



---

第 23 章  
APACHE2.0

---

無  
限  
參  
考

Linux

## 第 23 章 APACHE2.0

### 23-1 APACHE2.0

RedHat Linux Fadora 作業系統現在都是使用 APACHE2.0，因此我們也介紹 APACHE2.0 來架設網站。我們的網站 flash.aasir.com 就是使用 APACHE2.0 來架設。

啟動 apache

```
[root@flash chaiyen]# /etc/rc.d/init.d/httpd start
```

關閉 APACHE。

```
[root@flash chaiyen]# /etc/rc.d/init.d/httpd stop
```

停止 httpd:[ 確定 ]

重新啟動 APACHE。

```
[root@flash chaiyen]# /etc/rc.d/init.d/httpd restart
```

停止 httpd:[ 確定 ]

啟動 httpd:[ 確定 ]

APACHE2.0 伺服器的檔案是放在/etc/httpd 的目錄下。底下的 conf 目錄，主要是放置 APACHE2.0 主要的組態設定檔 httpd.conf、Makefile 和安全協定的檔案；底下的 conf.d 組態目錄是放置 APACHE2.0 要載入的組態檔，包含了 php.conf、perl.conf、python.conf 和安全協定組態檔案 ssl.conf。

#### 23-1-1 設定

/etc/httpd/conf/httpd.conf 主要組態設定檔

使用 vi 編輯器編輯 apache

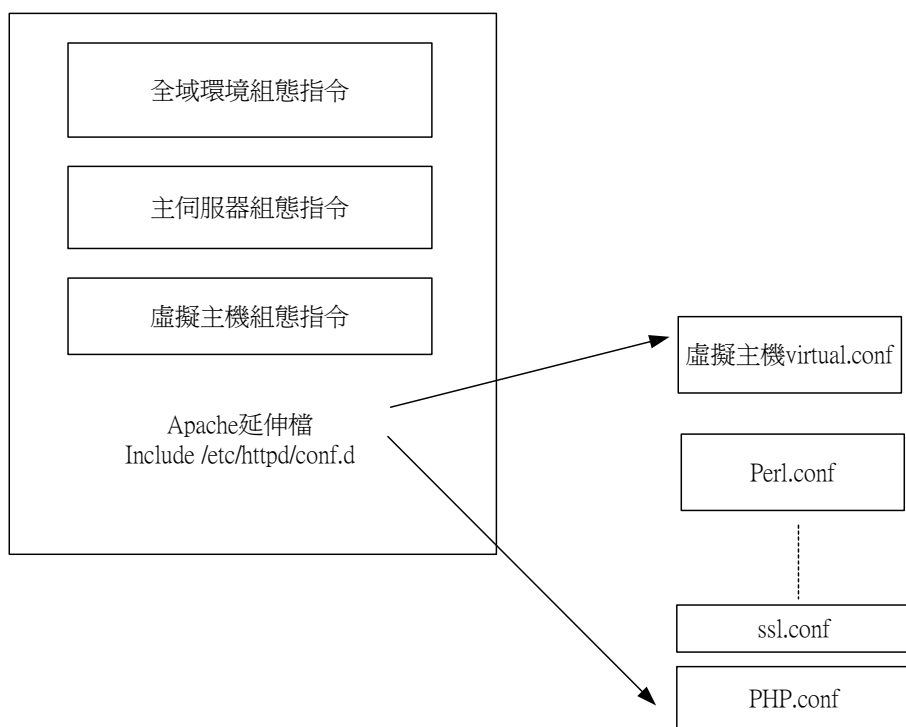
```
[root@aasir chaiyen]# vi /etc/httpd/conf/httpd.conf
```

這是 apache 的設定，所有的設定都是 flash.aasir.com 網站的設定。



## 23-1-2 APACHE 網站伺服器概念

我們 APACHE 的組態設定檔 `httpd.conf` 是由全域環境組態指令、主伺服器組態指令和包含檔 `Include` 組態指令所組成。我們也可以將一些組態檔使用包含的方式將組態設定檔包含進來。在圖中我們將其它虛擬主機的檔案設定在外部，再使用 `Include` 將該組態設定檔包含進來。我們也將 `Perl.conf`、`PHP.conf`、`Python.conf`、`ssl.conf`、`auth_mysql.conf` 和 `auth_pgsq.conf` 的組態檔包含進入我們的 `httpd.conf` 的組態檔。



/etc/httpd 就是 APACHE 網站伺服器所有檔案的父目錄

```
[root@flash httpd]# tree
.
|-- build -> ../../usr/lib/httpd/build
|-- conf
|   |-- Makefile -> ../../usr/share/ssl/certs/Makefile
|   |-- httpd-mailman.conf
|   |-- httpd.conf
|   |-- magic
|   |-- ssl.crl
|   |   |-- Makefile.crl
|   |-- ssl.crt
|   |   |-- Makefile.crt
|   |   |-- server.crt
|   |-- ssl.csr
|   |-- ssl.key
|   |   |-- server.key
|   |-- ssl.prm
|-- conf.d
|   |-- README
|   |-- auth_mysql.conf
|   |-- auth_psql.conf
|   |-- perl.conf
|   |-- php.conf
|   |-- python.conf
|   |-- squirrelmail.conf
|   |-- ssl.conf
|-- logs -> ../../var/log/httpd
|-- modules -> ../../usr/lib/httpd/modules
`-- run -> ../../var/run
```



## 23-2 設定網站伺服器的全域環境組態 Global Environment

我們使用 RedHat Linux Fedora 作業系統直接安裝時，則 apache 套件已經安裝在我們的電腦上了，需要修改的地方有幾點。

<<1>>Global Environment

第三十二行開始是設定網站伺服器 Apache 全域環境組態 Global Environment。

```
32 ### Section 1: Global Environment
```

<<2>>ServerTokens 指令

第四十三行的 ServerTokens 指令讓我們防止遠端的使用者觀看我們 Apache 伺服器正在執行的元件。如果我們設定註解#，則遠端的使用者都可以看到我們伺服器的資訊。ServerTokens 指令有四個值，分別是 Full、OS、Minimal 和 ProductOnly，預設是 OS。ServerTokens 指令控制伺服器是否回應送到使用者端包含伺服器資訊及作業系統形態的標頭檔欄位。

```
43 ServerTokens OS
```

<<3>>ServerRoot 指令

ServerRoot 指令設定安裝 APACHE 伺服器檔案位置的地方，預設是在"/etc/httpd"。

```
56 ServerRoot "/etc/httpd"
```

ServerRoot 指令設定伺服器放置的地方。一般而言，它會包含 conf logs conf.d modules 和 build 子目錄。

Conf 子目錄是放置 httpd.conf 伺服器組態檔的地方，logs 子目錄是放置伺服器記錄情況的地方、conf.d 子目錄是放置相關組態檔被 Include 指令包含到 httpd.conf 組態檔的地方、modules 子目錄是放置我們用 LoadModule 指令載入模組的地方。



## &lt;&lt;4&gt;&gt;ScoreBoardFile 指令

ScoreBoardFile 是設定在某些平台 Apache 父行程和子行程溝通的檔案位置。。ScoreBoard 是記錄我們內部伺服器行程的資訊，在 RedHat Linux Fadora 1 作業系統，他將這些狀態資訊儲存在記憶體中，這樣就可以提高存取效能。

```
66 #ScoreBoardFile run/httpd.scoreboard
```

## &lt;&lt;5&gt;&gt;PidFile 指令

PidFile 指令是設定行程 ID(Process ID)的檔案路徑，預設是在 run/httpd.pid。Apache 使用 PidFile 來儲存超級使用者行程的行程 ID。這個行程資訊就是我們 Apache 執行的常駐行程，當我們要啟動 Apache 伺服器時，就會有行程編號，這個行程編號就放在/etc/httpd/run/httpd.pid。當我們要關掉 Apache 伺服器時 /etc/init.d/httpd 就會使用到這個資訊。

```
72 PidFile run/httpd.pid
```

## &lt;&lt;6&gt;&gt;Timeout 指令、KeepAlive 指令和 KeepAliveTimeout 指令

從第七十五行到第九十行都是和連線時間 Timeout 有關，當使用者提出連線要求，伺服器會等待 Timeout 指令設定的秒數，若超過 Timeout 所設定的傳送或接收的時間，這個連線就會中斷，預設 Timeout 是 300 秒。

```
77 Timeout 300
```

Timeout 指令定義 Apache 會為了 Get 要求服務 Post 或 PUT 要求服務和 ACKs 的服務三種設定等待時間。

KeepAlive 指令設定是否允許持續連接 APACHE 伺服器，預設是關閉 OFF，如果要啟動則設為 On。當提出要求已經被處理了，KeepAlive 指令會指示 Apache 伺服器持續打開連接一段時間。

```
83 KeepAlive Off
```

KeepAlive 指令可以讓在同一個 TCP 連線的狀態來提供多種要求服務。當網頁文件有很多圖片時，這個功能可以讓我們提高將近 50%的加速效能。因此如果我們要讓



網站加速效能提高，可以將 KeepAlive 指令設為 On。

對於 HTTP/1.0 的使用者，Keep-Alive 持續連線不能用在動態內容 CGI、SSI 網頁和伺服器產生的目錄索引。對於 HTTP/1.1 的使用者持續連線是預設的。

MaxKeepAliveRequests 指令允許持續連接的數量，0 表示無限制。數字越大效能越高。MaxKeepAliveRequests 指令限制每次連線要求服務的上限。當 KeepAlive 為 On 時，MaxKeepAliveRequests 的預設值為 100 是較為合理的。為了最大的伺服器效能，我們建議將 MaxKeepAliveRequests 設定越高越好，可以設為 500。

```
90 MaxKeepAliveRequests 100
```

KeepAliveTimeout 指令會設定在持續連接中，伺服器會等待隨後要求服務的時間。預設是 15 秒。在持續的連接中，Apache 等待後來要求服務的時間，KeepAliveTimeout 指令指示這個時間秒數。一旦要求服務被接收，timeout 的數值將被 Timeout 指令使用。

```
96 KeepAliveTimeout 15
```

<<7>MPMs 的組態(多工模組)

第九十九行到第一百四十六行是設定伺服器行程的數量，為了讓 Apache 能夠滿足伺服器的需求服務，Apache 需能動態的改變伺服器正在執行的行程數量。下面是幾個我們必需指定的參數。

StartServers 指令是設定伺服器行程啟動的數量。

MinSpareServers 指令設定閒置行程的最小數量。

MaxSpareServers 指令設定閒置行程的最大數量。

Maxclients 指令設定同時可允許啟動的行程數量。

MaxRequestsPerChild 指令設定每一個子行程可以服務的請求數量。

啟動伺服器行程的數量。APACHE 的行程先啟動好，只要有使用者要求，我們就可以服務它。在這裏我們是預設 8 個 APACHE 的行程。



MPM 為多行程模組(Multiprocessing modules)。APACHE2.0 支援多執行緒，每個子行程還可產生多個特定的執行緒。而每個子行程的執行緒可以服務不同的請求。如果每個 Apache 可以產生 8 個子行程，而每個子行程又可以產生 25 個執行緒，則 Apache 可以產生  $8 \times 25 = 1000$  個執行緒的服務。當行程是要增加或減少是由閒置執行緒來決定，而其行程是介於 MinSpareServers(最小閒置行程和 MaxSpareServers(最大閒置行程之間)。

行程(Process)和執行緒(Thread)最大的不同在於，執行緒是屬於小行程。因為多個執行緒可以共用程式碼、檔案、和資料，因此執行緒的檔案比行程小很多。

這是行程的行程控制表。

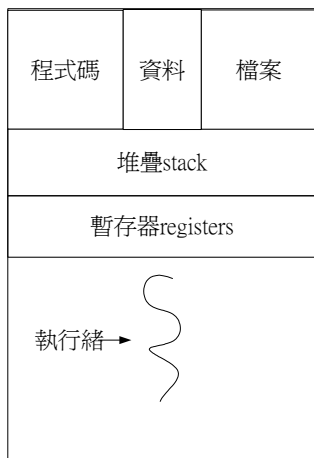
指標pointer	行程狀態 process state
行程號碼process number	
程式計數器program counter	
暫存器registers	
記憶體限制memory limits	
已開啓檔案表list of open files	
.	
.	
.	
.	

行程控制表PCB



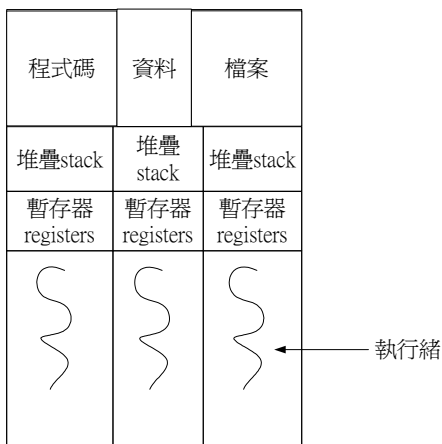


這是 Apache1.3 之前的單一執行緒行程，每一個行程都是單一的執行緒。



單一執行緒的行程

這是在 Apache2.0 的多個執行緒行程(MPM)。在 Apache2.0 中每一個行程可以產生多個執行緒，每個執行緒都有它自己的暫存器和堆疊，但是它們會共用程式碼、資料和檔案。這樣就可以用少量的資源來提供更多的服務。



多個執行緒的行程



Apache 在一開始會啟動一個主要的常駐行程(擁有者為超級使用者), 然後再轉換給一般沒有權限的使用者。Apache 會根據目前的負載來新增或移除伺服器的行程。假如流量減少或者是暫停的行程伺服器很多, Apache 將自動的 Kill(移除)行程。假如很多的要求服務, 或者很少的伺服器行程在線上等待, 則 Apache 會自動新增行程並啟動。

使用預設的 MPM(多工模組)環境, 當常駐 Apache 行程被啟動時, 有八個行程將開始 Run(跑起來), 並且等待著連接。當有更多的需求服務來到, Apache 會確定至少有八個行程已經準備好(ready)來提供服務。當需求服務已經滿了而新的連線又來到, Apache 將開始 Kill(移除)行程, 直到這間置暫停服務的網路伺服器行程低於 20。伺服器行程啟動的數量(StartServers)應該介於 MinSpareServers(指定閒置行程的最小數量)和 MaxSpareServers(指定閒置行程的最大數量)之間。

Apache 使用 MaxClients 指令(同時可允許啟動的行程數量)來限制同時存在行程的數量。這預設值是 150, 這已經是很高了。但是如果有超過 150 位的同時連線, 我們則應該提高這個上限。

為了減輕記憶體負擔, 而且讓這個記憶體緩衝池能夠更新, 則要限制每一個伺服器行程所能提供的要求服務。因此要設定 MaxRequestsPerChild(每一個子行程可以服務的請求數量)。當提供 1000 次要求後, 這個行程就會被 Kill(移除), 也就是每個行程只能服務 1000 個連線(因為 KeepAlive 指令是關掉的, 所以每個要求服務都是一個連線)。

在多工模組的環境(MPM), 我們可以藉由限制執行緒的數量來供幾或消費。我們可以在 MinSpareThreads 指定閒置執行緒的最小數量, 在 MaxSpareThreads 指定閒置執行緒的最大數量, 來控制所有執行緒的數量。當執行緒的數量少於 MinSpareThreads, 則多工模組會增加執行緒的數量。當執行緒的數量多於 MaxSpareThreads 時, 則多工模組會減少執行緒的數量。

IfModule 是表示載入模組, module\_name 為我們載入模組的名稱。

```
<IfModule module_name>
```

```
.....
```

```
</IfModule>
```



這是指定使用多執行緒行程 MPM。

```
98 ##  
99 ## Server-Pool Size Regulation (MPM specific)  
100 ##
```

IfModule prefork.c 是載入 prefork.c 的模組。

StartServers 是伺服器行程啟始的數量，預設為 8。

MinSpareServers 指定閒置行程的最小數量，預設為 5。

MaxSpareServers 指定閒置行程的最大數量，預設為 20。

Maxclients 是設定同時可允許最多的行程，預設為 150。

MaxClients\*ThreadsPerChild 就是 Apache 可以同時請求處理的總數量。

MaxRequestsPerChild 為每一個子行程可以服務的請求數量，預設為 1000。

```
108 <IfModule prefork.c>  
109 StartServers      8  
110 MinSpareServers  5  
111 MaxSpareServers  20  
112 MaxClients       150  
113 MaxRequestsPerChild 1000  
114 </IfModule>
```

IfModule work.c 是載入 work.c 的模組。

StartServers 是伺服器行程啟始的數量，預設為 2。

MinSpareServers 指定閒置執行緒的最小數量，預設為 25。

MaxSpareServers 指定閒置執行緒的最大數量，預設為 75。

Maxclients 是設定使用者端同步連接最大的數量，預設為 150。

MaxClients\*ThreadsPerChild 就是 Apache 可以同時請求處理的總數量。

ThreadsPerChild 為每個行程可以有多少常駐的執行緒，預設為 25。

MaxRequestsPerChild 為每一個行程可以服務的請求數量，預設為 0。



```
123 <IfModule worker.c>
124 StartServers          2
125 MaxClients           150
126 MinSpareThreads      25
127 MaxSpareThreads      75
128 ThreadsPerChild     25
129 MaxRequestsPerChild  0
130 </IfModule>
```

IfModule perchild.c 是載入 perchild.c 的模組。

NumServers 常駐的伺服器行程，預設是 5。

StartThreads 每一個行程一開始就執行的執行緒，預設為 5。

MinSpareThreads 指定閒置執行緒的最小數量，預設為 5。

MaxSpareThreads 指定閒置執行緒的最大數量，預設為 10。

MaxThreadsPerChild 每個行程最大的常駐執行緒數量，預設為 20。

ThreadsPerChild 為每個行程可以有多少常駐的執行緒，預設為 25。

MaxRequestsPerChild 為每一個伺服器行程可以連接的數量，預設為 0。

```
139 <IfModule perchild.c>
140 NumServers          5
141 StartThreads        5
142 MinSpareThreads     5
143 MaxSpareThreads     10
144 MaxThreadsPerChild 20
145 MaxRequestsPerChild 0
146 </IfModule>
```



## &lt;&lt;8&gt;&gt;Listen 連接埠指令

APACHE 伺服器指定的連接埠預設是 80。我們也可以指定 Apache 要連接的 IP 位址，如 61.218.29.2:80。Listen 61.218.29.2:80，就是設定 Apache 的位址為 61.218.29.2 的第 80 連接埠。Listen 指令比 BindAddress 和連接埠 port 指令更好用。多個 Listen 指令也可以讓我們指定多個 IP 位址。在/etc/service 中設定 Apache 的連接埠為 80。Service 服務連接埠可在低於 65536 的任何連街埠。只有超級使用者才有權限執行低於 1024 連接埠的程式。

157 Listen 80

設是在/etc/service 中設定 Apache 網站伺服器的連接埠為 80。

```

86 http          80/tcp          www www-http    # WorldWideWeb HTTP
87 http          80/udp          www www-http    # HyperText Transfer Pro
__ tocol

```

這是連接埠為 80 的網站。網址為 http://aasir.com:80。



## &lt;&lt;9&gt;&gt;Include 包含指令

我們可以使用 Include 指令從外部或/etc/httpd/conf.d 的組態目錄載入組態檔。/etc/httpd/conf.d 的目錄中包函了許多和組態相關的指定模組。Include 指令可以讓我們從外部載入檔案。假如 Include 指令指向目錄，Apache 將從該目錄讀取所有的檔案再分析語法。這檔案路徑是相對於 ServerRoot 指令所指定的路徑，例如 ServerRoot 指令指定根目錄為/etc/httpd，所以 Include 指令包含 conf.d 的路徑就是



/etc/httpd/conf.d 目錄下的所有組態檔。

## 162 Include conf.d/\*.conf

這是我們虛擬主機的設定檔 virtual.conf，我們將這個 virtual.conf 的檔案放到 /etc/httpd/conf.d 的目錄下。我們在 httpd.conf 組態檔的 162 行使用 Include 指令將 /etc/httpd/conf.d 所有的檔案包含進來我們 httpd.conf 的組態檔。

```
1NameVirtualHost 61.218.29.3
2<VirtualHost 61.218.29.3>
3    ServerAdmin wu.chaiyen@msa.hinet.net
4    DocumentRoot /var/www/virtual2
5    ServerName virtual2.aasir.com
6    ErrorLog logs/virtual2.com-error_log
7    CustomLog logs/virtual2.com-access_log common
8</VirtualHost>
9
10<VirtualHost 61.218.29.3>
11    ServerAdmin wu.chaiyen@msa.hinet.net
12    DocumentRoot /var/www/virtual3
13    ServerName virtual3.aasir.com
14    ErrorLog logs/virtual3.com-error_log
15    CustomLog logs/virtual3.com-access_log common
16</VirtualHost>
```

這些檔案會被 Include 指令包含進來 Apache 伺服器的組態檔。

```
[root@flash conf.d]# ls
auth_mysql.conf  perl.conf  python.conf  squirrelmail.conf  virtual.conf
auth_pgsqf.conf  php.conf  README      ssl.conf
```

我們也可以選擇性的將需要的檔案包含進來使用 include 指令。

Include conf/ssl.conf，這樣就可以將 ssl.conf 加密傳輸協定組態檔給包含。

Include conf/php.conf，這樣就可以將 php.conf 動態網頁 PHP 組態檔給包含進來。

<<10>>LoadModule 載入模組指令

從 176 行到 211 行為載入動態分享物件。\*.so 為我們的動態分享物件。

在我們編譯 Apache 的過程中，個別的 Apache 模組可以被選取來作動態連結。



只有我們在指定載入(LoadModule 指令)動態模組時, Apache 網站伺服器才會將它載入到我們的記憶體中。在第 176 行到 211 行為我們載入的指令(LoadModule)和所載入模組的路徑。如果我們不需要載入相關模組, 只要在該行的前面加上註解#就可以了。

```
176 LoadModule access_module modules/mod_access.so
177 LoadModule auth_module modules/mod_auth.so
178 LoadModule auth_anon_module modules/mod_auth_anon.so
179 LoadModule auth_dbm_module modules/mod_auth_dbm.so
180 LoadModule auth_digest_module modules/mod_auth_digest.so
181 LoadModule include_module modules/mod_include.so
182 LoadModule log_config_module modules/mod_log_config.so
183 LoadModule env_module modules/mod_env.so
184 LoadModule mime_magic_module modules/mod_mime_magic.so
185 LoadModule cern_meta_module modules/mod_cern_meta.so
186 LoadModule expires_module modules/mod_expires.so
187 LoadModule headers_module modules/mod_headers.so
188 LoadModule usertrack_module modules/mod_usertrack.so
189 LoadModule unique_id_module modules/mod_unique_id.so
190 LoadModule setenvif_module modules/mod_setenvif.so
191 LoadModule mime_module modules/mod_mime.so
192 LoadModule dav_module modules/mod_dav.so
193 LoadModule status_module modules/mod_status.so
194 LoadModule autoindex_module modules/mod_autoindex.so
195 LoadModule asis_module modules/mod_asis.so
196 LoadModule info_module modules/mod_info.so
197 LoadModule cgi_module modules/mod_cgi.so

198 LoadModule dav_fs_module modules/mod_dav_fs.so
199 LoadModule vhost_alias_module modules/mod_vhost_alias.so
200 LoadModule negotiation_module modules/mod_negotiation.so
201 LoadModule dir_module modules/mod_dir.so
202 LoadModule imap_module modules/mod_imap.so
203 LoadModule actions_module modules/mod_actions.so
204 LoadModule speling_module modules/mod_speling.so
205 LoadModule userdir_module modules/mod_userdir.so
206 LoadModule alias_module modules/mod_alias.so
207 LoadModule rewrite_module modules/mod_rewrite.so
208 LoadModule proxy_module modules/mod_proxy.so
209 LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
210 LoadModule proxy_http_module modules/mod_proxy_http.so
211 LoadModule proxy_connect_module modules/mod_proxy_connect.so
```



這是每個被 Apache 伺服器動態載入模組的功能。

模組	說明
mod_access	以主機名稱或 IP 位址來控制存取
mod_actions	以檔案的 MIME 形態或需求方法來條件式執行 CGI 程式。
mod_alias	允許經過網站伺服器來轉向或映對部份的實體檔案系統到邏輯實體。
mod_asis	讓檔案能夠被轉換而不用增加任何的 HTTP 標頭。
mod_auth	結合認證和授權。以使用者名稱和密碼來控制存取。密碼加密使用 crypt( )系統呼叫。
mod_auth_anon	就像匿名登錄 FTP 伺服器，這個模組讓使用者能夠以他的 e-mail 當作密碼來存取已授權的區域。
mod_auth_dbm	以使用者名稱和密碼來控制存取。這個資訊用加密的密碼來存放到 DBM 資料庫。
mod_auth_digest	使用 MD5 的授權。
mod_autoindex	自動實作產生目錄索引。
mod_cache	控制 URLs 的快取內容。
mod_cgi	控制由 CGI 程式來分析語法的檔案執行。
mod_dir	假如沒有明確的檔案名稱在 URL(資源定址)中被選取，設定檔案的列表將會被用到。
mod_env	控制傳送到 CGI 程式的環境變數。
mod_example	說明伺服器如何處理模組參考。
mod_expires	使用有時間限制的 HTTP 標頭來實作快取文件的限制時間。

模組	說明
mod_headers	可以建立或產生一般的 HTTP 標頭。
mod_imap	控制由 imap 處理函數分析的線上圖形檔。
mod_include	實作伺服器端的 HTML 包含文件(SSSI)。
mod_info	提供伺服器組態的詳細資訊，這些資訊包含所載入的模組和目前所設定的所有伺服器組態指令。
mod_log_config	在記錄檔裏面包含一般格式的資訊。
mod_mime	以先前定義的數值來改變文件的處理。
mod_mime_magic	就像 Unix 檔案的指令，mod_mime_magic 模組試著決定檔案的 MIME 形態。





mod_negotiation	以文件加密、語言內容、內容長度和 HTTP 標頭欄位來條件的顯示文件。
mod_proxy	實作 HTTP1.1proxy/gateway 伺服器

模組	說明
mod_rewrite	提供彈性和可以延伸的方法來轉向使用者端的請求，而且在檔案系統中對應進來的 URL(資源定址)到其它位置。
mod_setenvif	以各種 HTTP 標頭欄位的內容來條件式的設定環境變數。
mod_so	動態載入的指令必需靜態的編譯到這伺服器上。
mod_speling	試著去修正錯誤的網站位址拼字。
mod_ssl	使用 SSL 和 TLS 來加密。
mod_status	提供 HTTP 伺服器行程 CPU 和頻寬的使用說明。
mod_suexec	讓 CGI 程式只能讓授權的使用者或群組使用。
mod_unique_id	分配權杖給每一位要求服務的使用者。
mod_userdir	指定個別使用者網頁文件的位置。
mod_usertrack	使用 cookie 來追蹤使用者在網站的過程。
mod_vhost_alias	包含支援動態的安裝大量的虛擬主機。

### 23-3 設定主伺服器的組態指令

<<1>>設定主伺服器的組態。

第二段的 http.conf 檔案從第 220 行到 1000 行為設定主伺服器的組態。

```
220 ### Section 2: 'Main' server configuration
```

<<2>>User 使用者與 Group 群組指令

User 指令指定以 apache 使用者來執行 httpd。

Group 指令指定以 apache 群組來執行 httpd。

第 244 行設定 APACHE 伺服器的擁有者，預設是 APACHE。第 245 行設定 APACHE 伺服器的使用群組為 APACHE。一般使用建立 www 的使用者來管理 Apache 系統。



244 User apache

245 Group apache

這 httpd 常駐程式不必以超級使用者的模式來執行，因為這樣比較安全，因為它的權限不會這麼高。在 httpd 常駐程式的擁有者為 Apache。假如我們改變使用者和群組的指令，我們應該指定沒有特權的實體。這樣當我們的網站被滲透時，就可以減低被傷害的危機。因為網站伺服器的權限並沒有特權來修改我們的系統。Httpd 的常駐行程一開始是由 root 超級使用者啟動。連接伺服器到較低的連接埠，而且用使用者和群組指令來轉換使用者和群組是很重要的。其它的伺服器行程則以 User ID 或 Group ID 指令的方式來執行。

### <<3>>ServerAdmin 伺服器管理者指令

ServerAdmin 指令設定 APACHE 的管理者 E-mail 信箱位址。預設是超級使用者的 E-mail。當伺服器發生網頁錯誤時，就會將錯誤的資訊寄到下列的郵件信箱位址。預設是 root@localhost。我們可以將郵件信箱位址設到自己的信箱例如：  
wu.chaiyen@msa.hinet.net，是我們吳佳諺老師的信箱。

252 ServerAdmin root@localhost

當我們伺服器發生一些事情時，我們可以寄事件資訊給伺服器的管理者。在 ServerAdmin 指令中，我們可以設定收到此郵件的伺服器管理者。ServerAdmin 指令可以寄送郵件到使用者帳號，或者轉信到伺服器管理者。

### <<4>>ServerName 伺服器名稱指令

ServerName 指令設定 APACHE 伺服器名稱，在這裏可打上 IP。預設是使用註解符號暫時關閉，如果我們的主機名稱為 www.aasir.com，就將 Server Name 指令設成 www.aasir.com。

266 #ServerName new.host.name:80

ServerName 指令讓伺服器能夠使用名稱和連接埠來辨別它自己。我們建議指定正確的伺服器名稱來預防問題的發生。假如 ServerName 指令沒有指向主機正確的網域名稱，伺服器將沒有辦法轉向。假如主機名稱沒有登錄到 DNS(網域名稱伺服器)，則請輸入我們的 IP 位址。我們將靠這 IP 位址來存取這台電腦。



aasir.com就像是www.aasir.com的別名機台，而這真正的機台名稱為www.aasir.com(aasir.com和www.aasir.com的名子在DNS都有登錄)。我們可以將好幾個網域名稱都指向同一個IP位置，但是伺服器名稱則只有一個。例如aasir.com和www.aasir.com都指向 61.218.29.2 這個機台的位置，而它的伺服器名稱則為[www.aasir.com](http://www.aasir.com)。

Hostname (w/o domain)	IP 位置
.aasir.com	61.218.29.2
flash.aasir.com	61.218.29.3
ftp.aasir.com	61.218.29.2
mail.aasir.com	61.218.29.2
mandrake.aasir.com	61.218.29.6
virtual2.aasir.com	61.218.29.3
virtual3.aasir.com	61.218.29.3
www.aasir.com	61.218.29.2

## <<5>>UseCanonicalName 指令

UseCanonicalName 指令設定伺服器如何決定它自己的名稱和連接埠。當 Canonical 名稱設定 OFF 時 APACHE 會用主機名稱和連接埠來辨視自己的位址與伺服器名稱。

### 275 UseCanonicalName Off

UseCanonicalName 指令決定 Apache 如何建立參考自己的 URLs、伺服器名稱和連接埠變數。當設定為 Off 時，Apache 將使用使用者輸入的主機名稱和連接埠，當設定為 On 時，Apache 將使用 ServerName 指令所指定的名稱。

UseCanonicalName 指令提供命名一致的類型。當 UseCanonicalName 設定為



On 時，Apache 總是使用 ServerName 指令和 Port 指令來建立 URL(資源定址)的參考。當 UseCanonicalName 指令設定為 off 時，這 URL(資源定址)總是用使用者輸入的位址。

## <<6>>DocumentRoot 文件根目錄指令

DocumentRoot 指令設定 APACHE 伺服器存放網站網頁的地方。DocumentRoot 指令執行之後，會指示伺服器把指定的目錄當作是所有網站伺服器網頁的根目錄。如果我們設定/home/chaiyen，則 chaiyen 使用者的家目錄就是當作網站伺服器存取的根目錄。預設是放到/var/www/html 的目錄底下。我們也可以使用符號連結(Symbolic link)來連結到其它的位址。

```
282 DocumentRoot "/var/www/html"
```

我們可以將網站的網頁放到好幾個不同的目錄中，而主要的位置是/var/www/html 為網頁文件的主要目錄，我們可以使用 DocumentRoot 指令來指示網頁文件所放置的位置。

DocumentRoot "/var/www/html"設定網站的根目錄，因此當我們存取 http://www.aasir.com/index.html 時，就是存取/var/www/html/index.html 的網頁。

## <<7>><Directory>指令

<Directory>指令將一群指令作用在設定的檔案系統目錄和子目錄。Apache 有下列的容器指令<Directory> <Limit> <LimitExcept> <Directory> <DirectoryMatch> <Files>、<FilesMatch>、<Location>、<LocationMatch>和<VirtualHost>指令。

我們在這裏要說明<Directory>指令。

語法：

```
<Directory 目錄路徑>....</Directory>
```

<Directory>和</Directory>通常用來圍住一群用在被命名的目錄和子目錄的指令。

我們在網站目錄結構下可以設定每一個目錄的各別存取權限。這預設的權限是非常的嚴格。<Directory />....</Directory>之間的區段設定 Options 和 AllowOverride 的目錄存取允許權。/目錄預設為/var/www/html 的目錄。Options FollowSymlinks 指令



指示伺服器符號連結可以使用連結到其它位置的檔案。

```
292 <Directory />
293     Options FollowSymLinks
294     AllowOverride None
295 </Directory>
```

第 307 行到第 336 行是我們設定放置網站根目錄的屬性

我們由前面預設的目錄為/var/www/html。

```
307 <Directory "/var/www/html">
308
309 #
310 # Possible values for the Options directive are "None", "All",
311 # or any combination of:
312 #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
313 #
314 # Note that "MultiViews" must be named "explicitly" — "Options All"
315 # doesn't give it to you.
316 #
317 # The Options directive is both complicated and important. Please see
318 # http://httpd.apache.org/docs-2.0/mod/core.html#options
319 # for more information.
320 #
321     Options Indexes FollowSymLinks
322
323 #
324 # AllowOverride controls what directives may be placed in .htaccess files.
325 # It can be "All", "None", or any combination of the keywords:
326 #   Options FileInfo AuthConfig Limit
327 #
328     AllowOverride None
329
330 #
331 # Controls who can get stuff from this server.
332 #
333     Order allow,deny
334     Allow from all
335
336 </Directory>
```

在<Directory>....</Directory>指令當作容器將 options 指令、AllowOverride 指令、deny、allow 與 order 指令給包住。Options 指令是設定區塊的功能。Deny 指令和 allow 指令是拒絕或允許使用者可以從哪些地方連接到我們網站，order 指令是設定 deny 與 allow 的使用順序。



Options 指令設定特定目錄的組態特徵。

語法：

Options [+|-]option

一般而言多個 Options 指令的參數可以作用在特定目錄。假如在參數面前加上+的符號，則表示將該項特徵參數加到 Options 上。假如在參數面前加上-的符號，則表示將該項特徵參數從 Options 移除。

這是 Options 指令的參數功能說明。

Options	參數功能說明
indexes	伺服器可以產生此目錄中的檔案列表
includes	提供 ssi 的功能
followsymlinks	允許存取 symbolic link 連結的檔案
all	使用者可以在這網站目錄做任何事
execcgi	允許使用者執行 cgi 程式
none	不允許存取此目錄
IncludesNOEXEC	允許 ssi 功能，但不包括#exec 和#include 元素。
MultiViews	允許以內容檔案的範本尋找。
SymlinksIFOwnerMatch	假如目標和連結的擁有者都相同，則開始符號連結。



這是 AllowOverride 指令的參數說明。AllowOverride 指令設定可以在.htaccess 檔案的指令類型。

AllowOverride 的指令類型	參數功能說明
AuthConfig	啟動授權相關指令 (AuthName,AuthType,AuthUserFile,AuthGroupFile,Require)
FileInfo	啟動 MIME 檔案相關指令 (AddType,AddEncoding,AddLanguage,LanguagePriority)
Indexes	啟動目錄的檔案列表指令。 (FancyIndexing,DirectoryIndex,IndexOptions,IndexIgnore)
Limit	啟動指令控制主機存取(Allow,Deny,Order)
Options	啟動 Options 指令。
None	上面的任何指令都不能被覆寫。
All	所有的指令都能被覆寫

這是 AllowOverride 的指令語法。預設是 AllowOverride All。當我們有設定 AccessFileName 指令來設定目錄的存取檔.htaccess 時，伺服器會找尋.htaccess 檔的指令，而 AllowOverride 指令定義那些指令可以被.htaccess 檔案覆寫。

語法：

AllowOverride All | None | 指令類型

當 AllowOverride 指令設定為 None 時，.htaccess 就會完全被忽略，因此伺服器就不會去讀取檔案系統的.htaccess 檔。

當 AllowOverride 指令設定為 All 時，.htaccess 檔案裏面的指令就會被作用。

在<Directory>指令中不能有巢狀的<Directory>出現。

我們可以一些像是 Shell 的字元來表示路徑。萬用字元\*可以代表一連串的字元，但是不包括 '/' 字元。還有 '?' 字元，'?' 字元只代表任意一個字元。'[]' 代表字元的範圍。'~' 代表可以延伸的字元。'^' 代表第一個字元。'\$' 代表最後一個字元。

例如在<Dirctory /home/\*/public\_html>中使用\*萬用字元來對應在/home 底下的所有目錄。

在下面這個<Dirctory /home/\*/public\_html>指令中，它可以運用在每一個使用者



的 public\_html 目錄。因為每一個使用者的家目錄都是在/home 底下，\*萬用字元則代表所有的字元及使用者。

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

```
Allowoverride All
```

```
</Directory>
```

我們可以使用正規運算式來表示目錄或檔案路徑。這表示符合大寫字母和兩個字元的目錄，例如/A00/或/Z99。

```
<DirectoryMatch "[A-Z][0-9]{2}"/>
```

這是正規運算式常用的代換字元，如果我們要將代換字元當一般字元使用可以在前面加上\'反斜線。例如\'代表逗號，而不是代換字元。

代換字元	說明
.	代替任何字元（新列字元除外）
^	指定正規運算式的開頭
\$	指定正規運算式的結尾
\b	比對字的邊界
Good bad	找出 good 或 bad(為 or 運算子)
[A-Z]	滿足大寫 A-Z 的字元
~	代替前面任何字元
[^xyz]	任何不在 xyz 的字元

正規運算式是由代換字元、數量符號和一般字元所形成。

^apache 是指定符合任何以 apache 為開始的路徑。

apache\$是指定符合任何以 apache 為結尾的路徑。

^apache\$是指定符合只有一個字 apache 所組成的路徑。





這是代表數量符號。[0-9]{3}代表 0 到 9 的數字有三個。

數量符號	說明
{n}	n 個
{n,m}	至少 n 個但不超過 m 個
*	0 個或多個
+	1 個或多個
?	0 或 1 個

hello?符合 hell 或 hello。

hello+符合 hello、helloa、hellolll、hellooooo、heooolllll。

Hello\*符合 hell、hello、helloa、helloooo、hellolllll。

([a-z]+\.[a-z]\.([a-z]+) 代表 www.aasir.com 或 flash.aasir.com，指要裏面有兩個逗點。([a-z]+)代表任意 a 到 z 的字元至少有 1 個，所以 www 或 flash 都符合。  
\.代表逗號，而不是代換字元，\一般稱為跳脫字元。

(^index\.htm\$)運算式表示 index.htm，而不是 index.html 或 pndex.htm。因為在開頭使用^是指定 i 為起始字元，而在尾部使用\$代換字元是指定 m 為結尾字元。

下列是跳脫字元和說明。

跳脫字元	說明
\?	符合字元?
\+	符合字元+
\*	符合字元*
\.	符合字元.
\\	符合字元\\

<Directory ~"^\www/.\*/[0-9]{3}>表示該目錄包含/www/和 0 到 9 的三個數字的目錄，例如/var/www/public/123 或/usr/www/good/568 的目錄。~

<Directory ~ abc\$>.....</Directory>指令指定以 abc 結尾的任何目錄。例如



/var/www/goodabc 或 /usr/local/abc。

<Dirctory />是允許任何人存取我們的檔案，因為/為檔案系統的根目錄。所以我們 Order 指令順序是先禁止所有人來存取。

```
<Dirctory />
  Order Deny,Allow
  Deny from All
</Dirctory>
```

### <<8>>取消根目錄的檔案列表

從第 342 行到第 345 行是取消根目錄的自動檔案列表，假如該目錄沒有檔案列表的網頁，則提供預設的歡迎網頁。第 342 行到第 345 行的指令取消根目錄(預設 /var/www/html)的檔案列表，取而代之的是使用 noindex.html 來當預設放置到 Apache 的網頁。當我們進入到網站時，需求服務，Apache 就會提供預設的歡迎網頁，預設是 index.html。第 343 行關閉 Indexes 目錄的檔案列表。

```
342 <LocationMatch "^/$>
343     Options -Indexes
344     ErrorDocument 403 /error/noindex.html
345 </LocationMatch>
```

在第 292 行到第 295 行設定/根目錄的選項是 FollowSymLinks，我們在第 343 行使用 - 的符號關閉 Indexes(目錄的檔案檔案列表)。+ 是啟動 Options 指令的選項，- 是關閉 Options 指令的選項。

```
292 <Directory />
293     Options FollowSymLinks
294     AllowOverride None
295 </Directory>
```

<Location>指令就和<Dirctory>指令一樣，但是它是使用 URL(資源定址)而不是實體的檔案位置。例如<Location /server-info>要求/server-info 的 URL(資源定址)，這是會引發 server-info 處理函數來產生伺服器的資訊檔。因此這可以是/server-info



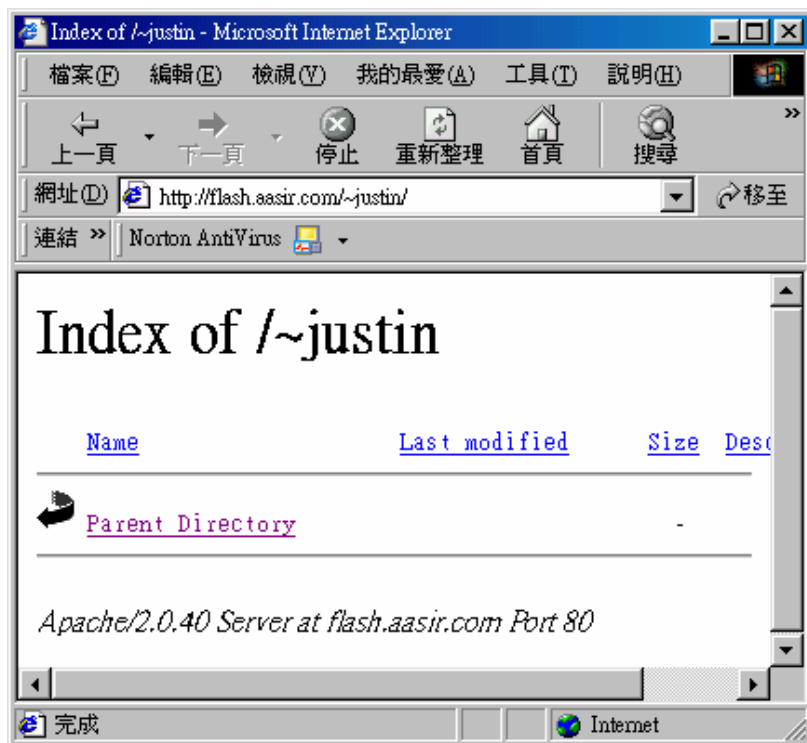
或/server-info/info.html。<Location>指令完全的獨立於檔案系統，所以它能夠符合最後的 URL(資源定址)部份，而<Dirctory>指令指能符合最後的目錄。

<LocationMatch>指令比<Location>好用，因為只要符合該正規運算式的都可以被指定。例如<LocationMatch “^[a-z]+/\$”>，表示/var/www/、/usr/local/、/home/chaiyen/和/etc/httpd/都符合。

<LocationMatch “^/\$”>則是指定符合’/’的目錄，也就是根目錄。

### <<9>>放置個人網頁的地方

當我們在網址後面輸入~user 時，就可以連接到該使用者的家目錄，在這裏我們在 http://flash.aasir.com/的後面輸入~justin，則會跳到/home/Justin/public\_html 的目錄去。



使用者目錄預設的為 `public_html`，然後它的功能已經被關掉。因此在第 372 行我們將註解拿掉，並且啟動使用者目錄，所以我們也在第 365 行將使用者目錄 `disable` 給關掉而加上註解。`UserDir` 在預設為 `disabled`(取消)。為了啟動當要求 `~/user/` 的服務時，能夠到使用者的 `public_html` 目錄，所以我們將第 365 行的 `UserDir disable` 註解給加上，然後再去除第 372 行 `UserDir public_html` 的註解。

我們可以使用 `mod_userdir.c` 模組提供使用者它們自己的網站。`Mod_userdir` 模組提供一個 `UserDir` 指令。當 Apache 看到正確格式的 URL(`~/使用者`)，`UserDir` 指令提供有限制的轉向到該使用者的家目錄。

```
359 <IfModule mod_userdir.c>
360     #
361     # UserDir is disabled by default since it can confirm the presence
362     # of a username on the system (depending on home directory
363     # permissions).
364     #
365     UserDir disable
366
367     #
368     # To enable requests to ~/user/ to serve the user's public_html
369     # directory, remove the "UserDir disable" line above, and uncomment
370     # the following line instead:
371     #
372     #UserDir public_html
373
374 </IfModule>
```

為了啟動當要求 `~/user/` 的服務時，能夠到使用者的 `public_html` 目錄，所以我們將第 365 行的 `UserDir disable` 註解給加上，然後再去除第 372 行 `UserDir public_html` 的註解。這樣就可以在載入 `mod_userdir.c` 模組時啟動 `~/user/` 該使用者目錄服務。

```
359 <IfModule mod_userdir.c>
360     #
361     # UserDir is disabled by default since it can confirm the presence
362     # of a username on the system (depending on home directory
363     # permissions).
364     #
365     #UserDir disable
366
367     #
368     # To enable requests to ~/user/ to serve the user's public_html
369     # directory, remove the "UserDir disable" line above, and uncomment
370     # the following line instead:
371     #
372     UserDir public_html
373
374 </IfModule>
```



每個在伺服器的帳號(有權使用機器的使用者),都可以在自己的目錄下建立自己的網頁,而 380 行到 391 行就是設定每個人網頁目錄的設定。\*是指每個使用者。到使用者目錄下,增加 public\_html 的目錄放置網頁,再將權限修改成 755 就可以了,如果我們不需要使用者個人網站就不需要設定。

```
380 #<Directory /home/*/public_html>
381 #   AllowOverride FileInfo AuthConfig Limit
382 #   Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
383 #   <Limit GET POST OPTIONS>
384 #       Order allow,deny
385 #       Allow from all
386 #   </Limit>
387 #   <LimitExcept GET POST OPTIONS>
388 #       Order deny,allow
389 #       Deny from all
390 #   </LimitExcept>
391 #</Directory>
```

例如我們可以只增加 justin 個人的網站,因此我們將第 380 行的星號改為 justin。

```
380 #<Directory /home/justin/public_html>
381 #   AllowOverride FileInfo AuthConfig Limit
382 #   Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
383 #   <Limit GET POST OPTIONS>
384 #       Order allow,deny
385 #       Allow from all
386 #   </Limit>
387 #   <LimitExcept GET POST OPTIONS>
388 #       Order deny,allow
389 #       Deny from all
390 #   </LimitExcept>
391 #</Directory>
```

我們在 justin 使用者的/home/justin 目錄下新增加 public\_html 的目錄,來放置 justin 的網站。在這前題下,我們必須先有使用 useradd justin 來增加 justin 這個使用者。

```
[root@flash root]# cd /home/justin
[root@flash justin]# mkdir public_html
[root@flash justin]# ls
public_html
```

我們改變 justin 目錄的權限,讓一般使用者可以觀看 public\_html 的目錄內容。

```
[root@flash chaiyen]# chmod 711 /home/justin
[root@flash chaiyen]# chmod 755 /home/justin/public_html
```



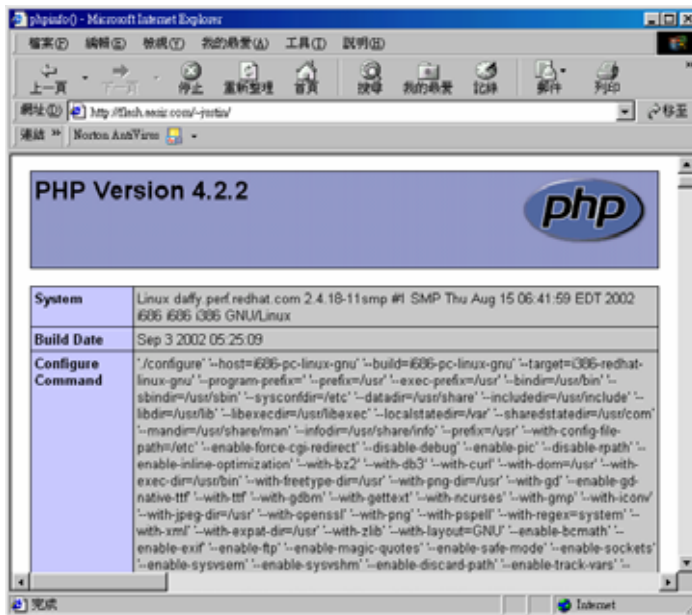
我們也可以在 public\_html 增加索引網頁 index.php，這樣當讀取目錄索引時，就會讀取這網頁的內容。

```
[root@flash public_html]# vi index.php
```

我們呼叫 phpinfo() 函數，這是可以顯示我們網頁的資訊。

```
<?PHP  
phpinfo();  
?>
```

我們在網站上打上 <http://flash.aasir.com/~justin/> 就可以連到 justin 的網站了，並且直接讀取 index.php 的網頁。



我們可以使用 UserDir 指令來指示參考路徑。當要求服務時，就會跳到該參考路徑去提供網頁服務。例如我們指令為 UserDir public\_html 時，其參考路徑就是 ~justin/public/www/index.php。下列是我們 UserDir 指令所設定的路徑名稱和所對應的參考路徑名稱。

UserDir 指令	參考路徑
UserDir public_html	~justin/public_html/index.php
UserDir /public/*/www	/public/justin/www/index.php
UserDir /user/local/web	/usr/loca/web/justin/index.php
UserDir http://flash.aasir.com/users	http://flash.aasir.com/users/justin/index.php
UserDir http://flash.aasir.com/~*	http://flash.aasir.com/~justin/index.php
UserDir http://flash.aasir.com/*/html	http://flash.aasir.com/justin/html/index.php

我們也可以指定 UserDir 不可以轉換到特定的使用者。例如我們可以使用

```
UserDir disable root
```

來禁止 URL-to-path 的名稱轉換到超級使用者 root。

我們也可以全面禁止 URL-to\_path 的轉換，除了 chaiyen、justin 使用者，使用

```
UserDir disable
```

```
UserDir enable chaiyen justin
```

## <<10>>DirectoryIndex 指令

使用 DirectoryIndex 來指定預設的索引檔。DirectoryIndex 指令設定該目錄的索引檔。當我們輸入該目錄的路徑時，DirectoryIndex 指令就會指定該索引檔來回應。當 client 端的使用者輸入位址為 http://flash.aasir.com/~justin 則會轉換到 http://flash.aasir.com/~justin/ 的目錄底下。然後目錄就會搜尋所有在 DirectoryIndex 指令所指的檔案為優先，然後再顯示該檔案。在第 401 行是設定先搜尋名稱為 index.html 的網頁，則當輸入 http://flash.aasir.com/~justin/ 時就會回應 http://flash.aasir.com/~justin/index.html。假如該目錄沒有這些檔案而且 index 選項又被啟動，則這伺服器會產生在這目錄的檔案索引。

設定每個目錄中預設的網頁檔案，已排前面的為優先。我們預設是 index.html。

```
401 DirectoryIndex index.html index.html.var
```



這是指定以 index.php 為預設的網頁，如果 index.php 不存在才是以 index.html 為預設的索引檔，如果 index.php 和 index.html 都不存在則會去執行/cgi-bin/run.cgi 的程式。DirectoryIndex 指令的優先順序是以前面的為優先索引。

```
DirectoryIndex index.php index.html /cgi-bin/run.cgi
```

## <<11>>AccessFileName 指令

AccessFileName 指令設定目錄每個目錄的組態。

我們可以在目錄上增加一個 access 存取檔案，來控制該目錄的存取。預設 AccessFileName 指令是設定.htaccess 檔為包含該目錄以及子目錄的控制組態。

AccessFileName 是設定每個目錄的控制資訊指令，在這裏我們設定.htaccess 檔。

### 407 AccessFileName .htaccess

我們除了可以在 httpd.conf 設定目錄的組態檔，我們也可以在每一個目錄裏面放置組態檔。

我們使用 locate .htaccess|more 來觀看所有目錄的組態檔.htaccess。每一個.htaccess 目錄組態檔都控制著該目錄的組態。 .htaccess 目錄組態是預設的每個目錄組態。在第 407 行 AccessFileName 指令設定存取檔案的名稱為.htaccess。 .htaccess 的逗點表示該檔案是隱藏檔，這樣可以保護該檔案的安全。 .htaccess 目錄組態檔的優先權高於主伺服器組態檔 httpd.conf，因此當.htaccess 存在時，該目錄的組態是依據.htaccess。

```
[root@flash /]# locate .htaccess|more
/usr/share/squirrelmail/plugins/squirrelspell/modules/.htaccess
/var/www/html/html/admin/.htaccess
/var/www/html/html/admin/case/.htaccess
/var/www/html/html/admin/language/.htaccess
/var/www/html/html/admin/links/.htaccess
/var/www/html/html/admin/modules/.htaccess
/var/www/html/html/language/.htaccess
/var/www/html/html/modules/AvantGo/language/.htaccess
/var/www/html/html/modules/WebMail/language/.htaccess
/var/www/html/html/modules/Content/language/.htaccess
/var/www/html/html/modules/Downloads/language/.htaccess
/var/www/html/html/modules/Encyclopedia/language/.htaccess
/var/www/html/html/modules/FAQ/language/.htaccess
```





一般我們將每個目錄的存取控置組態設定為.htaccess、.acl、.access、.config 或.perdirectory 組態檔。AccessFileName 指令會指伺 Apache 去讀取和分析這些目錄組態檔，因此定義目錄組態檔越多，Apache 伺服器的功就作得越多。

```
AccessFileName .htaccess .acl .access .config .perdirectory
```

第 413 行到第 416 行防止一般使用者透過網站看到.htaccess 和.htpasswd 的檔案。如果要改變隱藏檔的名稱，則修改“^\.ht”。<Files ~ “^\.ht”>是指任何由.ht 字元組合而成的檔案，這裏是指.htaccess 和.htpasswd 這兩個檔案。

```
413 <Files ~ "\.ht">
414     Order allow,deny
415     Deny from all
416 </Files>
```

<Files>指令就像<Directory>指令一樣，但是<Files>指令是用在指定檔案，而<Directory>指令則是指定目錄。<Files>指令經常和萬用字元\*用來參考一大群檔案，例如<Files \*.gif>。我們也可以將<Files>指令放到<Dirctory>和<Location>的容器內來限制部份使用的檔案系統。

<Directory /var/www/html> <Files \*.gif>將滿足而且解析這個格式 /var/www/html/good.gif 的任何檔案。

```
<Directory /var/www/html>
  <Files *.gif>
    SetHandler /cgi-bin/make-gif.cgi
  </Files>
</Dirctory>
```

<<12>>條件組態<IfModule>和<IfDefine>指令

TypeConfig 指定設定預設的檔案 MIME 形態。

TypeConfig 指令說明 mime.types 的檔案位置，預設是在/etc/mime.types。



## 422 TypesConfig /etc/mime.types

DefaultType 指令設定伺服器將使用的文件格式。假如我們的網站使用很多的文字檔(text)或 HTML 的文件，“text/plain”是一個很好的參數選擇。假如我們的網站有很多執行檔、圖片或應用程式，則使用“application/octet-stream”當參數，則使用者瀏覽時，會顯示該檔案為應用程式。

## 433 DefaultType text/plain

使用 mod\_mime\_magic.c 模組可以讓伺服器知到是什麼檔案被需求服務。我們在運用這項服務之前必需先確定這個模組 mod\_mime\_magic.c 已經被加入到這個系統。我們可此從/usr/share/magic.mime 或/etc/httpd/conf/magic 得到提示來決定被要求服務的檔案內容。預設是使用/etc/httpd/conf/magic。

```
440 <IfModule mod_mime_magic.c>
441 # MIMEMagicFile /usr/share/magic.mime
442 MIMEMagicFile conf/magic
443 </IfModule>
```

<IfDefine>或<IfModule>指令的區段內容為條件式的組態，也就是我們可以設定在特定的情況下才執行<IfDefine>或<IfModule>所包含的內容。

例如因為第 184 行的 LoadModule 指令有載入 mod\_mime\_magic.c 模組，所以第 440 行的<IfModule mod\_mime\_magic.c>指令會為真 TRUE，因次會執行<IfModule mod\_mime\_magic.c>所包含的第 441 行和第 223 行的內容。我們只要將第 184 行加上註解，也就是將 LoadModule mod\_mime\_magic.so 模組取消，則第 440 行的<IfModule mod\_mime\_magic.c>就會出現 false，因此就不會執行第 440 行到第 443 行的程式。Mod\_mime\_magic.so 是 mod\_mime\_magic.c 已經編譯過的動態物件。

```
180 LoadModule auth_digest_module modules/mod_auth_digest.so
181 LoadModule include_module modules/mod_include.so
182 LoadModule log_config_module modules/mod_log_config.so
183 LoadModule env_module modules/mod_env.so
184 LoadModule mime_magic_module modules/mod_mime_magic.so
185 LoadModule cern_meta_module modules/mod_cern_meta.so
```



```
440 <IfModule mod_mime_magic.c>
441 #   MIMEMagicFile /usr/share/magic.mime
442     MIMEMagicFile conf/magic
443 </IfModule>
```

使用<IfModule>指令可以讓我們藉著判段是否有載入指定模組來執行<IfModule>區段內的指令。我們可以輕易的取消<IfModule>區段內的指令，只要我們將該載入模組 LoadModule 指令設為註解即可。<IfModule>指令提供我們簡易的啟動和關閉該區段的指令，我們只要在指定的模組加上或去除註解。

我們也可以使用!表示相反的意義。<IfModule !mod\_mime\_magic.c>表示沒有載入 mod\_mime\_magic.c 的時後才會執行該容器的區段。

```
<IfModule !mod_mime_magic.c>
#當 mod_mime_magic.c 的模組沒有被載入時,這裏面的指令才會被執行
.....
.....
.....
</IfModule>
```

只要測試條件為真則執行<IfModule>區段的內容，只要測試條件為假則不執行區段的內容。

```
<IfModule 測試條件> ...</IfModule>
```

當參數名稱已存在，則<IfDefine>指令會傳回真，而執行<IfDefine>的區段內容。

```
<IfDefine 參數名稱>...</IfDefine>
```

我們可以使用 httpd -D ReverseProxy 來定義常數。

當 ReversProxy 常數存在時就會執行<IfDefine>指令區段內的 LoadModule。

```
<IfDefine ReversProxy>
```



```
LoadModule rewrite_module modules/mod_rewrite.so  
</IfDefine>.
```

### <<13>>HostnameLookups 指令

HostnameLookups 指令用使用者端 IP 位址來查詢 DNS 其主機名稱。預設是關閉 Off。HostnameLookups 指令能夠查詢使用者端的 IP 和主機名稱。預設是 Off，因為如果打開 On 則每次使用者要求服務時，伺服器就會去查詢名稱伺服器 DNS 該使用者的資料，這樣會讓費很多網路資源。如果要增加網路流量的效率，可以將 HostnameLookups 指令設為 Off。

## 453 HostnameLookups Off

假如 HostnameLookups 指令打開，每一個進入的連線將產生一個 DNS 的查詢，並且將使用者端的 IP 轉換成其主機名稱。假如我們的伺服器接收到很多的要求服務，又同時要去 DNS 查詢該要求端的名稱，將造成我們網路資源的浪費。一般而言我們都會將 HostnameLookups 給關閉，除非我們要作記錄檔的程式或者是作網站資源使用的統計分析才會將 HostnameLookups 給打開。這種記錄解析程式是安裝在 Apache 的套件上，在離峰時，我們經常使用它來分析與統計網站的經營效果。

### <<14>>伺服器的記錄指令 ErrorLog、LogLevel、LogLevel 和 Custom 指令

我們可以設定當 Apache 伺服器發生錯誤時所記錄的一些資訊。預設我們是將這些 Apache 的錯務資訊記錄到/etc/httpd/logs/error\_log 目錄。我們除了可以記錄錯誤的資訊，也可以設定警告記錄的等級和匿名的記錄。

ErrorLog 是設定錯誤記錄檔的指令，預設是記錄在/etc/httpd/logs/error\_log 的地方。當我們有錯誤的資訊時，就會將錯誤的資訊寫在 error\_log 檔案上。假如我們沒有在<VirtualHost>...</VirtualHost>內指定錯誤記錄檔指令，在虛擬網站上的錯誤資訊也會記錄到這個位置。假如我們在<VirtualHost>...</VirtualHost>內指定錯誤記錄檔指令，在虛擬網站上的錯誤資訊也會記錄到所指定的那個位置

## 462 ErrorLog logs/error\_log

第 469 行的 LogLevel 記錄等級指令控制記錄到錯誤記錄檔的資訊數量。這些資



訊包括除錯 debug、資訊 info、注意 notice、警告 warn、錯誤 error、評論 crit、說明 alert 和緊急 emerg。

## 469 LogLevel warn

第 475 行到第 478 行 LogFormat 指令定義伺服器的記錄的格式。

```

475 LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
    combined
476 LogFormat "%h %l %u %t \"%r\" %>s %b" common
477 LogFormat "%{Referer}i -> %U" referer
478 LogFormat "%{User-agent}i" agent

```

這是記錄檔符號的格式和說明。

代號	說明
%a	使用者端的 IP 位址
%b	送去使用者端不包含標頭檔的位元組
%{VAR}e	環境變數 VAR 的內容
%f	被需求端參考的檔案名稱
%h	使用者端主機的名稱
%{Header}i	在 HTTP 要求中的標頭檔內容。
%l	Identd 常駐程式報告使用端的情況。
%{Note}n	從不同模組的訊息註解內容。
%{Header}o	在 HTTP 回應中的標頭檔內容。
%p	要求接收端的連接埠編號。
%P	處理要求服務的伺服器編號。
%r	使用者端的 HTTP 要求服務
%s	由要求服務所產生的伺服器回應碼。
%t	目前當地的時間和日期。
%T	填滿使用者要求所需的秒數。
%u	假如存取控制權需要使用者名稱和密碼授權，這代表由使用者端支援的使用者名稱。
%U	被使用者端要求的 URL(位址)
%v	根據名稱伺服器定義伺服器的主機名稱和網域名稱。
%V	根據 ServerName 指令處理伺服器的主機名稱和網域名稱。

第 488 行 CustomLog 指令是設定存取記錄檔的位置和格式。假如我們沒有在



<VirtualHost>.....</VirtualHost>定義任何的存取記錄檔，則虛擬主機的存取記錄檔也將記錄在這邊。如果我們在<VirtualHost>.....</VirtualHost>定義存取記錄檔，則虛擬主機的存取記錄檔將記錄在<VirtualHost>區段那邊。

```
488 CustomLog logs/access_log combined
```

如果我們要參考 referer、代理 agent 記錄檔，則將第 494 行和第 495 行的註解符號#取消。

```
494 #CustomLog logs/referer_log referer
495 #CustomLog logs/agent_log agent
```

假如我們使用單一記錄檔 CustomLog 指令來存取、代理參考資訊，則我們可以使用第 501 行的 CustomLog logs/access\_log combined 來組合記錄檔格式。

```
501 #CustomLog logs/access_log combined
```

一般的格式包含使用者主機名稱、IP 位址、使用者名稱、被要求服務的時間、伺服器的回應碼和傳送的位元組大小，而這個格式是以網站伺服器的標準或 FTP 伺服器的標準。

為了追蹤連接，參考格式一般都是存在 URL(資源定址)位址上，而這 URL(資源定址)位址是和該主機的 ServerRoot 指令有關。例如，假如有一個網頁 <http://www.aasir.com/module/link.htm> 包含一個連接到 <http://flash.aasir.com/jigsaw/jigsaw.html> 的時後，當有一個使用者存取該連接 <http://www.aasir.com/module/link.htm> 時，這個在 <http://flash.aasir.com/> 的參考記錄檔為 <http://www.aasir.com/module/link.htm> - > </jigsaw/jigsaw.html>，而這個資訊可以決定使用者到達我們網站的路徑。

{User-Agent}i 這個代理格式儲存每個進入的使用者 HTTP 標頭檔內容。這個欄位指出使用者的瀏覽器名稱、版本、語言和作業系統。



組合格式包含所有其它三個記錄檔格式的資訊並且組合成一行。這個格式可以用一個集中檔儲存所有連接相關的進入記錄

CustomLog 指令指派 LogFormat 格式到一個檔名，而 CustomLog logs/access\_log combined 是唯一可以組態 HTTP 的記錄檔。

這是我們的網站伺服器存取記錄檔。

```
[root@aasir chaiyen]# vi /etc/httpd/logs/access_log
```

存取記錄檔/etc/httpd/logs/access\_log 的格式是以 Apache 伺服器的 LogFormat 格式定義。

```
475 LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
    combined
```

%h 是使用者端主機的名稱，因此 219.76.64.2 為使用者端主機的名稱，也是使用者端 IP 的位址。

%t 是目前當地的時間和日期，因此 02/Feb/2003:06:38:33 +0800 為當地的時間及日期。

%r 是使用者端的 HTTP 要求服務，因此 Get /modules/Downloads/images/down-logo.gif HTTP/1.0，是使用者端要求使用 HTTP 傳輸模式來要求服務，這個要求服務是從我們網站下載 down-logo.gif 的檔案，傳輸方法是用 Get。

%s 是由要求服務所產生的伺服器回應碼。

%b 是送去使用者端不包含標頭檔的位元組，在這裏是 3277 個位元組。

%{Header}i 是在 HTTP 要求中的標頭檔內容，在這裏是 http://aasir.com/modules.php?name=Downloads。

{User-Agent}i 這個代理格式儲存每個進入的使用者 HTTP 標頭檔內容。這個欄位指出使用者的瀏覽器名稱、版本、語言和作業系統，在這裏使用的瀏覽器是 IE6.0(相容於 Mozilla/4.0 的瀏覽器)。





```

219.76.64.2 - - [02/Feb/2003:06:38:33 +0800] "GET /modules/Downloads/ima
ges/down-logo.gif HTTP/1.0" 200 3277 "http://aasir.com/modules.php?name=
Downloads" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
219.76.64.2 - - [02/Feb/2003:06:38:40 +0800] "GET /modules.php?name=Down
loads&d_op=viewdownload&cid=11 HTTP/1.0" 200 4656 "http://aasir.com/modu
les.php?name=Downloads" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5
.1)"
219.76.64.2 - - [02/Feb/2003:06:38:43 +0800] "GET /modules.php?name=Down
loads&d_op=viewdownload&cid=12 HTTP/1.0" 200 4875 "http://aasir.com/modu
les.php?name=Downloads&d_op=viewdownload&cid=11" "Mozilla/4.0 (compatibl
e; MSIE 6.0; Windows NT 5.1)"
219.76.64.2 - - [02/Feb/2003:06:38:44 +0800] "GET /modules/Downloads/ima
ges/lwin.gif HTTP/1.0" 200 80 "http://aasir.com/modules.php?name=Downloa
ds&d_op=viewdownload&cid=12" "Mozilla/4.0 (compatible; MSIE 6.0; Windows
NT 5.1)"
219.76.64.2 - - [02/Feb/2003:06:38:47 +0800] "GET /modules.php?name=Down
loads&d_op=getit&lid=19 HTTP/1.0" 302 26 "http://aasir.com/modules.php?n
ame=Downloads&d_op=viewdownload&cid=12" "Mozilla/4.0 (compatible; MSIE 6
.0; Windows NT 5.1)"

```

這是來自 61.218.232.187 位址的存取記錄。我們藉由/etc/httpd/logs/access\_log 這個網站存取記錄檔來了解每個來自我們網站的使用者，以及這個使用者端在我們網站作了什麼事，以及使用者端的情況。

```

61.218.232.187 - - [02/Feb/2003:08:20:37 +0800] "GET /scripts/..%c1%9c..
/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 1040 "-" "-"
61.218.232.187 - - [02/Feb/2003:08:20:39 +0800] "GET /scripts/..%%35%63.
/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 400 973 "-" "-"

```

這是來自 210.85.116.46 的使用者，它是藉由

<http://www.web-desinger.com.tw/thing/thing.asp> 來連接到我們的網站。

```

210.85.116.46 - - [02/Feb/2003:10:18:19 +0800] "GET / HTTP/1.1" 200 7906
"http://www.web-designer.com.tw/thing/thing.asp?classid=2" "Mozilla/4.0
(compatible; MSIE 6.0; Windows NT 5.1)"

```

這個 61.229.53.148 是藉由雅虎搜尋引擎搜尋字串 access2002 而找到我們的網站 [http://tw.websearch.yahoo.com/search/web\\_kimo?p=access2002](http://tw.websearch.yahoo.com/search/web_kimo?p=access2002)。

```

61.229.53.148 - - [02/Feb/2003:16:36:56 +0800] "GET / HTTP/1.1" 200 7905
"http://tw.websearch.yahoo.com/search/web_kimo?p=access2002&Type=0&u=b&
hc=0&hs=0" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)"

```

211.74.218.16 是藉由 google 搜尋引擎搜尋 access 而找到我們的網站，這個使用者的作業系統是 Windows 98。因此，我們可以藉由存取記錄檔 access\_log 來分析我們網站的使用情況。



```
211.74.218.16 - - [03/Feb/2003:04:05:54 +0800] "GET /modules.php?name=Downloads&d_op=MostPopular HTTP/1.1" 200 7211 "http://www.google.com/search?hl=zh-TW&ie=UTF-8&oe=UTF-8&q=access+%E7%AF%84%E6%9C%AC&btnG=Google%E6%90%9C%E5%B0%8B&lr=lang_zh-TW" "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98; Win 9x 4.90)"
```

來自 66.57.250.223 的使用者，其作業系統是 Linux 其瀏覽器是 Konqueror/3 征服者。

```
66.57.250.223 - - [04/Feb/2003:20:03:25 +0800] "GET /themes/NukeNews/images/corner-bottom-right.gif HTTP/1.1" 200 111 "http://aasir.com/modules.php?name=News&new_topic=8" "Mozilla/5.0 (compatible; Konqueror/3; Linux)
```

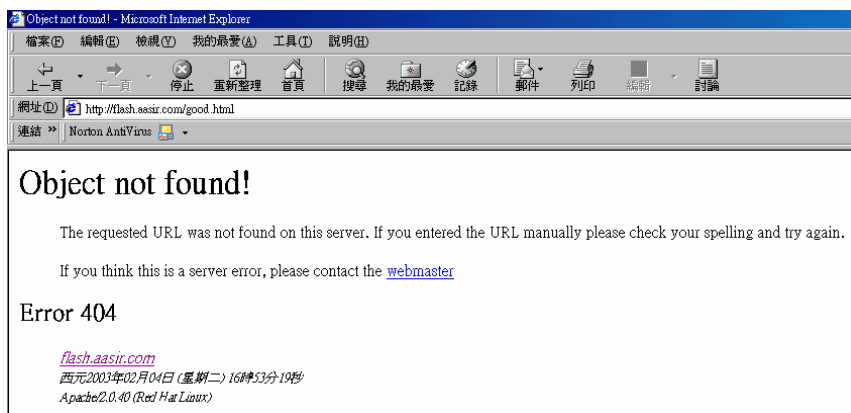
## <<15>>ServerSignature 指令

ServerSignature 指令設定增加識別名稱在伺服器所產生的頁面。

任何由 Apache 所產生的網頁可以在網頁底下增加識別名稱。我們可以選擇性的增加伺服器版本、虛擬主機名稱到這 Apache 所產生的網頁。這 ServerSignature 指令可以設定為 On、Off 和 Email。預設是 On。

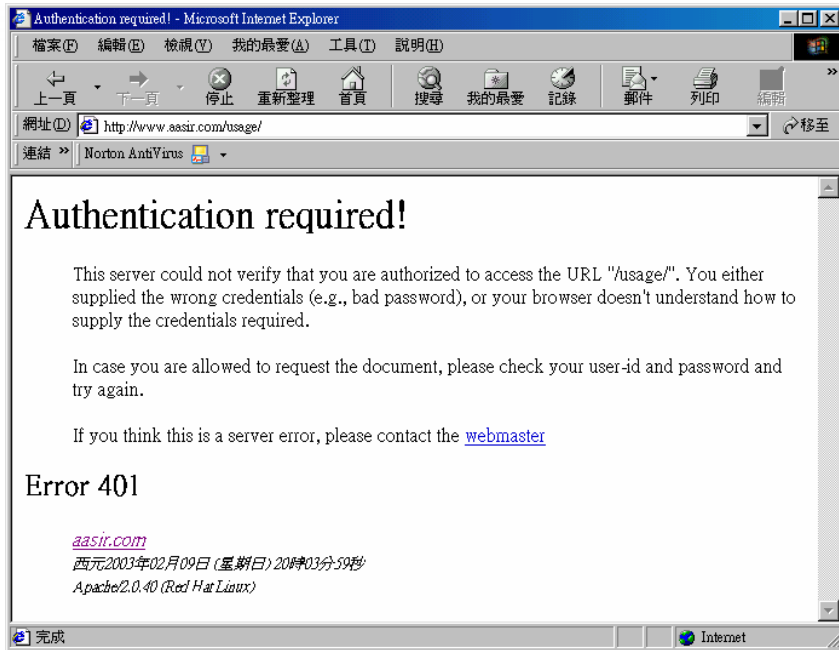
### 510 ServerSignature On

當 ServerSignature 指令設為 On 時，發生網頁錯誤時，在 Error 404 後面有網站名稱、日期、伺服器管理者和伺服器版本。這是由伺服器產生的網頁。



當 ServerSignature 指令設為 On 時，發生網頁錯誤時，在 Error 401 後面有伺服器名稱、日期、伺服器管理者和伺服器版本。這是由伺服器產生的網頁，我們在後面增加識別名稱。





當我們按下 `webmaster` 時，就可以將錯誤資訊寄給伺服器管理者。在第 252 行的 `ServerAdmin` 指令為設定伺服器管理者。

```
252 ServerAdmin wu.chaiyen@msa.hinet.net
```

## <<16>>Alias 別名和轉向

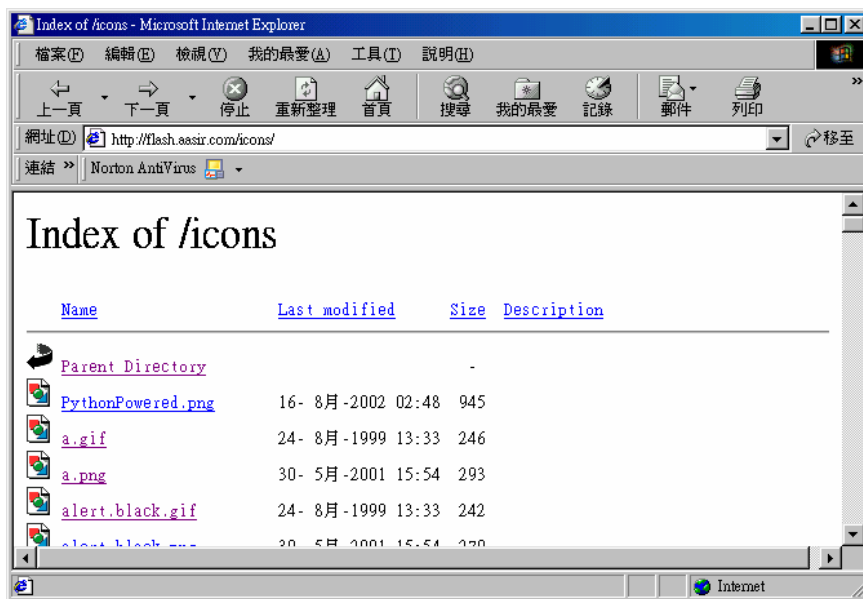
`Alias` 指令重新安置內容到別名，也就是轉向的意義。

我們可以在第 539 行到 551 行設定重新安置檔案內容到別名。`Alias` 指令可以將複雜的路徑簡化，讓我們只要輸入別名即可得到相同的檔案位置。`Alias` 別名會依照我們所輸入的名稱來取代原有的路徑名稱。假如在別名後面有 `slash(/)`，則在真實檔案後面也要有 `(/)`。在第 525 行的 `/icons/` 是重新安置檔案 `/var/www/icons/` 的別名，而不是 `icons`，因為在 `/icons/` 後面有 `slash(/)`。



```
525 Alias /icons/ "/var/www/icons/"
526
527 <Directory "/var/www/icons">
528     Options Indexes MultiViews
529     AllowOverride None
530     Order allow,deny
531     Allow from all
532 </Directory>
```

假如我們網站的根目錄是在/var/www/html/html，而當我們輸入 http://flash.aasir.com/icons/ 的位置就會連接到/var/www/icons/的位置，因此這就像是 Shell Script 的 ln -s 符號連結指令一樣，將/var/www/html/html/icons/連結到 /var/www/icons/的目錄中。



第 539 行的 Alias /manual "/var/www/manual" 會將 http://flash.aasir.com/manual 對應到/var/www/manual 去。即使我們改變網站的根目錄位址，Alias 別名指令還是可以將/manual 對應到/var/www/manual。我們可以將這加上註解符號，如果我們不需



要別名 Alias 指令。

```
539 Alias /manual "/var/www/manual"
540
541 <Directory "/var/www/manual">
542     Options Indexes FollowSymLinks MultiViews
543     AllowOverride None
544     Order allow,deny
545     Allow from all
546 </Directory>
547
548 <IfModule mod_dav_fs.c>
549     # Location of the WebDAV lock database.
550     DAVLockDB /var/lib/dav/lockdb
551 </IfModule>
```

ScriptAlias 和別名 Alias 指令一樣，都是重新安置內容到別名，只是 ScriptAlias 所重新安置的是執行檔的內容。假如網站伺服器的根目錄在 /var/www/html/html/，在第 561 行的 ScriptAlias /cgi-bin "/var/www/cgi-bin/" 將 http://flash.aasir.com/cgi-bin/ 對應到 /var/www/cgi-bin/ 的目錄去。在 /cgi-bin/ 的目錄中都是執行檔。

/var/www/cgi-bin 會對應到我們使用 ScriptAlias 指令所設定的別名路徑。第 575 行到 580 行是 /var/www/cgi-bin 目錄的選項設定。

```
561 ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
562
563 <IfModule mod_cgid.c>
564 #
565 # Additional to mod_cgid.c settings, mod_cgid has Scriptsock <path>
566 # for setting UNIX socket for communicating with cgid.
567 #
568 #Scriptsock          logs/cgisock
569 </IfModule>
570
571 #
572 # "/var/www/cgi-bin" should be changed to whatever your ScriptAliased
573 # CGI directory exists, if you have that configured.
574 #
575 <Directory "/var/www/cgi-bin">
576     AllowOverride None
577     Options None
578     Order allow,deny
579     Allow from all
580 </Directory>
```



## <<17>>Redirect 轉向指令

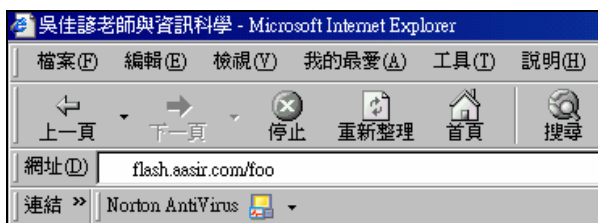
Redirect 指令允許我們告訴使用者以前舊網站的資料現在已經搬到其它新網站的位址。Redirect 指令告訴使用者重新轉向到新的位置。例如第 587 行的 Redirect 會將其伺服器網址/foo 轉向到 <http://www.example.com/bar>。

```
587 # Redirect permanent /foo http://www.example.com/bar
```

如果我們將 flash.aasir.com 的網站第 587 行 Redirect 指令的註解拿調，並將其導向到 <http://aasir.com>，則當使用者在瀏覽器上輸入 <http://flash.aasir.com/foo> 的網址，Redirect 指令就會轉向到 <http://aasir.com> 的網址去。

```
587 Redirect permanent /foo http://aasir.com
```

我們在瀏覽器上輸入 [flash.aasir.com/foo](http://flash.aasir.com/foo)，結果轉向 Redirect 到 <http://aasir.com> 去了。



這是轉向的地方 <http://aasir.com>。

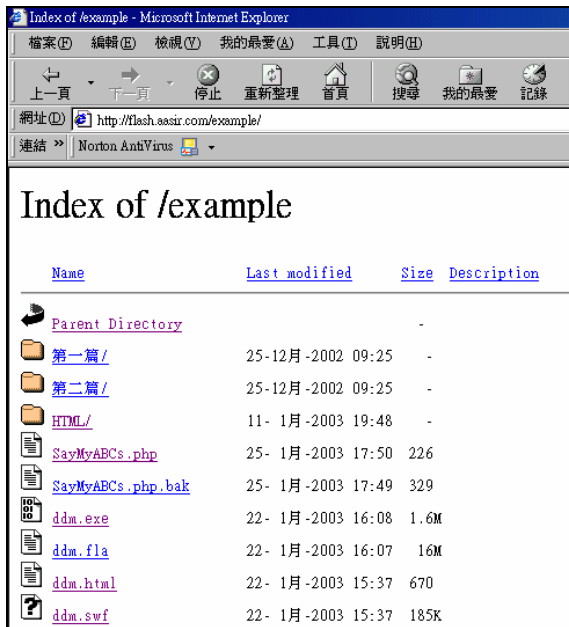


## <<18>>IndexOptions 定義索引

我們可以在 Apache 伺服器設定每一種檔案都有一種圖示。我們可以在 IndexOptions 指令設定 FancyIndexing(圖示索引), 而且 AddIconByEncoding、AddIconByType 和 AddIcon 指令應該被使用。

IndexOptions 指令控制伺服器產生的目錄串列顯示。FancyIndexing 是的選項是讓我們有標準的目錄索引(使用圖示索引)。在先前的選項區塊是和伺服器產生目錄索引有關。IndexOptions FancyIndexing VersionSort NameWidth=\*指令啟動一個自動產生目錄索引來包含每個目錄或檔案的資訊, 來包含圖示來表示檔案的類型、檔案名稱、最後更新的日期、檔案的大小和檔案的描述。VersionSort 選項允許檔案包含按照大小排列的版本編號(版本 2 會排在版本 10 的前面)。NameWidth=\*選項會允許任何檔案名稱長度的檔案名稱顯示。我們可以將\*改變成適度大小的數字, 來表示該檔案名稱字元的長度。假如 IndexOptions 沒有設定 FancyIndexing, 則只有檔案名稱會出現在索引上。

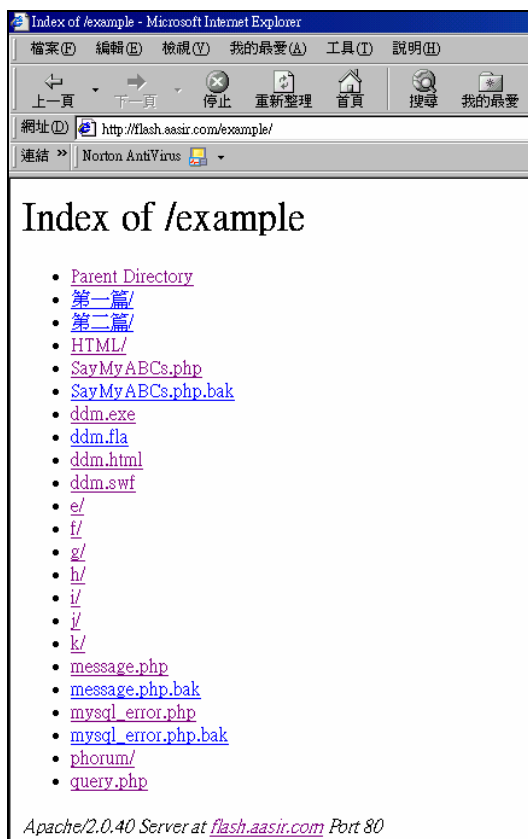
599 IndexOptions FancyIndexing VersionSort NameWidth=\*  
在這裏出現圖示、檔案名稱、最後修改日期、檔案大小和描述。



如果我們將 IndexOptions 選項的 FancyIndexing 去除，索引只出現檔案或目錄名稱。

## 599 IndexOptions VersionSort NameWidth=\*

圖示不見了，這是索引只出現檔案或目錄名稱。



### <<19>>AddIcon\*指令

AddIcon\*指令告訴伺服器什麼檔案類型顯示該檔案類型的圖示。這是只有用在 FancyIndexed 選項的索引模式目錄。

AddIconByEncoding 指令用來組態 FancyIndexing 指令的輸出。



AddIconByEncoding 強迫特殊的圖示來顯示符合特殊 MIME 資料編碼的檔案。在第 606 行/icons/compressed.gif 將用來顯示 x-compress 和 x-gzip 的資料編碼檔。AddIconByType 指令擁有相同的語法，將符合的 MIME 資料檔以適當的圖示表示。在 608 行將以/icons/text.gif 圖示來顯示文字檔。

AddIcon 指令執行相類似的功能，但是圖示是根據檔案名稱的範本來顯示。因此 AddIcon /icons/binary.gif .bin .exe 設定執行檔.bin、.exe(檔案名稱後面是\*.bin 或\*.exe)是以/icons/binary.gif 的圖示來顯示。.Z、.z、.tgz、.gz、.zip 壓縮檔是以/icons/compressed.gif 圖示來顯示。

假如 AddIcon、AddIconByEncoding 或 AddIconByType 指令有衝突，則以 AddIcon 指令為優先。

```
606 AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
607
608 AddIconByType (TXT,/icons/text.gif) text/*
609 AddIconByType (IMG,/icons/image2.gif) image/*
610 AddIconByType (SND,/icons/sound2.gif) audio/*
611 AddIconByType (VID,/icons/movie.gif) video/*
612
613 AddIcon /icons/binary.gif .bin .exe
614 AddIcon /icons/binhex.gif .hqx
615 AddIcon /icons/tar.gif .tar
616 AddIcon /icons/world2.gif .wrl .wrl.gz .vrm .vrm .iv
617 AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
618 AddIcon /icons/a.gif .ps .ai .eps
619 AddIcon /icons/layout.gif .html .shtml .htm .pdf
620 AddIcon /icons/text.gif .txt
621 AddIcon /icons/c.gif .c
622 AddIcon /icons/p.gif .pl .py
623 AddIcon /icons/f.gif .for
624 AddIcon /icons/dvi.gif .dvi
625 AddIcon /icons/uuencoded.gif .uu
626 AddIcon /icons/script.gif .conf .sh .shar .esh .ksh .tcl
627 AddIcon /icons/tex.gif .tex
628 AddIcon /icons/bomb.gif core

630 AddIcon /icons/back.gif ..
631 AddIcon /icons/hand.right.gif README
632 AddIcon /icons/folder.gif ^^DIRECTORY^^
633 AddIcon /icons/blank.gif ^^BLANKICON^^
```

這是.tar.gz 的壓縮檔圖示/icons/compress.gif。



[phorum-3.3.2c.tar.gz](#)

09- 1月 -2003 08:34 113K





這是執行檔的圖示/icons/binary.gif。



[ddm.exe](#)

22- 1月-2003 16:08 1.6M

這是網頁\*.html 的圖示/icons/layout.gif。



[ddm.html](#)

這是圖形\*.gif 的圖示



[account.gif](#)

/icons/blank.gif 的圖示是自動插入自動產生的串列來保留適當的格式。

假如有特別還未定義的檔案名稱，則使用 DefaultIcon 指令來指定圖形  
/icons/unknown.gif。

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

ddm.swf 檔是/icons/unknown.gif 的圖示，因為伺服器未定義\*.swf 的檔案。



[ddm.swf](#)

AddDescription 指令允許我們在伺服器產生的索引檔案後面放置簡單的描述，這是只有在 FancyIndexed 指令有作用下才會產生。

```
647 #AddDescription "GZIP compressed document" .gz
648 #AddDescription "tar archive" .tar
649 #AddDescription "GZIP compressed tar archive" .tgz
```

如果我們將第 647 行的註解給拿掉，並且在後面寫上文件壓縮檔，則伺服器產生的\*.tar.gz 檔案後面會有"GZIP 文件壓縮檔的描述"。

```
647 AddDescription "GZIP 文件壓縮檔" .gz
648 #AddDescription "tar archive" .tar
649 #AddDescription "GZIP compressed tar archive" .tgz
```



[phorum-3.3.2c.tar.gz](#)

09- 1月-2003 08:34 113K GZIP 文件壓縮檔



ReadmeName 指令會把 README.html 加到檔案串列的後面。

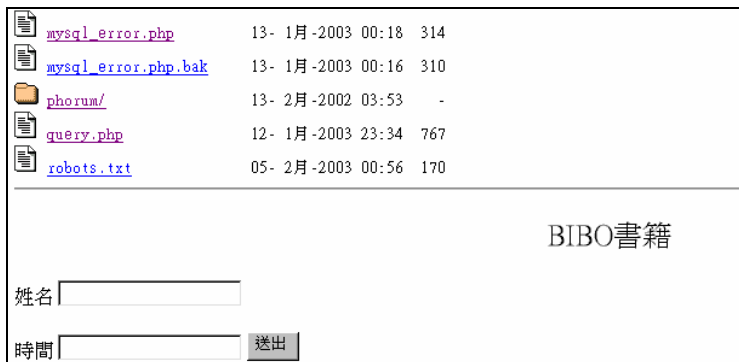
HeaderName 指令會把 HEADER.html 加到目錄索引的前面。

```
657 ReadmeName README.html
658 HeaderName HEADER.html
```

HeaderName 指令會把 HEADER.html 加到目錄索引的前面。阿瑪孫是 HEADER.html 被加到目錄索引的前面。



ReadmeName 指令會把 README.html 加到檔案串列的後面。BIBO 書籍為 README.html 被加到檔案串列的後面。



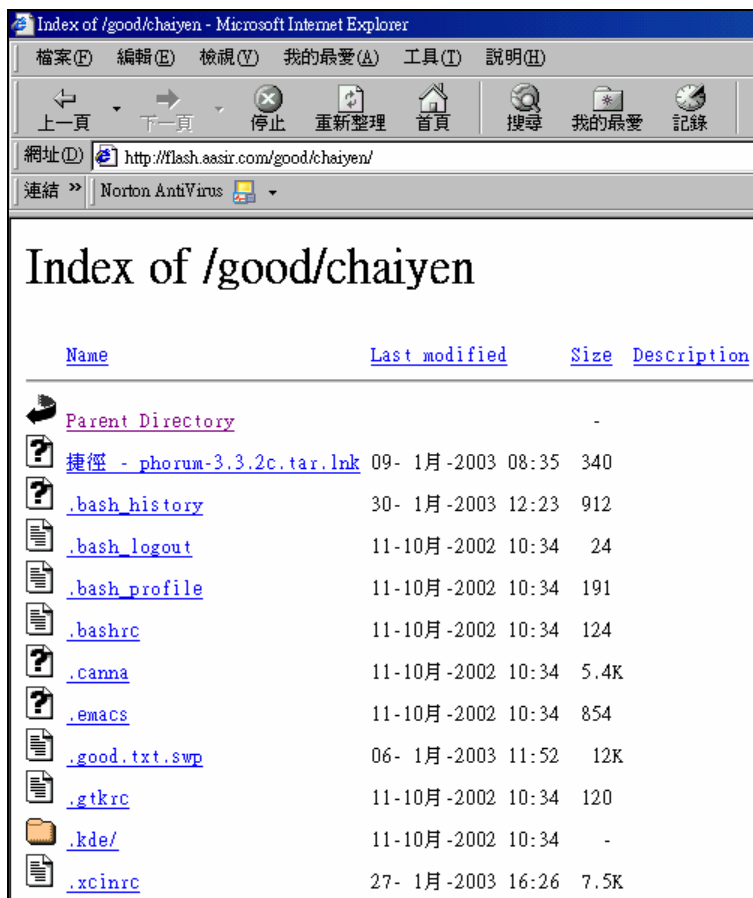
IndexIgnore 指令指定不應該出現在索引目錄的檔案。這些檔案包括由逗號開始且置少有兩個字元的檔案、以\*~符號或以\*#為結尾的檔案、目錄串列以 HEADER 或 README 為啟始的檔案和 RCS(修正控制系統)目錄。













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

如果我們將 IndexIgnore 指令加上註解，則 IndexIgnore 指令會沒有作用，因此當我們觀看使用者家目錄時，很多隱藏檔都會顯示出來。

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

.bash\_history、.bashrc 和 .bash\_profile 為我們使用者 chaiyen 的隱藏檔，.xcinrc 為中文輸入法的隱藏檔。



Name	Last modified	Size	Description
 <a href="#">Parent Directory</a>		-	
 <a href="#">捷徑 - phorum-3.3.2c.tar.lnk</a>	09- 1月 -2003 08:35	340	
 <a href="#">.bash_history</a>	30- 1月 -2003 12:23	912	
 <a href="#">.bash_logout</a>	11-10月 -2002 10:34	24	
 <a href="#">.bash_profile</a>	11-10月 -2002 10:34	191	
 <a href="#">.bashrc</a>	11-10月 -2002 10:34	124	
 <a href="#">.camna</a>	11-10月 -2002 10:34	5.4K	
 <a href="#">.emacs</a>	11-10月 -2002 10:34	854	
 <a href="#">.good.txt.swp</a>	06- 1月 -2003 11:52	12K	
 <a href="#">.gtkrc</a>	11-10月 -2002 10:34	120	
 <a href="#">.kde/</a>	11-10月 -2002 10:34	-	
 <a href="#">.xcinrc</a>	27- 1月 -2003 16:26	7.5K	

## &lt;&lt;20&gt;&gt; 定義語言 AddLanguage 和編碼 AddEncoding

AddEncoding 指令可以讓我們設定被瀏覽器使用來編碼資料的壓縮定義。

AddEncoding 指令補充或覆寫由 TypesConfig 檔所提供的對應。了解 MIME 的編碼可以允許瀏覽器管理被下載的這些 MIME 資料檔案。

```
672 AddEncoding x-compress Z
673 AddEncoding x-gzip gz tgz
```

DefaultLanguage 指令可以讓我們指定文件的語言。如果該文件沒有定義任何預設的語言，則當我們在 DefaultLanguage 指令設定為 tw 時，其文件的語言會以繁體中文出現(tw 為台灣)。

```
689 DefaultLanguage tw
```

AddLanguage 指令讓我們指定文件的語言，並且加在檔案的後面。

第 672 行到第 737 行是設定指定文件的語言國別。第 735 行第 736 行是將繁體中文給放入。

```
735 AddLanguage tw .tw
736 AddLanguage zh-tw .tw
```

LanguagePriority 是設定語言的優先權。我們可以將 tw 放在前面。

```
746 LanguagePriority tw en da nl et fr de el it ja kr no pl pt pt-br ltz
```

## &lt;&lt;21&gt;&gt; AddDefaultCharset 指令

AddDefaultCharset 指令設定網頁預設字元集。AddCharset 指令增加字元集。

AddDefaultCharset 指定伺服器送出的網頁預設字元集。如果我們設定以 iso-8859-1(拉丁字)字元集，則網頁送出時會以拉丁字的字元編碼出現，如果 AddDefaultCharset 指定以 Big5(繁體中文)字元集，則網頁送出時會以繁體中文的字元編碼出現。所以我們可以使用 AddDefaultcharset 來指定我們網站的預設字體。

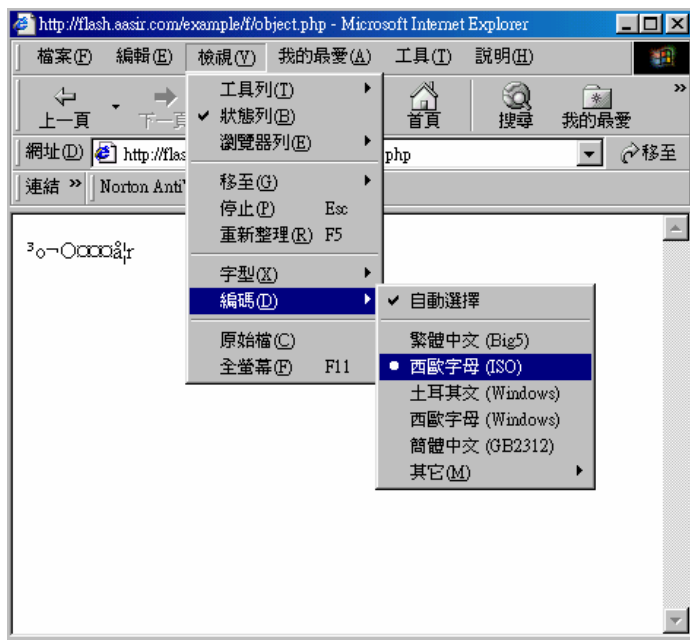
```
765 AddDefaultCharset iso-8859-1
```



這是 object.php 的網頁，我們沒有指定網頁文件的語言。

```
1 <?php
2 echo "這是中文字";
3 ?>
```

如果我們 AddDefaultCharset 指令使用 iso-8859-1 字元集，這裏 object.php 網頁會顯示拉丁文的亂碼，並且在瀏覽器上的檢視->編碼會出現西歐字母。

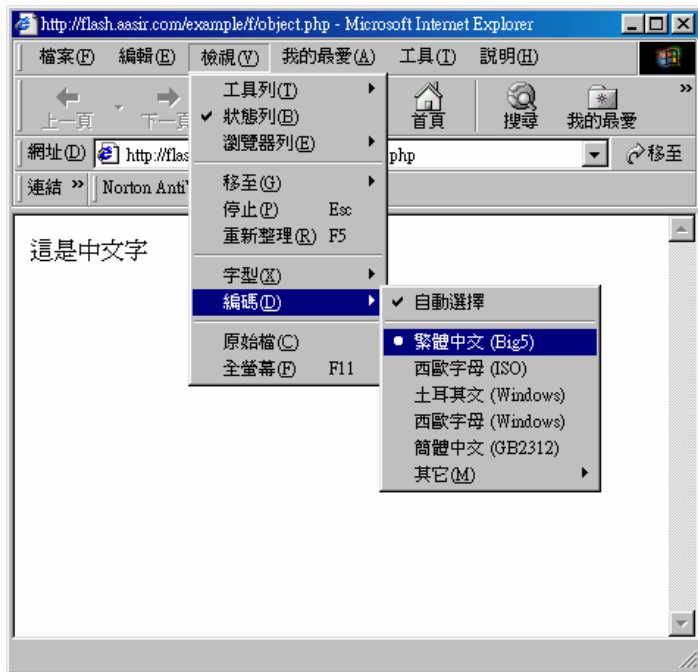


第 765 行是設定瀏覽器的編碼為繁體中文(Big 5)。

## 765 AddDefaultCharset Big5

如果我們 AddDefaultCharset 指令使用 Big5(大五碼)字元集，這裏 object.php 網頁會顯示中文字，並且在瀏覽器上的檢視->編碼會出現繁體中文。因此，如果我們的網站是繁體中文，則在 AddDefaultCharset 指令設定字元集時，請設定 Big5 大五碼。



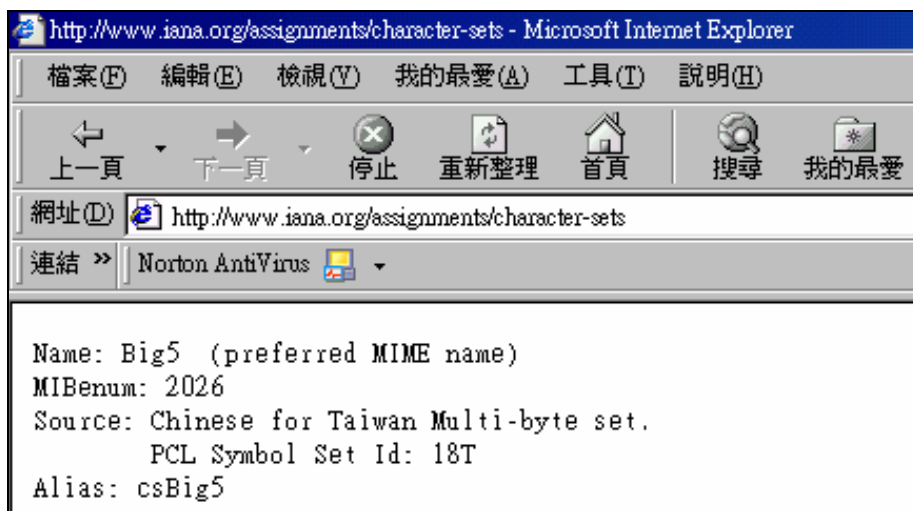


我們可以在第 786 行使用 AddCharset 指令增加 Big5 大五碼(繁體中文)字元集。  
第 774 行是設定增加拉丁字元集。

```

774 AddCharset ISO-8859-1 .iso8859-1 .latin1
775 AddCharset ISO-8859-2 .iso8859-2 .latin2 .cen
776 AddCharset ISO-8859-3 .iso8859-3 .latin3
777 AddCharset ISO-8859-4 .iso8859-4 .latin4
778 AddCharset ISO-8859-5 .iso8859-5 .latin5 .cyr .iso-ru
779 AddCharset ISO-8859-6 .iso8859-6 .latin6 .arb
780 AddCharset ISO-8859-7 .iso8859-7 .latin7 .grk
781 AddCharset ISO-8859-8 .iso8859-8 .latin8 .heb
782 AddCharset ISO-8859-9 .iso8859-9 .latin9 .trk
783 AddCharset ISO-2022-JP .iso2022-jp .jis
784 AddCharset ISO-2022-KR .iso2022-kr .kis
785 AddCharset ISO-2022-CN .iso2022-cn .cis
786 AddCharset Big5 .Big5 .big5
  
```

我們可以在 <http://www.iana.org/assignments/character-sets> 找到 Big5 大五碼的字元集。



## <<22>>AddType、AddHandler 指令

AddType 指令設定指定的檔案對應到 MIME 形態。

AddType 指令允許將指定的檔案形態增加或覆寫到 MIME 組態檔 `mime.types`。

\*.tgz 的檔案形態對應到 `application/x-tar` 為壓縮檔。

```
817 AddType application/x-tar .tgz
```

AddHandler 指令允許我們對應指定的檔案形態到指定處理函式。

第 827 行 AddHandler `cgi-script .cgi` 我們可以在非程式目錄執行 CGI 的檔案(我們也要在該目錄增加"ExecCGI"到 Options 指令)。

```
827 #AddHandler cgi-script .cgi
```

AddHandler `send-as-is asis` 讓檔案上增加自己的 HTTP 標頭。



```
832 #AddHandler send-as-is asis
```

AddHandler imap-file map 增加分析伺服器的圖形對應檔。

```
837 AddHandler imap-file map
```

AddHandler type var 允許 Apache 伺服器以多種語言來顯示其網頁，預設是啟動這項功能。

```
844 AddHandler type-map var
```

Addoutputfilter 指令允許我們在送出內容到使用者端時先將它處理並過濾(我們也要在該目錄增加"Includes"到 Options 指令。

```
851 AddOutputFilter INCLUDES .shtml
```

在/etc/mime.types 的 mime.types 檔案根據副檔名來控制傳送到使用者的網路媒體形態。傳送正確的檔案形態到使用者端是很重要的，因為使用者端的瀏覽器會依據檔案形態來執行相關的應用程式。我們除了可以使用 AddType 指令來增加額外的檔案形態，也可以直接在/etc/mime.types 的檔案設定該形態。

```
[root@flash /]# vi /etc/mime.types
```

### <<23>> Action 指令

Action 指令設定指定檔案媒體執行 CGI 程式。有一些媒體的類型當他們被打開時可以被特定程式執行。有一些處理函數當被打開時，可以設定執行特定的程式。Action 指令可以用來組態這一些程式。Action 指令將 MIME 形態或處理函式對應到特定 CGI 程式。這可以減輕我們經常使用重複的路徑來執行 CGI 檔案。

Action media/type /cgi 程式位置。

Action handler-name /cgi 程式位置。

```
857 # Format: Action media/type /cgi-script/location  
858 # Format: Action handler-name /cgi-script/location
```





我們也可以設定新的媒體檔案形態，我們再使用我們設定的函式來處理它。

我們先使用 `AddType` 指令來設定副檔名為 `.mytext` 的檔案為 `text/my_text_type` 檔案形態。我們再使用 `Action` 指令設定 `text/my_text_type` 檔案形態的處理函式路徑為 `/cgi-bin/my_text_type_handler`。

```
AddType text/my_text_type .mytext
Action text/my_text_type /cgi-bin/my_text_type_handler
```

`Action` 指令設定 `image/gif` 檔案內容的 CGI 程式為 `/cgi-bin/images.cgi`。

```
Action image/gif /cgi-bin/images.cgi
```

`AddHandler` 指令設定 `.xyz` 為 `my-file-type` 檔案形態，`Action` 指令設定 `my-file-type` 的執行程式為 `/cgi-bin/program.cgi`。

```
AddHandler my-file-type .xyz
Action my-file-type /cgi-bin/program.cgi
```

## <<24>>ErrorDocument 指令

對於錯誤的情況發生時，`ErrorDocument` 指令可以設定伺服器回應特定的資訊給使用者端。這個回應可以是完整文字、設定的訊息、轉到伺服器內部的網頁或是轉到外部的網頁來處理錯誤的訊息。

```
866 #ErrorDocument 500 "The server made a boo boo."
867 #ErrorDocument 404 /missing.html
868 #ErrorDocument 404 "/cgi-bin/missing_handler.pl"
869 #ErrorDocument 402 http://www.example.com/subscription\_info.html
```

如同第 866 行到第 869 行註解的建議，`ErrorDocument` 指令能定作任何伺服器的回應、轉向到外部的網頁、轉向到內部的檔案、簡單的文字語句或是 CGI 的程式。



這是錯誤發生時的回應碼及相關說明。

回應碼	說明
200 Ok	這個要求服務成功的被處理。
201 Created	使用 POST 要求方法，一個新的檔案成功的儲存在伺服器上。
202 Accepted	這個要求已經被接收，而且目前正在處理它。
204 No Content	這個要求已經成功，但是目前顯示給使用者的網頁並沒有改變。
301 Moved Permanently	這被要求服務的網頁已經被移動，以後參考這網頁應該使用新的 URL(資源定址)位址來顯示。
302 Moved Temporarily	這被要求服務的網頁已經暫時被移動，以後參考這網頁應該繼續使用同樣的 URL(資源定址)位址，但是目前的連接已經被轉向。
304 Not Modified	緩衝版本的網頁和被要求的網頁是相同的。
400 Bad Request	使用者端的要求服務包含不正確的語法。
401 Unauthorized	使用者端使用未經證實的使用者名稱和密碼組合。
402 Payment Required	使用者必需提供完成貨幣交易的方法。
403 Forbidden	存取控制機制拒絕使用者端的要求。
404 Not Found	被要求的網頁不存在於這個伺服器上。
500 Internal Server Error	當執行 CGI 程式發生錯誤，這個回應碼指出這程式包含不正確的程式碼或者是它無法處理的輸入。
501 Not Implemented	這個要求方法不被伺服器了解。這些方法包括 GET、POST、PUT、DELETE 和 HEAD)
502 Bad Gateway	使用網站當代理伺服器，當在執行要求服務到外部主機時錯誤發生。
503 Service Unavailable	這個伺服器目前正在處理太多的要求服務。
505 HTTP Version Not Supported	這個要求服務版本(HTTP/1.0 或 HTTP/1.1)無法被伺服器了解。

我們可以使用 Alias 指令轉向任何/error/HTTP\_<error>.html.var 回應到多樣語言的錯誤資訊集合。我們使用 includes 來代替適當的文字。

我們可以藉著 Alias /error/include/ "/your/include/path/"來修改顯示資訊而不用更



改任何預設的 HTTP\_<error>.html.var 檔案。Alias 指令允許我們建立我們自己的檔案設定(藉由/var/www/error/include/檔案開始)和複製它們到/your/include/path/。在虛擬主機裏，Alias 指令也允許我們這樣作。

```
882 # Alias /error/include/ "/your/include/path/"
```

為了讓錯誤的訊息讓大家都看得懂和標準化這些回應的訊息，Apache 包含了這些不同的網頁(以.var 為結尾)。這些不同的網頁提供以依據語言為變數輸出，並且存在於/var/www/error 目錄中。

這 ErrorDocument 指令結合特別的錯誤程式碼編號和指定的.var 檔，而這些.var 檔是依據語言包含多樣可能的回應。

```
889 Alias /error/ "/var/www/error/"
890
891 <IfModule mod_negotiation.c>
892 <IfModule mod_include.c>
893     <Directory "/var/www/error">
894         AllowOverride None
895         Options IncludesNoExec
896         AddOutputFilter Includes html
897         AddHandler type-map var
898         Order allow,deny
899         Allow from all
900         LanguagePriority en es de fr
901         ForceLanguagePriority Prefer Fallback
902     </Directory>
903
904     ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
905     ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
906     ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
907     ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
908     ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
909     ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
910     ErrorDocument 410 /error/HTTP_GONE.html.var
```

第 911 行的/error/HTTP\_LENGTH\_REQUIRED.html.var 是對應到 /var/www/error/HTTP\_LENGTH\_REQUIRED.html.var。因為第 889 行 Alias 指令將 /error/對應到/var/www/error。



```

911 ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
912 ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
913 ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
914 ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
915 ErrorDocument 415 /error/HTTP_SERVICE_UNAVAILABLE.html.var
916 ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
917 ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
918 ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
919 ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
920 ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var

```

這是在/var/www/error/目錄下的錯誤回應資訊檔。

```

[root@flash error]# ls
contact.html.var          HTTP_PRECONDITION_FAILED.html.var
HTTP_BAD_GATEWAY.html.var HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
HTTP_BAD_REQUEST.html.var HTTP_REQUEST_TIME_OUT.html.var
HTTP_FORBIDDEN.html.var  HTTP_REQUEST_URI_TOO_LARGE.html.var
HTTP_GONE.html.var       HTTP_SERVICE_UNAVAILABLE.html.var
HTTP_INTERNAL_SERVER_ERROR.html.var HTTP_UNAUTHORIZED.html.var
HTTP_LENGTH_REQUIRED.html.var HTTP_UNSUPPORTED_MEDIA_TYPE.html.var
HTTP_METHOD_NOT_ALLOWED.html.var HTTP_VARIANT_ALSO_VARIES.html.var
HTTP_NOT_ACCEPTABLE.html.var include
HTTP_NOT_FOUND.html.var  noindex.html
HTTP_NOT_IMPLEMENTED.html.var README

```

存取控制機制拒絕使用者端的要求(因為使用者沒有讀取該目錄的權限)，因此出現 Error 403 的錯誤編號。這是第 906 行將錯誤網頁導向 /var/www/error/HTTP\_FORBIDDEN.html.var 檔案。



被要求的網頁 `kk.html` 不存在於這個伺服器上，因此出現 `Error404` 的編碼錯誤。這是第 907 行 `ErrorDocument` 指令將其導向到

`/var /error/HTTP_NOT_FOUND.html.var` 的檔案。



### <<25>>BrowserMatch 指令設定給瀏覽器的回應

使用 `BrowserMatch` 指令，我們可以設定根據不同的瀏覽器類型來回應。`BrowserMatch` 指令修改一般 HTTP 回應方式來處理瀏覽器已知的問題。

第 929 行在小恐龍瀏覽器上停用 `KeepAlive` 的功能。

第 930 行在微軟瀏覽器 4 的版本上停用 `Keepalive` 的功能，並且強迫使用 `HTTP1.0` 的回應。



```
929 BrowserMatch "Mozilla/2" nokeepalive
930 BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
931 BrowserMatch "RealPlayer 4\.0" force-response-1.0
932 BrowserMatch "Java/1\.0" force-response-1.0
933 BrowserMatch "JDK/1\.0" force-response-1.0
```

第 941 行和第 942 行對於 non-Get 要求目錄服務(該目錄後面沒有接 slash)取消轉向。這修正微軟網路目錄不適用於使用 DAV 方法處理目錄的轉向。

```
941 BrowserMatch "Microsoft Data Access Internet Publishing Provider" redirect-carefully
942 BrowserMatch "^WebDrive" redirect-carefully
```

## <<26>> 監看伺服器的活動 Server-info 和 Server-status 函式

Apache 提供兩種獨特的內件方法來檢查我們伺服器的表現。Server-status 處理函式可以用來顯示伺服器行程的資訊。Server-info 可以用來顯示伺服器組態詳細的資訊。

Server-status 允許伺服器作狀態的報告。我們在第 951 行設定允許 61.218.29.0/255.255.255.248 的網域來關看我們伺服器的狀態，如果要關閉這項功能，我們只要將第 948 行到第 952 行加上註解就可以了。

```
948 <Location /server.status.content>
949     SetHandler server-status
950     Order deny,allow
951     deny from all
952     allow from 61.218.29.0/255.255.255.248
953 </Location>
```

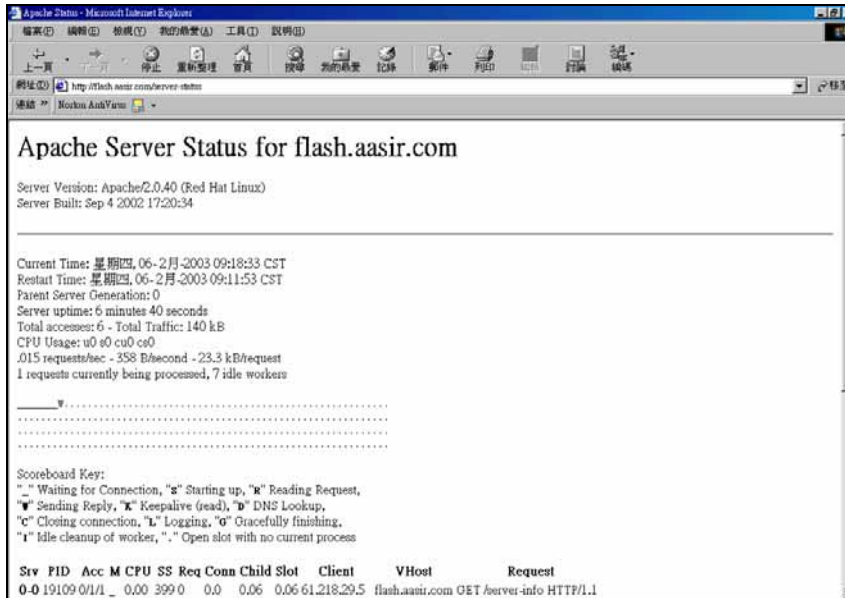
Server-info 的功能在使用前需先載入 mod\_info.c 的模組。Server-info 可以讓我們了解伺服器的組態報告。第 962 行是設定允許 61.218.29.0/255.255.255.248 的網域觀看。

```
960 <Location /server.info.content>
961     SetHandler server-info
962     Order deny,allow
963     deny from all
964     allow from 61.218.29.0/255.255.255.248
965 </Location>
```





我們輸入 `http://flash.aasir.com/server-status` 就可以觀看到 Apache 伺服器的狀態資訊。



因為 `server-status` 和 `server-info` 包含我們網站許多的重要資訊，為了防止駭客經由此項資訊來侵入我們的網站，我們可以將 `<Location /server-status>` 和 `<Location/server-info>` 作名稱的修改，這樣我們就可以用其它的名稱來存取 `server-status` 和 `server-info` 模組。例如我們使用 `server.status.content` 網頁和 `server.info.content` 網頁來存取這些資訊。Allow from 61.218.29.0/255.255.255.248 的 255.255.255.248 為網路遮罩，它可以設定區段網域的允許權。

```
948 <Location /server.status.content>
949     SetHandler server-status
950     Order deny,allow
951     deny from all
952     allow from 61.218.29.0/255.255.255.248
953 </Location>
```

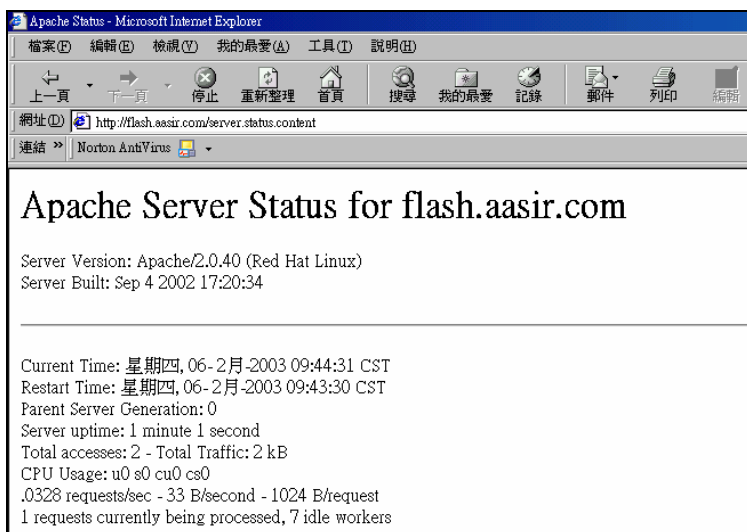


```

960 <Location /server.info.content>
961     SetHandler server-info
962     Order deny,allow
963     deny from all
964     allow from 61.218.29.0/255.255.255.248
965 </Location>

```

我們輸入 `http://flash.aasir.com/server.status.content`，則可以看到 Apache 伺服器的狀態資訊。Apache 狀態資訊包含 Apache 版本的資訊、伺服器建立的日期、目前的时间、連線的數量、CPU 使用率、和伺服器更新的时间。



Apache Status - Microsoft Internet Explorer

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

地址(D) http://flash.aasir.com/server.status.content

連結(N) Norton AntiVirus

## Apache Server Status for flash.aasir.com

Server Version: Apache/2.0.40 (Red Hat Linux)  
Server Built: Sep 4 2002 17:20:34

---

Current Time: 星期四, 06-2月-2003 09:44:31 CST  
Restart Time: 星期四, 06-2月-2003 09:43:30 CST  
Parent Server Generation: 0  
Server uptime: 1 minute 1 second  
Total accesses: 2 - Total Traffic: 2 kB  
CPU Usage: u0 s0 cu0 cs0  
.0328 requests/sec - 33 B/second - 1024 B/request  
1 requests currently being processed, 7 idle workers

Srv 是子伺服器的編號。

PID 是行程的編號(Process ID Number)。

Acc 是這個連接存取的數量。

CPU 是 CPU 的使用率，單位為使用多少秒。

SS 是從開始要求服務到現在已經過了幾秒。

Reques 是要求的服務。

Client 是使用者端的 IP 位址。



Scoreboard Key:

"\_" Waiting for Connection, "s" Starting up, "r" Reading Request,  
 "w" Sending Reply, "k" Keepalive (read), "d" DNS Lookup,  
 "c" Closing connection, "L" Logging, "G" Gracefully finishing,  
 "I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	19237	0/1/1	_	0.00	58	0	0.0	0.00	0.00	61.218.29.5	flash.aasir.com	GET /server-status HTTP/1.1
1-0	19238	0/1/1	_	0.00	50	0	0.0	0.00	0.00	61.218.29.5	flash.aasir.com	GET /server.status.content HTTP/1.1
3-0	19240	0/0/0	W	0.00	0	0	0.0	0.00	0.00	61.218.29.5	flash.aasir.com	GET /server.status.content HTTP/1.1

**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

## <<27>>ProxyRequests 指令啟動代理和緩充服務

Apache 可以提供代理伺服器或快取的功能。假如 ProxyRequests 指令被設為 Off，伺服器將只提供快取。假如 CacheRoot 指令沒有被定義，或者 NoCache 指令被設成\*，則快取將不會被執行。Apache 預設將代理和快取的功能取消，如果要讓它們運行，則將其前面的註解#移除。在使用快取之前，先要設定載入 mod\_cache.so 快取模組，一般是要自行設定(預設是沒有載入)。

假如我們啟動快取，則 CacheRoot 指令是指定放置快取檔案的地方，因此要有足夠大的地方來放置這些檔案。我們可以使用 CacheSize 指令來設定快取空間的大小(預設是 5K)。我們使用 CacheGcInterval 指令定期的清除快取檔案，因此我們在檔案系統需要額外的 20%到 40%的空間來定時清除快取。

CacheMaxExpire 指令指示文件最多可以儲存在快取的時間，單位是小時。

CacheRoot 是設定快速緩充暫存的目錄，預設是在/etc/httpd/proxy 目錄。

CacheLastModifiedFactor 讓文件可以在沒有設定期限來使用快取。

CacheLastModifiedFactor 指令介著指定比率來設定文件儲存的期限，而這比率是依據文件被修改的時間。例如，假如這文件在 5 天前(120)被修改，而 CacheLastModifiedFactor 為 0.1，則這文件儲存在快取的時間將在 12 個小時後期滿(120\*0.1=12)。

CacheDefaultExpire 指令指定文件儲存在快取將在幾個小時後期滿，而 CacheDefaultExpire 指令可以運用在沒有支援期限的協定上。

NoCache 指令可以設定沒有快取的網站名稱、IP 位址、主機名稱和關鍵字。

```
969 #<IfModule mod_proxy.c>
970 #ProxyRequests On
971 #
972 #<Proxy *>
973 #   Order deny,allow
974 #   Deny from all
975 #   Allow from .your-domain.com
976 #</Proxy>
977
978 #
979 # Enable/disable the handling of HTTP/1.1 "Via:" headers.
980 # ("Full" adds the server version; "Block" removes all outgoing Via: headers)
981 # Set to one of: Off | On | Full | Block
982 #
983 #ProxyVia On
984
985 #
986 # To enable the cache as well, edit and uncomment the following lines:
987 # (no cacheing without CacheRoot)
988 #
989 #CacheRoot "/etc/httpd/proxy"
990 #CacheSize 5
991 #CacheGcInterval 4
992 #CacheMaxExpire 24
993 #CacheLastModifiedFactor 0.1
994 #CacheDefaultExpire 1
995 #NoCache a-domain.com another-domain.edu joes.garage-sale.com
996
997 #</IfModule>
```

如果我們將快取或代理伺服器的註解拿掉，再載入 mod\_cache.so 模組，就可以執行快取。快取檔案可以設定於 /etc/httpd/proxy，檔案的儲存期限(CacheMaxExpire) 是 24 個小時。每四個小時伺服器會執行清除快取(CacheGcInterval)，來確保快取的檔案大小少於 5K(CacheSize)。

從 anonymous FTP 來的文件雖不支援期限協定，但也可以儲存 1 個小時(CacheDefaultExpire)。HTTP 支援期限，但 FTP 不支援。

如果文件沒有期限時間，我們將使用 CacheLastModifiedFactor 指令來決定文件儲存的時間(CacheDefaultExpire\*CacheLastModifiedFactor=1\*0.1=0.1)，預設是 0.1 個小時。

NoCache 指令限制 a-domain.com、another-domain.edu 和 joes.garage-sale.com 網域的文件，將不被快取儲存。



## &lt;&lt;28&gt;&gt;授權機制 Auth

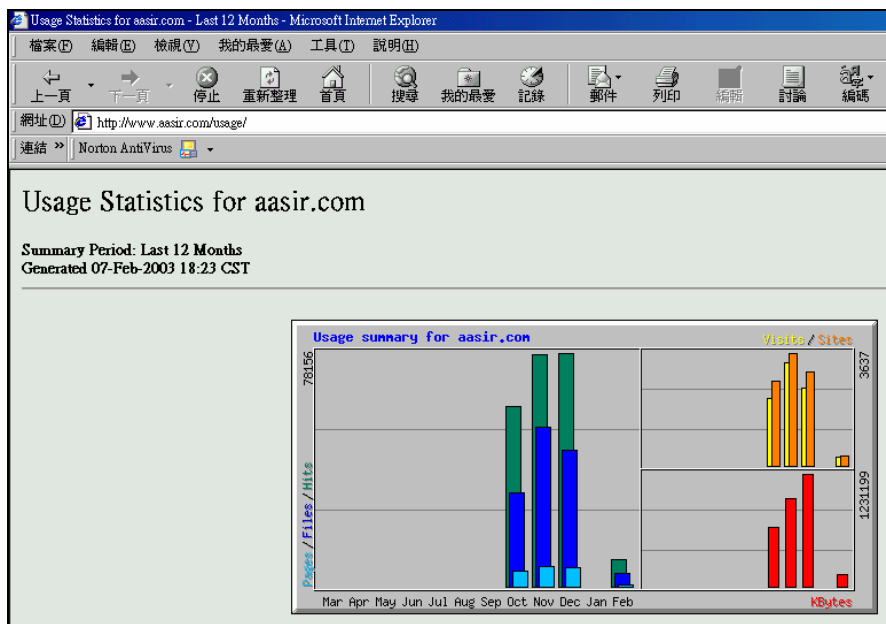
當我們要進入網站的特定目錄時，因為網站安全上的要求，會需要我們輸入特定的使用者名稱和密碼才能進入該網站的特定目錄。例如當我們要進入 www.aasir.com 網站的/usage/目錄時，因為該目錄限制特定的使用者才能使用，所以會跳出輸入網路密碼的視窗。



我們輸入我們使用者名稱 webmaster，還有輸入 webmaster 的密碼。



這樣我們就進入了統計網站流量的目錄了。



htpasswd 指令可以設定特定使用者的密碼，並且將該密碼放置到指定的地方。

Apache 提供 htpasswd 工具來作基本檔案的授權，就像是我們 Unix 的 passwd 指令一樣。

```
#htpasswd password-file 使用者。
```

我們可以建立 webmaster 使用者的密碼檔 password.file，並將這個密碼檔放置到/home/chaiyen 的目錄下，我們可以使用下列的指令。

```
[root@aasir chaiyen]# htpasswd -c /home/chaiyen/password.file webmaster
New password:
Re-type new password:
Updating password for user webmaster
```

為了安全，htpasswd 指令沒有在螢幕上顯示密碼，因此要我們輸入兩次來確認。htpasswd 指令-c 的選項是要建立一個新的密碼檔。當這個密碼檔已經存在時，我們可以直接增加使用者，而不用-c 的參數。在這裏我們直接增加使用者 goodman。



```
[root@aasir chaiyen]# htpasswd /home/chaiyen/password.file goodman
New password:
Re-type new password:
Adding password for user goodman
```

Htpasswd 指令和參數的功能。

參數	功能
-d	使用 Crypt 加密，為 Unix 預設
-m	使用 MD5 加密
-s	使用 SHA 加密

我們設定授權 Apache 使用者使用/var/www/html/html/usage 目錄的權限為可讀和可執行。

```
[root@aasir chaiyen]# chmod 755 /var/www/html/html/usage
```

我們設定授權 Apache 使用者使用/home/chaiyen/password.file 檔案的權限為可讀取。

```
[root@aasir chaiyen]# chmod 644 /home/chaiyen/password.file
```

我們編輯 Apache 的組態檔 httpd.conf。

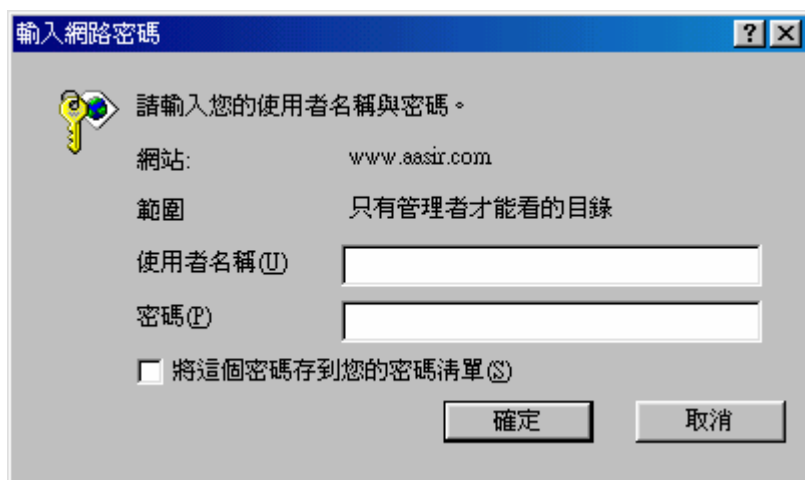
```
[root@aasir chaiyen]# vi /etc/httpd/conf/httpd.conf
```

我們只要再加上/var/www/html/html/usage 目錄的授權選項設定就可以了。

```
1034 <Directory /var/www/html/html/usage>
1035     AuthName "只有管理者才能看的目錄"
1036     AuthType basic
1037     AuthUserFile /home/chaiyen/password.file
1038     require user webmaster
1039     AuthAuthoritative on
1040 </Directory>
```



這就是網路授權要輸入使用者名稱和密碼。



這是我們設定安全目錄或地方，只有特定被授權的使用者或群組才能使用這個被保護區段的目錄或地方。

```
<Location /secure>
```

```
AuthName "realm 的名稱"
```

```
AuthType basic|Digest
```

```
AuthUserFile 到達被授權使用者的路徑
```

```
AuthGroupFile 到達被授權群組的路徑
```

```
require user user1 user2 group group1 group2
```

```
AuthAuthoritative on
```

```
</Location>
```

這是被保護區段的目錄。

```
<Directory /secure>
```

```
AuthName "realm 的名稱"
```

```
AuthType basic|Digest
```



AuthUserFile 到達被授權使用者檔案的路徑

AuthGroupFile 到達被授權群組檔案的路徑

```
require user user1 user2 group group1 group2
```

AuthAuthoritative on

```
</Directory>
```

這些檔案的授權都是由 mod\_auth 所提供。

AuthName 指令定義/secure 目錄的授權 realm 名稱。授權名稱使用""字串符號包起來。

AuthType 指令定義授權的機制，它的選項可以是 Basic 或者是 Digest。Basic 授權的機制是直接在網路上傳遞沒有加密的使用者名稱和密碼。而 Digest 則使用 MD5 將使用者名稱和密碼加密，所以比較安全。Basic 授權方式較為簡單，所以受到大家的使用。

AuthUserFile 指令指定被授權使用者檔案的路徑，例如我們指定密碼檔是放到/home/chaiyen/password.file。

AuthGroupFile 指令指定被授權群組檔案的路徑。

Require 指令設定在該保護的位置或資源可以存取的使用者，例如我們使用 require webmaster 來設定只有 webmaster 可以存取/var/www/html/html/usage 的目錄(webmaster 使用者的密碼也需正確)。

AuthAuthoritative 指令控制授權機制，當我們有多重的授權機制時，我們可以設定為 on 的選項，如果只是單一授全則使用 off 的選項。

## 23-4 設定虛擬主機組態指令

虛擬主機的設定可以分為以名稱為基礎的虛擬主機設定和以 IP 為基礎的虛擬主機設定。NameVirtualHost 指令是設定以名稱為基礎的虛擬主機設定。

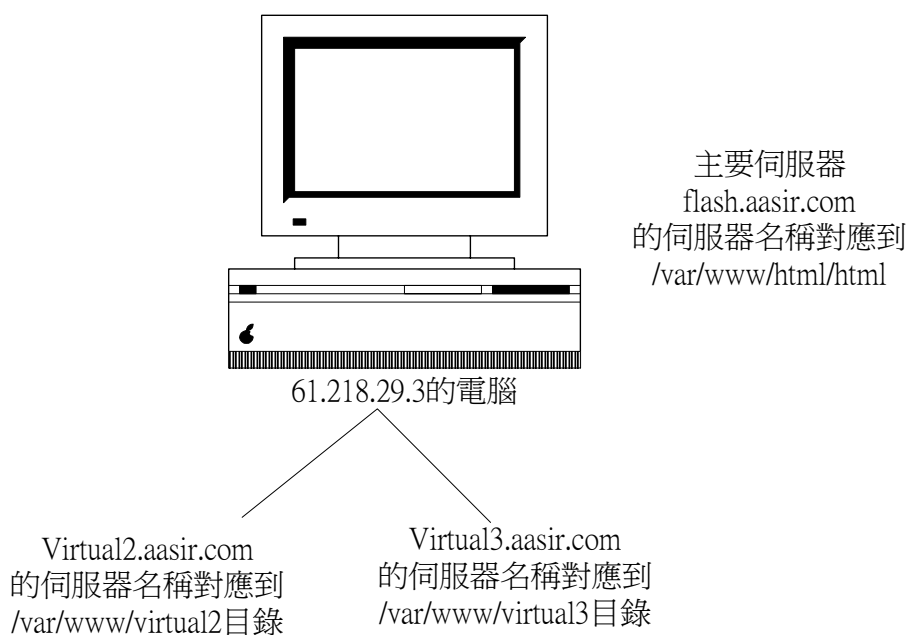
```
1019 #NameVirtualHost *
```





### 23-4-1 以名稱為基礎的虛擬主機設定

虛擬主機可以讓我們多個虛擬伺服器或網站共同使用一個 IP，來解決 IP 不足的問題。在 61.218.29.3 位址的電腦除了本身 flash.aasir.com 的網站(目錄是在 /var/www/html/html)，還有 Virtual2.aasir.com 的虛擬網站(其網站目錄是在 /var/www/virtual2)，還有 Virtual3.aasir.com 的虛擬網站(其網站目錄是在 /var/www/virtual3)。虛擬主機如果以名稱為基礎，相關的主機名稱對應到相關的目錄。



我們使用虛擬主機，當有一個瀏覽器的要求服務進來我們伺服器，Apache 會檢查網域名稱，並且將回應對應到特定的虛擬主機目錄去。假如我們網站要有提供虛擬主機的功能，則我們要設定相關的組態，這些組態包括網站內容、伺服器名稱、伺服器管理者、錯誤記錄檔和錯誤記錄檔放置的地方。我們將虛擬主機定義在 <VirtualHost>指令標籤內。

如果我們要在我們的機器上維持多個網域/主機名稱，我們可以設定虛擬主機來包含這些網域。大部份的組態都使用以名稱為主的虛擬主機，因此我們不需要擔心 IP



位置的問題。

在/etc/httpd/conf/httpd.conf 中組態的設定。

NameVirtualHost 指令是設定虛擬主機的 IP 位址。假如我們要設定以名稱為基礎的虛擬主機，NameVirtualHost 指令是必需的。雖然\*可以用主機名稱取代，但是最好還是用 IP 位址當作 NameVirtualHost 指令的參數。

```
1019 #NameVirtualHost *
```

我們可以指定以名稱為基礎的虛擬主機其連接埠的編號。在這裏我們設定 flash.aasir.com 其虛擬主機的位址與連接埠是 61.218.29.3:8080。

```
NameVirtualHost 61.218.29.3:8080
```

Ipv6 的位址形態必需封閉在括號裏面。

```
NameVirtualHost [fe80::a00:20ff:ccea]:8080
```

如果要在所有的介面接收要求服務，我們可以使用\*參數。

```
NameVirtualHost *
```

使用虛擬主機的參數方法。

```
NameVirtualHost 61.218.29.3
```

```
<VirtualHost 61.218.29.3>
```

```
#虛擬主機的組態設定
```

```
.....
```

```
</VirtualHost>
```

第 1027 行的\*可以讓我們設定 IP。

第 1028 行的 ServerAdmin 指令是設定該虛擬主機管理者的 E-mail。

第 1029 行的 DocumentRoot 指令是設定該虛擬主機存放網頁的地方。

第 1030 行的 ServerName 指令是設定該虛擬主機伺服器名稱。

第 1031 行的 ErrorLog 指令是設定該虛擬主機記錄檔的地方。



第 1032 行的 CustomLog 指令是設定虛擬主機一般的記錄檔。

```

1027 #<VirtualHost *>
1028 #   ServerAdmin webmaster@dummy-host.example.com
1029 #   DocumentRoot /www/docs/dummy-host.example.com
1030 #   ServerName dummy-host.example.com
1031 #   ErrorLog logs/dummy-host.example.com-error_log
1032 #   CustomLog logs/dummy-host.example.com-access_log common
1033 #</VirtualHost>

```

這是將 Virtual2.aasir.com 和 Virtual3.aasir.com 對應到同一台主機，主機的 IP 位址為 61.218.29.3。這是在 DNS 伺服器上的設定。

Hostname (w/o domain)	IP 位置
.aasir.com	61.218.29.2
flash.aasir.com	61.218.29.3
ftp.aasir.com	61.218.29.2
mail.aasir.com	61.218.29.2
mandrake.aasir.com	61.218.29.6
virtual2.aasir.com	61.218.29.3
virtual3.aasir.com	61.218.29.3
www.aasir.com	61.218.29.2
全球 DNS 正式生效約需 1-48 小時後， 打網址才會看到您的網站。	設定

NameVirtualHost 指令設定我們虛擬主機是以名稱為基礎，其 IP 位址是 61.218.29.3。如果是 IP 為基礎的虛擬主機則加上註解。

第一個虛擬主機的設定是在第 1 行到 8 行，第二個虛擬主機的設定是在 9 行到 15 行，第三個虛擬主機的設定是在第 16 行到第 22 行。

第一個 61.218.29.3 的虛擬主機其伺服器名稱為 flash.aasir.com(由 ServerName 指令設定)，其網站根目錄在/var/www/html/html。

第一個 61.218.29.3 的虛擬主機其伺服器名稱為 virtual2.aasir.com(由



ServerName 指令設定), 其網站根目錄在/var/www/virtual2。

第二個 61.218.29.3 的虛擬主機其伺服器名稱為 virtual3.aasir.com(由 ServerName 指令設定), 其網站根目錄在/var/www/virtual3。

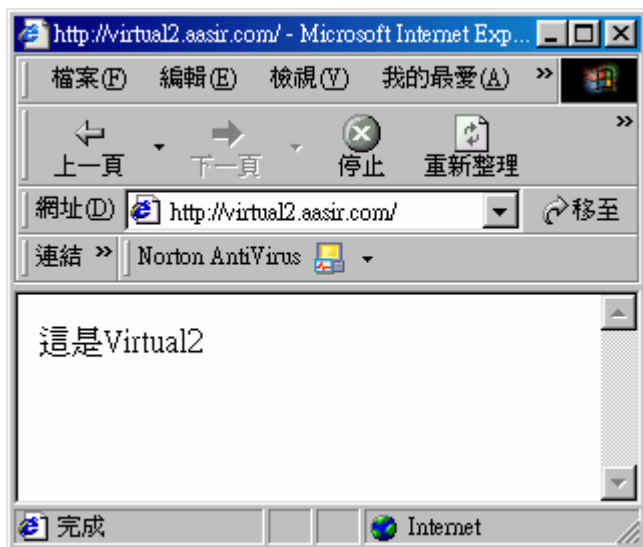
```
1 NameVirtualHost 61.218.29.3
2 <VirtualHost 61.218.29.3>
3     ServerAdmin wu.chaiyen@msa.hinet.net
4     DocumentRoot /var/www/html/html
5     ServerName flash.aasir.com
6     ErrorLog logs/virtual.com-error_log
7     CustomLog logs/virtual.com-access_log common
8 </VirtualHost>
9 <VirtualHost 61.218.29.3>
10     ServerAdmin wu.chaiyen@msa.hinet.net
11     DocumentRoot /var/www/virtual2
12     ServerName virtual2.aasir.com
13     ErrorLog logs/virtual2.com-error_log
14     CustomLog logs/virtual2.com-access_log common
15 </VirtualHost>
16 <VirtualHost 61.218.29.3>
17     ServerAdmin wu.chaiyen@msa.hinet.net
18     DocumentRoot /var/www/virtual3
19     ServerName virtual3.aasir.com
20     ErrorLog logs/virtual3.com-error_log
21     CustomLog logs/virtual3.com-access_log common
22 </VirtualHost>
```

這是在/var/www 目錄下建立 virtual2 和 virtual3 目錄。

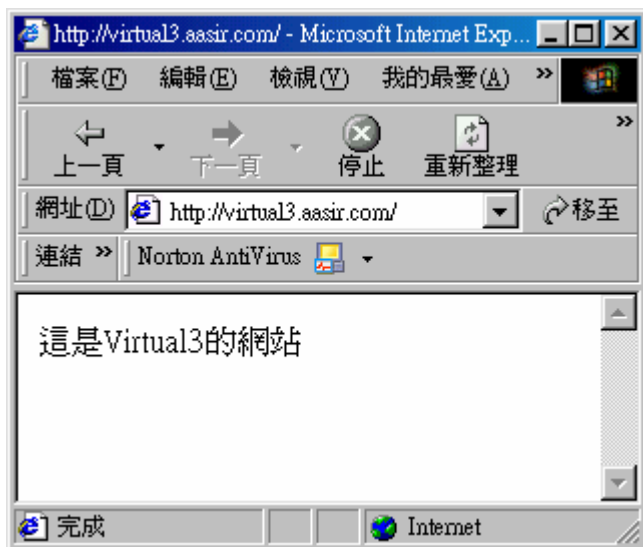
```
[root@flash www]# mkdir virtual2
[root@flash www]# mkdir virtual3
```



這是當我們執行 Virtual2.aasir.com 的網址服務時，它就會對應到 /var/www/virtual2 目錄下的檔案。



這是當我們執行 Virtual3.aasir.com 的網址服務時，它就會對應到 /var/www/virtual3 目錄下的檔案。



## 23-4-2 以 IP 為基礎的虛擬主機設定

我們可以在一台機器上設定多張網路卡，每一張網路卡都有一個 IP 位址而且也有連接埠編號。Listen 指令後面接 IP 位址當作它們的參數。

```
Listen 61.218.29.5:80
Listen 61.218.29.6:80
Listen 61.218.29.7:80
Listen 61.218.29.8:80
Listen 61.218.29.9:8080
```

我們可以定義以 ip 為基礎的虛擬主機，虛擬主機可以用<VirtualHost>容器指令來定義。

語法：

```
<VirtualHost IP 位址:連接埠 IP 位址:連接埠.....>
    #虛擬主機指令
    .....
</VirtualHost>
```

例如：我們將 61.218.29.5 和 61.218.29.7 兩個 IP 位置對應到同一個虛擬主機。

```
<VirtualHost 61.218.29.5 61.218.29.7>
    #虛擬主機指令
    .....
</VirtualHost>
```

例如：我們將 61.218.29.6:80、61.218.29.8:80 和 61.218.29.9:8080 三個 IP 位置對應到同一個虛擬主機。

```
<VirtualHost 61.218.29.5:80 61.218.29.6:80 61.218.29.8:8080>
```



```
#虛擬主機指令
```

```
.....
```

```
</VirtualHost>
```

## 23-5 Apache 的延伸檔

Apache 的延伸檔就是被 Include 指令包含進來 httpd.conf 的組態檔。

在/etc/httpd/conf/httpd.conf 第 168 行的 Include 指令將/etc/httpd/conf.d 的檔案包含進入。

```
162 Include conf.d/*.conf
```

### 1-5-1 設定/etc/httpd/conf.d/php.conf 動態網頁 PHP 組態檔

我們可以在/etc/httpd/conf.d 的目錄下編輯 php.conf 的檔案。

```
[root@flash conf.d]# vi php.conf
```

第六行 LoadModule php4\_module 是載入我們 PHP 的模組。預設的 libphp4.so 的 PHP 模組是放在/etc/httpd/modules/libphp4.so 第 11 行到第 15 行是認識我們 PHP 的檔案，使用 Files \*.php。第 21 行的 DirectoryIndex 是設定目錄的首頁。

```
1 #
2 # PHP is an HTML-embedded scripting language which attempts to make it
3 # easy for developers to write dynamically generated webpages.
4 #
5
6 LoadModule php4_module modules/libphp4.so
7
8 #
9 # Cause the PHP interpreter handle files with a .php extension.
10 #
11 <Files *.php>
12     SetOutputFilter PHP
13     SetInputFilter PHP
14     LimitRequestBody 524288
15 </Files>
16
17 #
18 # Add index.php to the list of files that will be served as directory
19 # indexes.
20 #
21 DirectoryIndex index.php
```



設定 PHP 的組態檔/etc/php.ini

將/etc/php.ini 的第 312 行的 register\_globals=off 改為 register\_globals=On，這樣 PHP 就能夠傳遞和記錄全域變數了。

```
[root@aasir chaiyen]# vi /etc/php.ini  
312 register_globals = on
```

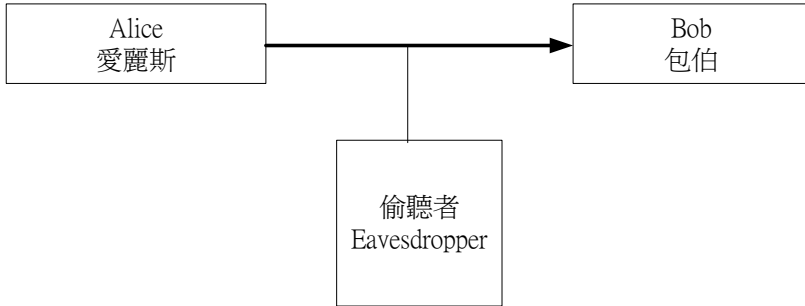
### 23-5-2 設定/etc/httpd/conf.d/ssl.conf 通訊安全協定組態檔

Apache HTTP 伺服器提供 mod\_ssl 模組介面給 OpenSSL 函式，mod\_ssl 模組提供 Apache 伺服器強大的加密方法給傳輸層，讓我們的資料可以在安全的情況下傳輸。在密碼學中未加密的資訊稱為明文，加密過後的資訊叫作密文。我們使用者將明文加密然後將密文傳給收件者，收件者只要有密鑰，就可以將密文轉換成明文，然後得取資訊。竊取者如果竊取傳輸中的密文，但沒有解密的密鑰，也是無法得到明文。簡單的解釋 P 為明文 E 為加密的函數 D 為解密的密鑰， $D(E(P))=P$ 。 $D(E(P))=P$  表示當 P 經 E 加密後再經過 D 解密就可以得到原來的 P 了。





SSL安全傳輸訊息概念



E:編碼演算法(Encoding algorithm)

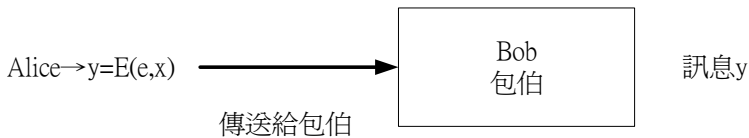
D:解碼演算法(Decoding algorithm)

這兩個是公開給大眾的

e:編碼之鑰Encoding key

d:解碼之鑰decoding key

X訊息是由 $\{0,1\}^*$ 二進位所組成。 $X \leftarrow \{0,1\}^*$   
 愛麗斯使用e編碼之鑰將x訊息編碼成y，再將它傳送給包伯。



當包伯收到訊息y然後解碼

$$\begin{aligned}
 D(d, y) &= x \\
 D(d, E(e, x)) &= x \\
 \because E(e, x) &= e \oplus x, D(d, y) = e \oplus y \\
 \therefore (x \oplus d) \oplus e &= x
 \end{aligned}$$

公開之鑰保密機:

愛麗斯介由計算和傳送 $E(e, x)$ 將訊息x送給包伯。

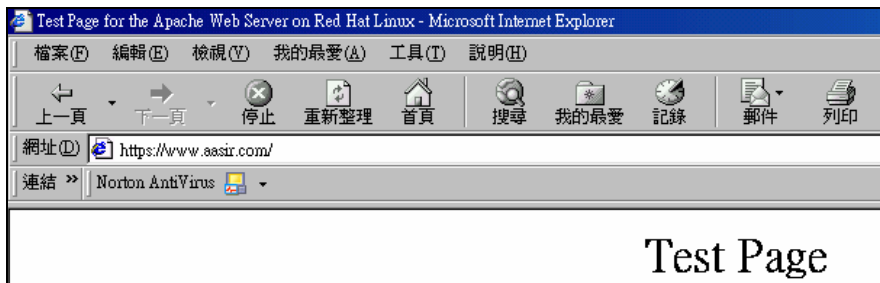
$D(D(e, x)) = x$ 只要解碼就可以得到x。

這個重點是偷窺者(駭客)很難從e來計算求出d，而且如果不知到d將很難計算從y來得到x。

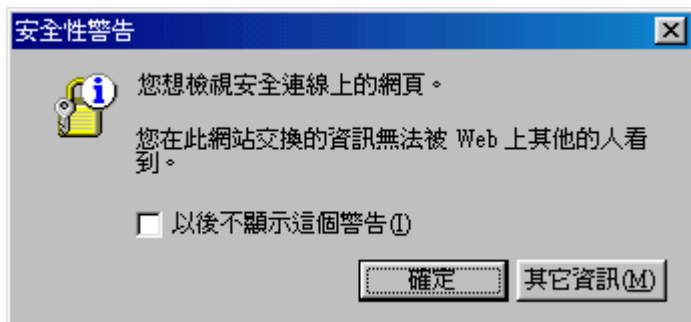
$$\begin{aligned}
 y &= x^e \pmod{pq} \\
 e * d &= 1 + k \phi(pq)
 \end{aligned}$$



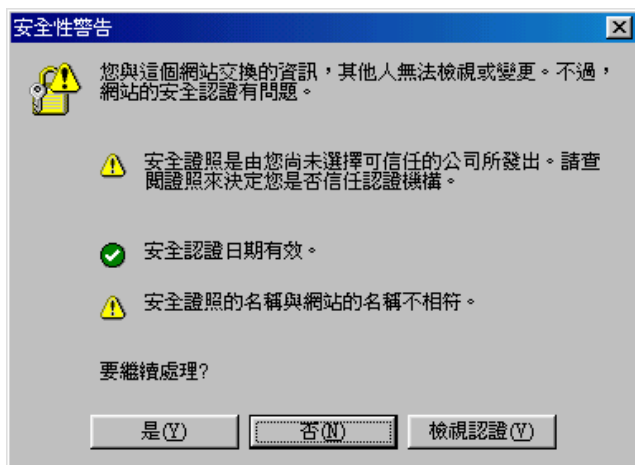
https://www.aasir.com/是經過加密的通訊協定。http://www.aasir.com 是沒有加密的。https 後面有加 s 是表示安全性 security。



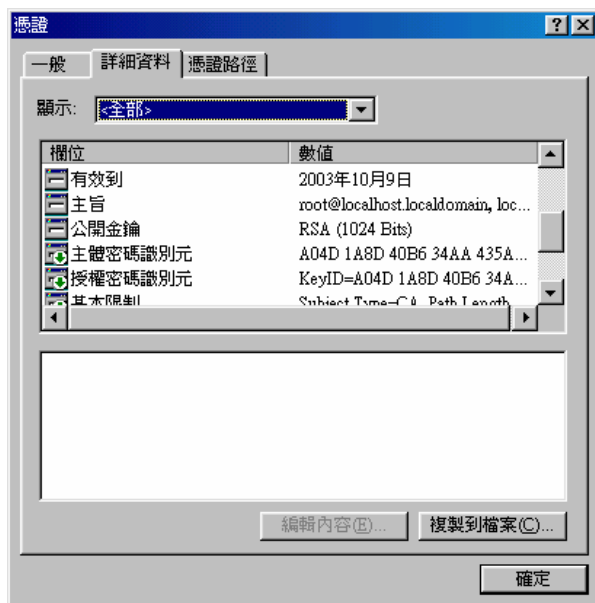
當我們輸入 https 時就會出現安全性警告。



這時會出現安全性警告。我們如果按下檢視認證，則可以觀看認證的資料。



這是憑證的詳細內容。



我們可以在網頁的右下方看到鑰匙，這就是經過加密傳輸到我們使用者端，然後再解密的網頁，網址 <https://www.aasir.com> 後面有加 s。



我們可以在 [http://httpd.apache.org/docs-2.0/mod/mod\\_ssl.html](http://httpd.apache.org/docs-2.0/mod/mod_ssl.html) 得到 SSL 通訊安全協定的資訊。我們也可以在 <http://www.modssl.org/docs/> 得到相關得資訊。

我們可以在 `/etc/httpd/conf.d` 的目錄下編輯 `ssl.conf` 通訊安全協定組態檔。

```
[root@flash conf.d]# vi ssl.conf
```

`LoadModule ssl_module` 是從 `/etc/httpd/modules/mod_ssl.so` 載入安全協定組態檔。

```
16 LoadModule ssl_module modules/mod_ssl.so
```

`Listen 443` 設定通訊安全協定的連接埠為 443。

```
27 Listen 443
```

第 33 行 `ErrorLog` 指令是設定錯誤資訊記錄的地方。

第 34 行 `CustomLog` 指令是設定存取記錄檔的位置和格式。

```
33 # ErrorLog logs/dummy-host.example.com-error_log
34 # CustomLog logs/dummy-host.example.com-access_log common
```

從第 36 行到第 85 行是設定 SSL 安全協定全域文件的地方，從第 86 行到第 256 行是設定 SSL 安全協定虛擬主機的地方。

`AddType` 指令設定 `.crt` 的檔案對應到 `application/x-x509-ca-cert` 形態。

`AddType` 指令設定 `.crl` 的檔案對應到 `application/x-pkcs7-crl` 形態。

```
46 AddType application/x-x509-ca-cert .crt
47 AddType application/x-pkcs7-crl .crl
```

當 Apache 開始時，它會讀取認證和啟動 SSL 虛擬伺服器 Private Key 的檔案。因為安全的考量，Private Key 一定要被加密，`mod_ssl` 模組需要去查詢管理者來解密這些檔案。這些查詢可以被分為 `builtin` 和 `exec:/path/to/program` 兩種。

`SSLPassPhraseDialog` 指令設定通過加密 Private Key 的對話類型。



## 53 SSLPassPhraseDialog builtin

SSLSessionCache 指令設定 global/inter-process SSL Session Cache 的類型。

預設是將 SSLSessionCache 指令設為 none，也就是取消 global/inter-process SSL Session Cache。

SSLSessionCacheTimeout 指定 SSL session 到期的時間，預設是 300。

```

58 #SSLSessionCache          none
59 #SSLSessionCache          shmht:/var/cache/mod_ssl/scache(512000)
60 #SSLSessionCache          shmcb:/var/cache/mod_ssl/scache(512000)
61 #SSLSessionCache          dbm:/var/cache/mod_ssl/scache
62 SSLSessionCacheTimeout    300

```

SSLMutex 指令設定運作的一致性訊號。SSLMutex 訊號可以分為三種 none、file:/path/to/mutex 和 scm。

## 67 SSLMutex file:logs/ssl\_mutex

SSLRandomSeed 指令設定在 OpenSSL 啟動(startup)和建立新的 SSL 連線(connect)時 PRNG 的播種資源。預設是 builtin。SSLRandomSeed 指令可以有 builtin、file:/path/to/source、egd:/path/to/egd-socket 和 exe:/path/to/program。

```

79 SSLRandomSeed startup builtin
80 SSLRandomSeed connect builtin
81 #SSLRandomSeed startup file:/dev/random 512
82 #SSLRandomSeed startup file:/dev/urandom 512
83 #SSLRandomSeed connect file:/dev/random 512
84 #SSLRandomSeed connect file:/dev/urandom 512

```

第 90 行<VirtualHost default:443>開始是定義加密虛擬網站。

第 93 行 DocumentRoot 是放置 SSL 加密傳輸網站目錄的地方。

第 95 行是設定 SSL 加密傳輸網站的管理者。



```
90 <VirtualHost _default_:443>
91
92 # General setup for the virtual host
93 DocumentRoot "/home/chaiyen/html"
94 ServerName aasir.com:443
95 ServerAdmin wu.chaiyen@msa.hinet.net
96 ErrorLog logs/ssl_error_log
97 TransferLog logs/ssl_access_log
```

### 23-6 分析網站流量 webalizer 指令

分析網站的流量可以讓我們了解網站的使用情況、流量、以及使用者的分佈。一般我們在 httpd.conf 組態檔使用 ErrorLog 指令和 LogLevel 指令來記錄網站的資訊。預設的檔案是在/etc/httpd/logs/error\_log，這是連接到/var/log/httpd/error\_log。

這是/etc/httpd/logs/error\_log 錯誤記錄檔的情況。這些錯誤檔可能包含了我們網站運行的一些錯誤資訊，因此我們可以按照這些錯誤資訊來修改我們網站的情況。

```
[Thu Feb 06 01:00:25 2003] [error] [client 61.218.29.5] File does not exist: /var/www/html/html/server-info
[Thu Feb 06 01:00:33 2003] [error] [client 61.218.29.5] client denied by server configuration: /var/www/html/html/server-status
[Thu Feb 06 01:02:29 2003] [warn] child process 16760 still did not exit, sending a SIGTERM
```

每一個進入網站的 HTTP 要求服務都會產生一個存取記錄檔

/etc/httpd/logs/access\_log(這個檔是連接到/var/log/httpd/access\_log)。這個存取記錄檔的格式在 httpd.conf 組態檔中是由 LogFormat 指令和 CustomLog 指令來產生與修改。這是存取記錄檔 access\_log 的格式。

```
218.165.141.157 - - [06/Feb/2003:07:33:33 +0800] "GET /modules.php?name=News&file=categories&op=newindex&catid=1 HTTP/1.1" 200 9688 "http://aasir.com/modules.php?name=Downloads" "Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)"
61.59.141.216 - - [06/Feb/2003:07:33:33 +0800] "GET /themes/NukeNews/images/corner-top-right.gif HTTP/1.1" 200 111 "http://www.aasir.com/modules.php?name=News&ew_topic=6" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)"
```



### 23-6-1 Webalizer 的組態

Webalizer 指令可以使用 Apache 伺服器所產生的記錄檔來產生報表。這些報表都是以網頁的格式出現，裏面有圖型也有表格，讓我們可以輕易的了解目前的網站情況。我們可以編輯 webalizer.conf 組態檔來設定網站記錄檔的報表。

```
[root@flash httpd]# vi /etc/webalizer.conf
```

LogFile 是 webalizer.conf 組態檔的指令，它指示從/var/log/httpd/access\_log 來讀取網站的記錄檔。

```
28 LogFile /var/log/httpd/access_log
```

OutputDir 是 webalizer.conf 組態檔的指令，它指示輸出 webalizer 處理後的網站報表到/var/www/html/usage。

```
42 OutputDir /var/www/html/usage
```

HistoryName 指令設定放置歷史記錄的地方，預設是放置到 /var/lib/webalizer/webalizer.hist。Webalizer 可以保存 12 個月的歷史記錄資料。

```
52 HistoryName /var/lib/webalizer/webalizer.hist
```

Incremental 指令設定成 yes 時可以讓我們有多種部份的記錄資料。

```
67 Incremental yes
```

IncrementalName 指令可以讓我們設定 Incremental 儲存資料的地方。

```
76 IncrementalName /var/lib/webalizer/webalizer.current
```

ReportTitle 指令設定報表的標題。

```
83 ReportTitle Statistics for AASIR
```

HostName 設定報表的名稱。

```
94 HostName www.aasir.com
```



HTMLExtension 設定網頁的副檔名 html。

```
100 HTMLExtension html
```

PageType 指令設定網頁的類型為 html 或 cgi。

```
108 PageType htm*
```

```
109 PageType cgi
```

UseHTTPS 指令使用安全傳輸來觀看報表，預設是沒有作用。

```
120 #UseHTTPS no
```

## 23-6-2 Webalizer 的報表析

我們使用 webalizer 指令就可以產生網站記錄檔的報表。

```
[root@flash www]# webalizer
```

這是使用 webalizer 指令產生的網站報表。

```
[root@flash httpd]# cd /var/www/html/usage
[root@flash usage]# ls
ctry_usage_200210.png    daily_usage_200301.png    msfree.png
ctry_usage_200211.png    daily_usage_200302.png    usage_200210.html
ctry_usage_200212.png    hourly_usage_200210.png  usage_200211.html
ctry_usage_200301.png    hourly_usage_200211.png  usage_200212.html
ctry_usage_200302.png    hourly_usage_200212.png  usage_200301.html
daily_usage_200210.png  hourly_usage_200301.png  usage_200302.html
daily_usage_200211.png  hourly_usage_200302.png  usage.png
daily_usage_200212.png  index.html                webalizer.png
```

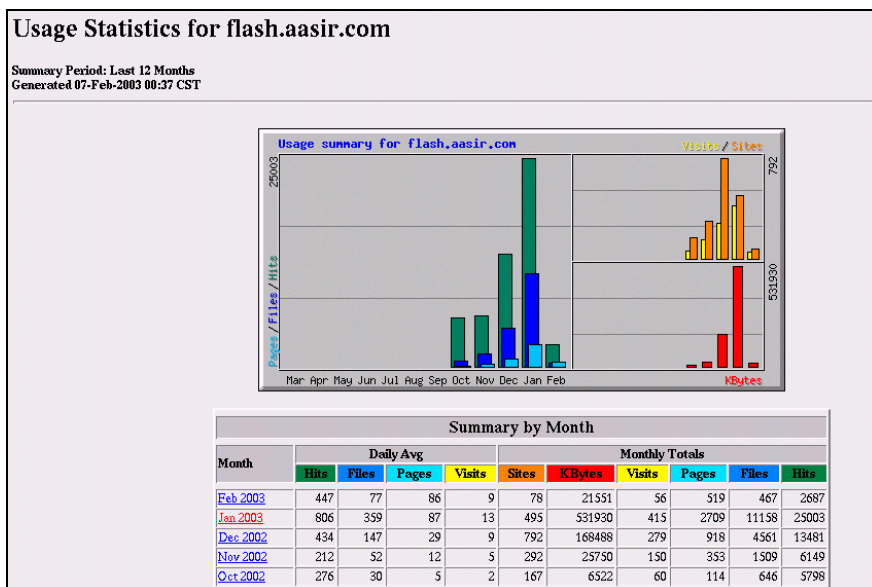
我們可以使用小恐龍瀏覽器來觀看報表，我們輸入 mozilla 指令和報表路徑。

```
[root@flash www]# mozilla /var/www/html/usage/index.html
```

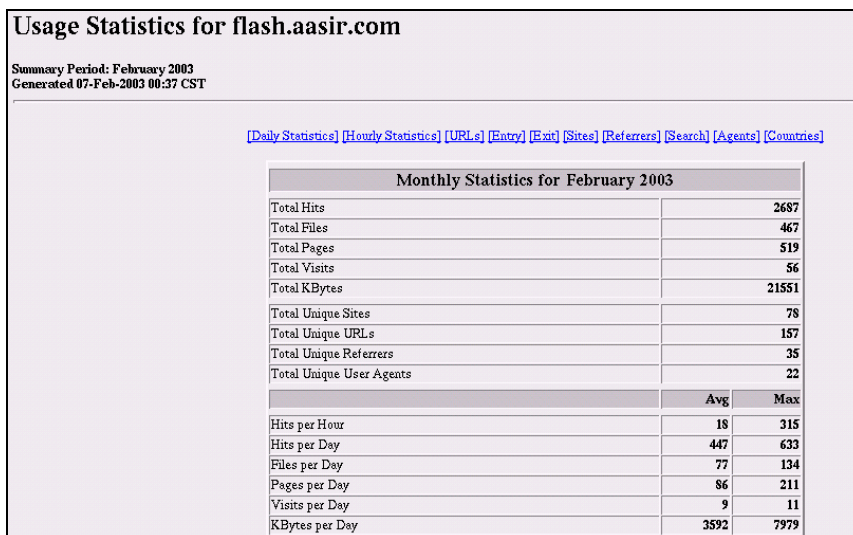




這是我們網站 flash.aasir.com 的使用率統計。這是月報表的總結。我們可以按下月份，來觀看該月份的網站使用詳細分析。



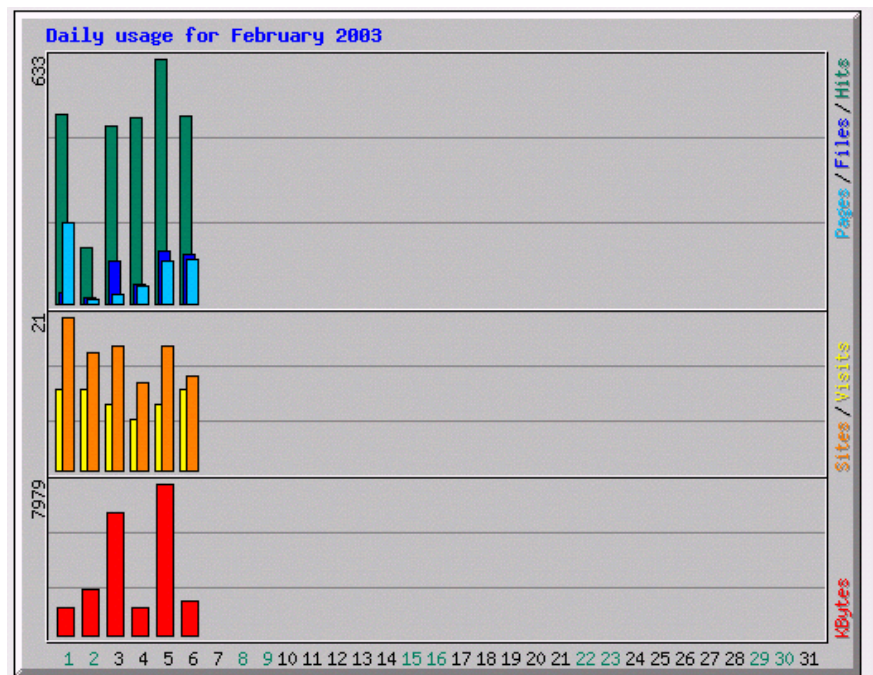
這是二月份網站使用的情况。



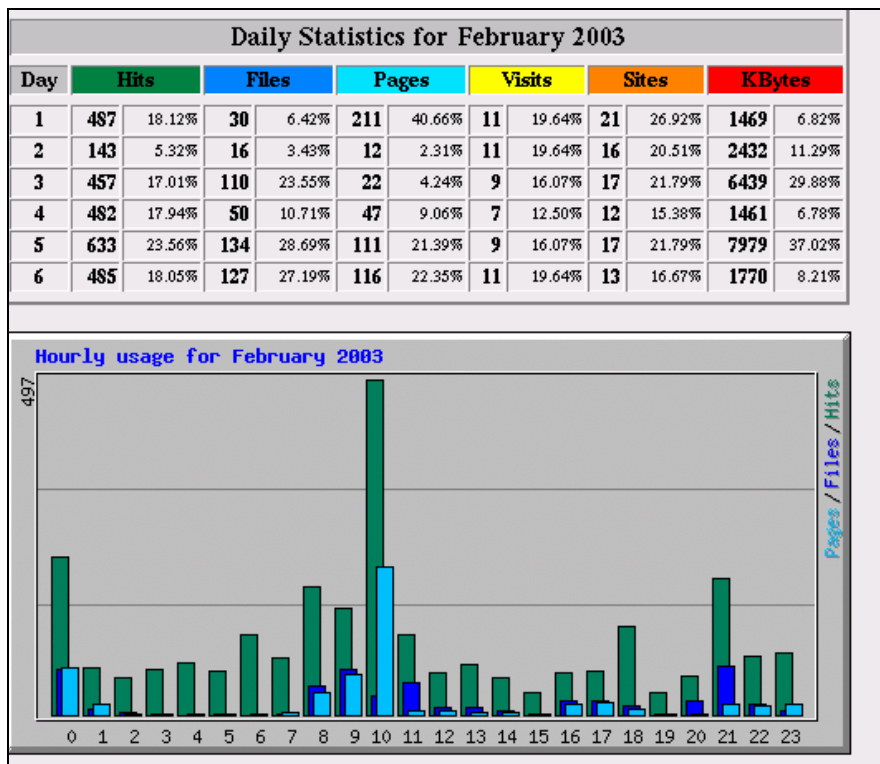
這是我們網站以編碼表示回應的情況。

Hits by Response Code	
Code 200 - OK	467
Code 206 - Partial Content	3
Code 301 - Moved Permanently	10
Code 302 - Found	3
Code 304 - Not Modified	310
Code 400 - Bad Request	166
Code 403 - Forbidden	197
Code 404 - Not Found	1529
Code 500 - Internal Server Error	1
Code 501 - Not Implemented	1

這是二月份網站每日使用的情況報告。



這是二月份每日統計網站的使用情況，以及二月份每天網站的使用率。我們可以看到二月份在早上 10 點的時後網站被點選的次數最高，總共有 497 次，而且網頁被要求的次數也最高，因此該網站在二月份的時後早上 10 點的使用率最高，這和該網站的屬性和客戶群有關。



Flash.aasir.com 的網站雖然在早上 10 點的點選次數最高，但是網站的流量則以在早上 12 點和晚上 10 點(22)的時後流量到達最高峰，分別占總流量的 25.01% 和 17.31%，再來就是凌晨 0 點的 10.92%，因此夜貓族也是該網站的客群。



Hourly Statistics for February 2003												
Hour	Hits			Files			Pages			KBytes		
	Avg	Total		Avg	Total		Avg	Total		Avg	Total	
0	39	234	8.71%	11	67	14.35%	11	71	13.68%	392	2352	10.92%
1	11	69	2.57%	1	9	1.93%	2	16	3.08%	82	491	2.28%
2	9	55	2.05%	0	3	0.64%	0	2	0.39%	11	63	0.29%
3	11	67	2.49%	0	2	0.43%	0	2	0.39%	62	371	1.72%
4	13	78	2.90%	0	0	0.00%	0	2	0.39%	13	78	0.36%
5	10	65	2.42%	0	2	0.43%	0	1	0.19%	61	368	1.71%
6	19	118	4.39%	0	0	0.00%	0	1	0.19%	20	121	0.56%
7	14	84	3.13%	0	1	0.21%	0	3	0.58%	14	87	0.40%
8	31	190	7.07%	7	42	8.99%	5	33	6.36%	65	387	1.80%
9	26	159	5.92%	11	67	14.35%	10	61	11.75%	80	483	2.24%
10	82	497	18.50%	4	28	6.00%	36	218	42.00%	34	204	0.95%
11	19	118	4.39%	8	48	10.28%	1	7	1.35%	292	1754	8.14%
12	10	62	2.31%	1	10	2.14%	0	5	0.96%	898	5390	25.01%
13	12	74	2.75%	1	10	2.14%	0	4	0.77%	15	90	0.42%
14	9	55	2.05%	0	5	1.07%	0	3	0.58%	289	1734	8.05%
15	5	32	1.19%	0	1	0.21%	0	1	0.19%	56	335	1.56%
16	10	63	2.34%	3	20	4.28%	2	16	3.08%	69	414	1.92%
17	10	65	2.42%	3	21	4.50%	3	19	3.66%	120	721	3.35%
18	21	131	4.88%	2	14	3.00%	1	8	1.54%	75	452	2.10%
19	5	33	1.23%	0	1	0.21%	0	1	0.19%	56	336	1.56%
20	9	58	2.16%	3	22	4.71%	0	2	0.39%	110	663	3.07%
21	33	201	7.48%	12	73	15.63%	2	15	2.89%	138	829	3.85%
22	14	88	3.28%	2	16	3.43%	2	13	2.50%	622	3731	17.31%
23	15	91	3.39%	0	5	1.07%	2	15	2.89%	16	96	0.45%

這是網站檔案被使用次數及流量排行的前 157 名。第一名是/example/object.php 的檔案，總共被使用 42 次。但是流量最大的檔案是/example/e/ddm.exe 的檔案，總共使用流量 13366Kbytes，流量使用率占總體的 62.02%。因此我們可以得知 /example/e/ddm.exe 雖然被要求服務次數不多，但是它的流量使用率確占了整體的 62.02%之多，ddm.exe 這個檔案和網站其它檔案比較起來其檔案太大了。

Top 30 of 157 Total URLs					
#	Hits		KBytes		URL
1	42	1.56%	1	0.00%	<a href="#">/example/f/object.php</a>
2	34	1.27%	141	0.65%	<a href="#">/server-status</a>
3	26	0.97%	210	0.97%	<a href="#">/</a>
4	22	0.82%	12	0.05%	<a href="#">/themes/DeepBlue/style/style.css</a>
5	20	0.74%	26	0.12%	<a href="#">/query/mysql_test.php</a>
6	18	0.67%	4852	22.51%	<a href="#">/good/chaiyen/jigsaw.swf</a>
7	12	0.45%	45	0.21%	<a href="#">/example/</a>
8	11	0.41%	6	0.03%	<a href="#">/good/chaiyen/jigsaw.html</a>
9	8	0.30%	13366	62.02%	<a href="#">/example/ddm.exe</a>
10	8	0.30%	15	0.07%	<a href="#">/shop/default.htm</a>
11	7	0.26%	15	0.07%	<a href="#">/arr.php</a>
12	6	0.22%	38	0.18%	<a href="#">/good/chaiyen/</a>
13	6	0.22%	1	0.00%	<a href="#">/robots.txt</a>
14	6	0.22%	36	0.17%	<a href="#">/shop/register.htm</a>
15	6	0.22%	13	0.06%	<a href="#">/shop2/default.htm</a>
16	5	0.19%	27	0.13%	<a href="#">/modules.php</a>
17	5	0.19%	8	0.04%	<a href="#">/mysql_test.php</a>
18	5	0.19%	316	1.47%	<a href="#">/server-info</a>
19	4	0.15%	10	0.05%	<a href="#">/example/HTML/</a>
20	4	0.15%	10	0.04%	<a href="#">/example/f/</a>
21	4	0.15%	10	0.05%	<a href="#">/example/%A4@%g/</a>
22	3	0.11%	2	0.01%	<a href="#">/query/query.php</a>
23	3	0.11%	11	0.05%	<a href="#">/server_status.content</a>
24	3	0.11%	61	0.29%	<a href="#">/~justin/</a>
25	2	0.07%	0	0.00%	<a href="#">/bs_insert.php</a>
26	2	0.07%	0	0.00%	<a href="#">/cgi-bin/htsearch</a>
27	2	0.07%	0	0.00%	<a href="#">/cgi-bin/namazugi</a>



這是 flash.aasir.com 網站使用前 10 名的檔案。

Top 10 of 157 Total URLs By KBytes					
#	Hits		KBytes		URL
1	8	0.30%	13366	62.02%	<a href="#">/example/ddm.exe</a>
2	18	0.67%	4852	22.51%	<a href="#">/good/chaiven/jigsaw.swf</a>
3	5	0.19%	316	1.47%	<a href="#">/server-info</a>
4	26	0.97%	210	0.97%	<a href="#">/</a>
5	34	1.27%	141	0.65%	<a href="#">/server-status</a>
6	3	0.11%	61	0.29%	<a href="#">/~justin/</a>
7	12	0.45%	45	0.21%	<a href="#">/example/</a>
8	6	0.22%	38	0.18%	<a href="#">/good/chaiven/</a>
9	6	0.22%	36	0.17%	<a href="#">/shop/register.htm</a>
10	5	0.19%	27	0.13%	<a href="#">/modules.php</a>

Top 10 of 10 Total Entry Pages					
#	Hits		Visits		URL
1	26	0.97%	14	35.00%	<a href="#">/</a>
2	11	0.41%	9	22.50%	<a href="#">/good/chaiven/jigsaw.html</a>
3	8	0.30%	5	12.50%	<a href="#">/shop/default.htm</a>
4	6	0.22%	4	10.00%	<a href="#">/shop2/default.htm</a>
5	12	0.45%	3	7.50%	<a href="#">/example/</a>
6	4	0.15%	1	2.50%	<a href="#">/example/HTML/</a>
7	4	0.15%	1	2.50%	<a href="#">/example/i/</a>
8	5	0.19%	1	2.50%	<a href="#">/server-info</a>
9	34	1.27%	1	2.50%	<a href="#">/server-status</a>
10	6	0.22%	1	2.50%	<a href="#">/shop/register.htm</a>

這是拜訪我們網站的使用者資料。其中有一些人是來自其它網域。

#	Hits		Visits		URL	
1	26	0.97%	11	27.50%	/	
2	11	0.41%	10	25.00%	/good/chaiven/jigsaw.html	
3	6	0.22%	5	12.50%	/shop/register.htm	
4	34	1.27%	4	10.00%	/server-status	
5	4	0.15%	2	5.00%	/example/!	
6	8	0.30%	2	5.00%	/shop/default.htm	
7	6	0.22%	2	5.00%	/shop2/default.htm	
8	12	0.45%	1	2.50%	/example/	
9	1	0.04%	1	2.50%	/example/!	
10	6	0.22%	1	2.50%	/good/chaiven/	

Top 30 of 78 Total Sites									
#	Hits		Files		KBytes		Visits		Hostname
1	642	23.89%	231	49.46%	2875	13.34%	13	23.21%	61-218-29-5.hinet-ip.hinet.net
2	641	23.86%	0	0.00%	654	3.04%	0	0.00%	61-218-232-188.hinet-ip.hinet.net
3	314	11.69%	3	0.64%	41	0.19%	1	1.79%	61.233.38.16
4	204	7.59%	0	0.00%	207	0.96%	0	0.00%	61-218-154-85.hinet-ip.hinet.net
5	127	4.73%	0	0.00%	130	0.60%	0	0.00%	61-218-232-187.hinet-ip.hinet.net
6	121	4.50%	67	14.35%	143	0.66%	1	1.79%	u175-95.u61-70.giga.net.tw
7	91	3.39%	0	0.00%	94	0.44%	0	0.00%	61-218-136-173.hinet-ip.hinet.net
8	59	2.20%	38	8.14%	166	0.77%	1	1.79%	localhost.localdomain
9	56	2.08%	14	3.00%	193	0.90%	2	3.57%	61-228-31-47.hinet-ip.hinet.net
10	32	1.19%	0	0.00%	33	0.15%	0	0.00%	61.177.181.90
11	30	1.12%	8	1.71%	13	0.06%	1	1.79%	61-228-30-247.hinet-ip.hinet.net
12	20	0.74%	4	0.86%	1681	7.80%	1	1.79%	61-228-30-81.hinet-ip.hinet.net
13	19	0.71%	19	4.07%	67	0.31%	1	1.79%	tc218-187-65-54-2-1.pl.apol.com.tw
14	16	0.60%	0	0.00%	17	0.08%	0	0.00%	61-07-122-20.okema.com





第六名的 u175-95.u61-70.giga.net.tw 的使用者，使用 giga 寬頻上到我們網站。

Top 30 of 78 Total Sites									
#	Hits		Files		KBytes		Visits		Hostname
1	642	23.89%	231	49.46%	2875	13.34%	13	23.21%	61-218-29-5.hinet-ip.hinet.net
2	641	23.86%	0	0.00%	654	3.04%	0	0.00%	61-218-232-188.hinet-ip.hinet.net
3	314	11.69%	3	0.64%	41	0.19%	1	1.79%	61.233.38.16
4	204	7.59%	0	0.00%	207	0.96%	0	0.00%	61-218-154-85.hinet-ip.hinet.net
5	127	4.73%	0	0.00%	130	0.60%	0	0.00%	61-218-232-187.hinet-ip.hinet.net
6	121	4.50%	67	14.35%	143	0.66%	1	1.79%	u175-95.u61-70.giga.net.tw
7	91	3.39%	0	0.00%	94	0.44%	0	0.00%	61-218-136-173.hinet-ip.hinet.net
8	59	2.20%	38	8.14%	166	0.77%	1	1.79%	localhost.localdomain
9	56	2.08%	14	3.00%	193	0.90%	2	3.57%	61-228-31-47.hinet-ip.hinet.net
10	32	1.19%	0	0.00%	33	0.15%	0	0.00%	61.177.181.90
11	30	1.12%	8	1.71%	13	0.06%	1	1.79%	61-228-30-247.hinet-ip.hinet.net
12	20	0.74%	4	0.86%	1681	7.80%	1	1.79%	61-228-30-81.hinet-ip.hinet.net
13	19	0.71%	19	4.07%	67	0.31%	1	1.79%	tc218-187-65-54-2-1.pl.apol.com.tw
14	16	0.60%	0	0.00%	17	0.08%	0	0.00%	61-97-122-29.oksung.com
15	16	0.60%	0	0.00%	16	0.08%	0	0.00%	61.11.75.249
16	16	0.60%	0	0.00%	17	0.08%	0	0.00%	61.128.219.241
17	16	0.60%	0	0.00%	16	0.08%	0	0.00%	61.130.78.170
18	16	0.60%	0	0.00%	16	0.08%	0	0.00%	61.146.237.46
19	16	0.60%	0	0.00%	16	0.08%	0	0.00%	61.146.78.146
20	16	0.60%	0	0.00%	16	0.08%	0	0.00%	61.150.120.4
21	16	0.60%	0	0.00%	16	0.07%	0	0.00%	61.184.193.204
22	16	0.60%	0	0.00%	16	0.07%	0	0.00%	61.185.33.21
23	16	0.60%	0	0.00%	16	0.07%	0	0.00%	61.241.149.249
24	16	0.60%	0	0.00%	17	0.08%	0	0.00%	61.52.7.140
25	16	0.60%	0	0.00%	17	0.08%	0	0.00%	61.60.89.58
26	14	0.52%	0	0.00%	14	0.07%	0	0.00%	61-218-96-99.hinet-ip.hinet.net
27	12	0.45%	0	0.00%	12	0.06%	0	0.00%	61.219.243.2



第 2 名的 u175-95.u61-70.giga.net.tw 的使用者，大量使用我們的流量資源。

Top 10 of 78 Total Sites By KBytes										
#	Hits		Files		KBytes		Visits		Hostname	
1	642	23.89%	231	49.46%	2875	13.34%	13	23.21%	61-218-29-5.hinet-ip.hinet.net	
2	3	0.11%	3	0.64%	1975	9.16%	1	1.79%	u58-191.user.giga.net.tw	
3	20	0.74%	4	0.86%	1681	7.80%	1	1.79%	61-228-30-81.hinet-ip.hinet.net	
4	2	0.07%	2	0.43%	1671	7.75%	0	0.00%	crawl13-public.alexa.com	
5	1	0.04%	1	0.21%	1671	7.75%	0	0.00%	212.199.11.10	
6	1	0.04%	1	0.21%	1671	7.75%	0	0.00%	218-163-34-76.hinet-ip.hinet.net	
7	1	0.04%	1	0.21%	1671	7.75%	0	0.00%	57.c210-58-135.ethome.net.tw	
8	1	0.04%	1	0.21%	1671	7.75%	0	0.00%	61.177.45.111	
9	641	23.86%	0	0.00%	654	3.04%	0	0.00%	61-218-232-188.hinet-ip.hinet.net	
10	9	0.33%	9	1.93%	314	1.46%	1	1.79%	61.180.27.169	

Top 30 of 35 Total Referrers		
#	Hits	Referrer
1	2073	77.15% - (Direct Request)
2	201	7.48% <a href="http://flash.aasir.com/">http://flash.aasir.com/</a>
3	95	3.54% <a href="http://flash.aasir.com/modules.php">http://flash.aasir.com/modules.php</a>
4	67	2.49% <a href="http://flash.aasir.com/example/">http://flash.aasir.com/example/</a>
5	34	1.27% <a href="http://flash.aasir.com/good/chaiyen/">http://flash.aasir.com/good/chaiyen/</a>
6	25	0.93% <a href="http://flash.aasir.com/index.php">http://flash.aasir.com/index.php</a>
7	22	0.82% <a href="http://flash.aasir.com/shop/default.htm">http://flash.aasir.com/shop/default.htm</a>
8	21	0.78% <a href="http://flash.aasir.com/example/i/">http://flash.aasir.com/example/i/</a>
9	19	0.71% <a href="http://flash.aasir.com/query/query.php">http://flash.aasir.com/query/query.php</a>
10	17	0.63% <a href="http://127.0.0.1/">http://127.0.0.1/</a>
11	17	0.63% <a href="http://localhost/">http://localhost/</a>
12	12	0.45% <a href="http://flash.aasir.com/shop2/default.htm">http://flash.aasir.com/shop2/default.htm</a>
13	10	0.37% <a href="http://aasir.com/">http://aasir.com/</a>



這是被使用網頁的情況。

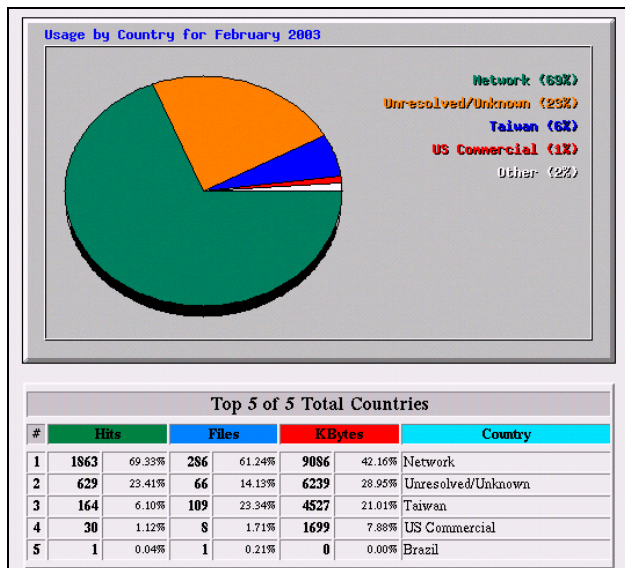
Top 30 of 35 Total Referrers		
#	Hits	Referrer
1	2073	77.15% - (Direct Request)
2	201	7.48% <a href="http://flash.aasir.com/">http://flash.aasir.com/</a>
3	95	3.54% <a href="http://flash.aasir.com/modules.php">http://flash.aasir.com/modules.php</a>
4	67	2.49% <a href="http://flash.aasir.com/example/">http://flash.aasir.com/example/</a>
5	34	1.27% <a href="http://flash.aasir.com/good/chaiven/">http://flash.aasir.com/good/chaiven/</a>
6	25	0.93% <a href="http://flash.aasir.com/index.php">http://flash.aasir.com/index.php</a>
7	22	0.82% <a href="http://flash.aasir.com/shop/default.htm">http://flash.aasir.com/shop/default.htm</a>
8	21	0.78% <a href="http://flash.aasir.com/example/">http://flash.aasir.com/example/</a>
9	19	0.71% <a href="http://flash.aasir.com/query/query.php">http://flash.aasir.com/query/query.php</a>
10	17	0.63% <a href="http://127.0.0.1/">http://127.0.0.1/</a>
11	17	0.63% <a href="http://localhost/">http://localhost/</a>
12	12	0.45% <a href="http://flash.aasir.com/shop2/default.htm">http://flash.aasir.com/shop2/default.htm</a>
13	10	0.37% <a href="http://aasir.com/">http://aasir.com/</a>
14	9	0.33% <a href="http://aasir.com/modules.php">http://aasir.com/modules.php</a>
15	9	0.33% <a href="http://flash.aasir.com/good/">http://flash.aasir.com/good/</a>
16	6	0.22% <a href="http://flash.aasir.com/icons/">http://flash.aasir.com/icons/</a>
17	6	0.22% <a href="http://flash.aasir.com/shop/register.htm">http://flash.aasir.com/shop/register.htm</a>
18	5	0.19% <a href="http://flash.aasir.com/example/%A%G%g/">http://flash.aasir.com/example/%A%G%g/</a>
19	5	0.19% <a href="http://flash.aasir.com/query.php">http://flash.aasir.com/query.php</a>
20	5	0.19% <a href="http://flash.aasir.com/query/">http://flash.aasir.com/query/</a>
21	5	0.19% <a href="http://flash.aasir.com/shop/links.htm">http://flash.aasir.com/shop/links.htm</a>
22	4	0.15% <a href="http://flash.aasir.com/manual/">http://flash.aasir.com/manual/</a>
23	3	0.11% <a href="http://flash.aasir.com/example/phorum/list.php">http://flash.aasir.com/example/phorum/list.php</a>
24	3	0.11% <a href="http://flash.aasir.com/shop/default_authenticated.htm">http://flash.aasir.com/shop/default_authenticated.htm</a>
25	2	0.07% <a href="http://flash.aasir.com/example/HTML/">http://flash.aasir.com/example/HTML/</a>
26	2	0.07% <a href="http://flash.aasir.com/manual/misc/security_tips.html">http://flash.aasir.com/manual/misc/security_tips.html</a>
27	2	0.07% <a href="http://flash.aasir.com/~justin/">http://flash.aasir.com/~justin/</a>

使用者使用 Windows98 作業系統來上網，而且使用 IE 瀏覽器上我們的網站。

Top 1 of 1 Total Search Strings			
#	Hits	Search String	
1	1	100.00%	php

Top 15 of 22 Total User Agents			
#	Hits	User Agent	
1	647	24.08%	Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
2	310	11.54%	Mozilla/5.0
3	132	4.91%	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
4	122	4.54%	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
5	61	2.27%	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.1) Gecko/20020
6	24	0.89%	Mozilla/4.0 (compatible; MSIE 5.01; Windows 98)
7	21	0.78%	Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
8	8	0.30%	ia_archiver
9	6	0.22%	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; KKman2.0)
10	4	0.15%	gnome-vfs/2.0.2
11	3	0.11%	FAST-WebCrawler/3.6 (atw-crawler at fast dot no; http://fast.
12	2	0.07%	Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)
13	2	0.07%	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; KKman2.0;
14	2	0.07%	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; T312461)
15	1	0.04%	BaiDuSpider



## 課後練習

1. APACHE2.0 伺服器的檔案是放在哪一個目錄下？ 底下的 conf 目錄，主要是放置 APACHE2.0 主要的組態設定檔 httpd.conf、Makefile 和安全協定的檔案；底下的 conf.d 組態目錄是放置 APACHE2.0 要載入的組態檔，包含了 php.conf、perl.conf、python.conf 和安全協定組態檔案 ssl.conf。

- (A). /etc/httpd
  - (B). /var/www/html
  - (C). /etc/rc.d
  - (D). D./root
- 

2. Apache 網站伺服器的主要組態檔是哪一個？

- (A). A./var/www/samba.conf
  - (B). B./usr/src/apache.conf
  - (C). C. /etc/httpd/conf/httpd.conf
  - (D). D./usr/src/httpd.conf
- 

3. 我們 APACHE 的組態設定檔 httpd.conf 是由全域環境組態指令、主伺服器組態指令和包含檔組態指令所組成。我們也可以將一些組態檔使用包含的方式將組態設定檔包含進來。請問包含檔組態指令為哪一個？

- (A). Documentroot
  - (B). include
  - (C). inside
  - (D). conclude
- 



4.為了讓 Apache 能夠滿足伺服器的需求服務,Apache 需能動態的改變伺服器正在執行的行程數量。下面是幾個我們必需指定的參數。

請問 StartServers 指令是何種意義?

- (A). 設定伺服器行程啟動的數量
- (B). 設定閒置行程的最小數量
- (C). 設定同時可允許啟動的行程數量
- (D). 設定同時可允許啟動的行程數量

5. MPM 為多行程模組(Multiprocessing modules)。而每個子行程的執行緒可以服務不同的請求。如果每個 Apache 可以產生 8 個子行程,而每個子行程又可以產生 25 個執行緒,則 Apache 可以產生  $8*25=1000$  個執行緒的服務。當行程是要增加或減少是由閒置執行緒來決定,而其行程是介於 MinSpareServers(最小閒置行程和 MaxSpareServers(最大閒置行程之間)。請問 Apache2 支援何種執行緒,Apache1.3 支援何種執行緒?

- (A). APACHE2 支援多執行緒, Apache1.3 支援單一執行緒。
- (B). APACHE2 支援多執行緒, Apache1.3 也支援多執行緒。
- (C). APACHE2 支援單一執行緒, Apache1.3 也支援單一執行緒。
- (D). APACHE2 支援單一執行緒, Apache1.3 支援多執行緒。

6.何種指令可以讓我們指定文件的語言?如果該文件沒有定義任何預設的語言,則當我們在 DefaultLanguage 指令設定為 tw 時,其文件的語言會以繁體中文出現(tw 為台灣)。

- (A). Addlanguage
- (B). Setlanguage
- (C). Listen
- (D). DefaultLanguage



7. Apache HTTP 伺服器提供何種模組介面給 OpenSSL 函式，該模組提供 Apache 伺服器強大的加密方法給傳輸層，讓我們的資料可以在安全的情況下傳輸？

- (A). mod\_php
- (B). mod\_secure
- (C). mod\_ssl
- (D). mod\_access

8. 我們可以設定當 Apache 伺服器發生錯誤時所記錄的一些資訊。預設我們是將這些 Apache 的錯誤資訊記錄到哪一個目錄？我們除了可以記錄錯誤的資訊，也可以設定警告記錄的等級和匿名的記錄。

- (A). A./var/www/html/error
- (B). B./usr/src/logs/error
- (C). C./usr/src/log/error\_log
- (D). D. /etc/httpd/logs/error\_log

9. 大大公司的銷售部門想和服務部門共用一台主機服務，那一種選擇最方便？

- (A). 使用 SSL 加密
- (B). 使用虛擬主機 virtual host
- (C). 使用 Webalizer
- (D). 使用 Include 指令

10. 大大公司的行銷部門想將廣告檔案放置給使用者下載，請問何種是可行的方式，複選？

- (A). Apache
- (B). Samba
- (C). FTP
- (D). DHCP



---

**【答案】**

1. A 2. C 3. B 4. A 5. A 6. D 7. C 8. D  
9. B 10. A與C



