

Enterprise Linux 實戰講座

RHEL 4 之 WWW 伺服器-Apache 2

Apache HTTP Server 是一個免費且 Open Source 的 HTTP 伺服器，主要目的在於提供一個安全並且有效率的 HTTP Server 環境。Apache HTTP Server 目前有 Unix-Based 和 Windows 的版本。根據統計，目前 Apache HTTP Server 已成為佔有率最高的 HTTP Server 軟體，而且它的功能與執行效率並不會比一般商業的 Web Server 軟體來的遜色，甚至領先許多。Apache HTTP Server 目前的版本分為 1.3.x 版與 2.0.x 版，兩個版本基本上功能都差不多，不一樣的地方在於 2.0.x 的執行效率會比 1.3.x 來得好，因為 2.0.x 結合了執行緒〈thread〉與程序〈process〉，讓熟悉系統的管理者能夠調校各種不同的組合，以達到最佳效能。

Apache 簡介

在網際網路初生時期，NCSA 研發了一個網站伺服器。大約在 1995 年左右，NCSA 的網站伺服器已成為世界排名第一的佼佼者。然而，同時期 NCSA 網站伺服器的主要研發群離開了 NCSA，結果這套網站伺服器的發展就延宕下來了。正因為如此，過去採用 NCSA 網站伺服器人開始彼此交換伺服器的修正程式，沒多久之後，眾人便體會出需要一個專門的組織來管理這些補漏程式(patch)才行；該團隊採用原先 NCSA 伺服器的程式碼，並將修補後的網站伺服器正式命名為 Apache。筆者一定會好奇為什麼會命名為 Apache，Apache 不是美國戰鬥直昇機嗎？之所以會命名 Apache 原因是因為這個網站伺服器是上了很多的 patch 的「A patchy server」。所以便取其諧音「Apache」。

Apache 的核心程式是來自於 NCSA 最初的伺服器，並搭配眾家自行開發出來的修正程式綜合而成。當今的 Apache 現已是網站伺服器社區(Web server community) 的主要討論話題了。短短三年，Apache 就獲得市場認同。

Apache 0.6.2 版於 1995 年四月釋出，而 1.0 版在 1995 年 12 月 1 日發表。Apache 團隊從成軍後，便不斷地擴張組織成員，進一步整合成公益團體；整個團隊都靠網際網路運作。Apache 雖然由 Apache 團隊發表，然而，Apache 團隊並沒有對 Apache 的發展定下任何的限制，任何人只要通曉事情該怎麼作，就可以參與 Apache 的開發或者內部元件的整修，Apache 團隊的成員都樂見其成。

不過，任何的更正或修補最後都得由 Apache 團隊審核通過才能加入 Apache 網站伺服器的標準版本，簡言之，Apache 團隊仍保留了 Apache 未來發展之路的掌控權。儘管如此，開放的理念使得世界各地的開發人員得以透過國際網路齊聚一堂，討論新的功能、修正錯誤的邏輯以及移植到別種作業平台等等。任何新生產出來的程式碼都得送交 Apache 團隊，經由成員檢視程式碼的細節、完成測試工作並執行嚴格的品管控制。如果團隊中的成員滿意新的程式碼，新的程式碼才能正大光明的登堂入室，融入 Apache 的標準範本。

Apache Server 是由 Apache Group 在開發與維護，1999.06.30 成立了 Apache Software Foundation (Apache 軟體基金會，簡稱 ASF)。從原本彙集各家的 Apache Group，到現在有幾家公司仍然持續的支持後續的發展計畫，使得 Apache Server 可以確保持續的發展。

Apache2.x 核心與模組新功能

自從 2000 年 3 月在 Apache Conference 2000 論壇發表第一個 Apache 2.0 的測試版後 (Alpha Version)，ASF 歷經了二年多的研究並發表數種測試版，終於在 2002 年 4 月 6 日正式推出 Apache 2.0。它較前一版的 1.3.24 最大的差異就在穩定度 (Scalability)，除此之外，尚包含以下許多的核心與模組新功能：

核心新功能

- **Unix Threading**
在支援 POSIX 執行緒 (Thread) 的 UNIX 系統上，Apache 目前可以執行混合式的多程序 (Multi-process) 和多執行緒 (Multi-thread) 模式，這有助於多種組態的穩定性 (但不適用於所有的組態)。
- **支援多種通訊協定**
Apache 2.0 可支援更多的通訊協定，以增加其網路功能。
- **新式的系統架構**
新式的系統架構已被改寫為 autoconf 與 libtool 的型態，這使用 Apache 的組態設定和其他套件更為相似。
- **對於非 Unix 平台有較佳的支援**

由於 MPM 和 Apache Portable Runtime (APR) 技術的使用，使得在 BeOs、os/2 和 Windows 等非 Unix 平台上，Apache 2.0 具有較快及較穩定的表現，因為這些平台可以使用原始的 API (Application Programming Interface)，以避免模擬 POSIX 環境時引起的效能下降。

- 新型的 API

在 2.0 版的 Apache 中，各模組的 API 有較明顯的改變，同時也改進了 1.3 版時的模組順序及優先等級的問題，因為 Apache 2.0 會自動處理有關先後順序的工作。

- 篩選功能 (Filtering)

Apache 模組現在已被改寫為 Filtering 類型，當它們作用在串流的內容時，就如同由伺服器接收或傳送一般。例如在「 mod_include 」模組中，可允許使用「 INCLUDES 」篩選器中的 SSI (Server Side Includes) 指令，以剖析 CGI 指令檔的輸出。

- 多種語言的錯誤回應訊息

利用 SSI 文件，管理員可以自訂不同與演版本的錯誤訊息，以回應至用戶端瀏覽器。

- 簡化的組態設定

在 Apache 2.0 中，簡化了許多容易混淆的指令，例如「 Port 」和「 BindAddress 」指令以取消，若要鏈結 (Binding) 伺服器的 IP 位址，只需要使用「 Listen 」指令，而利用「 ServerName 」指令所設定的伺服器名稱與連接埠號碼，僅適用於 Redirect 和 Virtual Host 的識別。

- IPv6 支援

根據預設值，在支援 APR 函式庫 IPv6 系統中，Apache 2.0 都可監聽 IPv6 的通訊連接埠。此外，〈 Listen 〉、〈 NameVirtualHost 〉和〈 VirtualHost 〉指令 (Directive) 也可支援 IPv6 中的數字位址字串，

例如「 Listen [fe80::1]:8080 」。

- 支援 Windows NT/2000/XP 萬國碼 (Unicode) 功能

安裝於 Windows NT 上的 Apache 2.0，使用「 UTF-8 」為所有檔案名稱的編碼方式，因此可以直接支援多種語系，但不支援 Windows95/98/ME。

- 正規表示法 (Regular Expression) 函式庫更新
因為 Apache 2.0 包含了「 Perl Compatible Refular Expression Library 」(PCRC)，所以目前在正規表示法中可使用較具效率的 Perl 5 語法。

模組新功能

- mod_auth_digest
使用 Share Memory 來支援程序間的交易快取，此外，使用 MD5 摘要試驗正法來驗證使用者帳號的正確性。但若網路中使用多種不同類型的瀏覽器時，不建議使用此模組的功能，因為當不支援 MD5 的瀏覽器存取時，Apache 伺服器將不會對其要求進行驗證。
- mod_auth_ldap
允許使用 LDAP 目錄來儲存 HTTP 基本驗證的資料庫。
- mod_auth_dbm
使用 DBM (DataBase Management) 檔案以驗證使用者身份，也就是利用使用者名稱與純文字密碼的對應方式，此功能與「 mod_auth_file 」模組相似。
- mod_autoindex
允許「目錄瀏覽」(Directory Indexing) 功能，所謂目錄瀏覽是指當網頁目錄沒有包含預設的網頁檔案時，例如 index.html，此功能便可以將此目錄中的所有檔案及目錄以排列的方式，就如同執行 Linux 中的「 ls 」或 Windows 中的「 dir 」指令一樣。也可以使用 HTML 標籤來設計清單的格式，並且支援排序的功能。
- mod_charset_lite
這是一個尚在實驗階段的模組，它可以在網頁傳送回用戶端前，指定來源物件及欲翻譯成的字元集，但它不直接翻譯 Data，而是告知 Apache 執行何種翻譯。
- mod_dav
此模組可以起用 WebDav (Web-base Distributed Authoring and Versioning) 的功能，它允許使用者存取貨管理位於遠端主機上的目

錄及檔案。

■ **mod_deflate**

此模組可提供「DEFLATE」輸出篩選器的功能，它可以在回應透過網路傳回用戶端前，先對檔案進行壓縮。

■ **mod_file_cache**

這個模組包含 Apache 1.3 中的「mod_mmap_static」模組功能，並且將某些經常被存取的網頁快取在個別的記憶體中，以提高快取的功能，這可減少伺服器啟動程序和回應要求的時間。

■ **mod_headers**

若需要自定由「mod_proxy」模組產生的 HTTP 表頭，您可以啟用 mod_headers 模組的功能，它會傳送額外的資訊給瀏覽器、在每個要求中附加時間標記，或是其他有用的資訊。

■ **mod_include**

這個模組是用於伺服器剖析的 HTML 文件（Server Side Includes，SSI），它可在網頁回應用戶端之前先傳送至篩選器，而由這個篩選器設定不同的變數（由 SSI 的設定所定義）。

■ **mod_negotiation**

新增的「ForceLanguagePriority」指令可用來確認用戶端接收的是單一的文件，而不是「NOT ACCEPTABLE」或「MULTIPLE CHOICES」的回應。此外，模組會先檢視用戶端要求中有關自己的資訊，然後傳送最適合或是尋找伺服器中最符合的檔案至用戶端，這種功能就稱為「內容協商」（Content Negotiation）。

■ **mod_proxy**

這個模組已被完全的改寫，使得 Apache 足以擔任 Proxy 或閘道器的角色，它可以代理 FTP、SSL、HTTP/1.1 等通訊協定，同時也可以連接到其它的 Proxy 模組。

■ **mod_ssl**

啟用 mod_ssl 模組可以提供 SSL（Secure Sockets Layer）v2/v3 和 TLS（Transport Layer Security）v1 等通訊協定的支援能力。

RHEL 4 Apache 相關套件及設定檔

Daemon:httpd

Daemon 類別：System V standalone daemon

所需套件：

➤httpd-*.rpm

➤system-config-httpd-*.rpm (httpd 圖形管理工具)

啟動 Script：/etc/rc.d/init.d/httpd

Port：80 (http)，443 (https)

設定檔：/etc/httpd/conf/httpd.conf，/etc/httpd/*

Log 檔：/var/log/httpd

安裝 Apache

在 RHEL 4 安裝 Apache 是件很容易的事，因為 RHEL 4 提供了一個很方便的 GUI 套件管理工具「**system-config-packages**」。只要以 root 的身份登入系統，鍵入 **system-config-packages** (圖 1)，勾選「網頁伺服器」即可，然後按「更新」，便會提示放入適當的光碟片以完成安裝 Apache 的工作。

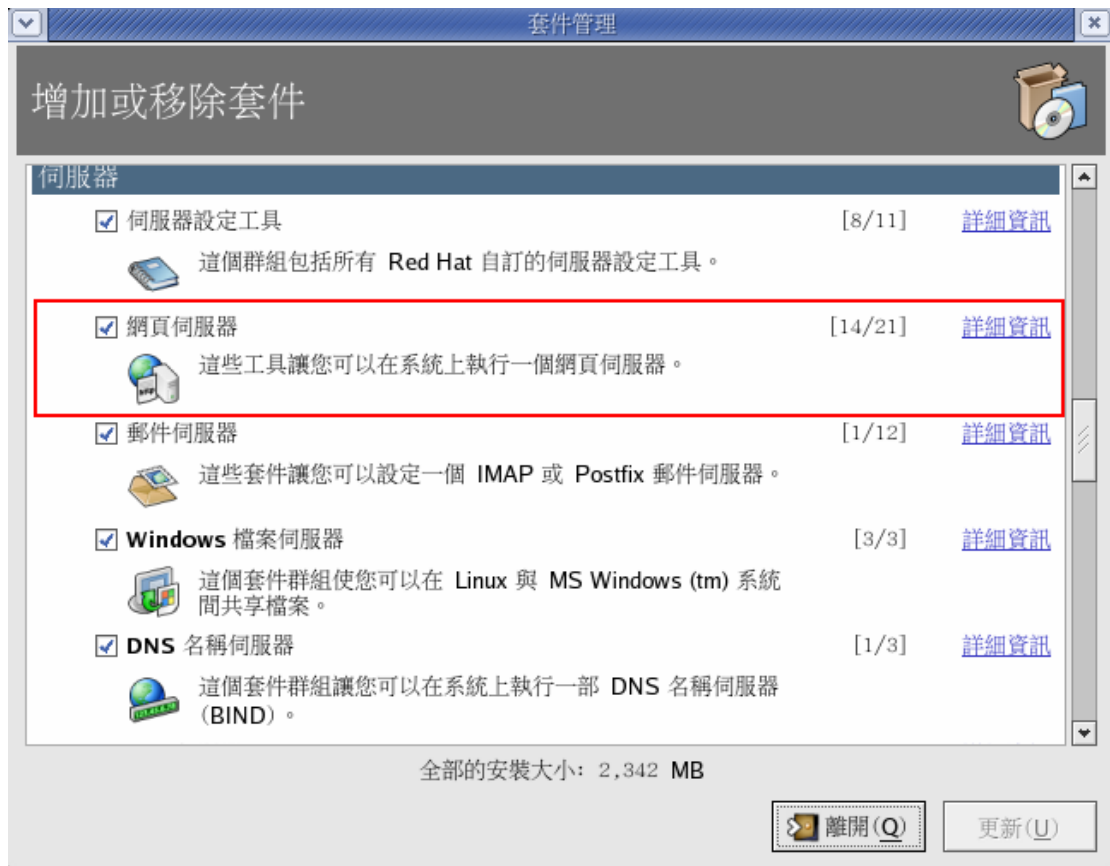


圖1、system-config-pacakges 畫面

待安裝後，可利用「rpm -qa | grpe '^httpd」檢查是否安裝成功。除了採用「system-config-packages」工具安裝外，亦可利用rpm -ivh 指令進行安裝。

```
[root@server1 ~]# rpm -qa | grep '^http'
httpd-2.0.52-9.ent
httpd-manual-2.0.52-9.ent
httpd-suexec-2.0.52-9.ent
```

要建置一個有內容的網站，除了安裝 Apache 外，html 網頁的製作撰寫也是不可或缺的一環。

HTML 就是超文件標示語言 (Hyper Text Markup Language)。主要是在瀏覽器中顯示一份文件的內容。它本身並非程式設計，只不過是標示(Mark-up)，用來強調及組織一般性的文字。HTML 是由一些標記 (tags) 所構成，每個標記對瀏覽器下一個特定的指令，這些指令會告訴瀏覽器如何顯示文件的內容。

HTML是由很多的文件標記(Document Tags)所組成，這些標記在你建立的每一個HTML文件都會用到。它們會對文件檔中不同的部份作不同的定義。大

部分的標記都是成對的(圍堵標記 Container Tags),即包含『起始標記』和『結束標記』。如<HTML>表示網頁的開始,而</HTML>代表網頁的結束。起始標記和結束標記差別就在於前面的斜線符號而已。標記本身沒有大小寫的區分,不過建議是使用大寫字母,以便在文字編輯器中容易識別。

不過本篇文章並不是教導網頁寫作的書籍,筆者採用幾個的實用的範例來實作 Apache 的基本功能。詳細 html 的語法,讀者可以參考相關的書籍。

實戰演練一、公司網頁

需求: <http://公司網頁主機>出現公司網站首頁

一般企業建置WWW Server就是希望客戶打開瀏覽器,在位址欄內鍵入公司的主機名稱時可看到該公司所設計的網頁。本演練的目的就是告訴讀者該將公司網站的首頁命名為何?置於何目錄?利用瀏覽器開啟<http://公司網頁主機>便可看到公司網站的首頁。

實作環境:

RedHat Enterprise Linux ES 4.0

SELinux關閉(可利用system-config-securitylevel工具關閉SELinux)

實作步驟:

步驟 1: 啟動 httpd server

讀者可利用下列指令啟動httpd server:

```
[root@server1 ~]# service httpd start
Starting httpd: [ OK ]
```

亦可直接執行httpd system V script

```
[root@server1 ~]# /etc/init.d/httpd start
Starting httpd: [ OK ]
```

待server啟動成功後,打開瀏覽器,在位址欄鍵入<http://localhost>便可看到如圖2的Apache預設網頁。若想重新開機時,系統預設會啟動Apache可執行下列指令:

```
[root@server1 ~]# chkconfig httpd on
```

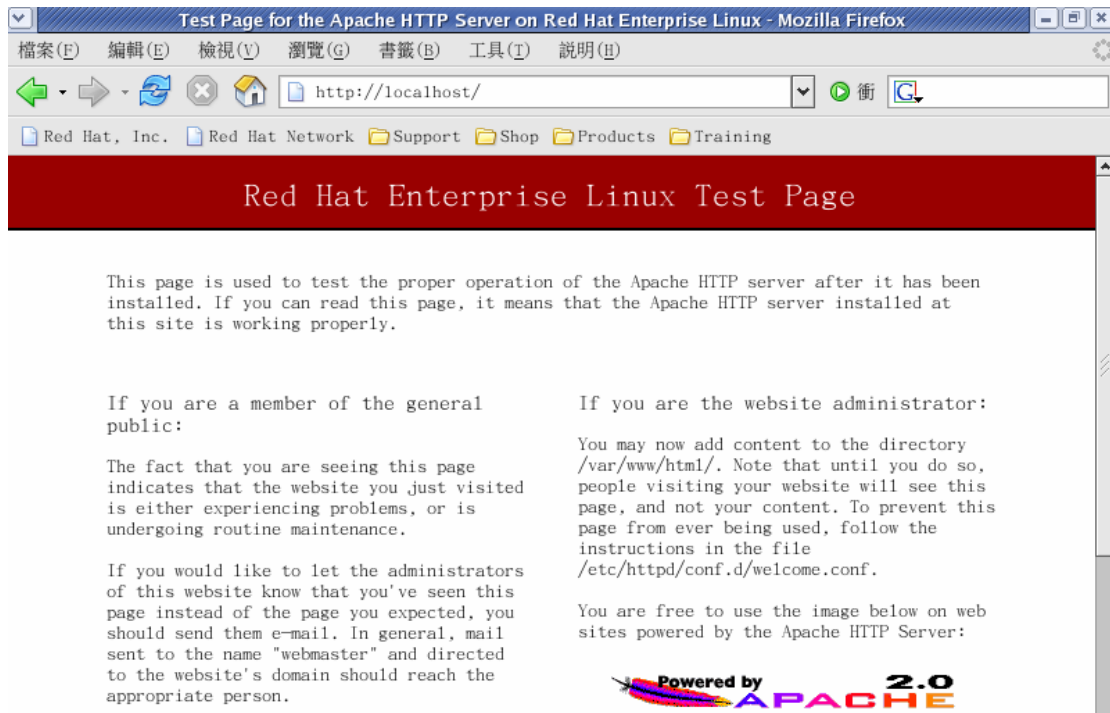



圖2、Apache預設網頁

步驟 2：修改預設網頁

讀者只需到/var/www/html（預設的DocumentRoot）下建立index.html。則<http://localhost>所看到的畫面就會是index.html的內容（圖3）。

```
[root@server1 ~]# cat /var/www/html/index.html  
Alex Lin's Homepage
```



圖3、修改Apache預設網頁

但若是index.html內有中文則遭遇無法正常顯示的問題。例如筆者利用記事本撰寫index.html內容如圖4所示，然後利用FTP方式上傳至WWW Server上的/var/www/html目錄內，打開Linux 上的瀏覽器Mozilla Firefox或Windows上的瀏覽器IE觀看此網頁，都是出現一行亂碼（圖5）。原因是RHEL 4上的Apache預設採用萬國碼（UTF-8），而使用記事本或一般的網頁編輯軟體存檔時預設是用Big5碼儲存文件，所以才會無法正常顯示。解決這個問題的方法有兩種：

方法一：利用UTF-8編碼方式儲存網頁

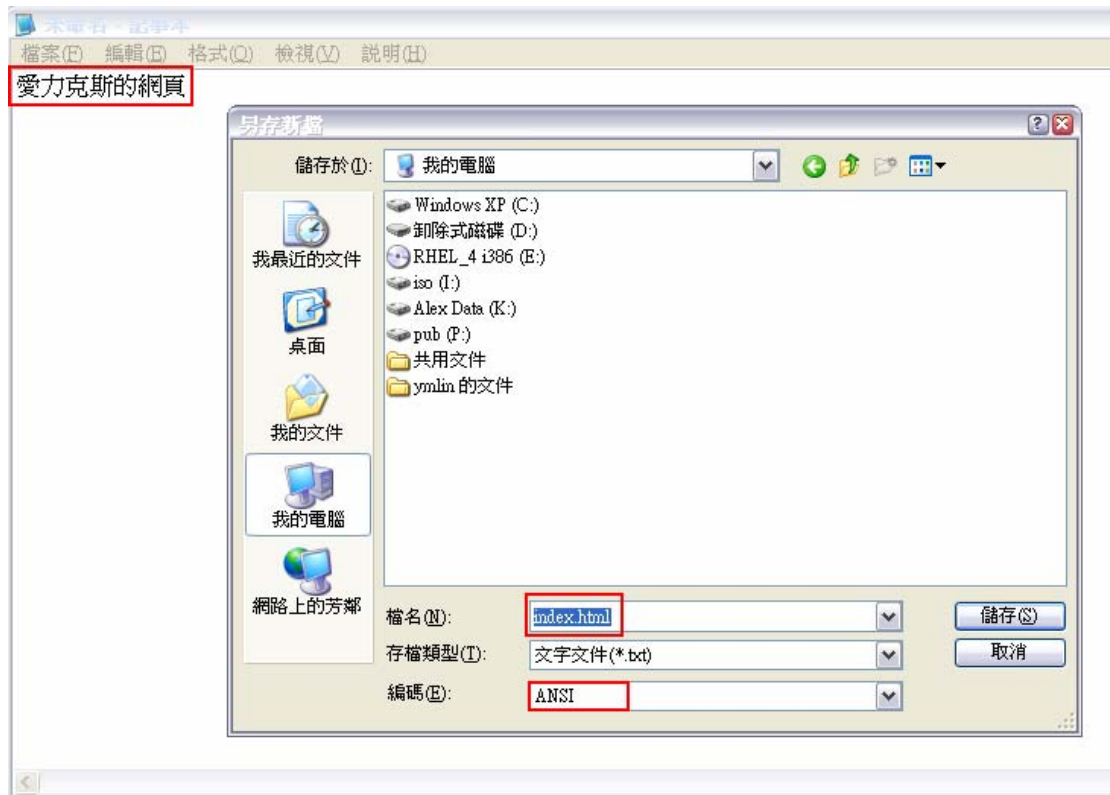


圖4、利用記事本編寫index.html



圖5、無法正常顯示含有中文字的網頁

筆者再利用記事本打開index.html，但存檔時在編碼欄位選「UTF-8」(圖6)，然後再利用FTP方式上傳至WWW Server上的/var/www/html目錄內，此時用瀏覽器觀看<http://localhost> 便正常顯示中文(圖7)。

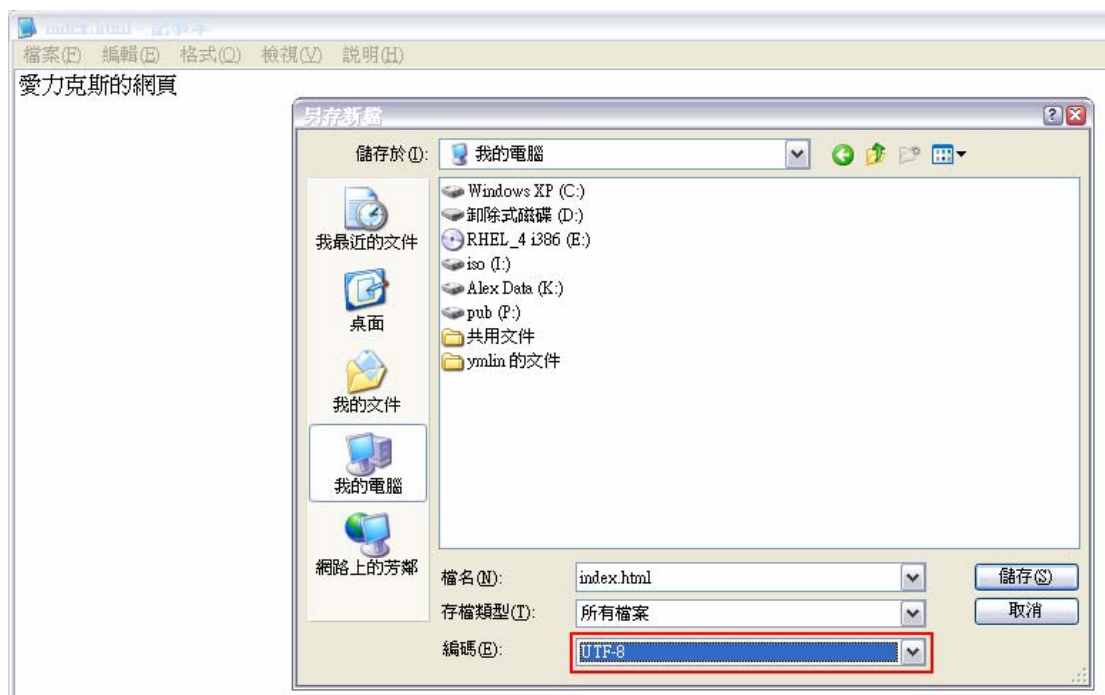


圖6、選擇UTF-8方式存放index.html

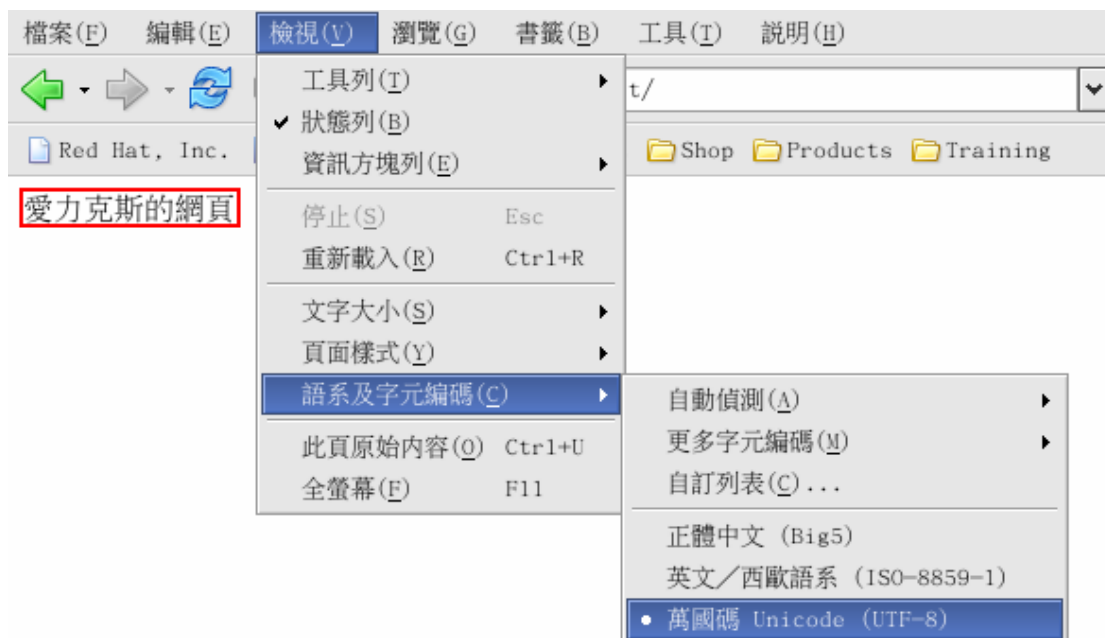


圖7、正確的顯示具有中文字的網頁

方法二：修改Apache上預設的編碼方式

方法一雖可解決中文網頁的問題，在現今的環境卻是個不可行的解法。因為很多網頁編輯軟體預設就是用Big5碼儲存網頁；還有網際網路上現存的中文網頁也是採用Big5編碼方式。所以另一個解決的方法便是修改Apache上預設的編碼方式。讀者可修改/etc/httpd/conf/httpd.conf中約730行的「AddDefaultCharset UTF-8」。

若是在此行前面加上#，註解掉此設定值，並執行「server httpd restart」重新啟動WWW伺服器。則會採用Browser上的語言設定值。（註：IE此設定值存放於「工具」→「網際網路選項」→「一般」→「語言」）。

或者是將此行改為「AddDefaultCharset Big5」，則瀏覽器將用Big5方式來解讀網頁內容。

```
[root@server1 ~]# vi /etc/httpd/conf/httpd.conf
```

```
...
```

```
721 # Specify a default charset for all pages sent out. This is
722 # always a good idea and opens the door for future internationalisation
723 # of your web site, should you ever want it. Specifying it as
724 # a default does little harm; as the standard dictates that a page
725 # is in iso-8859-1 (latin1) unless specified otherwise i.e. you
726 # are merely stating the obvious. There are also some security
727 # reasons in browsers, related to javascript and URL parsing
728 # which encourage you to always set a default char set.
729 #
```

```
730 AddDefaultCharset UTF-8
```

```
1.在AddDefaultCharset UTF-8加上#
```

```
2.或改為AddDefaultCharset Big5
```

```
731
```

```
732 #
```

實戰演練二、個人網頁

需求：<http://主機名稱/~使用者名稱>或<http://主機名稱/使用者名稱>出現個人網頁

學校或各ISP有時會提供空間給個人存放網頁，通常他們只是告訴使用者首頁必需命名為index.html或default.htm，然後放到某目錄下。然後在瀏覽器鍵入<http://主機名稱/~使用者名稱>（網頁主機通常是Unix Like）或<http://主機名>

[稱/使用者名稱](#)（網頁主機可能是 Windows）便可看到使用者專屬的網頁。本演練目的便是告訴讀者要開放使用者可自製網頁，需要那些設定工作。

步驟 1：Enable UserDir

Apache 預設關閉（disable）使用者可設定自己的網頁目錄（UserDir）的功能，所以首先要將此設定註解掉。約在 httpd.conf 第 339 行處，在此行前加上 # 符號，然後重新啟動 httpd。

```
[root@server1 ~]# vi /etc/httpd/conf/httpd.conf
```

```
335     # UserDir is disabled by default since it can confirm the presence
336     # of a username on the system (depending on home directory
337     # permissions).
338     #
339     # UserDir disable
```

```
[root@server1 ~]# service httpd restart
```

步驟 2：建立 UserDir 及 index.html

轉換至一般使用者身份，在其家目錄中建立 public_html（預設的 UserDir）子目錄，並於 public_html 目錄內撰寫 index.html。最後開放 Other 身份有進入家目錄的權限（即對目錄設定 x permission）；因為 Apache daemon 是利用 apache 這個身份到使用者家目錄下的 public_html 子目錄讀取網頁，但預設 /home/neo 的預設 permission 為「rwx-----」，apache 這個使用者是無法進入 /home/neo，所以必須執行「chmod o+x /home/neo」讓 apache 使用者可進入至 /home/neo/public_html 目錄讀取 index.html。

```
[root@server1 ~]# su - neo
[neo@server1 ~]$ mkdir public_html
[neo@server1 ~]$ cat > public_html/index.html
Neo's Homepage
← Ctrl+D，利用 cat > filename 方式建立檔案，最後按下「Ctrl+D」結束編輯。
[neo@server1 ~]$ chmod o+x /home/neo
```

此時在在瀏覽器鍵入在瀏覽器鍵入 <http://主機名稱/~neo> 便可看到 neo 的個人網頁。

步驟 3：建立 /var/www/html/neo soft link

為了在瀏覽器鍵入<http://主機名稱/neo>同樣可看到neo的個人網頁，筆者便建立/var/www/html/neo soft link 指向/home/neo/public_html

```
[root@server1 ~]# ln -s /home/neo/public_html/ /var/www/html/neo
[root@server1 ~]# ls -l /var/www/html/neo
lrwxrwxrwx  1 root root 22 Apr 25 08:26 /var/www/html/neo ->
/home/neo/public_html/
```

實戰演練三、網頁目錄保護機制（.htaccess 目錄認證保護）

需求：<http://主機名稱/exam>出現輸入帳號與密碼認證對話框

讀者一定有過進入某個網站之後，按下某些連結，竟然出現一個對話框，告訴你要登入該目錄，需要輸入帳號與密碼才能登入的經驗。要實作此功能的方法，可利用Apache所提供的『.htaccess目錄認證保護』機制。這種認證的模式彈性很大，不用寫程式控制就可限制某些帳號及密碼才可進入此目錄，使網頁資料只有註冊使用者才可存取。

步驟 1：建立/var/www/html/exam 目錄

既然是『進入某個目錄之後，才會出現認證對話視窗』，那麼首先當然就是要要有那個設定為認證網頁的『目錄』！而<http://主機名稱/exam/>實際上對應系統的 /var/www/html/exam。所以首先建立/var/www/html/exam目錄，並在此目錄下編輯index.html。

```
# mkdir /var/www/html/exam
# echo ".htaccess hands-on Lab" > /var/www/html/exam/index.html
```

步驟 2：編輯 httpd.conf 指定/var/www/html/exam 採用 .htaccess 認證

接下來，當然就是到 httpd.conf 檔案中去指定保護 /var/www/html/exam 這個目

錄。在檔案最後加入下列文字，然後重新啟動 httpd 伺服器。

```
[root@server1 ~]# vi /etc/httpd/conf/httpd.conf
```

...

```
<Directory "/var/www/html/exam">
    Allowoverride authconfig
</Directory>
```

```
[root@server1 ~]# service httpd restart
```

步驟 3：編寫 .htaccess

在 /var/www/html/exam 下產生一個 .htaccess 檔，內容如下：

```
AuthName ".htaccess test" ← 這個是顯示在視窗上面的提示字元
AuthType Basic ← 認證的類型
AuthUserFile /var/www/html/exam/wwwpasswd ← 密碼檔放置的地方
require valid-user
```

```
[root@server1 ~]# vi /var/www/html/exam/.htaccess 內容如下：
```

```
AuthName ".htaccess Lab"
AuthType Basic
AuthUserFile /var/www/html/exam/wwwpasswd
require valid-user
```

步驟 4：建立密碼檔及設定可進入此目錄的帳號及密碼

```
[root@server1 ~]# htpasswd -c /var/www/html/exam/wwwpasswd alex
第一次執行 htpasswd 指令時，因為還未有 wwwpasswd 檔案，所以必須指定 -c
(create) 參數以建立 wwwpasswd 檔案。
```

```
New password:
```

```
Re-type new password:
```

```
Adding password for user alex
```

```
[root@server1 ~]# htpasswd /var/www/html/exam/wwwpasswd bryan
```

```
New password:
```

```
Re-type new password:
```

```
Adding password for user bryan
```

```
[root@server1 ~]# htpasswd /var/www/html/exam/wwwpasswd eric
```

```
New password:
```

```
Re-type new password:
```

```
Adding password for user eric
```

```
[root@server1 ~]# cat /var/www/html/exam/wwwpasswd
```

```
alex:Hq4RNltrxFRc.
```

```
bryan:VjRgiOoWCvRUI
```

```
eric:eiRxZmeovP61o
```

步驟5：測試

打開瀏覽器在位址欄輸入http://localhost/exam時就會出現目錄認證視窗（圖8），必須輸入步驟4所建立的帳號及對應的密碼才可進入此網頁目錄。

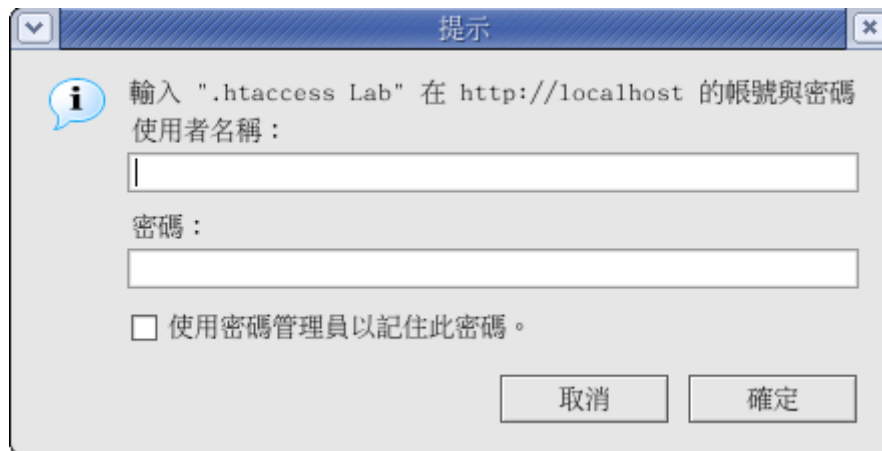


圖8、.htaccess認證視窗

彥明有感：

廣告詞：「小時候想趕快長大，想談戀愛，想獨立。當我長大了，卻想回家！以家的價值，啟動前進的動力。」。最近可能是因為工作量太大，右手竟不時會有發麻的情形發生。去年筆者的父親生了一場重病，今年母親身體也出了些狀況，深深感受到身體健康的重要，也讓筆者驚覺父母的年事已高，需人照料。以家的價值，啟動前進的動力，也期許自己能早點有能力及時間好好照顧家人，和大家互勉之！