

## 附錄一、PEAR 套件的介紹，安裝，管理，與使用

### 一、PEAR 套件簡介

由於參與開發套件的團隊人數高達一百四十八人，且還在不斷增加中，所以套件更新的速度非常快，而且不斷有全新功能的套件推出。目前為止共 32 類 219 個套件，以下將各類與各套件的概略功能以表格的方式條列出來：

表附一.1 PEAR 套件類別-權限認證類(Authentication)

套件名稱	套件功能概述
Auth	建立一個認證系統
Auth Enterprise	(尚未釋出)
Auth HTTP	用 HTTP 的方式做認證
Auth PrefManager	認證偏好(Preferences)的管理
Auth RADIUS	給 RADIUS PECL 用的包裹器
Auth SASL	各種不同的 SASL 機制的抽象層
LiveUser	提供使用者認證與權限管理

目前共有 7 個套件

表附一.2 PEAR 套件類別-執行時間測量類(Benchmarking)

套件名稱	套件功能概述
Benchmark	用來計算一個網頁程式執行完畢所需的時間。

目前共有 1 個套件

表附一.3 PEAR 套件類別-網頁快取類(Caching)

套件名稱	套件功能概述
Cache	快取各式各樣的資料
Cache Lite	快速且安全的小型快取系統

目前共有 2 個套件

表附一.4 PEAR 套件類別-設定類(Configuration)

套件名稱	套件功能概述
Config	幫助管理設定值的工具

目前共有 1 個套件

表附一.5 PEAR 套件類別-控制介面類(Console)

套件名稱	套件功能概述
Console Color	此類別可讓你在應用程式中使用 console 的 ANSI 顏色
Console Getopt	文字指令的解析
Console ProgressBar	此類別提供一個易用的進行狀態列
Console Table	可簡單的建立一個 console 樣式的表格
System Command	文字命令的執行介面

目前共有 5 個套件

表附一.6 PEAR 套件類別-資料庫類(DataBase)

套件名稱	套件功能概述
DB	資料庫的抽象層
DBA	柏克萊式資料庫的抽象層
DBA Relational	也是柏克萊式資料庫的抽象層
DB ado	用在微軟的 ADODB 的資料庫驅動層
DB DataObject	一個用來建立資料庫的 SQL 語法的物件介面
DB DataObject FormBuilder	可以從 DB_DataObject 類別自動建立 HTML_QuickForm 物件的類別
DB ldap	用來連上 LDAP 伺服器的資料庫介面
DB ldap2	用來連上 ldap v2 或 ldap v3 的資料庫驅動層
DB NestedSet	用來建立或存取巢狀結構的應用程式

DB Pager	用來接收或傳回資料庫存取結果的訊息
DB QueryTool	用來接收與設定資料庫中資料屬性的一個物件介面
Gtk MDB Designer	一個 Gtk 資料庫樣式設計的類別
MDB	資料庫的抽象層
MDB2	資料庫的抽象層
MDB QueryTool	一個簡易的接收與設定資料庫的資料屬性的物件導向介面
SQL Parser	SQL 語法的解析工具

目前共有 16 個套件

表附一.7 PEAR 套件類別-日期與時間類(Date and Time)

套件名稱	套件功能概述
Calendar	用來建立月曆的日期架構
Date	日期與時區的設定

目前共有 2 個套件

表附一.8 PEAR 套件類別-加密類(Encryption)

套件名稱	套件功能概述
Crypt CBC	用來模擬 perl 的 CBC 加密方式的類別
Crypt CHAP	用來產生 CHAP 包裹
Crypt Crypt	加密演算法的抽象類別
Crypt HMAC	用來計算符合 RFC2104 的雜湊的類別
Crypt RC4	做 RC4 加密的類別
Crypt Xtra	用在 TEA 加密演算法的類別
Message	會隨機產生提示訊息的類別

目前共有 7 個套件

表附一.9 PEAR 套件類別-檔案格式類(File Formats)

套件名稱	套件功能概述
Contact Vcard Build	建立或抓取 vCard 2.1 或 3.0 檔的文字區塊
Contact Vcard Parse	用來解析 vCard 2.1 和 3.0 的檔案
File DICOM	用來閱讀與設定 DICOM 檔的套件
File IMC	建立並解析混合式的電子郵件檔
File Ogg	存取 Ogg 串流
Genealogy Gedcom	Gedcom 檔的解析
MP3 ID	讀寫 MP3
Spreadsheet Excel Writer	用來產生 Excel 空白工作表的套件

目前共有 8 個套件

表附一.10 PEAR 套件類別-檔案系統類(File System)

套件名稱	套件功能概述
Archive Tar	用來管理 tar 檔的套件
Archive Zip	用來管理 zip 檔的套件
File	提供一般檔案與目錄常用的功能
File Find	用來做檔案搜尋的類別
File HtAccess	操作.htaccess 檔的套件
File Passwd	操作許多種 password 檔的套件
File SearchReplace	可執行檔案中搜尋與取代的工作
File SMBPasswd	用來管理 SAMBA 型態的 password 檔
VFS	虛擬的檔案系統的應用程式

目前共有 9 個套件

表附一.11 PEAR 套件類別-GTK 元件類(Gtk Components)

套件名稱	套件功能概述
Gtk VarDump	一個簡單的視窗介面(GUI)用來展示 PHP 資料的樹狀結構

目前共有 1 個套件

表附一.12 PEAR 套件類別-超文字標記語言類 (HTML)

套件名稱	套件功能概述
HTML BBCodeParse	這是一個語法解析器，用來將 UBB 樣式的標籤轉換為意義相同的 html 標籤
HTML Common	是給本大類中其他類別引用的 HTML 基本類別
HTML Crypt	用來將文字加密，可在 client 端用 javascript 解密
HTML CSS	用來產生 CSS 宣告的類別
HTML Form	簡單的 HTML 表單製作
HTML Javascript	提供一個建立簡單的 javascript 程式碼的介面
HTML Menu	可從多方面隨機產生 html 導覽列
HTML Page	用來產生 xhtml 網頁
HTML Progress	可以快速又簡單的引入一個執行狀態百分比的動態長條列
HTML QuickForm	此類別可提供方法來建立，檢驗，執行 html 表單
HTML QuickForm Controller	可以在 HTML_QuickForm 套件中建立「多頁式」的表單
HTML Select	可產生 HTML 表單的 select 元件
HTML Select Common	可處理 HTML 表單的 select 元件
HTML Table	可彈性的，簡單的設計 HTML 表格
HTML Table Sortable	可建立「可排序的」表格
HTML Template Flexy	一個相當有威力的網頁樣板的引擎
HTML Template IT	一種網頁樣板的類別
HTML Template PHPLIB	用在樣版系統的 preg_* 函式，用來解析語法。

HTML Template Sigma	一種網頁樣板的系統
HTML Template Xipe	一種簡單，但威力強大的樣板引擎
HTML TreeMenu	一個建立 HTML 樹狀導覽列的工具
Pager	可將資料編頁
Pager Sliding	可將資料編頁並頁碼列可滑動

目前共有 23 個套件

表附一.13 PEAR 套件類別-超文件傳輸協定類(HTTP)

套件名稱	套件功能概述
HTTP	各式各樣的與 http 相關的函式
HTTP Client	用來執行多種 http requests 且產生其結果
HTTP Download	傳送 http 的下載
HTTP Header	可用來修改網頁的 http 表頭
HTTP Request	提供一個簡單的方法來執行 http 的 requests
HTTP Server	可做 HTTP server
HTTP Session	可提供 session_* 函式的物件
HTTP Upload	可透過 HTML 表單簡單又安全的將檔案上傳
HTTP WebDAV Client	WebDEV 的串流的包裹器
HTTP WebDAV Server	WebDAV 伺服器的基本類別

目前共有 10 個套件

表附一.14 PEAR 套件類別-圖案處理類(Images)

套件名稱	套件功能概述
Images Barcode	可產生條碼
Image Color	管理與處理顏色相關資訊
Image GIS	將地理資訊系統的資料變成圖案
Image Graph	將數字轉換成圖案
Image GraphViz	用在一種 AT&T 的圖形工具的介面

Image IPTC	抽出，修改，儲存 IPTC 的資料
Image Remote	接收遠端的圖形檔的資訊
Image Text	可處理「圖型文字」
Image Transform	對不同的凸庫，可提供相同的介面來處理圖形

目前共有 10 個套件

表附一.15 PEAR 套件類別-國際化類(Internationalisation)

套件名稱	套件功能概述
I18N	將網頁變成多語言通用
Translation	用來處理多語言網站的管理
Translation2	多語言網站應用程式的管理

目前共有 3 個套件

表附一.16 PEAR 套件類別-紀錄類(Log)

套件名稱	套件功能概述
Log	將網頁的訊息紀錄下來
Log Parser	用在許多種紀錄檔的語法解析

目前共有 2 個套件

表附一.17 PEAR 套件類別-郵件類(Mail)

套件名稱	套件功能概述
Mail	提供多種寄信的介面
Mail Mbox	解析 Unix 的 mailbox 的語法
Mail Mime	提供一個針對 mime 的建立與解碼的類別
Mail Queue	大量郵件傳送的暫存佇列

目前共有 4 個套件

表附一.18 PEAR 套件類別-數學類(Math)

套件名稱	套件功能概述
------	--------

Math Basex	將數字型態轉換為文字型態
Math Complex	定義複雜的數字與操作
Math Fibonacci	計算與處理費柏南西數列
Math Histogram	計算長條圖的分佈比例
Math Intrger	處理整數
Math Matrix	處理矩陣
Math Quaternion	處理四元數
Math RPN	Reverse Polish Notation
Math Stats	統計學上計算
Math TrigOp	補充三角函數
Math Vector	處理向量

目前共有 11 個套件

表附一.19 PEAR 套件類別-網路類(Networking)

套件名稱	套件功能概述
Net CheckIP	檢查 Ipv4 的語法格式是否正確
Net Curl	提供一個 PHP 的 Curl 延伸功能
Net Dict	DICT 協定的介面
Net Dig	一個好用的 dig 命令的介面
Net DIME	DIME 編碼的工具
Net DNS	與 DNS 伺服器溝通的解析工具庫
Net Finger	提供一個存取 finger sever 的工具
Net FTP	可方便的與 ftp-server 溝通
Net Geo	由網路位址圖形化所在位置
Net Ident	鑑別協定(Identification Protocol)的工具
Net IMAP	提供一個 imap 協定的工具
Net Ipv4	IPv4 網路計算與驗證
Net Ipv6	驗證與使用 Ipv6
Net IRC	IRC 的客戶端類別
Net DNS	與 DNS 伺服器溝通的解析工具庫

Net LDAP	尋找與處理 LDAP 實體 (LDAP-entries) 的物件導向介面
Net LMTP	提供一個 RFC2033 LMTP 協定的工具
Net NNTP	與 NNTP 伺服器溝通
Net Ping	執行 ping 指令
Net POP3	提供一個存取 POP3 的類別
Net Portscan	掃描 port 的工具
Net Server	一般伺服器的類別
Net Sieve	處理與 timsieved 的交談
Net SmartIRC	與 IRC 網路的溝通
Net SMTP	提供一個 SMTP 協定的工具
Net Socket	網路的 socket 介面
Net Traceroute	執行 traceroute 指令
Net URL	簡單解析 Urls 語法
Net UserAgent Detect	用來判斷瀏覽器的種類，版本，平台
Net UserAgent Mobile	可移動的 http 使用者精靈字串的解析
Net Whois	提供一個詢問網域名稱 (domain name) 的工具

目前共有 30 個套件

表附一.20 PEAR 套件類別-數字類 (Numbers)

套件名稱	套件功能概述
Numbers Roman	將其他資料型態轉變為羅馬數字
Numbers Words	提供方法將文章中有多少字算出來

目前共有 2 個套件

表附一.21 PEAR 套件類別-付款類 (Payment)

套件名稱	套件功能概述
Payment Clieop	若有荷蘭銀行戶頭，可送資訊到該

	戶頭中
Payment DTA	建立一個含交易金額資料的檔
Payment Process	建立一個單一的付款程序

目前共有 3 個套件

表附一.22 PEAR 套件類別-PEAR 類

套件名稱	套件功能概述
PEAR	PEAR 的最底層套件
PEAR Frontend Gtk	Gtk 介面的 PEAR 套件管理
PEAR Fronted Web	網頁介面的 PEAR 套件管理
PEAR Info	可顯示出已安裝套件的訊息
PEAR PackageFileManager	可依每個套件中的 package.xml 檔來判斷版本訊息並做為是否可更新套件的依據，或將已安裝或移除的套件做記錄。

目前共有 5 個套件

表附一.23 PEAR 套件類別-PHP 類

套件名稱	套件功能概述
Inline C	允許線上(inline)引入 C 語言中定義的函式
PHPUnit	正規表達式(Regression)判別的架構
PHP Parse	PHP 語法的解析器
Validate	輸入的資料是否合乎有效性的驗證
Var Dump	將變數中所含的資料顯示出來的方法

目前共有 5 個套件

表附一.24 PEAR 套件類別-執行狀態類(Processing)

套件名稱	套件功能概述
------	--------

FSM	有限的狀態機器(Finite State Machine)
-----	-------------------------------

目前共有 1 個套件

表附一.25 PEAR 套件類別-科學類(Science)

套件名稱	套件功能概述
Science Chemistry	可處理化學的物件如原子，分子。等

目前共有 1 個套件

表附一.26 PEAR 套件類別-串流類(Stream)

套件名稱	套件功能概述
Stream SHM	可分攤記憶體的串流
Stream Var	允許串流對變數做一些基本的存取

目前共有 2 個套件

表附一.27 PEAR 套件類別-結構類(Structures)

套件名稱	套件功能概述
Games Chess	建構且驗證一個符合邏輯的西洋棋遊戲
OLE	讀寫 OLE 的容器
Text Statistics	計算文件中可以讀取的索引(indexs)
Tree	可將資料庫與 XML 作樹狀結構管理

目前共有 4 個套件

表附一.28 PEAR 套件類別-系統類(System)

套件名稱	套件功能概述
System ProcWatch	可監看系統的執行緒(Process)
Stream Var	允許串流對變數做一些基本的存取

目前共有 2 個套件

表附一.29 PEAR 套件類別-文字類(Text)

套件名稱	套件功能概述
Text Password	用 php 可產生密碼
Text Wiki	在文章結構中將 Wiki 標記抽象地解析與翻譯。

目前共有 2 個套件

表附一.30 PEAR 套件類別-工具與效能類(Tools and Utilities)

套件名稱	套件功能概述
PECL Gen	用來將 XML 轉為 PECL 擴增功能的工具
PhpDocumentor	從程式碼的注解用 php 程式自動產生說明文件

目前共有 2 個套件

表附一.31 PEAR 套件類別-網站服務類(Web Services)

套件名稱	套件功能概述
Service ExchangeRates	可執行貨幣的轉換
Services Weather	各種線上氣象服務的介面
SOAP	以 PHP 提供 SOAP 的 Client 與 Server
SOAP Interop	SOAP 可在不同平台運行的測試程式
UDDI	PHP 的 UDDI
XML RPC	XML RPC 協定的 PHP 工具

目前共有 6 個套件

表附一.32 PEAR 套件類別-XML 類

套件名稱	套件功能概述
DTD	可解析與驗證 xml 的 DTD 檔
XML Beautifier	可格式化 xml 文件

XML CSSML	提供建立 xml 的 css 的方法
XML DTD	可解析與驗證 xml 的 DTD 檔
XML fo2pdf	藉由 apache-fop 的幫助將 xsl-fo 檔轉為 pdf 或 ps 或 pcl, text, etc 檔
XML FOAF	可處理 FOAF 的 RDF 檔與 XML 檔
XML HTMLSax	可解析 HTML 和其他一些不合乎格式的 XML 文件的以 SAX 為基礎的解析器
XML image2svg	將圖案轉為 SVG
XML NITF	可解析 NITF 文件
XML Parser	可解析 XML 檔
XML RDDL	可以讀 RDDL(Resource Directory Description Language)文件
XML RSS	RSS 的解析器
XML SaxFilters	用 SAX 的應用程式建立的 XML filters
XML Serializer	讀寫 xml 檔的好用小工具，可將 data 轉為 xml 檔，反之亦然
XML sql2xml	從 SQL 的 Query 傳回 XML 格式的資料
XML SVG	XML_SVG 的應用程式
XML Transformer	XML 轉換成 php
XML Tree	將 xml 的資料用樹狀結構表現
XML UTil	一些 XML 工具
XML XPath	可提供 XPath 或 DOM 的 XML 操作與存取
XML XSLT Wrapper	對不同的 XSLT 介面或指令提供一個單一介面

目前共有 21 個套件

由以上的條列表格可以發現，為數十分龐大的 PEAR 套件所提供的功能涵蓋的範圍非常廣泛，並且與其他許多網路或通訊協定都可互相支援，例如：簡單認證與安全層(Simple Authentication and Security Layer, SASL)，輕量級目錄存取協定(Lightweight Directory Access Protocol)，延伸標記語言(Extensible Markup Language, XML)…等，不過這許多支援其他協定的套件，需要配合該協定的一些應用程式的安裝與設定，這些資訊在每個套件的說明文件上都會有記載。

這 219 個套件皆可從網站直接下載，解壓縮與解 tar 後，可看到第一層目錄中皆會有一個 package.xml 檔，這個檔是用來作套件的安裝與升級的比對之用，內容記載著所下載套件的詳細屬性，包括：套件名稱，版本編號，套件內所含的資料與函式。當升級套件時會由此檔判斷你的套件是否已升級到最新版。而第二層目錄中包括了該套件的主程式，以及範例程式與使用安裝說明。主程式皆以「物件導向性(Object-Oriented Programs, OOP)」方式撰寫，方便引用與延展，而範例程式與說明，是幫助使用者如何使用該套件的**最佳幫手**。

## 二、PEAR 基本套件與指令的安裝

如果是安裝 PHP4.3.0 以上版本的伺服器，就已內建基本的 PEAR 套件，(位於 /usr/local/lib/php 下)與 PEAR 的套件管理指令(位於 /usr/local/bin/pear 中)。若是安裝低於 PHP4.3.0 以下的 PHP 版本，或已安裝 PHP4.3.0 以上版本的 PHP，但想重新安裝 PEAR 基本套件與套件管理指令則需在 shell 的提示符號後鍵入以下指令：

```
$ lynx -source http://go-pear.org/ | php
```

當然前提是必須先安裝 lynx 應用程式。

接著出現以下歡迎畫面，安裝過程的名稱為「go-pear!」，並告訴安裝者將安裝那些 PEAR 的「基本」套件，請注意，只安裝一些基本套件，並不是全部套件都裝。

```

host2# lynx -source http://go-pear.org/ | php
Welcome to go-pear! /lib/php/
home# ls
Go-pear will install the 'pear' command and all the files needed by
it. This command is your tool for PEAR installation and maintenance.
registry Mail System.php tests
Go-pear also lets you download and install the PEAR packages bundled
with PHP: DB, Net_Socket, Net_SMTP, Mail, XML_Parser, PHPUnit-0.6.2.
DB OS data
If you wish to abort, press Control-C now, or press Enter to continue:

```

圖二十一 go-pear 安裝的歡迎畫面

可以用 Control-C 來中止安裝，若不想中斷則可以按 Enter 來進入安裝畫面，繼續安裝程序。

```

HTTP proxy (http://user:password@proxy.myhost.com:port), or Enter for none: :
Below is a suggested file layout for your new PEAR installation/go-pear.org/ in
change individual locations, type the number in front of the You can then run
directory. Type 'all' to change all of them or simply press Enter to
accept these locations.
1. Installation prefix : /usr/local
2. Binaries directory : $prefix/bin
3. PHP code directory ($php_dir) : $prefix/lib/php
4. Documentation base directory : $php_dir/docs
5. Data base directory : $php_dir/data
6. Tests base directory : $php_dir/tests
1-6: 'all' or Enter to continue:

```

圖二十二 go-pear 安裝的路徑

若有透過 proxy 上網需填入 proxy 的網址與使用者帳號，密碼，port 號等。若沒有透過 proxy 則直接按 Enter 跳過，接著出現要安裝的套件與指令的預設路徑，預設套件會安裝在 /usr/local/lib/php 下，而 pear 的指令是安裝在 /usr/local/bin 下，另外還會安裝一些文件，資料與範例，可按 Enter 鍵直接用預設值安裝。



若已改過 `include_path` 值，則只需寫：

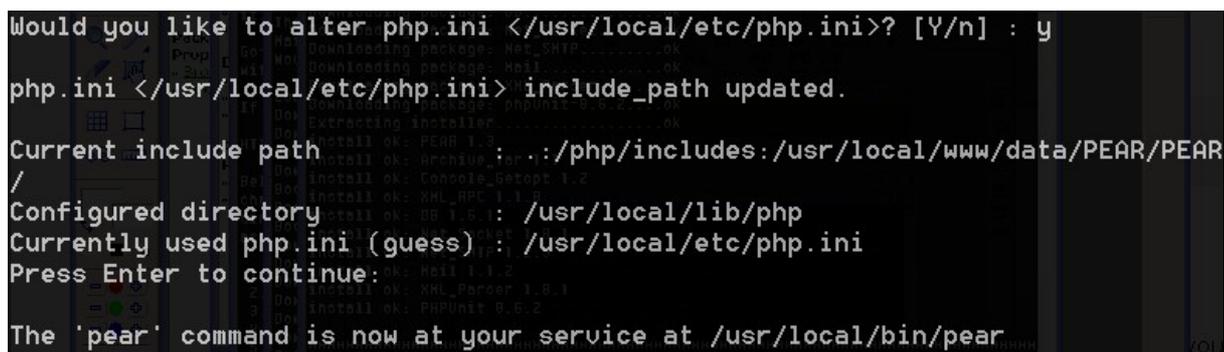
`<? require_once "PEAR.php" ; ?>`即可。

若不想修改 `php.ini` 檔，也可以在每一頁網頁的開頭加入以下的 php 程式碼

```
<?
ini_set('include_path', '/usr/local/lib/php'. PATH_SEPARATOR .
ini_get ('include_path'));
?>
```

即可達到相同的效果。

上圖圖二十四中只要填入 `y`，並按 `Enter`，將會自動修改 `php.ini` 中 `include_path` 值。建議安裝者修改，可節省許多麻煩。

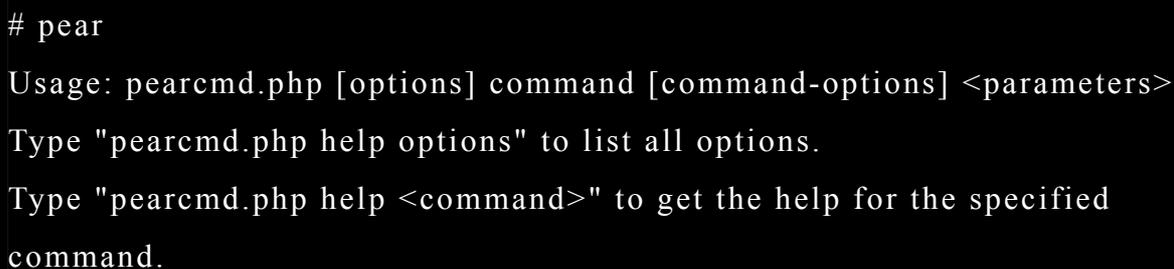


```
Would you like to alter php.ini </usr/local/etc/php.ini>? [Y/n] : y
php.ini </usr/local/etc/php.ini> include_path updated.
Current include path      : ../php/includes:/usr/local/www/data/PEAR/PEAR/
Configured directory     : /usr/local/lib/php
Currently used php.ini (guess) : /usr/local/etc/php.ini
Press Enter to continue:
The 'pear' command is now at your service at /usr/local/bin/pear
```

圖二十五 go-pear 完成安裝圖

OK，出現上圖二十五的畫面代表大功告成了。可以開始使用 `PEAR` 指令來安裝其他「非基本」的套件了。

只要在伺服器以 `root` 登入，在 `shell` 的提示符號後打入 `pear` 指令後按 `Enter` 鍵，就會列出指令使用的格式與所有可用指令加上指令說明，如下所示：



```
# pear
Usage: pearcmd.php [options] command [command-options] <parameters>
Type "pearcmd.php help options" to list all options.
Type "pearcmd.php help <command>" to get the help for the specified
command.
```

Commands:

build	Build an Extension From C Source
clear-cache	Clear XML-RPC Cache
config-get	Show One Setting
config-help	Show Information About Setting
config-set	Change Setting
config-show	Show All Settings
cvsdiff	Run a "cvs diff" for all files in a package
cvstag	Set CVS Release Tag
download	Download Package
info	Display information about a package
install	Install Package
list	List Installed Packages
list-all	List All Packages
list-upgrades	List Available Upgrades
login	Connects and authenticates to remote server
logout	Logs out from the remote server
makerpm	Builds an RPM spec file from a PEAR package
package	Build Package
package-dependencies	Show package dependencies
package-validate	Validate Package Consistency
remote-info	Information About Remote Packages
remote-list	List Remote Packages
run-tests	Run Regression Tests
search	Search remote package database
shell-test	Shell Script Test
sign	Sign a package distribution file
uninstall	Un-install Package
upgrade	Upgrade Package
upgrade-all	Upgrade All Packages

例如，若你想安裝 Net\_UserAgent\_Detect 套件，你只需鍵入以下指令

```
# pear install Net_UserAgent_Detec
```

不用在 Net\_UserAgent\_Detec 後加上版本編號，伺服器會自動幫你安裝最新版本的 Net\_UserAgent\_Detec 套件。如下：

```
# pear install Net_UserAgent_Detect
downloading Net_UserAgent_Detect-1.0.tgz ...
...done: 7,699 bytes
install ok: Net_UserAgent_Detect 1.0
```

若出現下列畫面，代表「相依性」出了問題，你需先安裝畫面所提示的套件，本例為需先安裝「Net\_UserAgent\_Detect」套件，再安裝想安裝的套件，本例為「PEAR\_Frontend\_Web」套件

```
# pear install PEAR_Frontend_Web
downloading PEAR_Frontend_Web-0.3.tgz ...
...done: 32,821 bytes
requires package 'Net_UserAgent_Detect'
PEAR_Frontend_Web: dependencies failed
```

下面的例子，若想移除套件，例如 PEAR\_Info 套件，只需鍵入以下指令：

```
# pear uninstall PEAR_Info
uninstall ok: PEAR_Info
```

下面的例子，若有新版本的 HTML\_QuickForm 套件釋出，而想升級到新版本的套件，只需鍵入以下指令

```
# pear upgrade HTML_QuickForm
downloading HTML_QuickForm-3.0.tgz ...
...done: 62,379 bytes
upgrade ok: HTML_QuickForm 3.0
```

預設用指令安裝的套件會放在 /usr/local/lib/php 下。

除了用指令管理套件外，PEAR 也提供了一個用瀏覽器來管理套件的工

具，使用方法如下，先用 PEAR 指令的方式安裝四個套件，先安裝 Net\_UserAgent\_Detect 套件，再安裝 Pager 套件，HTML\_Template\_IT 套件，PEAR\_Frontend\_Webw 套件，其中因相依性的關係，PEAR\_Frontend\_Web 套件必須最後裝。

```
# pear install Net_UserAgent_Detect
# pear install Pager
# pear install HTML_Template_IT
# pear install PEAR_FrontEnd_Web
```

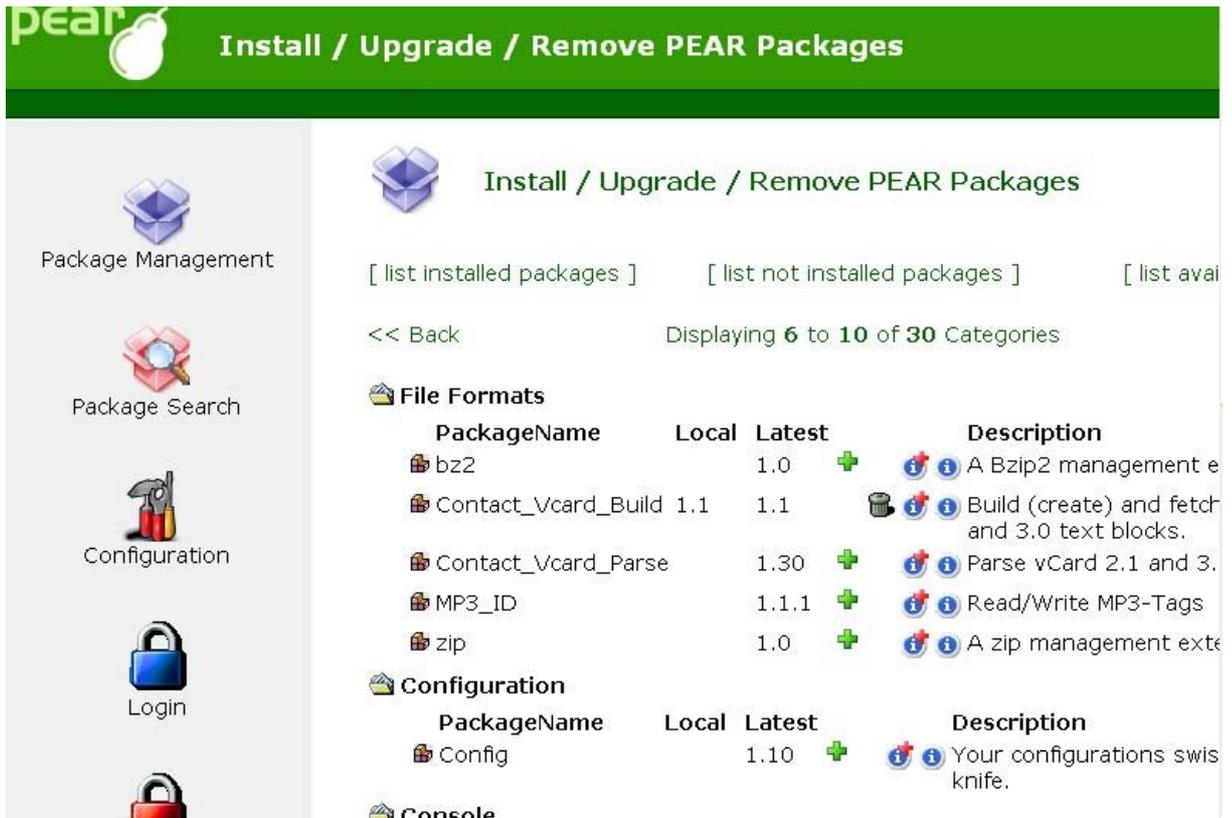
接著再將 /usr/local/lib/php/PEAR 中的 WebInstaller.php 檔，拷貝到你的網站的根目錄如下例：

```
# cp /usr/local/lib/php/PEAR/WebInstaller.php \
/path/to/my/server/admindir/
```

另外還要修改在網頁根目錄下的 WebInstaller.php 檔和以下兩個目錄的擁有者為網頁程式的執行者與其群組，否則因權限的關係將無法使用本功能，本例網頁程式的執行者為 www，群組也為 www。

```
# chown www.www /path/to/my/server/WebInstaller.php
# chown -Rf www.www /usr/local/lib/php
# chown www.www /usr/local/bin
```

接著打開瀏覽器，在網址列輸入  
http://YourWebSite/WebInstaller.php 即可看到以下畫面：



圖二十六. 以 WebInstall 管理 PEAR 套件

該畫面中，左側為導覽列，提供了「套件管理」，「套件搜尋」，「PEAR 屬性設定」，「登入 PEAR 官方網站」，「登出 PEAR 官方網站」等功能。其中最重要的是「套件管理」的功能，該功能又可分為，「列出已安裝套件」，「列出未安裝套件」，「列出可升級的套件」，「列出所有套件」。而上圖二十六，即為「列出所有套件」的畫面，若想安裝某一套件，只需在該套件後的綠色「加號」以滑鼠左鍵點一下即可安裝，若出現相依性問題，會跳出警告視窗加以提示。若想移除某一套件。只需以滑鼠左鍵點一下該套件後的「垃圾桶」圖案即可。若想知道某一套件的相關資訊，只需以滑鼠左鍵點選圓形的「i」圖案，即跳出該套件的訊息視窗。十分方便，可清楚看到目前套件的安裝情況，也可幫助一些對指令有恐懼症的電腦使用者，節省許多打指令的時間。

### 三、PHP 的物件導向程式設計(Object-Oriented Programs, OOP)

PEAR 是以物件導向程式設計的方式建構的，所以在研究 PEAR 之前，需先討論何謂「物件導向」。

物件導向程式設計已經出現一段時間了，由 SmartTalk 和 C++ 帶頭，接著就擴展到像是 Java 及 Python 等等的程式語言。當要開發軟體應用程式時，像是文字處理器或是電腦遊戲等，它就是必備的程式語言特性了。這在軟體架構上，是種開發維護性強，複雜度高及延展性好的軟體的標準方式，更經常用在私有的或是開放源碼的套件上。

從 PHP3 進步到 PHP4 最重要的是改良了該語言的物件導向特性，物件導向與以往程序導向程式設計的不同在於物件導向是以資料為中心，而程序導向是以程式碼為中心，差別在於物件持有所有應該在一起的「資料」與「行為」（或稱為「函式」），而這些資料及函式在程序導向中是完全分離的。這讓物件導向的程式碼在維護上變的非常容易去追蹤，並且增加了專案的模組性。

物件導向程式設計的美妙之處不只在於讓我們可以將程式碼和資料放置在一起，同時它也讓我們可以依照對真實世界的體認方式來分類及定義自己的程式。這讓我們可以深入程式的細節部份前，以更一般性的方式來思考問題。也讓我們在評估時間，風險及投入專案中的各種資源上更加容易。

當開發應用程式時，可以將它區分成各種零組件，或是大型模組，像是不同的表現層，資料庫存取物件，搜尋引擎及保全元件。在開發大型，單一元件的模組時，就能保證對單一物件的更動不會影響其他物件。更可能因此在許多不同的應用程式間重用這些元件。也可以將這些模組再往下區分成次模組，甚至區分成在物件導向程式中最小的單一物件-類別(class)。(Luis Argerich, 2002)

以下將 PHP 物件導向程式設計的專有名詞做一個解釋

- 類別(class): 在實際世界中，真實物體並非獨立而是有關聯性的。而物件導向對此關聯性的表達提出了解決之道：使用類別(Class)。而類別，乃是一種樣板，它定義了可能包含於個別物件的方法及變數。對於某一特定種類的物件所共同具有的方法和變數，只要將它們定義在類別中，然後再將這些物件宣告成屬於該類別即可。屬於同一個類別的這些物件便稱為該類別的「案例」(Instance)。在同一類別中案例的屬性皆是相同的，所不同的是變數的值。(吳學軒, 2003)
- 物件(object): 「物件」，為一個由軟體程式碼所構成的「包裝」，

包含了一組彼此相關的程序和資料。這些程序，稱為這個物件的「方法」(Method)，而資料，則稱之為「變數」或「成員」(Variable)。方法是該物件展現對外表達的一些行為，它告訴外界，它能夠做些什麼，以及外界該如何去和該物件進行溝通。變數則代表該物件內在的一面，存放了該物件的一些相關資料，這些資料是物件內部所使用，而不為外界所知的。如果一個物件打算讓它的某些變數開放讓外界存取的話，可以在其方法中加入適當的存取這些變數的方式。無論如何，外界只能透過物件的方法來和物件溝通，絕無法直接去存取物件的變數。因此，方法是一個物件和外界之間的窗口。物件的觀念看起非常的簡單，可是其功能卻很強大。我們可以定義出各種物件，來模擬真實世界的實體，並把各實體的行為、屬性、狀態等等，全部包裝在一起，而這些物件彼此都是獨立的小世界。一部再大的機器，也是由小零件所組成的，如果將物件看成軟體的小零件的話，我們便可以用這些小零件來組合成一個大軟體。這是物件導向技術最基本，也是最重要的觀念。(吳學軒，2003)

- 建構子(constructor):一種類別中的特別的方法，當類別被初始化(new)為物件時，建構子會被執行，建構子函式的名稱必須與類別名稱相同。
- 解構子(destructor):當網頁關閉時會自動執行的函式，解構子函式的名稱必須與建構子相同並在名稱前加上「底線」符號。
- 工廠(factory):讓物件本身可以一次產生一或多個其他物件，這些類別稱作 factories，而用來產生物件的函式就稱作 factory 方法。
- 封裝(encapsulation):將所有複雜的資料與邏輯放入物件中，並且只將使用者所期望物件能提供的有意義的服務開放給使用者。
- 繼承:將相似物件中自有的細節放到其各自的地方，並且由一個稱為基底物件(base class)的物件來放置相似的部份。而其他物件可繼承此基底物件來擁有這些共同的屬性與方法。
- 多型(Polymorphysm): 假設我們需要一串不同的物件；而這些物件是以相同的方式使用。在 OOP 中典型的範例就是幾何圖形。如果我們在正方形、三角形、圓形...加上一些資料結構並且所有圖

形皆可呼叫 Draw 方法以簡化操作（如檢查正確的类型）這就是多型。沒有繼承；多型沒有意義。沒有繼承；不可能讓一個（子）類別的物件替代另一個（基底）類別。

- 抽象方法(Abstract method):基底類別會代有不具備程式碼的方法，因為它無法用同一個方法處理不同的資料，因此，稱為抽象方法，而由每個子類別來實作這個函式。

本章節只簡單的說明物件導向的觀念，因本論文並非以研究物件導向為主題，故只條列出本論文會引用到的名詞加以解釋，相關物件導向的研究請參閱其他文件或書籍。

#### 四、PEAR 的基本套件程式的使用

##### 4-1 PEAR 的基本類別

所謂基本類別，就是大部分的 PEAR 套件程式中的類別大都是由此類別再延伸出來的，也就是此類別是許多類別的源頭，它提供了所有套件的基本功能，例如：解構子(destructor)，除錯，廣域變數(Global Variables)的使用等。底下利用官方網站提供的手冊中 Martin Jansen, Alexander Merz(2004)兩位所撰寫的範例程式來加以說明 PEAR 基本類別的用法。請注意，若以下範例以「//」作為一行的開頭，代表該行為「註解」，而研究者就利用註解來說明程式碼的意義。

##### 4-2 範例程式-範例一：模擬一個解構子

```
require_once "PEAR.php";  
//若在 php.ini 中已修改過 include_path，只要在網頁的前頭加入此行  
//即可引入 PEAR 基本類別的 PHP 檔，檔名為 PEAR.php  
  
class FileContainer extends PEAR  
//PEAR 的子類別，繼承(Inheritance)自 PEAR 基本類別，名為  
//FileContainer 類別  
{  
    var $file = '';  
    var $contents = '';
```

```

var $modified = 0;
//定義三個資料(Attributes)
function FileContainer($file)
//建立建構子函式(constructor)
{
    $this->PEAR();
    //需在建構子中呼叫父類別的 pear 方法，才可以使用解構
    //子，且解構子需命名為_FileContainer

    $fp = fopen($file, "r");
    if (!is_resource($fp)) {
        return;
    }
    while ($data = fread($fp, 2048)) {
        $this->contents .= $data;
    }
    //上述程式碼的意思為開啟一個檔案，並將檔案的內容
    //讀取到$this->contents 中
    fclose($fp);
}

function append($str)
//第二個函式，名為 append
{
    $this->contents .= $str;
    $this->modified++;
    //上述程式碼的意思為將$str 字串加到$this->contents 之後
}

function _FileContainer()
//解構子(destructor)的函式，名稱與建構子(constructor)一樣

```

```

//只是函式名稱前有加底線
{
    if ($this->modified) {
        $fp = fopen($this->file, "w");
        if (!is_resource($fp)) {
            return;
        }
        fwrite($fp, $this->contents);
        //上述程式碼的意思為將$this->contents 寫入檔案中
        fclose($fp);
    }
}

$fileobj =& new FileContainer("testfile");
$fileobj->append("this ends up at the end of the file\n");
//從 FileContainer 類別產生一個新物件名為$fileobj
//並使用該物件的 append 方法加入一行字串
//當此 php 檔關閉時，解構子函式將會被呼叫，檔案將會被 update 到
//磁碟中

```

以上範例可看到 PEAR 基本類別如何實作解構子的功能，請注意，PHP4 內建沒有解構子的功能，並不像 JAVA。

#### 4-3 PEAR 的除錯

除錯是所有類別都需要的功能，也是程式發展階段很重要的一向輔助機制，所以 PEAR 的除錯機制是放在 PEAR 的基本類別中，PEAR 可以把錯誤結果做成物件，而且不一定發生錯誤時一定要用某種形式表現出錯誤，可以不用將錯誤輸出在螢幕上，但卻可以除錯。例如：將錯誤的結果寫在某一個檔案，而不顯示在螢幕上。也可以將錯誤顯示出來，但卻不一定用 html 格式顯示出錯誤語(errors value)和錯誤碼(errors code)。

這個 error 物件可以被設定來做許多事，例如，印出錯誤語(errors value)和錯誤碼(errors code)但不停止程式執行，印出錯誤語(errors value)和錯誤碼(errors code)並停止程式執行，或用 trigger\_error()函式來處理錯誤的結果，或傳回一個回饋，或什麼都不做等方式。底下利用官方網站提供的手冊中 Martin Jansen, Alexander Merz(2004)兩位所撰寫的範例程式來加以說明 PEAR 除錯類別的用法。請注意，若以下範例以「//」作為一行的開頭，代表該行為「註解」，而研究者就利用註解來說明程式碼的意義。

#### 4-4 範例程式-範例二:除錯(一)

```
//建立一個名為 mysockopen 的函式，此函式可以連線到伺服器，透過
// 8090 port。
function mysockopen($host = "localhost", $port = 8090)
{
    $fp = fsockopen($host, $port, $errno, $errstr);
    if (!is_resource($fp)) {
        return new PEAR_Error($errstr, $errno);
        //若是有錯誤會傳回一個 PEAR_Error 物件(8)
    }
    return $fp;
}

$sock = mysockopen();
//執行 mysockopen 函式，將結果傳給 $sock 變數。若無法執行，則傳
//回一個所謂的「錯誤物件」給 $sock 變數

if (PEAR::isError($sock)) {
    print "mysockopen error: ".$sock->getMessage()."<BR>\n"
}

//PEAR::isError()是用於偵測 $sock 是否為 error 物件，若是，
```

```
//則顯示錯誤訊息
```

#### 4-5 範例程式-範例三:除錯(二)

```
require_once "PEAR.php";  
//若在 php.ini 中已修改過 include_path, 只要在網頁的前頭加入此行  
//即可引入 PEAR 基本類別  
  
class TCP_Socket extends PEAR  
//建立一個 PEAR 的子類別, 即繼承(Inheritance)自 PEAR 基本類別  
{  
    //定義一個名為 $sock 的資料  
    var $sock;  
    //建構子函式, 執行 PEAR 類別的建構子函式。  
    function TCP_Socket()  
    {  
        $this->PEAR();  
    }  
  
    //建立一個名為 connect 的函式, 用來與伺服器連線  
    function connect($host, $port)  
    {  
        $sock = fsockopen($host, $port, $errno, $errstr);  
        if (!is_resource($sock)) {  
            return $this->raiseError($errstr, $errno);  
            //將錯誤碼與錯誤語用 PEAR 基本類別中的 raiseError  
            //函式處理後傳回這是與前例不同的另一種錯誤處理  
            //方式。  
        }  
    }  
}
```

```
$sock = new TCP_Socket;
$sock->setErrorHandler(PEAR_ERROR_DIE);
//設定成當發生錯誤時，對錯誤的處理方式為 PEAR_ERROR_DIE，即為；
//將錯誤顯示出來並停止執行程式，PEAR 提供許多種處理方式，詳細請
//參閱 PEAR 說明文件
$sock->connect("localhost", 8090);
print "still alive<BR>\n";
```

#### 4-6 PEAR 的廣域變數(Global variables)

PEAR 類別註冊了一些預設的廣域變數，供所有類別使用。例如範例三的 PEAR\_ERROR\_DIE 即是，詳細的其他 PEAR 的廣域變數因數目繁多請參閱 PEAR 說明文件。

### 五、常用的 PEAR 套件

PEAR 的套件多達 219 個，除了基本套件外，程式開發者需要什麼套件皆需自行安裝。底下僅就常用的一些套件做說明。



#### 5-1 HTML\_QuickForm 套件

表單是動態網頁程式與使用者接觸的介面，藉由表單，使用者可以輸入各式各樣的資料，然後藉由後端動態網頁程式的處理，來產生各種結果。以往表單的製作最簡單的方法是以 HTML 來完成。但 HTML 無法達到一些動態的效果，例如：使用者輸入的資料格式是否正確的驗證；使用者若填寫錯誤，但已按「Submit」鍵，後端程式發現有錯後，往往要求使用者全部重新填寫，而不是可以僅就錯誤部份加以更改。

HTML\_QuickForm 套件就是提供製作動態表單的功能，讓程式開發者可以輕鬆製作出功能強大的表單。

底下利用 Keith Edmun(2004)所撰寫的範例程式來加以說明 HTML\_QuickForm 套件的用法。請注意，若以下範例以「//」作為一行的開頭，代表該行為「註解」，而研究者就利用註解來說明程式碼的意義。當程式開發者安裝好套件後，假設建立了一個檔案名為 quickform\_1.php，程式碼如下：

```
<?php
require_once "HTML/QuickForm.php";
//引入 PEAR 的 HTML_QuickForm 套件

$form = new HTML_QuickForm('frmTest', 'get');
//建立一個 HTML_QuickForm 物件，這是一個表單物件，此表單名為
//frmTest，其「action」的方式是 get。

$form->addElement('header', 'MyHeader', 'Testing QuickForm');
//在此表單物件中放入一個表單的表頭文字元件，此表頭文字元件的
//name 為 MyHeader，顯現出來的文字為 Testing QuickForm。

$form->addElement('text', 'MyTextBox', 'What is your name?');
//在此表單物件中放入一個文字元件，此文字元件的 name 為
//MyTextBox，此元件前方顯現出來的文字為 What is your name?。

$form->addElement('reset', 'btnClear', 'Clear');
//在此表單物件中放入一個 reset 按鈕，reset 按鈕元件的 name 為
//btnClear，此按鈕內顯現出來的文字為 Clear。

$form->addElement('submit', 'btnSubmit', 'Submit');
//在此表單物件中放入一個 submit 按鈕，submit 按鈕元件的 name 為
//btnsubmit，此按鈕內顯現出來的文字為 submit。

$form->addRule('MyTextBox', 'Your name is required',
'required');
//對這個表單中的 MyTextBox 元件加上一條驗證的規則-不可空白，若空
//白會出現「Your name is required」的警告語。
//驗證的規則在 HTML_QuickForm 中有許多預設的方式，例如：
//required, maxlength, rangelength, regex, email, emailorblank,
//lettersonly, alphanumeric, numeric, nopunctuation, nonzero。
```

```

//有加驗證規則的元件，前面會出現紅色「*」，請見圖
//這些規則的詳細說明與設定請參閱 PEAR 說明文件(18)，但可由文字的
//意思大概地推敲出其涵義。

$form->applyFilter('MyTextBox', 'trim');
//使用 trim 函式來處理 MyTextBox 中的值

if ($form->validate()) {
    # If the form validates then freeze the data
    $form->freeze();
}

//將表單內的資料加以驗證，若合乎格式，則這些資料直接顯示在//網
//頁上，且這些文字或數字無法修改

    $form->display();
//將此表單顯示在網頁上
?>

```

已上程式碼所建立的表單如下圖：

Testing QuickForm

\*What is your name?

Clear

Submit

\* denotes required field

圖二十七 用 HTML\_QuickForm 建立的表單範例(1)

接著假設建立了一個檔案名為 quickform\_2.php，此範例將在表單中使用「預設值」。程式碼如下，且開頭為「//」的那一行代表是「註解」。

```

<?php
require_once "HTML/QuickForm.php";

```

```

//引入 PEAR 的 HTML_QuickForm 套件

#####

$user = array("firstname"=>"Mickey" ,
              "lastname"=>"Mouse" ,
              "sal_id"=>4 ,
              "user_id"=>7);

//將預設的使用者資訊放入一個陣列中
$salutations = array("0"=>"Mr" ,
                     "1"=>"Miss" ,
                     "2"=>"Mrs" ,
                     "3"=>"Dr" ,
                     "4"=>"Sir");

//將 select 選單元件的選項放入一個名為$salutations 陣列
#####

$form = new HTML_QuickForm('frmTest' , 'get');
$form->addElement('header' , 'MyHeader' , 'Edit username');
$form->addElement('hidden' , 'user_id');
$form->addElement('select' , 'sal_id' , 'Address me as:' ,
                 $salutations);
//建立一個 select 選單元件，此元件名為 sal_id，前導文字為
//Address me as:而選單的內容選項為$salutations 陣列。

$form->addElement('text' , 'firstname' , 'First name:');
$form->addElement('text' , 'lastname' , 'Last name:');
$form->addElement('reset' , 'btnClear' , 'Clear');
$form->addElement('submit' , 'btnSubmit' , 'Submit');

if ($form->validate()) {
    // Form is validated, then processes the data

```

```

    $form->freeze();
    $form->process('process_data', false);
    //若驗證成功，則用 process_data 函式來處理
}
else {
    $form->setDefaults($user);
    //若驗證失敗則選單中的值用 $user 陣列為預設值。
    $form->display();
}

//建立一個函式
function process_data ($values) {
    echo "<pre>";
    foreach ($values as $key=>$value) {
        echo $key."=".$value."<br>";
        //將所有陣列的 key 與 value 列出
    }
    echo "</pre>";
}
?>

```

已上程式碼所建立的表單如下圖，每個元件中已有預設值：

**Edit username**

**Address me as:**

**First name:**

**Last name:**

圖二十八用 HTML\_QuickForm 建立的表單範例(2)

接著假設建了一個檔案名為 quickform\_3.php，此範例將在表單中使

用「CSS 樣式」程式碼如下，且開頭為「//」的那一行代表是「註解」。

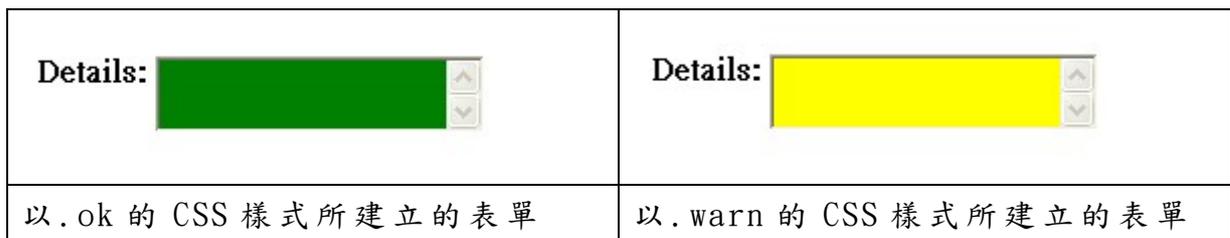
```
<style type="text/css">
    .ok { background: green; }
    .warn { background: yellow; }
    .error { background: red; }
</style>
<!--註解：建立三個 CSS 樣式，分別為.ok，.warn，.error。-->

</head>
<body>
<?php
    require_once "HTML/QuickForm.php";
    $form = new HTML_QuickForm('frmTest', 'get');

    //設定這個 tag 的樣式屬性
    $form->addElement('textarea', 'MyTextArea', 'Details:',
        "class=warn");
    //將表單中名為 MyTextArea 的 textarea 元件套用名為 warn 的樣式，
    //而此樣式定義在<style>與</style>中

    if ($form->validate()) {
        $form->freeze();
    }
    $form->display();
?>
```

已上程式碼所建立的表單如下圖：



圖二十九 使用 CSS+HTML\_QuickForm 建立的表單範例(3)

## 5-2 QuickForm Controller 套件

這個套件可以讓程式設計者用一套 PHP 的 script 卻可產生「多頁的表單」，也就是做出類似「精靈(Wizard)」的表單，此套件含有三個類別，分別是：

- HTML\_QuickForm\_Controller：這個類別可從 request 中抓出 action 的值，並選用適當的處理方法。
- HTML\_QuickForm\_Page：這個類別可以用來呈現多頁表單的某一頁。
- HTML\_QuickForm\_Action：這個類別提供一個用指令來設計的形式。

底下是套件中所提供的範例程式來加以說明 QuickForm Controller 套件的用法。請注意，若以下範例以「//」作為一行的開頭，代表該行為「註解」，而研究者就利用註解來說明程式碼的意義。

```
<?php
require_once 'HTML/QuickForm.php';
// 引入 HTML_QuickForm 類別

require_once 'HTML/QuickForm/Controller.php';
// 引入 HTML_QuickForm_Controller 類別

require_once 'HTML/QuickForm/Action.php';
// 引入 HTML_QuickForm_Action 類別

class FirstPage extends HTML_QuickForm_Page
// 建立一個名為 FirstPage 的類別，此類別繼承自 HTML_QuickForm_Page
//類別
{
    //建立一個函式
    function buildForm()
    {
```

```

$this->_formBuilt = true;
//設定_formBuilt 屬性為 true，以防止此方法被呼叫兩次
// 加入一些元件到表單中
$this->addElement('header', null, 'QuickForm tutorial
    example');
$this->addElement('text', 'name', 'Enter your name:',
    array('size' => 50, 'maxlength' => 255));
//文字元件，名稱為「name」
// 將元件的屬性作成陣列放入 addElement 的參數中

$this->addElement('submit', $this->getButtonName('submit'), 'Send');
//將 submit 的按鈕命名為 submit

$this->applyFilter('name', 'trim');
$this->addRule('name', 'Please enter your name', 'required',
null, 'client');

$this->setDefaultAction('submit');
//此設定代表若按「Enter」鍵，相當於按「submit」按鈕。
}
}
//類別 FirstPage 到此結束，此類別代表「表單內容」。

class ActionProcess extends HTML_QuickForm_Action
//一個處理 Action 的類別，繼承自 HTML_QuickForm_Action。
{
    function perform(&$page, $actionName)
    {
        echo '<h1>Hello' .
htmlspecialchars($page->exportValue('name')) . '!</h1>';
//將名為 name 的元件中的值，輸出在螢幕上。

```

```

    }
}
//類別 ActionProcess 到此結束，此類別代表使用者 Submit 後，處理的//
程序

//底下為執行的主程式。
$page =& new FirstPage('firstForm');

$page->addAction('process', new ActionProcess());
//若表單為有效的且使用者按「submit」則該網頁的處理程序(process)
//的 action。會被呼叫，而此 process 是一個 ActionProcess 物件。

$controller =& new HTML_QuickForm_Controller('tutorial');
//建立一個 HTML_QuickForm_Controller 物件，且這個物件名稱為
//tutorial。若有不同的 HTML_QuickForm_Controller 物件，可以用不同名
//稱命名

$controller->setDefaults(array(
    'name' => 'Joe User'
));
//設定這個 $controller 物件的預設值。

$controller->addPage($page);
//把 $page 放入 $controller 物件中，本範例只用一個表單，與
//HTML_QuickForm 的意義相同只是寫法不同，但若表單橫跨三，四頁，
//只要配合 session 與翻頁並多個 page 都放入一個 controller 即可。

$controller->run();
//執行此 controller，意思即為用 controller 來控制表單。
?>

```

已上程式碼所建立的表單如下圖：

**QuickForm tutorial example**  
\*Enter your name:   
  
\* denotes required field

圖三十 用 HTML\_QuickForm\_Controller 建立的表單範例(1)

以上範例只呈現一張表單，若一張以上表單該如何呈現呢？

若用 HTML\_QuickForm\_Controller 套件中的範例檔 wizard.php 為例，顯示結果如下：

<p><b>Wizard page 1 of 3</b> *Are you absolutely sure? <input checked="" type="radio"/> Yes <input type="radio"/> No <input type="button" value="Next &gt;&gt;"/> * denotes required field</p>	範例表單的第一頁若不選任何一個 radio 就按「next」鍵會有警告訊息出現
<p><b>Wizard page 2 of 3</b> Name (last, first): <input type="text" value="ben"/> , <input type="text" value="lin"/> <input type="button" value=" &lt;&lt; Back"/> <input type="button" value=" Next &gt;&gt;"/> * denotes required field</p>	範例表單的第二頁須填入 first name 與 last name。然後按「Next」，也可按「Back」回上頁
<p><b>Wizard page 3 of 3</b> *Parting words: <input type="text" value="ha!"/> <input type="button" value=" &lt;&lt; Back"/> <input type="button" value=" Finish"/> * denotes required field</p>	範例表單的第三頁最後填入想寫的話。若有紅色星號代表必填。填完後按「Finish」

	<p>結果將顯示出使用者填入表單的所有資訊。並以陣列的方式顯示出來。</p>
--	--

圖三十一 Wizard.php 範例檔的表單的三個頁面與執行成果

而此範例的原始碼如下：

```

<?php
//引入 HTML_QuickForm_Controller 檔案
require_once 'HTML/QuickForm/Controller.php';

//引入一些和 Action 相關的檔案，共有四種方式的檔案
//分別是「下一步」，「上一步」，「跳頁」，「顯示」
require_once 'HTML/QuickForm/Action/Next.php';
require_once 'HTML/QuickForm/Action/Back.php';
require_once 'HTML/QuickForm/Action/Jump.php';
require_once 'HTML/QuickForm/Action/Display.php';

//啟動 session 如此才可跨頁。
session_start();

//第一頁的表單類別
class PageFirst extends HTML_QuickForm_Page
{

```

```

//建立表單的函式，名為 buildForm
function buildForm()
{
    //表單元件的內容可對照上方範例圖三十一
    $this->_formBuilt = true;
    $this->addElement('header', null, 'Wizard page 1 of 3');
    $radio[] = &$this->createElement('radio', null, null,
'Yes', 'Y');
    $radio[] = &$this->createElement('radio', null, null,
'No', 'N');
    $this->addGroup($radio, 'iradYesNo', 'Are you
absolutely sure?');
    $this->addElement('submit',
$this->getButtonName('next'), 'Next >>');
    $this->addRule('iradYesNo', 'Check Yes or No',
'required');
    $this->setDefaultAction('next');
//若按「Enter」鍵等同於按「next」按鈕
}
}

//第二頁的表單類別
class PageSecond extends HTML_QuickForm_Page
{
    function buildForm()
    {
        $this->_formBuilt = true;
        $this->addElement('header', null, 'Wizard page 2 of 3');
        $name['last'] = &$this->createElement('text', 'last',
null, array('size' => 30));
        $name['first'] = &$this->createElement('text', 'first',

```

```

null, array('size' => 20));
    $this->addGroup($name, 'name', 'Name (last, first):',
, &nbsp;');

//建立一個 submit 按鈕的元件，名為「back」，按鈕上的
//文字為<<Back
    $prevnext[]=&$this->createElement('submit', $this->
    getButtonName('back'), '<< Back');

//建立一個 submit 按鈕的元件，名為「next」，按鈕上的
//文字為 Next>>
    $prevnext[]=&$this->createElement('submit', $this->
    getButtonName('next'), 'Next >>');

//將此兩個按鈕放入一個群組中
    $this->addGroup($prevnext, null, '', '&nbsp;', false);

//將 name 群組命加上規則，last 欄位為不可空白。
    $this->addGroupRule('name', array('last' =>
    array(array('Last name is required',
    'required'))));

    $this->setDefaultAction('next');
}
}

//第三頁的表單類別
class PageThird extends HTML_QuickForm_Page
{
    function buildForm()
    {

```