent: wg*`** means nothing at all. Useless.
- **`Disallow:`** means there is no restrictions at all.

Therfore: the **`*`** is never used in Allow or Disallow statements.

More examples:
To allow only one Web Robot in the site:

```
User-Agent: WebCrawler
Disallow:
User-Agent: *
Disallow: /
```

To Disallow only one Web Robot in the site:

```
User-Agent: WebCrawler
Disallow: /
```

27.4 - Caching of robots.txt

Many of the Web Robots will cache the robots.txt for up to une week.
If we want to change this to 3 days then we can add the following in the Apache config file.

```
<Location /robots.txt>
      ExpiresDefault "access 3 days"
</Location>
```

27.5 - Other methods of limiting access to Web Robots.

27.5.1 - Via HTML Headers
Although NOT all the Web Robots regards this as valid, we can limit the access by adding the following META headers in the HTML files (index.html)
eg.

```
<META NAME="ROBOTS" CONTENT="NOINDEX, NOINCLUDE">
```

This file will not be indexed by the WebRobots and the HyperLinks within it also not.

```
<META NAME="ROBOTS" CONTENT="NOFOLLOW">
```

This file WILL be indexed by The Web Robots but not the HyperLinks within it.

27.5.2 - Via Web Robot signature recognition and blockage.
Since the WebRobots Identify themselves in the User-Agent: HTTP header we can use the BrowserMatchNoCase Directive to prevent it from accessing some of the locations, or all of the locations! Here is the systax:

```
BrowserMatchNoCase "^robotname" Badrobot
SetEnvIf Remote_Host .*robotname.* Badrobot
<Location />
      order allow,deny
      deny from env=Badrobot
</Location>
```

27.5.3 - Via Rewrite Module.

We can also make a special redirection using the rewrite module to forbid certain resources. It goes like this:

```
RewriteCond %{HTTP_USER_AGENT} .*robotname1.*  [NC,OR]
RewriteCond %{HTTP_USER_AGENT} .*robotname2.*  [NC,OR]
RewriteCond %{REMOTE_HOST} badrobot.com$  [NC]
RewriteRule ^/not-indexable/ - [F]
```

27.5.4 - Getting information on Good and Bad Robots

To get up to date info on robots here is the right place:
`http://info/webcrawler.com/mak/projects/robots/robots.html`

27.5.5 - Via Allow/Deny Directives.

When an unwanted Web Robot's IP address is known, (by studiying the logs for example) then it is possible to block access of the whole site or part of it witht the regular Allow/Deny Directives as follows:

```
<Location />
      Order allow,Deny
      Deny From BadRobot's_IP_Addr.
</Location>
```

27.6 - Making sure the Robots index the right information.

These META entries help a lot the robots to make their index.

<META NAME="Author" CONTENT="The Computer">

<META NAME="Description" CONTENT="All about computers">

<META NAME="Keywords" CONTENT="Linux, Windows,Hardware">

27.7 - Submitting web sites to Web Robots.

One of the best ways to submit your web site to Robots is to visit the following site and make the appropriate entries:

```
      http://www.submit-it.com
```

28 - Search engine Web Robot: ht://Dig

28.1 - Description:

Htdig is a search engine program used to search for keywords in local or remote web sites. It can create a database of keywords of multiple URLs and therefore allow search through them.

28.2 - Components of Ht://Dig

Htdig is composed of 3 major components which are used in the following order:

- **Digging**: The gathering of unique words into a Database.
  The program used is **`htdig`** ...the search robot.
  It is located at : **`/opt/www/htdig/bin/htdig`**
  The databases files are in : **`/opt/www/htdig/db/`** dir.
  The 'digging' can be done in 2 modes:
  - 'Changes only' mode (Default)
  - Full initial mode (`rundig -i`)

Note: Htdig program can also authentify itself with a user and password for sites that require *basic* authentification It is done by calling the program with the following options:
`-u username:password`
It tells htdig to send the supplied username and password with each HTTP request. The credentials will be encoded using the 'Basic' authentication scheme. There HAS to be a colon (:) between the username and password.

- **Merging**: The merging of databases produced by htdig is done by the program **htmerge**. It is needed to merge the 'changes only' databases that htdig has created.
  The file is located at: **`/opt/www/htdig/bin/htmerge`**
- **Searching**: The searching of keywords is done by CGI **Htsearch**.
  The file is found at: **`/opt/www/cgi-bin/htsearch`**
  and at : **`/usr/local/httpd/cgi-bin/htsearch`**
  Htsearch is the actual search engine of the ht://Dig search system. It is a CGI program(compiled) that is expected to be invoked by an HTML form. It will accept both the GET and POST methods of passing data to the CGI program.
  Files used by htsearch are:

*CONFIG_DIR/***htdig.conf** The default configuration file.
*COMMON_DIR/***header.html** The default search results header file.
*COMMON_DIR/***footer.html** The default search results footer file.
*COMMON_DIR/***wrapper.html** The default search results wrapper file. that contains the header and footer together in one file.
*COMMON_DIR/***nomatch.html** The default 'no matches found' HTML file.
*COMMON_DIR/***syntax.html** The default file that explains boolean expression syntax errors.

The *CONFIG_DIR* and *COMMON_DIR* are paths already defined when the programs were compiled. In the case of SuSE, the path for *CONFIG_DIR* is **`/opt/www/htdig/conf/`** and the path for *COMMON_DIR* is **`/opt/www/htdig/common/`**

28.3 - Other programs included with ht://Dig:

/opt/www/htdig/bin/**rundig** Script used to generate an Ht://Dig database as per htdig.conf.
Use **`rundig -v`** for verbose
Type **`rundig -vvv`** for long debugging.

/opt/www/htdig/bin/**htfuzzy** Htfuzzy creates indexes for different "fuzzy" search algorithms. These indexes can then be used by the htsearch program.

The algorithms can be:

- exact
- soundex
- metaphone

- endings
- synonyms

`/opt/www/htdig/bin/`**`htnotify`** Htnotify scans the document database created by htmerge and sends an email message for every page that is out of date. Look in the notification manual for instructions to set up this service.

28.4 - Invoking the htsearch program from an HTML Form:

The parameters htsearch needs to proceed to the search are passed via the **GET** or **POST** methods data. The syntax of this data is defined in the HTML form as NAME and VALUE of the option. Eg.

```
<form method="GET" action="/cgi-bin/htsearch">
     <font size=-1><H3>Start eine Suche mit</H3><center>
      <select name=method>
           <option value="and">Und-Verknuepfung</option>
            <option value="or" Selected>Oder-Verknuepfung</option>
      </select>
      <Select name=config>
           <option value="bashshell">bashshell.conf</option>
           <option value="forms">forms.conf</option>
           <option value="htdigv">htdigv.conf</option>
           <option value="linuxkurs">linuxkurs.conf</option>
           <option value="manual">manual.conf</option>
           <option value="samba">samba.conf</option>
           <option value="selfhtml">selfhtml.conf</option>
           <option value="webalizer">webalizer.conf</option>
     </Select>
      , Suchbegriffe:
     <input type="text" size="30" name="words" value="">
     <input type="submit" value="Search">
 </form>
```

28.5 - HTML Form input syntax.

The primary interface to htsearch is through an HTML. When the form is submitted, the htsearch program will take values from the form and perform the actual search. The search can be modified in many ways with either hidden input fields or other HTML form tags. Study the examples to get a feel of what things are possible.
The HTML form is expected to contain at least an input text field named words. This is where the user will enter the search words. Other values are also recognized but have appropriate defaults in case they are not used:

`config`

Specifies the name of the configuration file. The name here is the name without the path and without the .conf at the end. This file is assumed to be located in the CONFIG_DIR directory. (SuSE- /opt/www/htdig/conf/) Periods are not allowed in this field for security reasons (to prevent HTML authors from pointing all around at your files).
The default is **htdig**

`exclude`

This value is a pattern that all URLs of the search results cannot match.
The default is ***blank***.

`format`

This specifies the name of the template to display the search results in. There are two builtin templates named **`builtin-long`** and **`builtin-short`** which can be used, but any number of custom templates can also be defined. Find out more about the templates in the Output Templates section. The format value can be specified as either a hidden input field or a drop down menu.
The default is specified by the **`template_name`** attribute in the configuration file.

`keywords`

Used to specify a list of required words that have to be in the documents. This list of words is added to the normal words value using logical "and"s. An example use for this value is to make it a drop down menu with a limited set of predetermined categories or keywords to restrict the search. This can be very useful for very structured pages.
Note that the words may appear anywhere in the document. The scope of these required words is not limited to words in META tags with the "**`keywords`**" or "**`htdig-keywords`**" property, despite what the parameter name may suggest.

`matchesperpage`

Specifies how many matches will be displayed on each page of results.
The default is specified by the **`matches_per_page`** attribute in the configuration file. Since this value has to be a number, it either needs to be set using a hidden input field or a with a drop down menu.

`method`
This can be one of **`and`**, **`or`**, or **`boolean`**. It determines what type of search will be performed. The default is specified by the **`match_method`** attribute in the configuration file. It is quite useful to make this item a drop down menu so the user can select the type of search at search time.

`page`
This should normally not be used. It is generated by the paged results display.

`restrict`
This value is a pattern that all URLs of the search results will have to match. This can be used to restrict the search to a particular subtree or subsection of a bigger database.
The default is blank.

`sort`
This can be one of **`score`**, **`time`**, **`date`**, **`title`**, **`revscore`**, **`revtime`**, **`revdate`**, or **`revtitle`**. It determines what type of sort will be performed on the search results. The types **`time`** and **`date`** are synonymous, as are **`revtime`** and **`revdate`**, as all four sort on the time that the documents were last modified, if this information is given by the server. The sort methods that begin with **`rev`** simply reverse the order of the sort.
The default is specified by the **`sort`** attribute in the configuration file. It is quite useful to make this item a drop down menu so the user can select the type of sort at search time.

28.6 - Running Ht://Dig for Multiple VirtualHosts:

Here are the steps needed to setup the Ht://Dig for a whole Apache server including all of its Virtual Hosts.

- Using YaST, install the **htdig** package from the series '**n**'.
- Edit the **`/etc/htdig/htdig.conf`** and enter the following:

1- All URLs of Virtual Hosts existing in the server. Each URL should be separated by at least a space.
Syntax:

```
    start_url: http://VHost1.Name http://VHost2.Name ....
```

eg. for 2 VirtualHosts

```
    start_url: http://samba.linux.local/
http://selfhtml.linux.local/docs/
```

IMPORTANT: Do not forget the last '/' after the URL

2 - The *DocumentRoot* of all the above Virtual Hosts. It should all be written on the same line. This directive tells **`htdig`** program to look in the file system for the URL of the VirtualHost and not ask the local Apache server for it. It prevents Apache from serving all the URLs and then not be able to manage which results in an incomplete search database.

Syntax:
`local_urls:` **`http://`*`VHost1.Name`*`/=/VHost1`*`DocumentRoot/`*** (*space*)
`http://`*`VHost2.Name`*`/=/VHost2`*`DocumentRoot/`* `....`

(notice the '/' at the end of each ***VHostx.Name/*** and ***DocumentRoot/*** They are important.

eg.

```
local_urls: http://samba.linux.local/=/www/samba/
            http://selfhtml.linux.local/=/www/selfhtml/
```

(Important:The above example should be entered all on one single line)

3 - Tell to use only the URL's existing in local file system.

```
local_urls_only: true
```

4 - All the VirtualHost's `DirectoryIndex` file names.
(First page sent to browser when accessing the VirtualHost's Site)
Syntax: (all on one line)
`local_default_doc:` ***`VHost1DirectoryIndex VHost2DirectoryIndex ......`***

eg.

```
local_default_doc: index.html selfhtml.htm
```

The default is **`index.html.`**

5- (optional) To tell htdig to scan PDF files do the following:
in Configuration file:
`max_doc_size:  100000000` (100MB . Must be bigger than the largest file)
`external_parsers: application/pdf /etc/htdig/parsepdf.pl`

In the above line we are using a Perl script(`parsepdf.pl`) as external parser.
The content of the external parser follows this section:

6- Give the database directory and the basename(name prefix) of the database filename to create.

```
database_dir:       /opt/www/htdig/db
database_base:      /opt/www/htdig/db/public4e
```

7- Run the **`rundig`** with the parameter **`-v -c`** ***`configuration_filename`***
eg.

```
/opt/www/htdig/bin/rundig -v
-c /opt/www/htdig/conf/public4e.conf
```

Note: The best is to run this command in an `xterm` and watch the 'digging' process.

28.7 - Running Ht://Dig for individual VirtualHosts:

The steps needed to make use of HT://Dig for VirtualHosts are more complex than to use it for the whole server. Here is the minimum to do to achieve it:

- Install the **`htdig`** package
- Create a configuration file for each VirtualHost and store it in the same location as

the original: in the **/opt/www/htdig/conf/** directory
eg. **/opt/www/htdig/conf/samba.conf**
Simply use a copy of the htdig.conf file as template for each Vitrtual Host

– In each Virtual Host configuration file, enter the following information:

1- The full URL of the Virtual Host.
Syntax:
start_url: http://*Virtual.Host.Name/*
eg.
start_url: http://samba.linux.local/docs/

2 - The *DocumentRoot* of the Virtual Host
Syntax:
local_urls: http://*Virtual.Host.Name/=/DocumentRoot/*
(notice the '/' at the end of ***Virtual.Host.Name/*** and ***DocumentRoot/***
They are important.
This directive tells **htdig** program to look in the file system for the URL of the VirtualHost and not ask the local Apache server for it. It prevents Apache from serving all the URLs and then not be able to manage which results in an incomplete search database.
eg. http://samba.linux.local/=/www/samba/

3 - Tell to use only the URL's existing in local file system.
local_urls_only:true

4 - The filename prefix of the Virtual Host of the database files.
Syntax:
database_dir: /opt/www/htdig/db
database_base: /opt/www/htdig/db/*VHostDatabasePrefix*
eg.
database_base: /opt/www/htdig/db/samba
This is the filename prefix of the the 4 files that are created by the **htdig** and **htmerge** for the VirtualHost. The 4 files would then be:
samba.docdb
samba.docs.index
samba.wordlist
samba.words.db
Theyl would be located in the **/opt/www/htdig/db/** directory.

5 - The VirtualHost's DirectoryIndex file name.
(First page sent to browser when accessing the VirtualHost's Site)
Syntax:
local_default_doc: *VirtualHostDirectoryIndex*
eg.
local_default_doc: selfhtml.htm
The default is **index.html.**

6- (optional) To tell htdig to scan PDF files do the following:
in Configuration file:
max_doc_size: 100000000 (100MB . Must be bigger than the largest file)

```
external_parsers: application/pdf /etc/htdig/parsepdf.pl
```

In the above line we are using a Perl script(`parsepdf.pl`) as external parser. The content of the external parser follows this section:

Important: If the directories have PDF files in it they MUST be referenced by a `href=.......pdf` in an HTML file to be detected by the rundig program.

7 - Run the **`rundig`** with the parameter **`-v -c`** ***`configuration_filename`***
eg.
`/opt/www/htdig/bin/rundig -v -c /opt/www/htdig/conf/samba.conf`

The best is to run this command in an Xterm and watch the 'digging' process.

Extertnal PDF file parser:

```
#!/usr/bin/perl --
#
# Name : parsepdf.pl
# parse pdf files for htdig
#
# - generate anchor tags
# - do site specific rewriting url to title
#   for missing or bad titles
# - I suppose it is faster then parse_doc.pl
#
# based on:
#       - htdig documentation
#       - parse_doc.pl
#       - pdftodig.py (http://po.gaillard.free.fr/pdftodig.py)
#
# Stefan Nehlsen  sn@parlanet.de
#
# external tools from the xpdf package
$parser = "/usr/bin/pdftotext";
$info   = "/usr/bin/pdfinfo";

my($infile, $content_type, $url, $config) = @ARGV;

# paranoid
die "pdfinfo \"$info\" not executable!\n"  unless -x $info;
die "parser \"$parser\" not executable!\n" unless -x $parser;
die "\"$infile\" not readable\n" unless -f $infile;
open PDF, $infile or die "opening $infile failed\n";
$text = <PDF>; # read first line
close PDF;
die "\"$infile is not a PDF-File!\n" unless $text=~/^%PDF-\d\.\d/;
# everything seems to be ok

# use pdfinfo to retrieve meta information
open INFO, "$info \"$infile\" 2>/dev/null |" or warn "$info \"$infile\"
failed\n";
while (<INFO>) {
        chop;
        if(s/^Title:\s*//){
             s/\s+$//; s/\s+/ /g; s/[\376\377]//g; # delete unicode (?) marker
             # if title is a filename we better use the real filename
             $title = $_ unless /\.pdf$|Microsoft\s+Word\s+-/i or
                        (length($_)> 16 and /\.\.\.$/);
```

```
                last;
        }
}
close INFO;

# At this point I do some site-specific rewriting of the title
# based on structured urls and/or an external database.

# read text from pdftotext
undef $/;
open PDF, "$parser -raw -q \"$infile\" - 2>/dev/null |"
        or die "error opening pdf \"$infile\"\n";
$text = <PDF>; # read whole file
close PDF;

# the point of no return
($title = $url) =~ s#^.*/(.*?\.pdf$)#PDF Dokument $1#i unless $title;
$title =~ s/&/\&amp\;/g; $title =~ s/</\&lt\;/g; $title =~ s/>/\&gt\;/g;
print "t\t", $title, "\n";

$text =~ s/^[\s\n\f]*//s; $text =~ s/[\s\n\f]*$//s;
$text =~ s/-\s*\n+\s*([a-z\340-\377])/$1/gs; # dehyphen

($header = $text) =~ s/[\s\n\f]+/ /gs;
if( $header ){
  $header =~ s/&/\&amp\;/g; $header =~ s/</\&lt\;/g; $header =~ s/>/\&gt\;/g;
  print "h\t", $header, "\n";
}

@words = grep { /\f|.{3,}/ } split /[^A-Za-z\300-\377\f]+/, $text;
$n = 0; $page = 2; $k = 1000 / @words if @words;
foreach $word ( @words){
        if( $word eq "\f" ){
                printf "a\tpage=%d\n", $page++;
        } else {
                printf "w\t%s\t%d\t0\n", $word, $n++ * $k;
        }
}
```

Example of htdig.conf for english linux info site:

```
start_url:         http://www.linuxint.com/english/
local_urls:
http://www.linuxint.com/english/=/var/www/michel/linux_info/english/
local_urls_only:    true
database_dir:      /var/www/michel/htdig/db
database_base:     /var/www/michel/htdig/db/public4e
local_default_doc: welcome.html
```

– In each web page HTML Form where we wan to have a serach field, tell which configuration file will be used to search the VirtualHost database. Naturally we need to give the VirtualHost Configuration file without the **`.conf`** extention. NO dots '**`.`**' are allowed in this name as well. The parameter name is **`config`** .
eg.
    `<input type=hidden name=config value=samba>`
This search would use the configuration file:
`/opt/www/htdig/conf/samba.conf` for its search.

– Make sure that the VirtualHost configuration in Apache has and `alias` that points to the `htdig` pictures directory.

eg.

```
alias /htdig/ /var/www/htdig/
```

28.8 - HTML Web pages optional META headers:

As the ht://Dig system will index all HTML pages on a system, individual authors of pages may want to control some of the aspects of the indexing operation. To this end, ht://Dig will recognize some special <META> tag attributes. The following things can be controlled in this manner:

- Do not index the document
- Notify a user that the document has expired
- Set keywords for the document

28.8.1 - General <META> tag use

In HTML, any number of <META> tags can be used between the <HEAD> and </HEAD> tags of a document. There are three possible attributes in this tag, two of which are recognized by ht://Dig:

• NAME	Used to name a specific property.
• CONTENT	Used to supply the value for a named property.

A document could start with something like the following:

```
<HTML>
<HEAD>
<META NAME="htdig-keywords" CONTENT="phone telephone online contact">
<META NAME="htdig-email" CONTENT="pat.user@nowhere.net">
<TITLE>Some document title</TITLE>
</HEAD>
<BODY> Body of document</BODY>
</HTML>
```

28.8.2 - Recognized properties

The following properties are recognized by ht://Dig:

- htdig-keywords
- htdig-noindex
- htdig-email
- htdig-notification-date
- htdig-email-subject
- robots
- keywords
- description

29- Compiling and Installing Apache from a downloaded file

(page 67 Professional Apache)

29.1 - Preparation

- Get it from : www.apache.org
- Copy it into `/usr/local` and untar it.
- Make a link called `apache` in `/usr/local/` subdirectory.
- Description of difference between core and module features
- Deciding wich modules will be compiled in and which will be loaded dynamically.
  Recompile, Speed, or size?

29.2 - Compiling Apache

`# ./configure --help`	Lists all modules that will be built-in Apache by default (see results of --help on another page)

`# ./configure --enable-module=most`	Compile almost all modules as built-in except:
	`mod_auth_db`
	`mod_mmap_static`
	`mod_so(dynamic module support)`
	`mod_example(for developers only)`
	`mod_auth_digest(new mod_digest)`
	`mod_log_agent(replaced by mod_log_config)`
	`mod_log_referer(replaced by mod_log_config)`

`# ./configure --enable-module=all`	Compile all modules, listed in --help, as built-in
`# ./configure --enable-shared=most`	Compile almost all modules, listed in --help, as Dynamic Shared Object(DSO)
`# ./configure --enable-shared=max`	Compile all possible modules built as Dynamic Shared Object(DSO)

To resume:

The help lists all names of recognized modules and whether they will be built-in or not.
If we want to build-in a module that would not be built-in as per --help then

```
# ./configure --enable-module=<Modulename> or most or all.
```

If we want to make a module or many as dynamically loadable instead of built-in:

```
# ./configure --enable-shared=<Modulename> or max or most.
```

If we want to exclude a module then:

```
# ./configure --disable-module=<Modulename>
```

Best of both worlds is most regular ones built-in and the rest loadable dynamically.

```
# ./configure --enable-module=most --enable-shared=max
```

Then do the last command: **`make install`**

29.3 - Configuring Apache Modules

- Edit the **httpd.conf** file**:**
- Note :
  During `make install`, the **`LoadModules`** and **`AddModules`** are written automatically in the httpd.conf file for the dynamically loadable modules. The following directives apply:
- **LoadModule** <xxx_module> libexec/<mod_xxx.so> Loads an Apache Module as available in the internal module list
- Sequence of modules being run is in reverse order as defined in LoadModule list of the httpd.conf file.
  To change this sequence:
  - **ClearModuleList** Clears the Module list
    (Normally used before defining the AddModule directives)
  - **AddModule** <mod_xxx.c> Defines the sequence in which the module will be in the module list.
    The last module in the LoadModule list will be processed first so to change the sequence this series of AddModule is used with the *mod_xxx.c* name.
    Normally the list is cleared with ClearModuleList before the AddModule directives are defined.
    Modules are located in `/usr/local/apache/libexec/` dir.

30 - Adapting a downloaded version of Apache to SuSE Distribution:

This adaptation keeps old files installed and allows to run the new version of Apache.

NOTE: We assume here that you have installed and compiled the downloaded Apache
into /usr/local/apache/(link) to /usr/local/apache_1.3.12/ then do the following:

- In /sbin/init.d/ dir. Rename the apache script to apache.SuSE

```
mv /sbin/init.d/apache /sbin/init.d/apache.SuSE
```

- Copy the script /usr/local/apache/bin/apachectl to /sbin/init.d/apache

```
cp /usr/local/apache/bin/apachectl /sbin/init.d/apache
```

- Edit the script /sbin/init.d/apache and at line 28 add the config. file parameter as follows:

```
# the path to your httpd binary, including options if necessary
HTTPD="/usr/local/apache/bin/httpd -f /usr/local/apache/conf/httpd.conf"
Note: The quotes "...." around the parameter are IMPORTANT. Originally not there.
```

- From now on the new apache will have the following settings:

  - Configuration file is `/usr/local/apache/conf/httpd.conf`

  - The daemon(httpd) is located in `/usr/local/apache/bin/httpd`

  - The ServerRoot directory is `/usr/local/apache` and should never be changed!!!
    If you need to change it then:
    - make a new directory somewhere else
    - copy the bin/, conf/, icons/, libexec/ and logs/ to the new directory.
    - edit the new httpd.conf file and change the ServerRoot directive to new dir.
    - edit the /sbin/init.d/apache script (line 28) to load the new config. file (httpd -f <*newdir*>/conf/htpd.conf)

  - The manually run rcapache command still works but uses the following arguments:
    - start, stop, restart, fullstatus, status, graceful, configtest, help
      (instead of start, stop, restart, full-status, status, reload)

  - The links in /sbin/init.d/rc2.d/ dir. for starting Apache at boot-up are also still valid.

- Edit the `/usr/local/apache/conf/httpd.conf` and set the appropriate parameters for:

  - Global Settings

  - Individual Virtual Hosts settings etc

Appendix A - Global Server Directives:

Underlined directives and containers are ONLY allowed as Global**.** The rest are considered as general defaults and are used also for containers that don't define them within the container.

Containers:

<**Directory** */dir* >	Directory access container .
<**DirectoryMatch** "*regex*" >	Directory access container with regular expressions.(regex)
<**Files** "[*path*]*file*" >	File access container. Note the " surrounding the filename !
<**FilesMatch** "*regex*" >	File access container with regular expressions.(regex)
<**Location** *URI* >	URI access container.
<**LocationMatch** "*regex*" >	URI access container with regular expressions (regex)
<**Limit** *METHOD(s)* >	HTTP Methods container.
<**LimitExcept** *METHOD(s)* >	HTTP Methods container for undefined Methods
<**IfModule** *module.c* >	Conditional directives processed only if specific module is loaded
<**IfDefine** *defined name* >	Conditional directives processed only if defined name is given on the command line of httpd following a -D option. e.g. httpd -f /etc/httpd/httpd.conf -D testname
<**VirtualHost** *IP#[Port]*>	Virtual Host directives container

Directives:

AccessFileName <Filename>	The Per-directory access control file name. Default: .htaccess
DocumentRoot <Html docs>	Default Landing Zone of documents for HTTP requests Default is the htdocs dir from the ServerRoot directory.
ErrorDocument <errorNo> <Filename>.	Document (.html) sent to client if a request error occurs.
Options <option1 option 2 .....>	Default options applied to container that don't use options.
DefaultType <default MIME type>	Default MIME type for untypable files.
ServerType <type>	Standalone(Daemon) or inetd(Loadable from inetd).
Port <port nr>	Normally = 80
HostnameLookups<on, off or double>	Enable(on) or Disable(off) or Double reverse DNS lookup.
User <Login_username>	Normally = nobody
Group <Login_group>	Normally = nogroup
ServerAdmin <admin_email_addr>	Email of administrator e.g. mario@doggydo.net
ServerName <hostname>	Server hostname.
ServerSignature <on , off, email>	Enable(on) or disable (off) server signature.
ServerRoot <Server Root path>	Path of the server base dir. where essential files are kept as well as the relative base dir. for any non-absolute directives in config. file.
ErrorLog <Filename>	Filename of the error log.
PidFile <Filename>	Where the Process ID of the root started Daemon is stored Default is logs/httpd.pid
ScoreBoardFile <Status filename>	Running Status file name used to communicate with children. Can be moved to a RAM Disk for speed. Default is logs/apache_status
LockFile <LockFilename>	Where the lock file will be saved. Apache won't start if it can't write this file. Used only to prevent multi instances of Apache. Default is logs/accept.lock
AccessConfig <Filename>	Access configuration file. (deprecated). Default is access.conf
ResourceConfig <Filename>	Resource configuration file. (deprecated). Default is srm.conf
ServerAlias <alias1 alias2 etc>	Alias name(s) used to access the server.
ServerPath <Path>	The pathname the server can be reached at. For HTTP1.0 only See page 54 in O'Reilly Apache.
Timeout <time in sec>	Timeout in sec. server waits for the next packet before connection is broken. Default is 300 (5 minutes)
KeepAliveTimeout <time in sec>	KeepAlive timeout in seconds before a child closes a connection.
MaxKeepAliveRequests <0/1/2/...>	Maximum number of requests per connection. 0 for infinite.
KeepAlive <On or Off>	Whether persistent connections should be On or Off.
IdentityCheck	Enables the user lookup identity check(RFC 1413)
ContentDigest	Whether or not to send a Content-MD5 header with each request

- UseCanonicalName.............................. How to work out the ServerName : Port when constructing URLs
- StartServers <Nr. of servers>................ Number of child processes launched at server startup
- MinSpareServers <Nr. of servers>........ Minimum number of idle children, to handle request spikes
- MaxSpareServers <Nr. of servers>....... Maximum number of idle children
- MaxServers <Nr. of servers>................. Deprecated equivalent to MaxSpareServers
- ServersSafetyLimit <Nr. of clients>....... Deprecated equivalent to MaxClients
- MaxClients <Nr. of clients>..................... Maximum number of requests running at the same time.
- MaxRequestsPerChild<Nr. of requests>Maximum number of requests a particular child serves before dying.
- RLimitCPU <limit in sec. per process>.... Soft/hard limits for max CPU usage in seconds per process.
  See Page 75 of Apache Server Bible
- RLimitMEM <limit in bytes per process> Soft/hard limits for max memory usage per process.
- RlimitNPROC <Nr. of processes>.......... Soft/hard limits for max number of processes per user (uid).
- BindAddress <addr1 addr2 addr3...>...... Limits the server to listening to specific IP Addr.
  Good to make Virtual Hosts using multi daemons
- Listen <IP#:port>..................................... Replaces BindAddress and port all in one.
  Can also be used more than once.
- SendBufferSize <Size in Bytes>........... Transmit(send) buffer size in bytes.
- AddModule <module_name.c>.............. Adds a module at the bottom of the module list for execution order.
- ClearModuleList.................................... Clears the module execution order list.
- ThreadsPerChild <Nr. of threads>......... Number of threads a child creates. (Windows only)
- ExcessRequestsPerChild <Nr. req.>... Maximum number of requests a child serves after it is ready to die.
- ListenBacklog Maximum length of queue of pending connections, used by listen.
- CoreDumpDirectory <CoreDump Dir>.. The location of the directory Apache changes to before dumping core
  Default is the *ServerRoot* directory
- Include <Filename>................................ Name of the config file to be included.
  The file is read as if being part of the present config file.
- LogLevel <level Nr.>............................... Level of verbosity in error logging
- NameVirtualHost <IP#[:Port]>............... IP Number (or the is name:not recommended) of a virtual host.
- ServerTokens Determine information header level returned about the Server itself:
  Values: Min(imal), OS or Full(default)
- LimitRequestLine Limit on maximum size of an HTTP request line
- LimitRequestFieldsize Limit on maximum size of an HTTP request header field
- LimitRequestFields Limit (0=unlimited) on max no. of header fields in a request message
- LimitRequestBody Limit (in bytes) on maximum size of request message body
- LoadModule <name> <object>.............. A module name and the name of a shared object file to load it from.
- LoadFile <Filename>.............................. Shared object file or library to load into the server at runtime
- DirectoryIndex <Filename(s)>............ Sets the file name(s) that will be automatically sent to clients when
  accessing a directory only. e.g. www.mydomain.de/mysubdir/
  This will display the index.html file if present in this dir.
- Redirect <requested URL> <new URL>.. Redirects a URL(can be a location) to a full new URL
- RedirectMatch<requested URL> <new URL>... Same as redirect but with regular expressions

NOTE: relative directory paths(without a leading /) always refer to ServerRoot directory.

Appendix B - Directives allowed in <Directory> <Files> and <Location>

Containers:

- <**Files** *path/file(s)*> File access directives container.
- <**FilesMatch** *regex*> File access directives container with regular matching expressions.
- <**Limit** *METHOD(s)*> HTTP Methods Directive container.
- <**LimitExcept** *METHOD(s)*> HTTP Methods Directive container for undefined Methods
- <**IfModule** *module.c*> Conditional directives processed only if specific module is loaded
- <**IfDefine** <defined name> Conditional directives processed only if defined name is given on the command line of httpd following a -D option.
  e.g. httpd -f /etc/httpd/httpd.conf -D testname

Directives:

- AuthType <type> An HTTP authorization type (e.g., "Basic")
- AuthName <Auth Realm> The authentication realm (e.g. "Members Only")
- Require Selects which authenticated users or groups may access a protected space.
- Satisfy <access policy>........................... Access policy if both allow and require used (all or any)
- ErrorDocument <errorNo> <Filename>. Document (.html) sent to client if a request error occurs.
- AllowOverride <options>....................... Tells which directives can be overridden by the the ones contained in the .htaccess file. The options can be:

All	Enables all overrides...Dangerous.
AuthConfig	Allows use of authorization directives: AuthName, AuthType and AuthUserFile. Note: Requires the mod_auth and equiv.
FileInfo	Allows directives controlling the file types like: AddType, DefaultType,AddEncoding,AddLanguage ErrorDocument etc.
Indexes	Allow use of directives controlling the appearance of the directory indices as generated by Apache.
Limit	Allow use of mod_access directives: order, allow and deny
Options	Allows the use of Options and XbitHack directives
None	Disallow all directives in .htaccess and prevents Apache to search and read for .htaccess files.

- Options <option1 option 2 .....>............. Default options applied to container that don't use options.
- DefaultType <default MIME type>......... Default MIME type for untypable files.
- HostnameLookups<on, off or double> Enable(on) or Disable(off) or Double reverse DNS lookup.
- ServerSignature <on , off, email>........ Enable(on) or disable (off) server signature.
- IdentityCheck... Enables the user lookup identity check(RFC 1413)
- ContentDigest .... Whether or not to send a Content-MD5 header with each request
- RLimitCPU <limit in sec.>....................... Soft/hard limits for max CPU usage in seconds.
- RLimitMEM <limit in bytes per process> Soft/hard limits for max memory usage per process.
- RlimitNPROC <Nr. of processes>.......... Soft/hard limits for max number of processes per user (uid).
- Include <Filename>................................ Name of the config file to be included.
  The file is read as if being part of the present config file.
- LimitRequestBody Limit (in bytes) on maximum size of request message body
- DirectoryIndex <Filename(s)>............. Sets the file name(s) that will be automatically sent to clients when accessing a directory only. e.g. www.mydomain.de/mysubdir/
  This will display the index.html file if present in this dir.

Specific Directives for **<Directory>** and **<DirectoryMatch>**

- order <read 1, read 2>........................... Sets the order of which the access rights will be read:
  allow, deny or deny, allow
- allow from <client_1 client_2...>........... Allows access to the defined directory to the following clients:
  IP# or hostname or all or none
- deny from <client_1 client_2...>........... Denies access to the defined directory to the following subjects:
  IP# or hostname or all or none

Appendix C - Directives allowed in .htaccess file

(the name of this file(.htaccess) is the default and can changed to something else through the AccessFileName
global directive. Multiple file names can be defined as well on the same line.
e.g.
`AccessFileName .default .htaccess .restrictions` etc.....

To hide .htaccess from browsers then:

```
<Files .htaccess>
    order allow, deny
    deny from all
</Files>
```

Containers:

<**Files** *path/file(s)*>	File access directives container.
<**FilesMatch** *regex*>	File access directives container with regular matching expressions.
<**Limit** *METHOD(s)*>	HTTP Methods Directive container.
<**LimitExcept** *METHOD(s)*>	HTTP Methods Directive container for undefined Methods
<**IfModule** *module.c*>	Conditional directives processed only if specific module is loaded
<**IfDefine** <defined name>	Conditional directives processed only if defined name is given on the command line of httpd following a -D option. e.g. httpd -f /etc/httpd/httpd.conf -D testname

Directives:

AuthType <type>	An HTTP authorization type (e.g., "Basic")
AuthName <Auth Realm>	The authentication realm (e.g. "Members Only")
Require	Selects which authenticated users or groups may access a protected space.
Satisfy <access policy>...........................	Access policy if both allow and require used (all or any)
ErrorDocument <errorNo> <Filename>.	Document (.html) sent to client if a request error occurs.
Options <option1 option 2 .....>..............	Default options applied to container that don't use options.
DefaultType <default MIME type>.........	Default MIME type for untypable files.
ServerSignature <on , off, email>........	Enable(on) or disable (off) server footer signature for served docs. Info in doc. is Server ver. No. and VirtualHost Name. email notifies the administrator(set by Server Admin) by email.
ContentDigest	Whether or not to send a Content-MD5 header with each request
LimitRequestBody	Limit (in bytes) on maximum size of request message body
DirectoryIndex <Filename(s)>.............	Sets the file name(s) that will be automatically sent to clients when accessing a directory only. e.g. www.mydomain.de/mysubdir/ This will display the index.html file if present in this dir.
RLimitCPU <limit in sec. per process>....	Soft/hard limits for max CPU usage in seconds per process. See Page 75 of Apache Server Bible
RLimitMEM <limit in bytes per process>	Soft/hard limits for max memory usage per process.
RlimitNPROC <Nr. of processes>..........	Soft/hard limits for max number of processes per user (uid).
ExpiresActive <On or Off>......................	Tells(On) the browser that the files generated cannot be refreshed, They will need to be reloaded. Useful when using PHP3.
SetHandler <handler name>	Sets the Handler module for a directory

Appendix D - Directives allowed in <*VirtualHost*> container.

(Page 81 of Apache Server Bible)

Containers:

- <**Directory** */dir* > Directory access container .
- <**DirectoryMatch** "*regex*" > Directory access container with regular expressions.(regex)
- <**Files** "[*path*]*file*" > File access container. Note the " surrounding the filename !
- <**FilesMatch** "*regex*" > File access container with regular expressions.(regex)
- <**Location** *URI* > URI access container.
- <**LocationMatch** "*regex*" > URI access container with regular expressions (regex)
- <**Limit** *METHOD(s)* > HTTP Methods container.
- <**LimitExcept** *METHOD(s)* > HTTP Methods container for undefined Methods
- <**IfModule** *module.c* > Conditional directives processed only if specific module is loaded
- <**IfDefine** *defined name* > Conditional directives processed only if defined name is given on the command line of httpd following a -D option.
  e.g. httpd -f /etc/httpd/httpd.conf -D testname

Directives:

- ServerName <Name>.......................... Name of the VirtualHost Server.
- DocumentRoot <Path to docs>........... Landing zone for documents served by this VirtualHost
- ServerAlias <Other name(s)>.............. Define other names that will be valid for this VirtualHost
- ServerAdmin <admin. email addr>....... Sets the email of the administrator of this VirtualHost
- UseCanonicalName.............................. How to work out the ServerName : Port when constructing URLs
- ErrorDocument <Filename>................. Document (.html) sent to client if a request error occurs.
- Redirect <requested URL> <new URL>.. Redirects a URL(can be a location) to a full new URL
- RedirectMatch<requested URL> <new URL>... Same as redirect but with regular expressions.
- and All Proxy Server directives

Appendix E - Options (used inside containers)

Syntax: Options [+|-]option [+|-]option ...
Context: server config, virtual host, directory, .htaccess

The Options directive controls which server features are available in a particular directory.
option can be set to **None**, in which case none of the extra features are enabled, or one or more of the following:

All All options included except for MultiViews. This is the default setting.
ExecCGI Execution of CGI scripts is permitted.
FollowSymLinks The server will follow symbolic links in this directory.
Note: even though the server follows the symlink it does not change the pathname used to match against other <Directory> sections.
Note: this option gets ignored if set inside a <Location> section.
Includes Server Side Includes(SSI) commands are permitted in HTML files.
IncludesNOEXEC Server Side Includes(SSI) are permitted, but the #exec and #include commands are disabled.
Indexes If a URL which maps to a directory is requested, and the there is no DirectoryIndex (e.g., index.html) in that directory, then the server will return a formatted listing(index) of the directory.
MultiViews Content negotiated MultiViews are allowed. This feature is a mechanism for guessing what the client wants when the URL requested doesn't exist.

SymLinksIfOwnerMatch The server will only follow symbolic links for which the target file or directory is owned by the same user id as the link.
Note: this option gets ignored if set inside a <Location> section.

Normally, if multiple Options could apply to a directory, then the most specific one is taken complete; the options are not merged. However if all the options on the Options directive are preceded by a + or - symbol, the options are merged. Any options preceded by a + are added to the options currently in force, and any options preceded by a - are removed from the options currently in force.

For example, without any + and - symbols:

```
<Directory /web/docs>
      Options Indexes FollowSymLinks
</Directory>

<Directory /web/docs/spec>
      Options Includes
</Directory>
```

then only Includes will be set for the /web/docs/spec directory. However if the second Options directive uses the + and - symbols:

```
<Directory /web/docs>
      Options Indexes FollowSymLinks
</Directory>

<Directory /web/docs/spec>
      Options +Includes -Indexes
</Directory>
```

then the options FollowSymLinks and Includes are set for the /web/docs/spec directory.
Note: Using -IncludesNOEXEC or -Includes disables server-side includes completely regardless of the previous setting. The default in the absence of any other settings is All.

Appendix F - Building 3rd party dynamically loadable modules with apxs

apxs script contains all the API header files info to allow to build modules without the need of Apache source code. The apxs is located in /usr/local/apache/bin/ dir.
see example of PHP compiling.

Adding the PHP3 module:

- Download the PHP module source for i386 from the web site http://www.php.net/download-php.php3
- Copy it to /usr/local/ directory `cp php-3.0.16.tar.gz /usr/local/`
- Uncompress it `cd /usr/local/` and `tar fvxz php-3.0.16.tar.gz`
- Create a php link in the same directory: `ln -s /usr/local/php-3.0.16 /usr/local/php`
- Compile PHP Module as per current Apache source header files:

```
#  ./configure --with-mysql --with-apxs=/usr/local/apache/bin/apxs --with-xml
#  make
#  make install
```

Copy the newly compiled PHP module to the apache module directory.

```
# cp /usr/local/php/libphp3.so   /usr/local/apache/libexec
```

- Edit the /usr/local/apache/conf/httpd.conf:

Uncomment the following lines or add them if needed:

```
<IfDefine PHP>
   AddType application/x-httpd-php3 .php3
   AddType application/x-httpd-php3 .php
   AddType application/x-httpd-php3-source .phps
   AddType application/x-httpd-php3 .phtml
</IfDefine>
```

After the LoadModule List, add the following lines:

```
<IfDefine PHP>
   LoadModule php3_module /usr/local/apache/libexec/libphp3.so
</IfDefine>
```

After the AddModule List, add the following lines:

```
<IfDefine PHP>
   AddModule  mod_php3.c
</IfDefine>
```

Add the underlined part to the following directive:

```
DirectoryIndex index.html index.htm index.php index.php3
```

- Restart or reload the Apache httpd Daemon:

```
# rcapache reload or
# rcapache restart
```

Adding the DAV module

"WebDAV stands for 'Web-based **D**istributed **A**uthoring and **V**ersioning'. It is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers."

DAV functionality includes creating, moving, copying, and deleting files and directories on a remote web server. Utilizing DAV requires both a DAV-aware client and server. mod_dav provides complete class 1 and 2 DAV services to DAV clients via the Apache Web Server (1.3.4 or later). The number of DAV-aware clients is growing and includes the 'Web Folders' used in Microsoft Internet Explorer 5.0 and Office 2000.

- Download the DAV module source for i386 from the web site
http://www.webdav.org/mod_dav/mod_dav-0.9.16-1.3.6.tar.gz

- Copy it to `/usr/local/ directory` `cp mod_dav-0.9.16-1.3.6.tar.gz /usr/local/`
- Uncompress it `cd /usr/local/` and `tar fvxz mod_dav-0.9.16-1.3.6.tar.gz`
- Create a dav link in the same directory:

```
ln -s /usr/local/mod_dav-0.9.16-1.3.6  /usr/local/dav
```

- Compile DAV Module as per current Apache source header files (all parameters on one line):

```
#  ./configure --with-apxs=/usr/local/apache/bin/apxs
#  make
#  make install
```

- The newly compiled DAV module(`libdav.so`) will automatically be copied to the apache module directory and some of the appropriate parameter (LoadModule) will be written to the httpd.conf file.
- To enable mod_dav, add the following directive to the appropriate container(s) in the httpd.conf file:

```
<Directory /usr/local/apache/htdocs>
        Options ........
        #
        # don't use DAV without access control !!
        #
        <IfDefine DAV>
                DAV On
        </IfDefine>
</Directory>
```

- Specify a location for the DAV lock database by adding a line similar to this to the httpd.conf file:

The DAVLockDB directive can be outside of any container; it only needs to appear **once**; and a file extension should not be supplied.

```
# To enable mod_dav, add the following directive to the
# appropriate container(s) in the httpd.conf file:
#
<IfDefine DAV>
        DavLockDB /var/lock/DAVLock
</IfDefine>
```

An optional directive, DAVMinTimeout, specifies the minimum lifetime of a lock in seconds. If a client requests a lock timeout less than DAVMinTimeout, then the DAVMinTimeout value will be used and returned instead. For example, Microsoft's Web Folders defaults to a lock timeout of 2 minutes; 10 minutes could be used to reduce network traffic and the chance that the client might lose a lock due to network latency.

A sample configuration segment might look like:

```
...
DAVLockDB /usr/local/apache/var/DAVLock
DAVMinTimeout 600

<Location />
     DAV On
     AuthType Basic
     AuthName DAV
  AuthUserFile dav.passwd
  <LimitExcept GET HEAD OPTIONS>
    require user webadmin
  </LimitExcept>
</Location>
```

The DAV spec (RFC 2518) does not incorporate a security model. It relies on any web server and file system security that the administrator configures. On Unix machines, the web server process must have permission to write to the DAV-enabled directories and any files to be modified. Local manipulation of files in a DAV-enabled directory is a bad thing. Specifically, DAV file locks are implemented by mod_dav, not the file system.

Appendix G - Options of Apache compiling program (*configure*)

```
# ./configure --help
```

```
Usage: configure [options]
Options: [defaults in brackets after descriptions]
General options:
 --quiet, --silent      do not print messages
 --verbose, -v          print even more messages
 --shadow[=DIR]         switch to a shadow tree (under DIR) for building

Stand-alone options:
 --help, -h             print this message
 --show-layout          print installation path layout (check and debug)

Installation layout options:
 --with-layout=[F:]ID   use installation path layout ID (from file F)
 --target=TARGET        install name-associated files using basename TARGET
 --prefix=PREFIX        install architecture-independent files in PREFIX
 --exec-prefix=EPREFIX  install architecture-dependent files in EPREFIX
 --bindir=DIR           install user     executables in DIR
 --sbindir=DIR          install sysadmin executables in DIR
 --libexecdir=DIR       install program  executables in DIR
 --mandir=DIR           install manual pages in DIR
 --sysconfdir=DIR       install configuration files in DIR
 --datadir=DIR          install read-only  data files in DIR
 --includedir=DIR       install includes files in DIR
 --localstatedir=DIR    install modifiable data files in DIR
 --runtimedir=DIR       install runtime data in DIR
 --logfiledir=DIR       install logfile data in DIR
 --proxycachedir=DIR    install proxy cache data in DIR

Configuration options:
 --enable-rule=NAME     enable  a particular Rule named 'NAME'
 --disable-rule=NAME    disable a particular Rule named 'NAME'
                        [DEV_RANDOM=default EXPAT=default   IRIXN32=yes ]
                        [IRIXNIS=no      PARANOID=no     SHARED_CHAIN=de]
                        [SHARED_CORE=default SOCKS4=no       SOCKS5=no  ]
                        [WANTHSREGEX=default                            ]
 --add-module=FILE      on-the-fly copy & activate a 3rd-party Module
 --activate-module=FILE on-the-fly activate existing 3rd-party Module
 --permute-module=N1:N2 on-the-fly permute module 'N1' with module 'N2'
 --enable-module=NAME   enable  a particular Module named 'NAME'
 --disable-module=NAME  disable a particular Module named 'NAME'
                        [access=yes      actions=yes     alias=yes      ]
                        [asis=yes        auth=yes        auth_anon=no   ]
                        [auth_db=no      auth_dbm=no     auth_digest=no ]
                        [autoindex=yes   cern_meta=no    cgi=yes        ]
                        [digest=no       dir=yes         env=yes        ]
                        [example=no      expires=no      headers=no     ]
                        [imap=yes        include=yes     info=no        ]
                        [log_agent=no    log_config=yes  log_referer=no ]
                        [mime=yes        mime_magic=no   mmap_static=no ]
                        [negotiation=yes proxy=no        rewrite=no     ]
                        [setenvif=yes    so=no           speling=no     ]
                        [status=yes      unique_id=no    userdir=yes    ]
                        [usertrack=no    vhost_alias=no                 ]
 --enable-shared=NAME   enable  build of Module named 'NAME' as a DSO
 --disable-shared=NAME  disable build of Module named 'NAME' as a DSO
 --with-perl=FILE       path to the optional Perl interpreter
 --without-support      disable the build and installation of support tools
 --without-confadjust   disable the user/situation adjustments in config
 --without-execstrip    disable the stripping of executables on installation

suEXEC options:
 --enable-suexec        enable the suEXEC feature
 --suexec-caller=NAME   set the suEXEC username of the allowed caller [www]
 --suexec-docroot=DIR   set the suEXEC root directory [PREFIX/share/htdocs]
 --suexec-logfile=FILE  set the suEXEC logfile [PREFIX/var/log/suexec_log]
 --suexec-userdir=DIR   set the suEXEC user subdirectory [public_html]
 --suexec-uidmin=UID    set the suEXEC minimal allowed UID [100]
 --suexec-gidmin=GID    set the suEXEC minimal allowed GID [100]
 --suexec-safepath=PATH set the suEXEC safe PATH [/usr/local/bin:/usr/bin:/bin]

Deprecated options:
 --layout               backward compat only: use --show-layout
 --compat               backward compat only: use --with-layout=Apache
```

Appendix H - Apache Full Status

Command: **`rcapache full-status`**

```
================================================================

  Apache Server Status for idefix.michel.home

  Server Version: Apache/1.3.9 (Unix) (SuSE/Linux) PHP/3.0.12
  Server Built: Nov 9 1999 02:46:17

  Current Time: Tuesday, 28-Mar-2000 16:16:47 CEST
  Restart Time: Tuesday, 28-Mar-2000 12:10:11 CEST
  Parent Server Generation: 1
  Server uptime: 4 hours 6 minutes 36 seconds
  Total accesses: 3 - Total Traffic: 4 kB
  CPU Usage: u.01 s.01 cu0 cs0 - .000135% CPU load
  .000203 requests/sec - 0 B/second - 1365 B/request
  1 requests currently being processed, 1 idle servers
W_..............................................................
................................................................
................................................................
................................................................

  Scoreboard Key:
  "_" Waiting for Connection, "S" Starting up, "R" Reading Request,
  "W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
  "L" Logging, "G" Gracefully finishing, "." Open slot with no current
  process

  Srv PID Acc M CPU SS Req Conn Child Slot Host VHost Request
  0-1 1367 0/2/2 W 0.02 14784 0 0.0 0.000 0.000 127.0.0.1
  idefix.michel.home GET /server-status HTTP/1.0
  1-1 1368 0/1/1 _ 0.00 63 54 0.0 0.00 0.00 localhost idefix.michel.home
  GET /server-status HTTP/1.0

  Srv Child Server number - generation
  PID OS process ID
  Acc Number of accesses this connection / this child / this slot
  M Mode of operation
  CPU CPU usage, number of seconds
  SS Seconds since beginning of most recent request
  Req Milliseconds required to process most recent request
  Conn Kilobytes transferred this connection
  Child Megabytes transferred this child
  Slot Total megabytes transferred this slot


   Apache/1.3.9 Server at idefix.michel.home Port 80
=================================================================
```

Appendix I - httpd Daemon and options

Command : `man httpd`

NAME

httpd - Apache hypertext transfer protocol server

SYNOPSIS

httpd [-X] [-R libexecdir] [-d serverroot] [-f con
fig] [-C directive] [-c directive] [-D parameter]

httpd [-h] [-l] [-L] [-v] [-V] [-S] [-t] [
-T]

DESCRIPTION

httpd is the Apache **H**yper**T**ext **T**ransfer **P**rotocol (HTTP)
server program. It is designed to be run as a standalone daemon process. When used like this it will create a pool of child processes to handle requests. To stop it, send a TERM signal to the initial (parent) process. The PID of this process is written to a file as given in the configuration file. Alternatively httpd may be invoked by the Internet daemon inetd(8) each time a connection to the HTTP service is made.

This manual page only lists the command line arguments.

For details of the directives necessary to configure httpd see the Apache manual, which is part of the Apache distribution or can be found at http://www.apache.org/. Paths in this manual may not reflect those compiled into httpd.

OPTIONS

-R <libexecdir>

This option is only available if Apache was built with the SHARED_CORE rule enabled which forces the Apache core code to be placed into a dynamic shared object (DSO) file. This file is searched in a hardcoded path under ServerRoot per default. Use this option if you want to override it.

-d <serverroot>

Set the initial value for the ServerRoot directive to serverroot. This can be overrid den by the ServerRoot command in the configu ration file. The default is /usr/local/apache.

-f <config>

Execute the commands in the file config on startup. If config does not begin with a /, then it is taken to be a path relative to the ServerRoot. The default is conf/httpd.conf.

-C <directive>

Process the configuration directive before reading config files.

-c <directive>

Process the configuration directive after reading config files.

-D <parameter>

Sets a configuration parameter which can be used with <IfDefine>...</IfDefine> sections in the configuration files to conditionally skip or process commands.

-h Output a short summary of available command line options.

-l Output a list of modules compiled into the server.

-L Output a list of directives together with expected arguments and places where the directive is valid.

-S Show the settings as parsed from the config file
(currently only shows the virtualhost settings).

-t Run syntax tests for configuration files only.
The program immediately exits after these syntax parsing with either a return code of 0 (Syntax OK) or return code not equal to 0 (Syntax Error).

-T Same as option -t but does not check the configured document roots.

-X Run in single-process mode, for internal debugging purposes only;
the daemon does not detach from the terminal or fork any children.
Do NOT use this mode to provide ordinary web service.

-v Print the version of httpd , and then exit.

-V Print the version and build parameters of httpd , and then exit.

FILES

/usr/local/apache/conf/httpd.conf
/usr/local/apache/conf/srm.conf
/usr/local/apache/conf/access.conf
/usr/local/apache/conf/mime.types
/usr/local/apache/conf/magic
/usr/local/apache/logs/error_log
/usr/local/apache/logs/access_log
/usr/local/apache/logs/httpd.pid

SEE ALSO

inetd(8).

Appendix J - Apache Configuration Core Directives

Command: **`/usr/sbin/httpd -L`**

```
---------------------------------
<Directory (http_core.c)
        Container for directives affecting resources located in the specified directories
        Allowed in *.conf only outside <Directory>, <Files> or <Location>
</Directory> (http_core.c)
        Marks end of <Directory>
        Allowed in *.conf only inside <Directory>, <Files> or <Location>
<Location (http_core.c)
        Container for directives affecting resources accessed through the specified URL paths
        Allowed in *.conf only outside <Directory>, <Files> or <Location>
</Location> (http_core.c)
        Marks end of <Location>
        Allowed in *.conf only inside <Directory>, <Files> or <Location>
<VirtualHost (http_core.c)
        Container to map directives to a particular virtual host, takes one or more host addresses
        Allowed in *.conf only outside <Directory>, <Files> or <Location>
</VirtualHost> (http_core.c)
        Marks end of <VirtualHost>
        Allowed in *.conf only outside <Directory>, <Files> or <Location>
<Files (http_core.c)
        Container for directives affecting files matching specified patterns
        Allowed in *.conf anywhere and in .htaccess
        when AllowOverride isn't None
</Files> (http_core.c)
        Marks end of <Files>
        Allowed in *.conf anywhere and in .htaccess
        when AllowOverride isn't None
<Limit (http_core.c)
        Container for authentication directives when accessed using specified HTTP methods
        Allowed in *.conf anywhere and in .htaccess
        when AllowOverride isn't None
</Limit> (http_core.c)
        Marks end of <Limit>
        Allowed in *.conf anywhere and in .htaccess
        when AllowOverride isn't None
<LimitExcept (http_core.c)
        Container for authentication directives to be applied when any
        HTTP method other than those specified is used to access the resource
        Allowed in *.conf anywhere and in .htaccess
        when AllowOverride isn't None
</LimitExcept> (http_core.c)
        Marks end of <LimitExcept>
        Allowed in *.conf anywhere and in .htaccess
        when AllowOverride isn't None
<IfModule (http_core.c)
        Container for directives based on existance of specified modules
        Allowed in *.conf anywhere and in .htaccess
        when AllowOverride isn't None
</IfModule> (http_core.c)
        Marks end of <IfModule>
        Allowed in *.conf anywhere and in .htaccess
        when AllowOverride isn't None
<IfDefine (http_core.c)
        Container for directives based on existance of command line defines
        Allowed in *.conf anywhere and in .htaccess
        when AllowOverride isn't None
</IfDefine> (http_core.c)
        Marks end of <IfDefine>
        Allowed in *.conf anywhere and in .htaccess
        when AllowOverride isn't None
<DirectoryMatch (http_core.c)
        Container for directives affecting resources located in the specified directories
        Allowed in *.conf only outside <Directory>, <Files> or <Location>
</DirectoryMatch> (http_core.c)
```

Marks end of <DirectoryMatch>
Allowed in *.conf only inside <Directory>, <Files> or <Location>

<LocationMatch (http_core.c)
Container for directives affecting resources accessed through the specified URL paths
Allowed in *.conf only outside <Directory>, <Files> or <Location>

</LocationMatch> (http_core.c)
Marks end of <LocationMatch>
Allowed in *.conf only inside <Directory>, <Files> or <Location>

<FilesMatch (http_core.c)
Container for directives affecting files matching specified patterns
Allowed in *.conf anywhere and in .htaccess
when AllowOverride isn't None

</FilesMatch> (http_core.c)
Marks end of <FilesMatch>
Allowed in *.conf anywhere and in .htaccess
when AllowOverride isn't None

AuthType (http_core.c)
An HTTP authorization type (e.g., "Basic")
Allowed in *.conf only inside <Directory>, <Files> or <Location> and in .htaccess
when AllowOverride includes AuthConfig

AuthName (http_core.c)
The authentication realm (e.g. "Members Only")
Allowed in *.conf only inside <Directory>, <Files> or <Location> and in .htaccess
when AllowOverride includes AuthConfig

Require (http_core.c)
Selects which authenticated users or groups may access a protected space
Allowed in *.conf only inside <Directory>, <Files> or <Location> and in .htaccess
when AllowOverride includes AuthConfig

Satisfy (http_core.c)
access policy if both allow and require used ('all' or 'any')
Allowed in *.conf only inside <Directory>, <Files> or <Location> and in .htaccess
when AllowOverride includes AuthConfig

AccessFileName (http_core.c)
Name(s) of per-directory config files (default: .htaccess)
Allowed in *.conf only outside <Directory>, <Files> or <Location>

DocumentRoot (http_core.c)
Root directory of the document tree
Allowed in *.conf only outside <Directory>, <Files> or <Location>

ErrorDocument (http_core.c)
Change responses for HTTP errors
Allowed in *.conf anywhere and in .htaccess
when AllowOverride includes FileInfo

AllowOverride (http_core.c)
Controls what groups of directives can be configured by per-directory config files
Allowed in *.conf only inside <Directory>, <Files> or <Location>

Options (http_core.c)
Set a number of attributes for a given directory
Allowed in *.conf anywhere and in .htaccess
when AllowOverride includes Options

DefaultType (http_core.c)
the default MIME type for untypable files
Allowed in *.conf anywhere and in .htaccess
when AllowOverride includes FileInfo

ServerType (http_core.c)
'inetd' or 'standalone'
Allowed in *.conf only outside <Directory>, <Files> or <Location>

Port (http_core.c)
A TCP port number
Allowed in *.conf only outside <Directory>, <Files> or <Location>

HostnameLookups (http_core.c)
"on" to enable, "off" to disable reverse DNS lookups, or "double" to enable double-reverse DNS lookups
Allowed in *.conf anywhere

User (http_core.c)
Effective user id for this server
Allowed in *.conf only outside <Directory>, <Files> or <Location>

Group (http_core.c)
Effective group id for this server
Allowed in *.conf only outside <Directory>, <Files> or <Location>

ServerAdmin (http_core.c)

The email address of the server administrator
Allowed in *.conf only outside <Directory>, <Files> or <Location>
ServerName (http_core.c)
The hostname of the server
Allowed in *.conf only outside <Directory>, <Files> or <Location>
ServerSignature (http_core.c)
En-/disable server signature (on|off|email)
Allowed in *.conf anywhere and in .htaccess
when AllowOverride isn't None
ServerRoot (http_core.c)
Common directory of server-related files (logs, confs, etc.)
Allowed in *.conf only outside <Directory>, <Files> or <Location>
ErrorLog (http_core.c)
The filename of the error log
Allowed in *.conf only outside <Directory>, <Files> or <Location>
PidFile (http_core.c)
A file for logging the server process ID
Allowed in *.conf only outside <Directory>, <Files> or <Location>
ScoreBoardFile (http_core.c)
A file for Apache to maintain runtime process management information
Allowed in *.conf only outside <Directory>, <Files> or <Location>
LockFile (http_core.c)
The lockfile used when Apache needs to lock the accept() call
Allowed in *.conf only outside <Directory>, <Files> or <Location>
AccessConfig (http_core.c)
The filename of the access config file. Default: access.conf
Allowed in *.conf only outside <Directory>, <Files> or <Location>
ResourceConfig (http_core.c)
The filename of the resource config file. Default: srm.conf
Allowed in *.conf only outside <Directory>, <Files> or <Location>
ServerAlias (http_core.c)
A name or names alternately used to access the server
Allowed in *.conf only outside <Directory>, <Files> or <Location>
ServerPath (http_core.c)
The pathname the server can be reached at
Allowed in *.conf only outside <Directory>, <Files> or <Location>
Timeout (http_core.c)
Timeout duration (sec)
Allowed in *.conf only outside <Directory>, <Files> or <Location>
KeepAliveTimeout (http_core.c)
Keep-Alive timeout duration (sec)
Allowed in *.conf only outside <Directory>, <Files> or <Location>
MaxKeepAliveRequests (http_core.c)
Maximum number of Keep-Alive requests per connection, or 0 for infinite
Allowed in *.conf only outside <Directory>, <Files> or <Location>
KeepAlive (http_core.c)
Whether persistent connections should be On or Off
Allowed in *.conf only outside <Directory>, <Files> or <Location>
IdentityCheck (http_core.c)
Enable identd (RFC 1413) user lookups - SLOW
Allowed in *.conf anywhere
ContentDigest (http_core.c)
whether or not to send a Content-MD5 header with each request
Allowed in *.conf anywhere and in .htaccess
when AllowOverride includes Options
UseCanonicalName (http_core.c)
How to work out the ServerName : Port when constructing URLs
Allowed in *.conf only outside <Directory>, <Files> or <Location>
StartServers (http_core.c)
Number of child processes launched at server startup
Allowed in *.conf only outside <Directory>, <Files> or <Location>
MinSpareServers (http_core.c)
Minimum number of idle children, to handle request spikes
Allowed in *.conf only outside <Directory>, <Files> or <Location>
MaxSpareServers (http_core.c)
Maximum number of idle children
Allowed in *.conf only outside <Directory>, <Files> or <Location>
MaxServers (http_core.c)
Deprecated equivalent to MaxSpareServers

Allowed in *.conf only outside <Directory>, <Files> or <Location>

ServersSafetyLimit (http_core.c)
Deprecated equivalent to MaxClients
Allowed in *.conf only outside <Directory>, <Files> or <Location>

MaxClients (http_core.c)
Maximum number of children alive at the same time
Allowed in *.conf only outside <Directory>, <Files> or <Location>

MaxRequestsPerChild (http_core.c)
Maximum number of requests a particular child serves before dying.
Allowed in *.conf only outside <Directory>, <Files> or <Location>

RLimitCPU (http_core.c)
Soft/hard limits for max CPU usage in seconds
Allowed in *.conf anywhere and in .htaccess
when AllowOverride isn't None

RLimitMEM (http_core.c)
Soft/hard limits for max memory usage per process
Allowed in *.conf anywhere and in .htaccess
when AllowOverride isn't None

RLimitNPROC (http_core.c)
soft/hard limits for max number of processes per uid
Allowed in *.conf anywhere and in .htaccess
when AllowOverride isn't None

BindAddress (http_core.c)
'*', a numeric IP address, or the name of a host with a unique IP address
Allowed in *.conf only outside <Directory>, <Files> or <Location>

Listen (http_core.c)
A port number or a numeric IP address and a port number
Allowed in *.conf only outside <Directory>, <Files> or <Location>

SendBufferSize (http_core.c)
Send buffer size in bytes
Allowed in *.conf only outside <Directory>, <Files> or <Location>

AddModule (http_core.c)
The name of a module
Allowed in *.conf only outside <Directory>, <Files> or <Location>

ClearModuleList (http_core.c)
Allowed in *.conf only outside <Directory>, <Files> or <Location>

ThreadsPerChild (http_core.c)
Number of threads a child creates
Allowed in *.conf only outside <Directory>, <Files> or <Location>

ExcessRequestsPerChild (http_core.c)
Maximum number of requests a particular child serves after it is ready to die.
Allowed in *.conf only outside <Directory>, <Files> or <Location>

ListenBacklog (http_core.c)
Maximum length of the queue of pending connections, as used by listen(2)
Allowed in *.conf only outside <Directory>, <Files> or <Location>

CoreDumpDirectory (http_core.c)
The location of the directory Apache changes to before dumping core
Allowed in *.conf only outside <Directory>, <Files> or <Location>

Include (http_core.c)
Name of the config file to be included
Allowed in *.conf anywhere

LogLevel (http_core.c)
Level of verbosity in error logging
Allowed in *.conf only outside <Directory>, <Files> or <Location>

NameVirtualHost (http_core.c)
A numeric IP address:port, or the name of a host
Allowed in *.conf only outside <Directory>, <Files> or <Location>

ServerTokens (http_core.c)
Determine tokens displayed in the Server: header - Min(imal), OS or Full
Allowed in *.conf only outside <Directory>, <Files> or <Location>

LimitRequestLine (http_core.c)
Limit on maximum size of an HTTP request line
Allowed in *.conf only outside <Directory>, <Files> or <Location>

LimitRequestFieldsize (http_core.c)
Limit on maximum size of an HTTP request header field
Allowed in *.conf only outside <Directory>, <Files> or <Location>

LimitRequestFields (http_core.c)
Limit (0 = unlimited) on max number of header fields in a request message
Allowed in *.conf only outside <Directory>, <Files> or <Location>

LimitRequestBody (http_core.c)
Limit (in bytes) on maximum size of request message body
Allowed in *.conf anywhere and in .htaccess
when AllowOverride isn't None

LoadModule (mod_so.c)
a module name and the name of a shared object file to load it from
Allowed in *.conf only outside <Directory>, <Files> or <Location>

LoadFile (mod_so.c)
shared object file or library to load into the server at runtime
Allowed in *.conf only outside <Directory>, <Files> or <Location>

Appendix K - HTTP Status Codes

(returned to client's browser)

- 100-199 Information Status Codes
  - 100 continue-ready to receive the rest of the request.
  - 101 switching protocols-for old or new HTTP protocols
- 200-299 Client successfull request
  - 200 OK
  - 201 URI successfully created
  - 202 Request accepted
  - 203 Meta-info in header is from another server
  - 203 Request accepted but nothing to send to client
  - 205 Request to reset document content at client's side
  - 206 Sucessfull partial retrival of a GET request
- 300-399 Request redirected. Server needs more info to perform the request
  - 300 Client need to chose one of the proposesd choices in document
  - 301 Requested resource doesn't exist on the server. Redirecting request
  - 302 Requested resource is temporarily moved from the server. Redirecting request
  - 303 Requested resource is found in different location. Please use this new one.
  - 304 Client should use it's cached copy. The requested doc has not been changed
  - 305 Use proxy specified by the Loction header to retrieve the requested resource
- 400-499 Client request incomplete
  - 400 Bad request. Syntax error in request.
  - 401 Unauthorised. Request can be performed only if user is authorized
  - 402 Payment required....(not implemented yet).
  - 403 Forbidden. Access to requested resource is forbidden.
  - 404 Not found. The requested document is not found on this server
  - 405 Method Not Allowed.
  - 406 Not acceptable.
  - 407 Proxy authentiction required
  - 408 Timeout of Request
  - 409 Request conflict
  - 410 Requested resource is permanently gone from the server
  - 411 Content-length header required from client
  - 412 Precondition failed
  - 413 Requested resource too large
  - 414 Requested URI too long
  - 415 Unsuppoeted media type.
- 500-599 Server Errors
  - 503 Service Unavailable. May be due to server is overloaded
  - 504 Gateway or proxy has timed out.
  - 505 HTTP version not supported

Appendix L - Configuring Apache using 3rd party programs:

Comanche Best of all: for Linux and Windows95/98/NT(english and spanish only)

To install it:

- Download the **`Comanche_xxxx.rpm`** file from internet.
- Issue the command: **`rpm -hiv  Comanche_xxxx.rpm`**
- Important: Make sure that the Include directives for configuration files for modules not loaded in Apache (in httpd.conf) are commented out with '#'.
  These Include directives are often found at the end of the httpd.conf.
- Start the program with the command :
  `comanche`
- Folllow the instructions of the wizzard.....and have fun.

- Binaries are found at:
  `http://www.covalent.net/projects/comanche` or
  `http://www.comanche.org`

LinuxConf Mainly for Linux but has a very good section on Apache Config.
Binaries found at:
ftp://ftp.solucorp.qc.ca/pub/linuxconf/devel/suse-7.3

Webmin Very good and adapted to various Distributions
http://www.webmin.com

Appendix M - Examples of FORMS and CGIs (used in exercises)

Short description of forms systax:

```
<!-- WHAT TO DO WHEN SUBMIT TYPE INPUT BUTTON IS PRESSED -->

  <FORM ACTION="./test2.mycgi" METHOD="GET">


<!-- INPUT TYPE=TEXT -->
     <B>Ihre Name: </B>
     <INPUT NAME="Name" TYPE="text" SIZE="53"><BR>
     <B>Ihre Addresse: </B>
     <INPUT NAME="Email" TYPE="Text" SIZE="53"><BR>
     <HR>

<!-- INPUT TYPE=TEXTAREA -->
     <TEXTAREA NAME="Address" ROWS="6" COLS="53"></TEXTAREA><BR>
     <B>Your E-Mail: </B>

<!-- INPUT TYPE=RADIO -->
    Geben Sie Ihre Zahlungsweise an:
    <input type=radio name="Zahlmethode" value="Mastercard"> Mastercard
    <br>
    <input type=radio name="Zahlmethode"  checked  value="Visa"> Visa
    <br>
    <input type=radio name="Zahlmethode" value="AmericanExpress">
    American Express

<!-- INPUT TYPE=CHECKBOX -->
    Ich mag:
    <input type=checkbox name="Vorliebe"  value="Urlaub"> Urlaub
    <input type=checkbox name="Vorliebe" checked value="Geld"> Geld
    <input type=checkbox name="Vorliebe" checked value="Fahrad"> Fahrad
    </p>

<!-- SELECT FROM LIST -->
    <p>Ihr Favorit:</p>
    <select name="top5" size=3>
         <option> Heino
         <option selected> Michael Jackson
         <option> Tom Waits
         <option> Nina Hagen
         <option> Marianne Rosenberg
    </select>
</FORM>

<!--  SENDING A FILE to CGI-->
<FORM action="/cgi-bin/upload.pl" method=post enctype="multipart/form-
data">
     <p>Senden Sie eine Text- oder HTML-Datei!</p>
     <input type=file size=50 maxlength=100000 name="Datei"
          accept="text/*"><br>
     <input type=submit value="Absenden">
</FORM>
```

```
<!-- HIDDEN ITEM IN FORM --->
     <FORM name="Feedback" action="mailto:abc@xy.com" method=post
enctype="text/plain">
     Ihr Name: <input name="UserName>"
          <input type=hidden name="UserBrowser" value="">
          <input type=submit value="Absenden">
     </FORM>

<!-- RESET THE FORM -->
     <input type=reset value="Abbrechen">

<!-- INPUT TYPE=SUBMIT -->
     <INPUT TYPE="submit" VALUE="Senden"></CENTER>


------------------------------------------------------------
```

Filename: **anmeldung.html**

```
<HTML>
<HEAD><TITLE>Teilnehmer Formulare</TITLE></HEAD>
<BODY>
  <FORM ACTION="/cgidir/test1" METHOD="GET">
      <HR>

      <B>Ihre Name: </B>
      <INPUT NAME="Name" TYPE="text" SIZE="53"><BR>

      <B>Ihre Addresse: </B>
      <TEXTAREA NAME="Address" ROWS="6"COLS="53">
      </TEXTAREA><BR>

      <B>Ihre E-Mail: </B>
      <INPUT NAME="Email" TYPE="Text" SIZE="53"><BR>
      <HR><P>
      <CENTER>
           <INPUT TYPE="submit" VALUE="Senden">
      </CENTER>
  </FORM>
</BODY>
</HTML>
```

Filename: test1.mycgi

```
#!/bin/sh
echo Content-type: text/html
echo

# This above header and empty echo after it is VERY important.
# Otherwise Apache declare an error...it can't find the Content-type header
# The HTML code enclosed in quotes is also very important......
echo "<HTML>"
echo "<HEAD>"
echo "<TITLE>This is a CGI test</TITLE>"
echo "</HEAD>"
echo "<BODY>"
echo "<FONT SIZR=4>"
echo "<Center><H1>CGI Environment Variables</H1></Center><BR>"
echo "<HR>"
echo "<FONT SIZE=4>"
echo "<Table border=0>"
echo "<TR><TD>SERVER_NAME:</TD><TD>$SERVER_NAME</TD></TR>"
echo "<TR><TD>HTTP_HOST:</TD><TD>$HTTP_HOST</TD></TR>"
echo "<TR><TD>HTTP_ACCEPT:</TD><TD>$HTTP_ACCEPT</TD></TR>"
echo "<TR><TD>HTTP_ACCEPT_CHARSET:</TD><TD>$HTTP_ACCEPT_CHARSET</TD></TR>"
echo "<TR><TD>HTTP_ACCEPT_LANGUAGE:</TD><TD>$HTTP_ACCEPT_LANGUAGE</TD></TR>"
echo "<TR><TD>HTTP_USER_AGENT:</TD><TD>$HTTP_USER_AGENT</TD></TR>"
echo "<TR><TD>HTTP_REFERER:</TD><TD>$HTTP_REFERER</TD></TR>"
echo "<TR><TD>HTTP_CONNECTION:</TD><TD>$HTTP_CONNECTION</TD></TR>"
echo "<TR><TD>SERVER_PORT:</TD><TD>$SERVER_PORT</TD></TR>"
echo "<TR><TD>REMOTE_HOST:</TD><TD>$REMOTE_HOST</TD></TR>"
echo "<TR><TD>REMOTE_PORT:</TD><TD>$REMOTE_PORT</TD></TR>"
echo "<TR><TD>REMOTE_ADDR:</TD><TD>$REMOTE_ADDR</TD></TR>"
echo "<TR><TD>REMOTE_USER:</TD><TD>$REMOTE_USER</TD></TR>"
echo "<TR><TD>SERVER_PROTOCOL:</TD><TD>$SERVER_PROTOCOL</TD></TR>"
echo "<TR><TD>REQUEST_METHOD:</TD><TD>$REQUEST_METHOD</TD></TR>"
echo "<TR><TD>REQUEST_URI:</TD><TD>$REQUEST_URI</TD></TR>"
echo "<TR><TD>REMOTE_IDENT:</TD><TD>$REMOTE_IDENT</TD></TR>"
echo "<TR><TD>AUTH_TYPE:</TD><TD>$AUTH_TYPE</TD></TR>"
echo "<TR><TD>CONTENT_TYPE:</TD><TD>$CONTENT_TYPE</TD></TR>"
echo "<TR><TD>CONTENT_LENGTH:</TD><TD>$CONTENT_LENGTH</TD></TR>"
echo "<TR><TD>SCRIPT_NAME:</TD><TD>$SCRIPT_NAME</TD></TR>"
echo "<TR><TD>SCRIPT_FILENAME:</TD><TD>$SCRIPT_FILENAME</TD></TR>"
echo "<TR><TD>QUERY_STRING:</TD><TD>$QUERY_STRING</TD></TR>"
echo "<TR><TD>PATH_INFO:</TD><TD>$PATH_INFO</TD></TR>"
echo "<TR><TD>PATH_TRANSLATED:</TD><TD>$PATH_TRANSLATED</TD></TR>"
echo "</TABLE><BR><HR>"
echo "</FONT>"

#--Display all the CGI Environment Variables list and values ----------
echo "<Center><H1>Environment variables (All of them!)</H1></Center><BR>"
printenv | sort | sed -e 's/.*$/&\<BR\>/'
echo "<HR>"

#---Display only CGI Environment Variables created by 'uncgi'------------
echo "<Center><H1>uncgi generated Environment variables</H1></Center><BR>"
printenv | grep "WWW_" | sed -e 's/.*$/&\<BR\>/'
echo "<HR>"

#---Change the special codes given by browser for space, newline, @ etc ,
#---Convert the + to space, %0D%0A to <BR>, %40 to @
ConvertedSTR=`echo "$QUERY_STRING"| sed -e 's/\%0A/\<BR\>/g' -e 's/\%0D//g' -e
's/\%40/\@/g' -e 's/\+/\ /g'`
```

```
#---Separate the 3 NAME=DATA ----------------------------------------
param1=`echo "$ConvertedSTR" | cut -d "&" -f 1 `
param2=`echo "$ConvertedSTR" | cut -d "&" -f 2 `
param3=`echo "$ConvertedSTR" | cut -d "&" -f 3 `

#---Separate the NAME and the DATA from the NAME=DATA -------------
kw1=`echo "$param1" | cut -d "=" -f 1`
val1=`echo "$param1" | cut -d "=" -f 2`
kw2=`echo "$param2" | cut -d "=" -f 1`
val2=`echo "$param2" | cut -d "=" -f 2`
kw3=`echo "$param3" | cut -d "=" -f 1`
val3=`echo "$param3" | cut -d "=" -f 2`


#---Display the CGI Environment Variables list and values ----------------
echo "<Center><H1>CGI Parameters List</H1></Center><BR>"
echo "Parameter 1 =  $kw1<BR>"
echo "Value 1 = $val1<BR>"
echo "Parameter 2 = $kw2<BR>"
echo "Value 2 = $val2<BR>"
echo "Parameter 3 = $kw3<BR>"
echo "Value 3 = $val3<BR>"
echo "<HR>"
```

Filename: **test2.mycgi** (This file includes above **test1.mycgi** and the following)

```
#---Adding the Name,Address,e-mail to the visitors file -----------
echo "$val1,$val2,$val3-IPAddr: $REMOTE_ADDR" >> visitors.cvs
echo "<Center><H1>Visitors List</H1></Center>"
cat visitors.cvs | sed -e 's/.*$/&\<BR\>/'
echo "<HR>"


#---Display which user and group the CGI is identied in Linux system -----
echo "<Center><H1>This CGI is identified as: <BR>"
user=`id -nu`
group=`id -ng`
echo "User = $user <BR>"
echo "Group = $group <BR>"
echo "</H1></Center><BR>"
echo "<HR>"


#---Display all the system Processes -----------------------------------
echo "<Center><H1>System Processes</H1></Center><BR>"
ps -ax | sed  -e 's/\ \ PID/\<B\>&/' -e 's/.*COMMAND$/&\<\/B\>/' -e 's/.*
$/&\<BR\>/'
echo "<HR>"

#---Display free space of all mounted disks in Linux ---------------------
echo "<Center><H1>Disk Space</H1></Center><BR>"
df -h | sed -e 's/.*$/&\<BR\>/'
echo "<HR>"

#---Display the Kernel Processes list -----------------------------------
echo "<Center><H1>Kernel Process Info</H1></Center><BR>"
procinfo -a | sed -e 's/.*$/&\<BR\>/'
echo "<HR>"

#---Display who is logged-in now ----------------------------------------
echo "<Center><H1>Who is logged now</H1></Center><BR>"
w | sed -e 's/.*$/&\<BR\>/'
echo "<HR>"


#---Display who were the last 20 logins (incl reboots)---------------------
echo "<Center><H1>Who were the last 20 logins (incl reboots)</H1></Center><BR>"
last -20 | sed -e 's/.*$/&\<BR\>/'
echo "<HR>"

#---wwwrun lauft eine ROOT SYSTEM programme ----------
#---Das is nur m.glish durch sudo und /etc/sudoers einstellung
##/etc/sudoers inhalt
#root   ALL=(ALL) ALL
#Host_Alias THIS_HOST=hof400
#Cmnd_Alias SYSTEM=/sbin/fdisk -l,/sbin/modprobe ppa
#wwwrun THIS_HOST=NOPASSWD:SYSTEM

echo "<Center><H1>Festplatteliste auf dem Server</H1></Center><BR>"
sudo /sbin/fdisk -l | sed -e 's/.*$/&\<BR\>/'
echo "<HR>"

#------------ END of CGI Script -----------------------------------------
echo "</BODY>"
echo "</HTML>"
```

Example of a search engine FORM using Ht://Dig

```
<HTML>
   <HEAD>
         <TITLE>Suche durch </TITLE>
   </HEAD>
   <BODY BGCOLOR="#ffffff">    <HR>
    <Table>
        <TR>
          <img align="center" src="./htdig.gif">
      </TR>
      <TR>
          <center>
          <form method="GET" action="/cgi-bin/htsearch">
            <font size=-1>
            <H3>Start eine Suche mit </H3>
            <center>

            <select name=method>
                <option value="and">Und-Verknuepfung der Worte</option>
                <option value="or" Selected>
                  Oder-Verknuepfung der Worte</option>
            </select>

                  <Select name=config>
                        <option value="bashshell">bashshell.conf</option>
                        <option value="forms">forms.conf</option>
                        <option value="htdigv">htdigv.conf</option>
                        <option value="linuxkurs">linuxkurs.conf</option>
                        <option value="manual">manual.conf</option>
                        <option value="samba">samba.conf</option>
                        <option value="selfhtml">selfhtml.conf</option>
                        <option value="webalizer">webalizer.conf</option>
                  </Select>

                  , Suchbegriffe:
                  <input type="text" size="30" name="words" value="">
                  <input type="submit" value="Search">

                  <select name="sort">
                        <option value="score" selected>Score
                        <option value="time">Time
                        <option value="title">Title
                        <option value="revscore">Reverse Score
                        <option value="revtime">Reverse Time
                        <option value="revtitle">Reverse Title
                  </select>
            </form>
          </center>
      </TR>
    </Table>
   </BODY>
</HTML>
```

Appendix N - Using mod_gzip to speed-up html connections:

Want to make your web server faster without getting a faster connection? All common browsers will transparently download content with gzip compression, but your out-of-the-box Apache probably doesn't have mod_gzip installed and turned on. Get the source from `http://www.schroepl.net/projekte/mod_gzip/` and add the following lines to your httpd.conf to turn it on:

```
LoadModule gzip_module /usr/lib/apache/1.3/mod_gzip.so

mod_gzip_on                 Yes
mod_gzip_maximum_file_size  0
mod_gzip_keep_workfiles     No
mod_gzip_temp_dir           /tmp
mod_gzip_item_include       mime ^text/.*
```

Appendix O - PDO support for PHP5 and MySQL database(Debian)

Standard Debian (Sarge) doesn't provide packages for PDO support in PHP.
Here are the steps to get it going for PHP5:

1) add the following lines in the /etc/apt/sources.list

```
deb http://dotdeb.pimpmylinux.org/ stable all
deb-src http://dotdeb.pimpmylinux.org/ stable all
deb http://dotdeb.netmirror.org/ stable all
deb-src http://dotdeb.netmirror.org/ stable all
```

2) issue the following commands:

```
apt-get update
apt-get install libapache2-mod-php5
```

It should automatically install the `dotdeb` versions the following way:

```
The following extra packages will be installed:
  php5-common php5-gd php5-mysql php5-xsl
Suggested packages:
  php-pear
The following packages will be upgraded:
  libapache2-mod-php5 php5-common php5-gd php5-mysql php5-xsl
```

3) Make sure that the modules will be loaded by including the symlinks in:

```
/etc/apache2/modules-enabled
```

4) Notice that the extra php new modules `.ini` files that are automatically read are in `/etc/php5/conf.d/` which will contain at least the following instructions:

```
extension=pdo.so
extension=pdo_mysql.so
```

5) Restart Apache2

6) Using a browser somehow load through this apache2 a php file that has the content:

```
<?
     phpinfo ();
?>
```

Then, in this generated PHP web page look for the sections:

```
     PDO
PDO drivers                                            mysql

     pdo_mysql
PDO Driver for MySQL, client library version           4.1.11
```

If you see this, the PDO for MySQL is loaded and ready.

--

Appendix P - Configuring mod_security module

```
# Example configuration file for the mod_security Apache module

LoadModule security_module modules/mod_security.so

<IfModule mod_security.c>

    # Turn the filtering engine On or Off
    SecFilterEngine On

    # The audit engine works independently and
    # can be turned On of Off on the per-server or
    # on the per-directory basis
    SecAuditEngine RelevantOnly

    # Make sure that URL encoding is valid
    SecFilterCheckURLEncoding On

    # Unicode encoding check
    SecFilterCheckUnicodeEncoding On

    # Only allow bytes from this range
    SecFilterForceByteRange 1 255

    # Cookie format checks.
    SecFilterCheckCookieFormat On

    # The name of the audit log file
    SecAuditLog logs/audit_log

    # Should mod_security inspect POST payloads
    SecFilterScanPOST On

    # Default action set
    SecFilterDefaultAction "deny,log,status:406"

    # Simple example filter
    # SecFilter 111
    # Prevent path traversal (..) attacks
    # SecFilter "\.\./"

    # Weaker XSS protection but allows common HTML tags
    # SecFilter "<( |\n)*script"

    # Prevent XSS atacks (HTML/Javascript injection)
    # SecFilter "<(.|\n)+>"

    # Very crude filters to prevent SQL injection attacks
    # SecFilter "delete[[:space:]]+from"
    # SecFilter "insert[[:space:]]+into"
    # SecFilter "select.+from"

    # Require HTTP_USER_AGENT and HTTP_HOST headers
    SecFilterSelective "HTTP_USER_AGENT|HTTP_HOST" "^$"

    # Only accept request encodings we know how to handle
    # we exclude GET requests from this because some (automated)
    # clients supply "text/html" as Content-Type
    SecFilterSelective REQUEST_METHOD "!^GET$" chain
    SecFilterSelective HTTP_Content-Type "!(^$|^application/x-www-form-
urlencoded$|^multipart/form-data)"

    # Require Content-Length to be provided with
    # every POST request
    SecFilterSelective REQUEST_METHOD "^POST$" chain
    SecFilterSelective HTTP_Content-Length "^$"

    # Don't accept transfer encodings we know we don't handle
    # (and you don't need it anyway)
    SecFilterSelective HTTP_Transfer-Encoding "!^$"

    # Some common application-related rules from
    # http://modsecrules.monkeydev.org/rules.php?safety=safe

    #Nuke Bookmarks XSS
    SecFilterSelective THE_REQUEST "/modules\.php\?
name=Bookmarks\&file=(del_cat\&catname|del_mark\&markname|edit_cat\&catname|
edit_cat\&catcomment|marks\&catname|
uploadbookmarks\&category)=(<[[:space:]]*script|(http|https|ftp)\:/)"

    #Nuke Bookmarks Marks.php SQL Injection Vulnerability
    SecFilterSelective THE_REQUEST "modules\.php\?
name=Bookmarks\&file=marks\&catname=.*\&category=.*/\*\*/(union|select|delete|
insert)"

    #PHPNuke general XSS attempt
    #/modules.php?name=News&file=article&sid=1&optionbox=
    SecFilterSelective THE_REQUEST "/modules\.php\?*name=<[[:space:]]*script"

    # PHPNuke SQL injection attempt
    SecFilterSelective THE_REQUEST "/modules\.php\?*name=Search*instory="

    #phpnuke sql insertion
    SecFilterSelective THE_REQUEST
"/modules\.php*name=Forums.*file=viewtopic*/forum=.*\'/"

    # WEB-PHP phpbb quick-reply.php arbitrary command attempt

    SecFilterSelective THE_REQUEST "/quick-reply\.php" chain
    SecFilter "phpbb_root_path="

    #Topic Calendar Mod for phpBB Cross-Site Scripting Attack
    SecFilterSelective THE_REQUEST "/calendar_scheduler\.php\?
start=(<[[:space:]]*script|(http|https|ftp)\:/)"

    # phpMyAdmin: Safe

    #phpMyAdmin Export.PHP File Disclosure Vulnerability
    SecFilterSelective SCRIPT_FILENAME "export\.php$" chain
    SecFilterSelective ARG_what "\.\."

    #phpMyAdmin path vln
    SecFilterSelective REQUEST_URI "/css/phpmyadmin\.css\.php\?GLOBALS\[cfg\]\
[ThemePath\]=/etc"
```