PHP 中文手册

王继纲 编

目录

PHP 的来龙去脉

讲到 PHP 的全名就蛮有趣的,它是一个递归的缩写名称,"PHP: Hypertext Preprocessor",打开缩写还是缩写。PHP 是一种 HTML 内嵌式的语言 (类似 IIS 上的 ASP)。而 PHP 独特的语法混合了 C、Java、Perl 以及 PHP 式的新语法。它可以比 CGI 或者 Perl 更快速的执行动态网页。

PHP 最初是在公元 1994 年 Rasmus Lerdorf 开始计划发展。在 1995 年以 Personal Home Page Tools (PHP Tools) 开始对外发表第一个版本。在这早期的版本中，提供了访客留言本、访客计数器等简单的功能。随后在新的成员加入开发行列之后，在 1995 年中，第二版的 PHP 问市。第二版定名为 PHP/FI(Form Interpreter)。PHP/FI 并加入了 mSQL 的支持，自此奠定了 PHP 在动态网页开发上的影响力。在 1996 年底，有一万五千个 Web 网站使用 PHP/FI ;在 1997 年中,使用 PHP/FI 的 Web 网站成长到超过五万个。而在 1997 年中，开始了第三版的开发计划，开发小组加入了 Zeev Suraski 及 Andi Gutmans,而第三版就定名为 PHP3。

PHP3 跟 Apache 服务器紧密结合的特性 加上它不断的更新及加入新的功能；并且它几乎支持所有主流与非主流数据库；再以它能高速的执行效率，使得 PHP 在 1999 年中的使用网站超过了十五万!!它的源代码完全公开,在 Open Source 意识抬头的今天,它更是这方面的中流砥柱。不断地有新的函数库加入，以及不停地

更新的活力，使得 PHP 无论在 UNIX 或是 Win32 的平台上都可以有更多新的功能。它提供丰富的函数，使得在程序设计方面有着更好的支持。

PHP 的第四代 Zend 核心引擎已经进入测试阶段。整个脚本程序的核心大幅改动，让程序的执行速度，满足更快的要求。在最佳化之后的效率，已较传统 CGI 或者 ASP 等程序有更好的表现。而且还有更强的新功能、更丰富的函数库。无论您接不接受，PHP 都将在 Web CGI 的领域上，掀起巅覆性的革命。对于一位专职 Web Master 而言，它将也是必修课程之一。

PHP 的功能概述

PHP 在数据库方面的丰富支持，也是它迅速走红的原因之一，它支持下列的数据库或是资料表：

Adabas D
DBA
dBase
dbm
filePro
Informix
InterBase
mSQL
Microsoft SQL Server
MySQL
Solid
Sybase

ODBC

Oracle 8

Oracle

PostgreSQL

而在 Internet 上它也支持了相当多的通讯协议 (protocol)，包括了与电子邮件相关的 IMAP, POP3；网管系统 SNMP；网络新闻 NNTP；帐号共用 NIS；全球信息网 HTTP 及 Apache 服务器；目录协议 LDAP 以及其它网络的相关函数。

除此之外，用 PHP 写出来的 Web 后端 CGI 程序，可以很轻易的移植到不同的系统平台上。例如，先以 Linux 架的网站，在系统负荷过高时，可以快速地将整个系统移到 SUN 工作站上，不用重新编译 CGI 程序。面对快速发展的 Internet，这是长期规划的最好选择。

在加入其它的模块之后,提供了更多样的支持如下：

英文拼写检查

BC 高精确度计算

公元历法

PDF 文件格式

Hyperwave 服务器

图形处理

编码与解码功能

哈稀处理

WDDX 功能

qmail 与 vmailmgr 系统

压缩文件处理

XML 解析

除此之外，一般语言有的数学运算、时间处理、文件系统、字符串处理、行程处理等功能，它一样都不缺。再加上它是免费的系统，使得成本与效益比，几乎等于无限大！

PHP 与其它 CGI 的比较

无可置疑的，写 CGI 的方式有很多种，而 PHP 只是其中的一种选择罢了。对资深的 Webmaster 而言，CGI 的写作界面应是随着需求而改动。毕竟，在一个对系统反映速度要求极严格的系统而言，恐怕只有 NSAPI 界面写的 CGI 程序才能符合要求了。在其它的场合，相信使用 PHP 来作为 CGI 的界面是游刃有余,而且是最适合的。

程序界面	PHP	ASP	CGI	NSAPI	ISAPI
操作系统	均可	Win32	均可	均可	Win32
Web 服务器	数种	IIS	均可	Netscape Server	IIS
执行效率	快	快	慢	极快	极快
稳定性	佳	中等	最高	差	差
开发时间	短	短	中等	长	长
修改时间	短	短	中等	长	长
程序语言	PHP	VB	不限	C/C++	C/Delphi
网页结合	佳	佳	差	差	差

学习门槛	低	低	高	极高	高
函数支持	多	少	不定	中等	少
系统安全	佳	极差	最佳	佳	尚可
使用网站	超多	多	多	极少	少
改版速度	快	慢	无	慢	慢

其中的 PHP 可用在数种 Web 服务器上；传统 CGI 就不限是哪种操作系统或 Web 服务器平台；NSAPI 一定要在 Netscape 的服务器 (如 Netscape Enterprise Server 或 FastTrack Server) 上才可以执行，但可支持多种操作系统 (UNIX 或 Win32)；ASP 及 ISAPI 只在 IIS 上有完整的功能。

在稳定性上，由于 NSAPI 或 ISAPI 是动态链接的方式，因此在执行若出现问题，会使得 Web 服务器一起瘫痪。而 ASP 在我实际应用经验上，隔阵子就会使系统不稳定，需要重新启动操作系统。PHP 在许多的网站使用上，不但长期使用都没有问题，而且程序的稳定性也不错。当然最稳的还是传统 CGI 程序，因为它是由操作系统负责控制，不会因 CGI 程序的错误导致 Web 服务器的不稳定。

在开发及维护时间上，PHP 及 ASP 都有不错的表现。而 NSAPI 及 ISAPI 则需要长时间的开发过程，在稳定上线后，这两种界面反倒是效率最佳的方法。传统的 CGI 程序则要视开发工具语言而定了，用 Perl 或是 shell script 不需要编译的过程，直接就可以执行，若用 Delphi 或 VC/BCB 甚至用组合语言等都要经过编译才能执行，至于用 VB 来写传统 CGI，唉....。

要比较和网页结合的能力，PHP 和 ASP 是并驾齐驱的，其它的方式就不能内嵌 HTML 语法了。而这也是影响开发时间的因素之一。

就系统安全性而言，ASP 是最差的，在没有经过微软的 IIS Service Pack 处理过，使用 ::$DATA 就可以看到 ASP 的源代码，这真是叫人不敢领教。当然，传统 CGI 的程序，由于是由操作系统直接管理，要破解的难度最高，黑客必须由操作系统下手，而不能由 Web 服务器下手。PHP 在许多商业及非商业使用时，也没有听过有什么安全的问题。

在新增功能及改版方面，传统的 CGI 由于不受任何语言限制，没有这方面的问题。PHP 是最有活力的，数天至数周就有一个新版本出现，每次的新版，就代表更多的功能及修正更多的错误。其它的 ASP、NSAPI、ISAPI 就视它的 Web 服务器改版速度了，ASP 要等到 IIS 5.0 出现时才会有 ASP 3.0，也就是要等到 Windows 2000 正式上市。

总而言之，在 Web 的后端 CGI 程序，就像鱼与熊掌一般，没有高效率又开发方便的选择。不过相信 PHP 是处于开发容易、效率也不错的平衡点上。

环境需求与准备工作

在安装 PHP 做为 WWW 服务器的一部份时，我们可以考虑用 UNIX 操作系统；或者是 Windows NT/95 等 Win32 API 的平台。当然，大部份的人都会使用 UNIX 来当作 PHP 的执行平台（在 Windows NT 的用户大多

数都会选择 IIS + ASP)，因此，本书的所有内容以及范例程序都是在 UNIX 上为主。实际上，Linux + Apache + PHP 应是最经济的选择，因为这样的组合几乎是不用钱的，成本与效益比这也是最好的选择。而许多成功网站的经验，更是采用这种组合最好的佐证。

Linux 操作系统方面，您可以选择各式的 Linux 套件，包括 Slackware Linux、RedHat、OpenLinux、SuSE....等等,反正这方面的软件在店里也是很容易而且很便宜就可以买到。对学生而言，也可以去各大 FTP 站下载完整的系统安装。

Apache 服务器则是目前最多 WWW 网站所采用的服务器。您可以到 http://www.apache.org 下载最新版的程序及相关文件，若您觉得从国外下载要很久的话，也可以用它的 Mirror 网站下载。

PHP 则可以去它的官方网站 http://www.php.net 下载所需要的程序。

虽然目前 WindowsNT 或者 Windows98 等 Win32 的系统平台也能安装 PHP 及 Apache 服务器,不过这似乎没什么道理，因为 PHP 和 Apache 在 UNIX 下可以跑得更快更好。

当然，若想使用商业化的系统平台，SUN、IBM、HP、DEC、SGI、NEC 等公司都提供相关的 UNIX 或者是 WindowsNT 的系统平台。加上高安全性调整过后的 Apache 服务器：Stronghold 或是其它支持 SSL 的 Apache 版本。这种组合，相信能满足商业化的需求。而 PHP 就扮演着快速方便的 CGI 角色,让客户对网站的服务品质更加满意。

快速配置及安装

以下是基本的安装步骤,运行环境是 UNIX 系列的系统平台。在安装之前，要先下载 apache_1.3.x.tar.gz 及 php-3.0.x.tar.gz 两个文件。可以将这两个文件放在 /usr/src 中再开始执行以下的步骤。下面每个行号后是一个步骤，步骤中的所有选项是连在一起的，请不要分开执行。

```
gzip -d -c apache_1.3.x.tar.gz | tar xvf -
gzip -d -c php-3.0.x.tar.gz | tar xvf -
cd apache_1.3.x
./configure --prefix=/www
cd ../php-3.0.x
./configure --with-mysql --with-apache=../apache_1.3.x --enable-track-vars
make
make install
cd ../apache_1.3.x
./configure --prefix=/www --activate-module=src/modules/php3/libphp3.a
make
make install
```

第一、二行利用 gzip 及 tar 加上管道功能，将压缩文件解压还原。然后在 Apache 的原始文件目录中执行环境配置，--prefix 选项指示 Apache 的安装目录路径。之后进入 PHP3 的原始文件目录中，若没有 MyS

QL 数据库，则可省略 --with-mysql 的选项，重要的是一定要加入 --with-apache 选项，而且 Apache 原始文件的路径要正确。配置完 PHP3 之后就编译、安装到 Apache 的原始文件目录中。之后在 Apache 原始文件目录中再加入 PHP 的模块文件。在编译及安装 Apache 之后就初步完成了。之后就是要配置 Apache 才能让 Web Server 顺利运作。

需要注意的是，PHP 要和任何数据库连接，都要在执行这些步骤之前先将数据库设好，并确定 Web Server 上可以顺利存取数据库系统。如果需要其它的一些 PHP 外部模块也要先配置好这些模块。

```
cd ../php-3.0.x
cp php3.ini-dist /usr/local/lib/php3.ini
```

之后将 php3.ini 放在指定的目录，如果需要，也可以手动修改 php3.ini 文件文件，以符合使用的要求。

在 Apache 服务器的配置方面，要在 Apache 的配置文件 httpd.conf 或 srm.conf 文件中加入下面的字符串。告诉 Apache 服务器，扩展名 php3 是一个特殊的程序文件。当然扩展名可以设成别的扩展名，还有一些网站将 php 的程序扩展名设为 phtml 也是不错的选择，反正这就要看 Webmaster 的规划了。

```
AddType application/x-httpd-php3 .php3
```

在 PHP 4.x 版的方法大致和 PHP 3.0.x 版相同，不同的地方在于 PHP 4.x 的目录名称及编译后的模块放置目录不同。此外，默认的扩展名也由 .php3 变成了 .php。当然在安装前还要先下载 PHP 4.x 的程序才行。

gzip -dc apache_1.3.x.tar.gz | tar xvf -

gzip -dc php-4.0.x.tar.gz | tar xvf -

cd apache_1.3.x

./configure --prefix=/www

cd ../php-4.0.x

./configure --with-mysql --with-apache=../apache_1.3.x --enable-track-vars

make

make install

cd ../apache_1.3.x

./configure --prefix=/www --activate-module=src/modules/php4/libphp4.a

make

make install

cd ../php-4.0.x

cp php.ini-dist /usr/local/lib/php.ini

在 httpd.conf **或** srm.conf **加入**

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

而 PHP 4.x **版中对** Apache **服务器加入了新的环境变量配置项。**

```
php_value [PHP directive name] [value]
php_flag [PHP directive name] [On|Off]
php_admin_value [PHP directive name] [value]
```

```
php_admin_flag [PHP directive name] [On|Off]
```

在 PHP 3.0.x 版中，有些目录可能会有 .htaccess 的文件，使用 PHP 4.0.x 版的系统，必须将这个文件拿掉，可以使用改名字的方式或者直接删除。

当一切配置好了之后，重新执行 Apache 服务器。在 Apache 目录下有 bin 或是 sbin 的目录，其中会有 apachectl 的 shell 程序，输入 apachectl restart 就可以重新启动 Apache 服务器了。赶快试看看 hello, world 程序吧！

PHP 的编译配置详细选项

在详细选项上，除了上述的安装简介外，也可以在编译时加入其它的选项。

apache 模块

语法: **--with-apache=DIR**

说明: 用本选项可以让 PHP 以 apache 的模块方式使用，DIR 的字符串可以是 /usr/local/apache 或其它安装 apache 的目录

范例: --with-apache=/var/lib/apache

fhttpd 服务器模块

语法: **--with-fhttpd=DIR**

说明: 若使用 fttpd 服务器，可以使用本指令编译 PHP。用模块的方式配合 fttpd 服务器，可以有较好的效率。

Adabas D 数据库

语法: --with-adabas=DIR

说明: 数据库系统为 Adabas D 数据库时需要加本选项。关于 Adabas D 数据库的细节，可以参考 http://www.adabas.com。

范例: --with-adabas=/usr/local/adabasd

dBase 资料表

语法: --with-dbase

说明: 只要加本选项,不用其它的参数或函数库，PHP 就会让系统有存取 dBase 资料表的功能。

filePro 数据库

语法: --with-filepro

说明: 不用指定数据库路径及其它函数库等，可以读取 filePro 数据库 (唯读)。

mSQL 数据库

语法: --with-msql=DIR

说明: 提供存取 mSQL 数据库。更多的细节请参考 mSQL 的网站 http://www.hughes.com.au。

范例: --with-msql=/usr/local/Hughes

MySQL 数据库

语法: --with-mysql=DIR

说明: 提供存取 MySQL 数据库。更多的细节请参考 MySQL 的网站 http://www.tcx.se。

范例: --with-mysql=/usr/local/mysql

iODBC 数据库装置

语法: --with-iodbc=DIR

说明: 提供 ODBC 数据库装置，用来存取后

端数据库。更多的细节请参考 iODBC 的网站 http://www.iodbc.org。

范例: --with-iodbc=/usr/local/iodbc

OpenLink ODBC 数据库装置

语法: **--with-openlink=DIR**

说明: 使用 OpenLink ODBC 数据库装置，用来存取后端数据库。更多的细节请参考 OpenLink ODBC 的网站 http://www.openlinksw.com。

范例: --with-openlink=/usr/local/openlink

Oracle 数据库

语法: **--with-oracle=DIR**

说明: 使用 Oracle 数据库。Oracle 的版本要在 7.3 版以上。您也可以在 PHP 程序中使用环境变量 ORACLE_HOME 来指定 Oracle 的路径。更多有关 Oracle 的信息请参考 Oracle 的网站 http://www.oracle.com。

范例: --with-oracle=/export/app/oracle/product/7.3.2

PostgreSQL 数据库

语法: **--with-pgsql=DIR**

说明: 使用 PostgreSQL 数据库。更多有关 PostgreSQL 的信息请参考 PostgreSQL 的网站 http://www.postgreSQL.org 或台湾的 Mirror 站 http://postgresql.ccit.edu.tw。

范例: --with-pgsql=/usr/local/pgsql

Solid 数据库

语法: **--with-solid=DIR**

说明: 使用 Solid 数据库。更多有关 Solid

的信息请参考 Solid 的网站 http://www.solidtech.com。

范例: --with-solid=/usr/local/solid

Sybase 数据库

语法: **--with-sybase=DIR**

说明: 使用 Sybase 数据库。更多有关 Sybase 的信息请参考 Sybase 的网站 http://www.sybase.com。

范例: --with-sybase=/home/sybase

Sybase-CT 数据库

语法: **--with-sybase-ct=DIR**

说明: 使用 Sybase-CT 数据库。

范例: --with-sybase-ct=/home/sybase

Velocis 数据库

语法: **--with-velocis=DIR**

说明: 使用 Velocis 数据库。有关 Velocis 数据库的进一步资料请参考 Raima 公司的网站 http://www.raima.com。

范例: --with-velocis=/usr/local/velocis

自订 ODBC 数据库驱动程序

语法: **--with-custom-odbc=DIR**

说明: 使用自订的 ODBC 函数库。当然，在使用本方式时要指定 CUSTOM_ODBC_LIBS 及 CFLAGS 变量。例如在 QNX 机器上使用 Sybase SQL Anywhere 时可能要配置系统环境变量 CFLAGS=-DODBC_QNX、LDFLAGS=-lunix 及 CUSTOM_ODBC_LIBS="-ldblib -lodbc"，并要在 PHP 配置加入 --with-custom-odbc=/usr/lib/sqlany