

넥스트라인 기술지원부 김삼수([kiss@nextline.co.kr](mailto:kiss@nextline.co.kr))

httpd.conf 설정

설정파일 경로

소스 컴파일 버전 : /usr/local/apache/conf/httpd.conf

RPM 설치 버전 : /etc/httpd/conf/httpd.conf

전체 환경 (Global Environment)

아파치 웹서버 프로세스의 전체적인(global) 부분에 영향을 미치는 부분입니다. 이 부분의 지시자들을 수정(변경)하시면 아파치 메인서버(Section 2)와 가상호스트(Section 3)에 설정된 모든 웹사이트에 영향을 미치게 됩니다.

ServerType standalone

서버의 구동방법으로는 standalone 과 inetd 방식이 있는데, standalone 방식은 하나의 웹데몬(아파치서버)이 클라이언트의 접속을 모두 처리하는 방식으로 응답속도가 빠른 방법으로 주로 이방식을 사용합니다. inetd 방식은 inetd 라는 시스템의 /etc 디렉토리 끝에 존재하는 inetd 라는 슈퍼데몬이 클라이언트의 접속요구가 있을 때마다 웹서버를 구동하는 방식이다. 일반적으로 응답속도가 빠르고 효율적인 standalone 으로 설정하여 사용합니다.

ServerRoot "/usr/local/apache/htdocs"

아파치서버의 홈디렉토리를 지정하며 절대경로로 지정합니다. 이후로 나오는 대부분의 패스들은 이 경로에 대한 상대경로로 지정이 됩니다. 예를 들어 환경설정파일, 에러로그 파일등의 상대경로의 기준이 되는 위치이다.

LockFile logs/accept.lock

아파치 컴파일 시 USE\_FCNTL\_SERIALIZED\_ACCEPT 나 USE\_FLOCK\_SERIALIZED\_ACCEPT 으로 컴파일 했을 때 사용되는 LockFile 의 경로 지정시에 사용됩니다. 가급적 기본값으로 사용합니다.

PidFile /var/run/httpd.pid

PidFile 설정은 ServerType 을 Standalone 으로 설정했을때만 유효한 것으로 아파치 서버의 프로세스가 생성되어 있을 때 그 프로세서 ID(PID)를 기록하는 파일을 지정합니다. 당연히 아파치서버가 재시작되거나 과부하로 인해 PID 가 바뀌게 될 경우에는 이 파일의 PID 값도

바뀌게 됩니다. 즉 다시말해서 여기서 지정된 파일(httpd.pid)에 실행되고 있는 아파치 서버의 프로세스번호(PID)값이 기록된다고 하면 정답이다. ServerRoot 를 기준으로한 상대경로로 지정됩니다. 절대경로로 지정하려면 "/"로 시작하는 절대경로를 적어주면 됩니다.

```
ResourceConfig conf/srm.conf
```

```
AccessConfig conf/access.conf
```

아파치 서버의 환경설정파일은 3 개이며 httpd.conf, srm.conf, access.conf 가 그것입니다. 그러나 하나의 설정파일로 하는 것이 효율적이기 때문에 지금은 httpd.conf 파일안에 3 개의 파트(Section)로 나누어서 하나의 파일안에서 설정을 하고 있습니다. srm.conf 와 access.conf 파일의 내용은 현재 비어있는 상태이지만, 필요한다면 이 파일 내에도 설정을 할 수 있습니다. 아파치 서버가 실행이 될 때는 httpd.conf, srm.conf, access.conf 순으로 언제나 이 3 개의 파일을 모두 읽고 난뒤에 실행이 되기 때문이다. 만약 이 두 개의 파일을 서버가 무시하도록 하려면 다음과 같이 하거나 ""으로 붙여 두면 주석 처리되어 무시됩니다.

```
Timeout 300
```

클라이언트의 요청에 의해 서버와 연결이 되었을 때 클라이언트와 서버간에 아무런 메시지가 발생하지 않았을 때 오류로 처리될 시간을 초단위로 설정합니다. 초기값은 1200 이며 보통은 300 초로 지정을 합니다. 네트워크의 속도가 나쁠수록 수치값은 높게 설정하는 것이 좋다.

```
KeepAlive On
```

HTTP 1.0 에서는 요청이 일어날 때마다 client 와 Server 간에 새로운 연결이 만들어 지는데 이 설정으로 인하여 하나의 연결에서 여러 요청이 가능하므로 요청을 처리하는 시간을 증진 시킨다. 이 기능을 끄려면 off 로 합니다.

대형 site 일 경우에는 이 기능을 끄기를 권장합니다. 이 기능을 Off 를 시킬 경우에 system 의 부하가 상당히 늘어 나는 것을 느낄 수는 있으나 Web 상의 속도에서는 속도가 On 일 경우보다 더욱 빠르다는 것을 체감할 수 있습니다. 즉 Web Server 만을 돌린다면 Off 로 하는 것을 권장하며 여러 서비스를 할 경우에는 On 으로 하라는 의미이다. 참고로 이런 것을 체험할 정도의 대형 서비스란 하루 웹로그가 1G 이상 쌓이는 경우를 의미 하며 웬만한 site에서는 이 부분에 대해서 신경을 쓰지 않아도 상관없다.

```
MaxKeepAliveRequests 100
```

접속된채로 특별한 요청이 없음에도 계속 연결을 유지시킬 수치를 지정합니다. 값이 너무 크면 하나의 client 가 Server 의 resource 를 독점 할 수 있으므로 적당 하게 잡는 것이

좋다.

KeepAliveTimeout 15

아파치 서버는 같은 접속상태의 클라이언트에서 여기서 지정한 초만큼의 요청이 없었을 때 접속을 끊게 됩니다.

MinSpareServers 8

MaxSpareServers 20

아파치 웹서버는 성능향상과 빠른 응답속도를 위해 유틸리티(현재 서비스대기 중인 프로세스)를 만들게 되는데 이 유틸리티의 개수는 시스템의 상황에 따라 달라지게 됩니다. 유틸리티가 MinSpareServers 의 개수(5) 보다 적게되면 추가로 생성을 하게 되며 MaxSpareServers 의 개수(10)보다 많게 되면 죽이게 됩니다. 즉, 유틸리티의 개수를 적절히 조절하기 위한 것이라 생각하면 됩니다.

StartServers 10

httpd 서버를 처음 실행시킬때, 여분의 프로세스를 생성시킬 수치를 지정합니다. 반응 시간을 짧게 하기 위해 StartServers 항목에서 말하는 만큼의 Server process 를 이미 만들어 두는 것인데, 실제로 Service 를 하고 있지 않을 경우에는 잠자고 있으므로 System 에 부하를 주거나 하지는 않는다.

MaxClients 150

아파치웹서버에 접근할 수 있는 클라이언트의 최대갯수는 이 상한값으로 제한합니다. 여기서 지정한 개수이상의 클라이언트의 요청이 생긴다면 아파치는 응답하지 않고 이 요청을 무시합니다. 이를 제한하는 이유는 시스템의 자원을 아파치 웹서버가 무한정 차지하는 것을 방지하기 위한 것입니다.

MaxRequestsPerChild 100

아파치 웹서버의 자식프로세스들이 클라이언트의 요청 개수를 지정합니다. 만약 자식프로세스가 이 값만큼의 클라이언트요청을 받았다면 이 자식프로세스는 자동으로 죽게됩니다. 이 값이 0 으로 설정이 된다면 자식프로세스가 자동으로 죽는일은 없을 것입니다. 그러나 0 아닌 다른 값으로 설정함으로써 프로세스의 수를 적절히 조절하여 시스템의 부하조절과 자원낭비를 어느정도 방지 할 수 있습니다.

Listen 3000

Listen 12.34.56.78:80

시스템의 기본값 이외에 다른 IP Address 와 포트에 대해서도 연결할 수 있도록 해 줍니다.

환경설정파일(httpd.conf) 맨뒤에 나오는 가상호스트(Virtual Host)부분에서 설정되는 가상호스트를 설정하기 위해 필요합니다.

#### BindAddress \*

서버가 응답할 수 있는 IP Address 를 설정하는 것입니다. 하나의 시스템에있는 아파치웹서버 하나로 여러 웹서버처럼 관리하는 웹호스팅서비스등에서 많이 이용하는 것으로 여러 IP Address 를 인식할 수 있게 합니다. "\*"으로 설정이 되었다면 모든 IP Address 에 대해 응답할 수 있으며, IP Address 를 지정한다면 지정한 IP Address 에 대해서만 응답할 수 있게 됩니다. 여러개의 IP Address 를 ISP 로부터 할당받아서 웹호스팅서비스를 하고자 한다면 이부분에서 지정해 주면 됩니다.

이 설정파일의 맨 뒷부분에 나오는 <VirtualHost> ~ </VirtualHost>부분의 IP bind 가상호스트부분에서 아파치 웹서버가 응답할 수 있도록 하려면 여기서 IP Address 를 지정해 줘야 합니다.

#### ExtendedStatus On

server-status 로 아파치웹서버의 상태를 상태를 모니터링 할 때 "자세한상태정보"기능을 제공할 것인지(On) 아닌지(Off)를 설정하는 것입니다.

#### '주(Main)' 서버 설정

이 부분은 가상호스트에서 핸들링되지 않는 요구들에 응답하는 아파치 메인(기본, default)서버의 설정부분입니다. Section 2 에서 설정하는 항목들은 아파치의 주된 서버가 사용할 값들을 지정합니다. Section 3 에 설정하는 <VirtualHost>에 정의된 가상호스트들에서 지정하지 않는 것은 여기서 지정된 값이 기본값으로 적용됩니다.

또한 여기서 지정하는 값을 각 <VirtualHost>내에도 지정할 수 있으며 이경우엔 각 <VirtualHost>내에서 지정한 값이 우선적으로 적용됩니다. 즉, 이 섹션의 지시자들은 <VirtualHost>정의에 의해 서비스되지 않는 요구들에 대한 부분만 응답하는 아파치메인서버 설정값들을 설정합니다. 또한 중요한 것은 여기서 설정하는 값들은 이 파일의 뒷부분에서 설정하고 있는 <VirtualHost>내의 기본값을 제공하는 역할도 한다는 것을 기억하시기 바랍니다. 그리고 Section 2 에서 설정하는 모든 지시자들은 Section 3 부분의 <VirtualHost>내에서도 별도로 사용될 수 있습니다. 만약 동일한 지시자가 Section 2와 Section3의 <VirtualHost>에 모두 설정이 될 경우에는 Section 3 의 <VirtualHost>내의 설정이 적용된다는 점도 반드시 알아두셔야 합니다.

#### Port 80

아파치웹서버의 기본포트를 지정합니다. 특별하게 사용하는 것이 아니라면 80 번으로

해줘야 합니다. 사용가능한 포트는 0 ~ 65535 이며 1024 이하의 포트번호는 시스템에서 특별하게 예약되어 있으므로 80 번 이외의 다른 포트를 사용하려면 1024 이상의 포트번호를 지정해서 사용해야 할 것입니다.

특별한 지정이 없다면 <VirtualHost>에 정의된 각각의 가상호스트들의 기본포트가 됩니다. 만약 <VirtualHost> 내에서 Port 가 지정이 된다면 그 포트번호가 우선합니다.

(특별히 PORT 를 따로 지정해 줄 필요가 있을 때는 따로 지정해 주며, 이때는 웹서버로 접근할 때 반드시 따로지정한 PORT 번호로 접근해야 합니다. 예를들어 Port 1234 로 지정했다면, 접근시 : http://www.domain.co.kr:1234 로 접속해야합니다. 단, 80 번은 default 이므로 Port 번호를 입력하지 않아도 도메인만으로 그냥 접근할 수 있습니다. 예: http://www.domain.co.kr )

User nobody

Group nobody

아파치 웹데몬이 요청을 받았을 때 여기서 지정한 user 와 group 으로 응답을 하게됩니다. 이 설정은 ServerType 이 Standalone 방식이며, 아파치의 실행이 root 권한으로 실행이 되었을 때 유효한 것입니다. 많은 웹서버관리자들이 nobody 로 설정을 해 두고 있으며, 만약 시스템에 nobody user 가 없다면 새로생성(useradd)을 해야 할 것입니다. 단, root 로 설정하는 것은 절대로 있어서는 안되며 nobody 이외의 다른 시스템사용자 id 로 지정을 한다면 정말 신중히 모든면(시스템 보안 및 자원사용등)에서 깊게 고려를 해보아야 합니다.

ServerAdmin admin@nextline.co.kr

여기서 지정하는 email address 는 웹 문서 로딩 에러 등의 문제에서 클라이언트측으로 보내질 메일 주소 값 이다. 대부분 웹 서버관리자의 email address 로 설정을 합니다.

ServerName www.nextline.co.kr

클라이언트에게 보여주는 호스트이름을 지정합니다.

www 를 쓰지 않는 호스트에서 www 를 쓰는 것처럼 보이게 할 수 있습니다.

예를 들어 bbs.nextline.co.kr 을 www.nextline.co.kr 로 지정해서 쓸 수 있습니다. 이곳에 IP Address 를 적게 되면 클라이언트에는 Ip Address 를 보여줍니다.

DocumentRoot "/usr/local/apache/htdocs"

아파치 웹서버의 웹문서가 있는 경로를 지정합니다.

예를 들어 "http://www.nextline.co.kr/index.html"의 초기 문서라면 이 초기문서의 절대 경로는 여기서 지정된 "/usr/local/apache/htdocs/index.html"이 됩니다.

경로의 맨 마지막에 "/"를 추가해서는 안됩니다. Alias 를 사용하여 다른 위치를 지정할 수도 있습니다.

```
<Directory />
  Options FollowSymLinks
  AllowOverride None
  Order allow,deny
  Allow from all
  Deny from env=no_access
</Directory>
```

<directory> tag에 의하여 각 directory마다 적절하게 permission을 걸 수가 있습니다. "Deny from env=no\_access" 설정에 대해서는 뒤에 다루기로 합니다.

```
<Directory "/usr/local/apache/htdocs">
  Options Indexes FollowSymLinks Includes
```

#### ① Options

None

어떤 옵션도 이용할 수 없습니다.

All

지정한 directory에서 모든 명령을 이용할 수 있습니다.

Indexes

URL에 지정된 디렉토리에 (index.html 같은) 지정된 파일이 없을 경우 디렉토리의 파일 목록을 보여주는 옵션입니다.

Includes

서버측의 추가적인 정보를 제공할 수 있게 합니다.

IncludesNoExec

서버측의 추가적인 정보를 제공할 수 있게 하지만, 어떠한 실행 파일을 실행하는 것을 방지합니다.

FollowSymLinks

디렉토리상의 심볼릭 링크를 사용 가능하게 합니다.

#### ExecCGI

CGI 스크립트를 실행할 수 있게 합니다.

#### MultiViews

All 옵션이 설정되었을 때만 지정된 목록의 multiviews 를 허용합니다.

#### ② AllowOverride None

.htaccess 파일은 서버의 각 디렉토리에 만들어서 각 디렉토리에 대한 접근을 제어하기 위한 것으로 디렉토리에 .htaccess 파일이 있으면, 서버 전체에 작용하는 access.conf 보다 우선권을 가집니다. .htaccess 파일에 대한 Override 에 대한 옵션입니다.

#### None

.htaccess 파일을 읽을 수 없게 합니다.

#### All

모든 지정에 대해 가능하게 합니다.

#### Options

규정된 디렉토리 형식을 컨트롤하는 지정의 사용을 허락합니다.

#### FileInfo

문서형식을 컨트롤하는 지정의 사용을 허용합니다.

#### AuthConfig

사용자 인증 지정의 사용을 허용합니다. 사용자 인증 변수를 사용합니다.

#### Limit

호스트 접근을 컨트롤하는 지정을 허용합니다.

Order allow,deny

Allow from all

Deny from env=no\_access

Limit 에 관련된 부분을 설정을 합니다.

### ③ Order

서버가 access control 을 수행하는 순서를 나타낸다. 여기서는 allow 기능을 먼저 수행하고, deny 기능을 수행하라는 것입니다.

deny, allow

deny 지시자 부터 검사하고 allow 지시자를 검사 합니다.

allow, deny

allow 지시자 부터 검사하고 deny 지시자를 검사 합니다.

mutual-failure

allow 목록에 없는 모든 host 에게 접속을 거부 합니다.

allow from

나열되는 주소들에 대한 access control 을 가능하게 합니다. 사용 가능한 주소는 도메인 네임, 호스트 이름 주소, 호스트 ip 주소, ip 주소의 앞부분 3 바이트, 모든 주소에 해당하는 all 이 있습니다.

deny from

allow from 과 반대되는 개념이며, 사용가능한 주소는 allow from 과 같다.

Require

사용자, 그룹에 대한 접근을 통제할 수 있습니다.

사용방법 : require entity en1 en2 ... enn

entity 에 들어갈 수 있는 것은 user , group, valid-user 의 세가지이다.

User

지정된 사용자들에게만 접근을 허용하는 것으로, 지정된 사용자에 대한 정보는 AuthUserFile 에서 지정한 파일에 있습니다.

Group

지정된 그룹에게만 접근을 허용하는 것으로, 지정된 그룹에 대한 정보는 AuthGroupFile 에서 지정한 파일에 있습니다.

valid-user

AuthUserFile 에 있는 모든 사용자에 대해 접근을 허용합니다.



```
</Directory>
```

```
UserDir public_html
```

계정 사용자들의 home directory 를 지정합니다. 보통 http://url/~계정 이런식으로 접속했을 때 계정에 이곳에 지정된 이름의 directory 를 만들고 home directroy 로 사용하면 됩니다. 보통 public\_html 이나 htdocs, home 을 많이 쓴다. 다른 이름으로 해도 상관은 없다. 계정 사용자가 사용을 못하게 하려면 DISABLED 옵션을 주면 됩니다.

참고를 할것은 RedHat 계열에서는 user 를 생성을 하면 user 의 home directory 는 700 의 권한을 갖게 되므로 httpd.conf 에서 아무리 설정을 해 줘도 forbidden error 만 만나게 됩니다. 그러므로 계정에서 homepage 를 운영하기 위해서는 꼭 chmod a+x ~accountname 명령을 실행을 해 줘야 합니다.

```
<Directory /*/public_html>
```

```
AllowOverride FileInfo AuthConfig Limit
```

```
Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
```

```
<Limit GET POST OPTIONS PROPFIND>
```

```
Order allow,deny
```

```
Allow from all
```

```
</Limit>
```

```
<Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
```

```
Order deny,allow
```

```
Deny from all
```

```
</Limit>
```

```
</Directory>
```

UserDir 디렉토리에 대한 접근을 제어합니다. 다음은 사용자 홈 페이지에 대하여 읽기만 가능하도록 한 예제 설정 내용입니다.

위의 설정 중 "\*" 사용에 관해서 언급을 할 것이 있습니다.

보통 "\*"는 all 의 의미로 많이 사용이 됩니다.

즉 위에서 /\*/public\_html 은 어떠한 경로에 있는 모든 public\_html 을 뜻한다고 할수 있습니다. 하지만 이는 그렇게 호락호락하게 되지는 않는 듯 합니다.

필자의 경험에 의하면 같은 레벨의 깊이에 있는 public\_html 만 적용이 되는듯 합니다.

즉 위의 설정대로라면 /home/public\_html 은 적용이 되지만 /home/nextline/publci\_html 은 적용이 되지 않더라란 말입니다.

```
DirectoryIndex index.html index.htm index.shtml index.php3 index.cgi
```

디렉토리만을 지정했을 경우에 그 디렉토리에서 찾게 될 문서의 순서를 지정해 줍니다. 즉, 디렉토리 이름만을 지정하더라도 여기서 지정한 index.html 을 찾아서 웹브라우저에 보여줍니다. 여러개의 파일을 지정할 수 있으며, 이런 경우에는 순서대로 찾아서 보여줍니다. 예를 들어 "DirectoryIndex index.html index.htm"로 지정했다면 먼저 "index.html"을 찾아서 있다면 이 파일을 로딩하고, "index.html"이 없다면 "index.htm"을 찾아서 로딩해 줍니다.

#### AccessFileName .htaccess

디렉토리별로 접근 제어할 정보(ID, Password)를 담고 있는 파일을 지정합니다. 디렉토리별로 인증을 거쳐서 접근할 수 있는 설정을 하기 위한 것입니다. 예를 든다면 어떤 홈페이지의 전부나 혹은 일부에서 접근하려고 할 때 ID, Password 를 묻는 창이 뜨면서 맞게 입력한 경우에만 접근 허용하는 것입니다. 보안상의 이유로 이 파일의 이름을 다른 이름으로 바꾸고 싶다면 ".htaccess" 대신에 다른 이름을 적어주면 됩니다.

```
<Files "^W.ht">
  Order allow,deny
  Deny from all
</Files>
```

바로 위에서 설정한 파일(".htaccess")의 내용을 볼 수 없게 할 때 사용하는 옵션이다. 보안상의 이유로 이 옵션은 설정해 두는 것이 좋다. 만약 이 옵션을 주석 처리해 둔다면 ".htaccess" 파일에 대한 보안은 누구도 장담할 수 없을 것입니다.

#### CacheNegotiatedDocs

이 설정은 Proxy Server 를 거치는 사용자들을 위한 것입니다. Proxy Server 가 "교섭" 문서, 즉 CGI script 의 출력이나 Server 가 생성한 index page 처럼 직접 수신되지 않는 문서를 cache 하도록 합니다. Default 로 주석 처리 되어 있고 그냥 이대로 두면 됩니다.

#### UseCanonicalName On

웹서버의 mime type 을 지정한 파일을 지정합니다. mime.types 파일은 서버에 의해 리턴될 수 있는 파일명과 mime 형식을 기술해 놓은 파일이다.

#### TypesConfig /etc/mime.types

웹서버의 mime type 을 지정한 파일을 지정합니다. mime.types 파일은 서버에 의해 리턴될 수 있는 파일명과 mime 형식을 기술해 놓은 파일이다.

#### DefaultType text/plain

mime.types 파일에 정의 되어있지 않은 파일형식에 대한 요청을 받았을 때 알 수 없는 문서타입에 대하여 사용할 기본적인 mime 타입을 정해준다.

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

이것 역시 1.3.12 에서 부터 새로 추가된 module 이다. 사용을 하기 위해서는 DS0 설정에서 mod\_mime\_magic line 의 주석을 해제해야 합니다.

HostnameLookups Off

웹서버의 로그(access\_log)를 지정하는 Format 에서 "DNS Lookup"으로 지정하였을 때, domain 으로 남길 것인가, IP Address 로 남길 것인가를 지정합니다. Default 로 Off 는 IP Address 로 남기는 것이며, Domain 으로 변경할 필요가 없으므로 on 으로 설정한 것보다는 속도가 조금빠르다.on 으로 하게 되면 IP address 를 IP Domain 으로 변환해야 하므로 속도가 조금 느릴 수 있습니다.

ErrorLog logs/error\_log

아파치 웹서버의 에러로그 기록파일을 지정합니다. 참고할 사항은 맨 마지막에 설정하는 <VirtualHost>부분에서 각서버에 대한 에러파일을 지정해 두지 않으면 그에 대한 에러로그도 여기에 기록되며, 지정해 두게 되면 그에 해당하는 로그는 이 파일에 기록되지 않는다. 만약 ErrorLog 를 남기고 싶지 않다면 /dev/null 로 지정을 해 주면 됩니다.

예: ErrorLog /dev/null

LogLevel warn

바로위에서 설정한 에러로그 파일에 얼마나 자세하게 적을 것인지를 결정합니다. 다음에 해당하는 순서대로 중요도가 정해진다.

" debug → info → notice → warn → error → crit → alert → emerg "

```
LogFormat "%h %l %u %t W"%rW" %>s %b W"%{Referer}iW"
```

```
W"%{User-Agent}iW" " combined
```

```
LogFormat "%h %l %u %t W"%rW" %>s %b" common
```

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

```
LogFormat "%{User-agent}i" agent
```

바로 아래에서 사용할 CustomLog 에서 사용할 몇가지 로그형식의 별명을 정한 곳이다.

웹서버의 관리자나 서버관리자는 이 부분을 특히 유심히 봐줘야 합니다. 웹서버의 로그를 어떤 식으로 남길 것인가를 결정하는 Format 을 지정하는 곳이다. 원하는 정보를 지정해서 볼 수 있으므로, 관리자에게 필요한 Format 으로 설정해야 하며, 또한 접속통계를 내기에 적당한 Format 으로 설정해 줘야 합니다.

```
CustomLog logs/access_log common
```

위에서 정한 로그형식(여기선 common)대로 로그를 남기게 됩니다.

맨마지막에서 지정하는 <VirtualHost>부분에서도 아파치 1.3.9 버전 부터는 CustomLog 를 가상호스트별로 지정할수 있도록 CustomLog 를 제공합니다. <VirtualHost>에서 CustomLog 를 지정하지 않으면 여기서 지정한 형식대로 로그를 남기게 되며 <VirtualHost>부분에서 CustomLog 를 지정했을 경우에는 여기서 지정한 로그형식은 무시됩니다.

```
CustomLog logs/referer_log referer
```

```
CustomLog logs/agent_log agent
```

위에서 지정한 4 가지의 로그형식(combine, common, referer, agent)중에서 원하는 부분의 (주석행)을 제거하면 지정됩니다.

```
CustomLog logs/access_log combined
```

이 설정은 위의 CustomLog logs/referer\_log referer, CustomLog logs/agent\_log agent 를 합쳐 놓은 것이라 생각하면 됩니다. 이 행의 주석을 풀어 주기 위해서는 위의 두행이 주석처리가 되어 있어야 합니다.

```
ServerSignature On
```

서버가 생성하는 문서(error documents, FTP directory listings, mod\_status and mod\_info output etc., but not CGI generated documents)의 trailing footer line 의 설정을 가능하게 합니다.

```
Alias /icons/ "/usr/local/apache/icons/"
```

```
<Directory "/usr/local/apache/icons">
```

```
Options Indexes MultiViews
```

```
AllowOverride None
```

```
Order allow,deny
```

```
Allow from all
```

```
</Directory>
```

필요한 만큼의 디렉토리 별칭을 만들어 쓸 수 있습니다. 사용하는 형식은 다음과 같다.

```
Alias fakename(가상이름) realname(진짜이름)
```

```
ScriptAlias /cgi-bin/ "/usr/local/apache/cgi-bin/"
```

```
<Directory "/usr/local/apache/cgi-bin">
```

```
    AllowOverride None
```

```
    Options ExecCGI
```

```
    Order allow,deny
```

```
    Allow from all
```

```
</Directory>
```

`ScriptAlias` 는 서버스크립트를 포함합니다. `ScriptAlias` 는 실제디렉토리 안에 들어있는 문서를 서버에 의해 응용프로그램으로 취급되어 실행되는 것을 제외하고는 근본적으로 `Aliases` 와 같다.

```
IndexOptions FancyIndexing
```

`IndexOptions`는 디렉토리목록을 표시할 때 사용할 옵션을 지정합니다. `Standard`는 표준적인 디렉토리를 나타내며, `FancyIndexing`은 좀더 예쁜 디렉토리목록을 표시해 줍니다.

아래에서 지정하는 `AddIcon`으로 시작하는 설정은 바로위에서 설정한 디렉토리인덱싱 옵션을 `FancyIndexing`으로 한 경우에 해당하며 디렉토리 목록을 표시할 때 각 파일 확장자에 따라서 어떤 아이콘을 선택하여 보여줄 것인지를 지정합니다.

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

```
AddIconByType (TXT,/icons/text.gif) text/*
```

```
AddIconByType (IMG,/icons/image2.gif) image/*
```

```
AddIconByType (SND,/icons/sound2.gif) audio/*
```

```
AddIconByType (VID,/icons/movie.gif) video/*
```

`IndexOptions` 에서 `FancyIndexing` 이 지정되었을 때 이 지시자는 file 마다 MIME type 에 따라 어떤 icon 을 사용 할지 정합니다.

```
AddIcon /icons/binary.gif .bin .exe
```

```
AddIcon /icons/binhex.gif .hqx
```

```
AddIcon /icons/tar.gif .tar
```

```
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
```

```
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
```

```
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
```

```
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
```

Server가 file과 Directory를 표시하는데 어떤 icon을 사용할 지 지정합니다.

```
DefaultIcon /icons/unknown.gif
```

여기서 지정한 확장자가 아닌 경우에 여기서 지정한 기본아이콘으로 보여줍니다.

```
AddDescription "GZIP compressed document" .gz
AddDescription "tar archive" .tar
AddDescription "GZIP compressed tar archive" .tgz
```

AddDescription은 서버가 생성한 인덱스의 파일 뒤에 간단한 설명을 표시할 때 사용됩니다. 이 설정은 IndexOptions가 FancyIndexing으로 설정되었을때만 표시되며, 설정형식은 다음과 같습니다.

형식 : AddDescription "표시할 설명" 파일확장자

```
AddDefaultCharset Off
AddDefaultCharset euc-kr
```

이 설정은 apache 1.3.12 에 들어와서 Cross Site Scripting 보안 문제 패치를 하면서 기본 글꼴을 iso-8859-1 로 하는 현상이 발생 하므로 한글 사용을 위해서는 어쩌면 필요한 설정일지도 모른다. 일단 문제가 없으면 그냥 놔두는 것도 괜찮을지 모르겠지만 문제가 된다면 설정을 해보는 것도 괜찮을것 같다. 일단 필자의 테스트 로는 euc-kr 로의

설정은 문제가 발생합니다.

```
ReadmeName README
```

```
HeaderName HEADER
```

디렉토리 목록을 보여줄 때 목록의 마지막 부분 뒤와 목록의 시작 전에 보여줄 내용을 담고 있는 파일을 지정합니다. 여기서는 목록의 뒤에 README, 목록의 처음에 HEADER 를 보여주게 지정되어 있습니다.

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

디렉토리를 출력할 때 무시할 파일들을 지정합니다.

```
AddEncoding x-compress Z
```

```
AddEncoding x-gzip gz
```

압축 코드에 대한 인코딩 정보를 지정합니다.

```
<IfModule mod_negotiation.c>
```

```
  AddLanguage ko .ko
```

```
  AddLanguage en .en
```

```
  AddLanguage fr .fr
```

```
  AddLanguage de .de
```

```
  AddLanguage da .da
```

```
  AddLanguage el .el
```

```
  AddLanguage it .it
```

```
</IfModule>
```

Server 가 다국 언어로 문서를 제공한다면 이 지시자를 사용해 file 확장자를 언어를 지정 하는 약어에 대응시킨다. 언어의 약어는 보통 인터넷 국가 코드를 사용합니다. client 가 home.html 이라는 file 을 요청하면 browser 는 프랑스어 사용자라는 것을 전하고 server 는 이 지시자를 찾아서 어떤 file 확장자가 프랑스어 문서에 사용되는지 알아본다. 프랑스어 사용자라면 home.html.fr 을 받게 됩니다. 다만 home.html 이라는 file 이 존재 합니다면 home.html.fr 을 받지 못하고 home.html 을 받아 버리게 됩니다.

```
<IfModule mod_negotiation.c>
```

```
  LanguagePriority ko en fr de
```

```
</IfModule>
```

site 에 위에서 지정한 것과 같은 home.html.fr 과 같은 문서가 있고 client 가 언어를 선택하지 않고 home.html 을 요청할때(home.html 이 없을 경우) 서버가 보낼 문서를

지정합니다. 이 지시자는 여러 언어를 내림 차순으로 나열합니다.

```
AddHandler cgi-script .cgi .pl .sh
```

서버의 어떤 위치에 있던지 '.cgi' 확장자를 가진 파일은 cgi-script 로 인식하게 합니다. '.pl', '.sh' 등의 다른 확장자도 추가할 수 있습니다. 보안을 이유로 account user 들에게 CGI 권한을 주지 않으려면 이행을 주석 처리해야 합니다. 반대로 account user 들에게 CGI 권한을 주기 위해서는 이 행의 주석을 풀어 줘야 합니다.

```
AddType text/html .shtml
```

```
AddHandler server-parsed .shtml
```

```
AddHandler text/x-server-parsed-html .html .txt
```

Server Side Includes (SSI)를 사용할 때 필요합니다. SSI 는 HTML 파일 속에 어떤 실행 프로그램의 결과나 특정 파일을 포함할 수 있게 합니다.

shtml 확장자가 아닌 파일에서도 SSI 를 사용할 수 있게 하려면 위에서와 같이 x-server-parsed-html mime type 설정을 이용합니다.

```
MetaDir .web
```

MetaDir은 아파치가 찾을 메타정보파일들의 디렉토리 이름을 지정합니다. 이 파일들은 문서를 전송할 때 포함되는 HTTP 헤더정보가 포함되어 있습니다.

```
MetaSuffix .meta
```

MetaSuffix 는 메타정보를 포함하고 있는 접미어의 이름을 지정합니다.

```
ErrorDocument 400 /message/400error.html
```

```
ErrorDocument 401 /message/401error.html
```

```
ErrorDocument 403 /message/403error.html
```

```
ErrorDocument 404 /message/404error.html
```

```
ErrorDocument 405 /message/405error.html
```

```
ErrorDocument 500 /message/500error.html
```

```
ErrorDocument 501 /message/501error.html
```

또는 스크립트를 이용하여 변수를 전달 시킬 수 도 있습니다.

```
ErrorDocument 400 /message/error.php?ecode=400
```

```
ErrorDocument 401 /message/error.php?ecode=401
```



```
ErrorDocument 403 /message/error.php?ecode=403
ErrorDocument 404 /message/error.php?ecode=404
ErrorDocument 405 /message/error.php?ecode=405
ErrorDocument 500 /message/error.php?ecode=500
ErrorDocument 501 /message/error.php?ecode=501
```

서버에러에 대한 응답을 지정해 주는 부분이다. 각 에러 코드에 대한 응답을 cgi 나 일반 텍스트로 만들어서 사용자에게 보여줄 수 있습니다. 어떤 서버에 접속하면 해당 URL 이 없다는 등의 한글 메시지가 가능한 것도 이것을 이용하는 것입니다.

아래는 (2)번의 내부 redirects 를 사용한 예이다. 나의 경험 상으로 외부 방향 전환을 이용했을때 CGI 와 htaccess 인증시에 505 Internal Server error 가 발생 했다. 왜 그런지 이유는 잘 모르겠다. 내부 방향 전환에는 전혀 이상이 없었다.

```
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4W.0b2;" nokeepalive downgrade-1.0 W
force-response-1.0
```

다음 지시자는 보통의 HTTP 반응 방식을 수정합니다.

첫번째 것은 네스케이프 2.x 또는 그를 흉내내는 브라우저에 대하여 KeepAlive 기능을 쓰지 않도록 합니다. 이 브라우저들은 KeepAlive 구현에 문제점을 갖고 있기 때문이다.

두번째 것은 HTTP/1.1 을 잘못 구현하였고 301 또는 302 (redirect) 반응에 대하여 KeepAlive 를 제대로 지원하지 못하는 Microsoft 인터넷 익스플로러 4.0b2 를 위한 것입니다.

```
BrowserMatch "RealPlayer 4W.0" force-response-1.0
BrowserMatch "Java/1W.0" force-response-1.0
BrowserMatch "JDK/1W.0" force-response-1.0
```

다음은 기본적인 1.1 반응도 제대로 처리하지 못함으로써 HTTP/1.1 스펙을 위반하고 있는 브라우저에 대하여 HTTP/1.1 반응을 하지 않도록 합니다.

```
BrowserMatch "WebZIP" no_access
BrowserMatch "Teleport" no_access
BrowserMatch "NamoWebEditor" no_access
BrowserMatch "WebTrack-HTTP" no_access
BrowserMatch "WebSymmetrix" no_access
```

BrowserMatch 지시자를 이용하여 User Agent 별로 접근을 제어 하도록 합니다.

```
<IfModule mod_status.c>
```

```

<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from localhost
    Allow from all
    AuthName "administrator Area"
    AuthType Basic
    AuthUserFile /home/.htpasswd
    AuthGroupFile /dev/null
    require valid-user
    satisfy all
</Location>
</IfModule>

```

서버의 상태결과를 `http://servername/server-status` 의 URL 에 접근 하면 볼 수 있게 해주는 옵션이다. 'allow from 서버 도메인네임'의 형식 으로 접근이 가능합니다.

하단의 주석 처리가 되어진 부분은 .htpasswd file 에 기록되어 있는 User 와 Password 로 인증을 하여 보게 하는 설정의 예이다. .htpasswd 인증을 위해서는 mod\_auth.c 가 활성화되어 있어야 합니다. 그럼 여기서 살짝 공수를 써보도록 하겠다. 일정 ip address 에서는 ip address check 만하고 인증을 안하도록 하고 다른 ip address 에서 는 인증을 하게 하는 방법을 설정해 보도록 하겠다.

```

<IfModule mod_status.c>
    <Location /server-status>
        SetHandler server-status
        Order deny,allow
        Deny from all
        Allow from localhost
        AuthName "administrator Area"
        AuthType Basic
        AuthUserFile /home/.htpasswd
        AuthGroupFile /dev/null
        require valid-user
        satisfy any
    </Location>

```

```
</IfModule>
```

원래의 설정과 위의 설정과의 차이를 보도록 하면, 일단 원래의 설정에서 주석 처리 되어 있는 부분이 모두 주석이 제거가 되어 있습니다. 그리고 제일 마지막 option 인 satisfy 의 값이 all 에서 any 가 되어 있습니다. 이것이 바로 키포인트이다. 즉 Allow from 에 지정된 ip address 나 domain name 은 인증 을 안하고 바로 보여주며, 그 외의 주소들은 /home/.htpasswd 에 있는 유저의 이름과 패스워드를 비교하여 인증을 해서 출력을 하게 됩니다. 이것은 아래의 server-info 에도 적용이 가능합니다.

```
<IfModule mod_info.c>
```

```
<Location /server-info>
```

```
    SetHandler server-info
```

```
    Order deny,allow
```

```
    Deny from all
```

```
    Allow from localhost
```

```
    Allow from all
```

```
    AuthName "administrator Area"
```

```
    AuthType Basic
```

```
    AuthUserFile /home/.htpasswd
```

```
    AuthGroupFile /dev/null
```

```
    require valid-user
```

```
    satisfy all
```

```
</Location>
```

```
</IfModule>
```

1.3.4 에서 부터 새로 추가된 module 이다. 이 기능은 http://servername/server-info 의 url 로 접근을 했을 경우 apache 에서 실행이 가능한 module 들의 목록 등 apache 의 전반적인 정보를 보여줍니다.

```
<Location /cgi-bin/phf*>
```

```
    Deny from all
```

```
    ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi
```

```
</Location>
```

1.1 버전 이전의 오래 된 버그를 악용하려는 사람들이 있습니다는 보고를 받았다. 이 버그는 아파치 일부분으로 제공한 CGI 스크립트와 연관 있습니다. 이 부분의 주석 처리를 없애면 이 버그를 악용하는 공격이 있을 때 phf.apache.org 상의 기록 스크립트로 방향 전환시킬 수 있습니다. 또는 support/phf\_abuse\_log.cgi 스크립트를 사용하여 여러분 직접 기록할 수도 있습니다.

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

```
ProxyRequests On
```

아파치 웹서버를 Proxy 서버로 사용할 때 on 을 해줘야 합니다.

즉 프락시 서버 지시자로서 프락시 서버를 on 시킵니다.

```
<Directory proxy:*>
```

```
Order deny,allow
```

```
Deny from all
```

```
Allow from .your_domain.com
```

```
</Directory>
```

```
ProxyVia On
```

HTTP/1.1 "Via:" 헤더를 처리할 것인지 여부를 결정합니다.

("Full"은 서버 버전을 포함하고 "Block"은 나가는 모든 자료에서 Via: 헤더를 제거합니다.) Off | On | Full | Block 중 하나의 값을 지정합니다.

```
CacheRoot "/usr/local/apache/proxy"
```

```
CacheSize 5
```

```
CacheGcInterval 4
```

```
CacheMaxExpire 24
```

```
CacheLastModifiedFactor 0.1
```

```
CacheDefaultExpire 1
```

```
NoCache a_domain.com another_domain.edu joes.garage_sale.com
```

캐쉬 기능도 사용하기 위해서는 다음 행의 주석을 풀어줍니다

(CacheRoot 가 없으면 캐쉬하지 않음)

```
</IfModule>
```

## 가상 호스트

아파치 가상호스트 설정부분입니다. 하나의 아파치 웹서버에서 여러 개의 도메인 또는 멀티도메인, 또는 서브도메인등으로 여러 개의 웹사이트를 운영하고자 하신다면 가상호스트 설정지시자를 이용하시면 됩니다. 대부분의 경우 여러 개의 IP주소를 사용할 필요없이 하나의 IP주소를 사용하여 여러 개의 웹사이트를 운영할 수 있기 때문에 이름기반의 가상호스트(name-based virtual host)를 사용합니다.

VirtualHost: 여러분의 리눅스 박스에 여러 개의 도메인/호스트 이름을 관리하고 싶다면

각각에 대하여 VirtualHost 컨테이너를 설정 합니다. 가상 호스트를 설정하기에 앞서 자세한 설명을 <URL:http://www.apache.org/docs/vhosts/>에 들러 읽어보기 바란다.

가상 호스트 설정 내용을 점검해보기 위해서는 아파치를 실행할 때 명령행 옵션으로 '-S'를 사용합니다.

```
NameVirtualHost 12.34.56.78:80
```

```
NameVirtualHost 12.34.56.78
```

이름 기반의 가상 호스트를 사용하려면 사용할 IP 주소 (최소 1 개, 그리고 포트 번호)를 정의해주어야 합니다.

```
<VirtualHost xxx.xxx.xxx.xxx>
```

```
    ServerAdmin ask@nextline.co.kr
```

```
    DocumentRoot /home/nextline/public_html
```

```
    ServerName www.nextline.co.kr
```

```
    ErrorLog /home/nextline/public_html/logs/error_log
```

```
    CustomLog /home/nextline/public_html/logs/access_log common
```

```
</VirtualHost>
```

ServerAdmin은 해당서버의 관리자 전자우편이며,

DocumentRoot는 해당서버의 홈디렉토리이며,

ServerName은 해당서버의 도메인이며,

ErrorLog는 해당서버의 에러파일 위치이며

CustomLog는 로그파일위치와 포맷을 지정한 것입니다.

```
<VirtualHost _default_*>
```

```
</VirtualHost>
```

Default 가상호스트 설정으로 위에서 설정되지 않은 다른 모든 호스트에 대해서 응답을 하고자 할 경우 설정해 줍니다.