

Web

网站实例导学系列丛书

PHP 开发 Web 网站 实例导学

中国民航出版社

余东峰等 编 著

Web 网站实例导学系列丛书

PHP 开发 Web 网站实例导学

策划：余东峰 何学仪

主编：陆昌辉

编者：胡 勇 钟 平

陆昌辉 蔡 勇

汪 毅 倪文志

中国民航出版社

图书在版编目(CIP)数据

PHP 开发 Web 网站实例导学/陆昌辉等编著. —北京: 中国民航出版社, 2002.1
ISBN 7-80110-254-1

I. P…

II. 陆…

III. 计算机网络—基本知识

IV. TP36

中国版本图书馆 CIP 数据核字(2002)第 12664 号

PHP 开发 Web 网站实例导学

陆昌辉 等编著

出版	中国民航出版社
社址	北京市朝阳区光熙门北里甲 31 号楼 (100028)
发行	中国民航出版社
电话	64290477
印刷	中国电影出版社印刷厂
开本	787×1092 1/16
印张	19.25
字数	462 千字
版本	2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

书号	ISBN 7-80110-254-1/G-096
定价	28.00 元

(如有印装错误, 本社负责调换)

前 言

几年前,也就是在 1994 年,Rasmus Lerdorf 为了监视看他在线简历的人而创造了“个人主页工具(Personal Home Page Tools)”。后来越来越多的人对这种语言感兴趣,并开始对其扩展提出了各种建议。在众多职业的和业余的程序员们的无私奉献下,它演变成特点丰富的 PHP 语言。根据 Netcraft 在 1999 年 10 月的报告,有 931122 个域和 321128 个 IP 地址利用 PHP 技术。

PHP 是完全免费的,任何人都可以从 PHP 站点(<http://www.php.net>)自由下载。PHP 遵守 GNU 公共许可(GPL),可以不受限制的获得源代码。PHP 在 UNIX、GUN/Linux 和微软 Windows 平台上均可以运行,对于各种 Web 服务器和数据库,它都能广泛而良好地支持。PHP 与同样免费而又优秀的 Web 服务器 Apache 可以以静态编译的方式结合起来,这样极大的利用了系统资源,从而达到了最快的运行速度,同时也可让数据库发挥最佳效能。而且,PHP 具有公认的良好安全性。所以,PHP + Apache 正成为人们开发网站的第一选择。

最新的 PHP 4.0 是更有效的、更可靠的动态 Web 页开发工具,其脚本描述更强大并且更复杂,最显著的特征是速率比的增加。

对于初学者,只要了解一些基本的语法和语言特色,就可以开始你的 PHP 编码之旅了。PHP 的语法接近于 C、Perl,运行原理接近于 ASP 或者 JSP,所以,对于较熟悉上述语言之一的人来说,PHP 就显得特别简单了,不需要多少时间就可以基本掌握 PHP 的语言特点,就可以用 PHP 开始创建真正意义上的动态网站了。同样地,如果你学会了 PHP,那么你现在学习其他几种语言也会感到很简单的。

请您相信,选择 PHP 将是一个十分正确的选择。

本书由余东峰、何学仪策划,陆昌辉主编,胡勇、钟平、陆昌辉、蔡勇、汪毅、倪文志等编写。此书在编写过程中,还得到了陈天霞、舒振等同志的大力支持,在这里一并表示感谢。由于时间紧促,作者水平有限,书中存在错误在所难免,恳请广大读者指正批评。

目 录

第一章 预备知识	1	2.2.8 文件处理及数据存储	74
1.1 PHP 技术简介	1	2.3 思考与练习	79
1.2 建立 PHP 与 Web 数据库的运行平台	5	第三章 PHP 高级应用	80
1.2.1 Win32 平台下 PHP 系统的安装	5	3.1 发送和接收电子邮件	80
1.2.2 在 Unix 和 Linux 系统下 PHP 系统的安装	7	3.1.1 简单邮件发送	80
1.2.3 PHP 配置	9	3.1.2 使用 SMTP 发送邮件	81
1.3 HTML 简介	12	3.1.3 接收 POP3 邮件	86
1.3.1 HTML 代码结构	13	3.2 动态图形生成	94
1.3.2 格式化文本	13	3.2.1 线形图	95
1.3.3 字体修饰	16	3.2.2 柱状图	99
1.3.4 超文本链接的实现	18	3.2.3 饼状图	101
1.3.5 添加列表	20	3.2.4 3D 饼状图	104
1.3.6 表格设计	25	3.3 PHP 与 LDAP 目录服务	108
1.3.7 图像插入	30	3.3.1 什么是 LDAP	108
1.3.8 表单	34	3.3.2 PHP 对 LDAP 的支持	112
1.3.9 框架	36	3.3.3 一个简单的 LDAP 浏览器	114
1.4 思考与练习	39	3.4 PHP 与 XML 语言	121
第二章 PHP 基础	40	3.4.1 什么是 XML	121
2.1 PHP 概述	40	3.4.2 PHP 对 XML 的支持	124
2.1.1 Web 编程技术简介	40	3.4.3 XML 应用	125
2.1.2 PHP 程序结构及执行流程	41	3.5 思考与练习	129
2.1.3 PHP 的客户端代码	42	第四章 Web 数据库	130
2.1.4 PHP 的使用规则	42	4.1 数据库技术概述	130
2.2 PHP 的脚本语言	43	4.1.1 数据库技术	130
2.2.1 数据类型	44	4.1.2 SQL 语言	132
2.2.2 运算符	51	4.2 目前流行的数据库产品	138
2.2.3 流程控制语句	54	4.3 PHP 与 MySQL	142
2.2.4 函数	62	4.4 数据库应用例子	145
2.2.5 数组	66	4.5 思考与练习	150
2.2.6 字符串	70	第五章 客户认证	151
2.2.7 面向对象的编程	72	5.1 cookie	151
		5.2 session 基础	154

目 录

5.3 思考与练习	162	8.4.2 答案数据库的建立 ...	205
第六章 聊天室	163	8.4.3 主观题批改分析及界面建立	205
6.1 聊天室的界面	163	8.4.4 脚本源代码	206
6.2 注册与登录	164	8.5 考分的统计和公布	212
6.3 发言	168	8.5.1 基本查询的实现	212
6.4 显示	173	8.5.2 详细情况查询	216
6.5 思考与练习	175	8.6 思考与练习	222
第七章 留言本	176	第九章 图书管理系统	223
7.1 留言本的功能分析	176	9.1 需求分析及模块划分	223
7.2 数据库的建立	177	9.2 数据库的建立	225
7.3 查看留言模块的实现	179	9.3 通用模块的建立	226
7.4 用户提交留言	184	9.4 管理员模块	227
7.5 回复留言	188	9.4.1 借出	231
7.6 删除留言	193	9.4.2 还入	233
7.7 思考与练习	197	9.4.3 续借	235
第八章 在线考试	198	9.4.4 用户管理	236
8.1 功能分析	198	9.4.5 新书入库	239
8.2 Web 试卷的实现	199	9.5 读者模块	242
8.2.1 界面设计	199	9.5.1 借书记录查询	244
8.2.2 试卷结构分析及源代码	199	9.5.2 网上续借	246
8.3 建立数据库	204	9.5.3 书籍查询	247
8.4 批改试卷	204	9.6 思考与练习	251
8.4.1 批改方法分析	204	附录 PHP 函数索引	252

第一章 预备知识

本章学习目标:

- ◆了解 PHP 技术发展历程与运行原理
- ◆建立 PHP 的运行平台
- ◆HTML 的基础知识

PHP 技术是当今最流行的网站构建语言之一。本章将对 PHP 技术做一个概要的浏览,使读者知道它的来龙去脉,并给出一个最简单的 PHP 程序,让读者对 PHP 的程序结构有清晰的认识,为本书的进一步学习打下较好的基础。

PHP 的安装比较复杂,与之配套的 Web 服务器和数据库的安装也不简单。另外,由于 PHP 是一种 HTML 内嵌式语言,所以想要熟练地应用 PHP,对 HTML 也应有一定的掌握。当然,预备知识只是为初学者准备的,如果具备了这些知识,完全可以跳过这一章。

1.1 PHP 技术简介

1. PHP 技术的历史起源

PHP 的全名是一个递归的缩写名称,很有趣,“PHP Hypertext Preprocessor”,打开缩写还是缩写,里面的“PHP”的意思是“Personal Home Page”。PHP 是一种 HTML 内嵌式的语言(类似 JSP:Java Server Pages 或者 IIS 上的 ASP:Active Server Pages)。而 PHP 的语法混合了 C、Java、Perl 以及 PHP 自己独特的新语法。它可以比 CGI 或者 Perl 更快速的执行动态网页。

PHP 最初是 Rasmus Lerdorf 作为个人爱好开始出现在 Web 上的。在 1995 年他以 Personal Home Page Tools(PHP Tools)为名开始对外发表第一个版本。在这个版本中,提供了访客留言本、访客计数器等简单的功能。随后有新的成员加入到开发行列之中,1995 年中,第二版的 PHP 问世。第二版改名为 PHP/FI(Form Interpreter)。PHP/FI 加入了对 mSQL 的支持,自此奠定了 PHP 在动态网页开发上的基础。在 1996 年底,有一万五千个 Web 网站使用了 PHP/FI;在 1997 年中,使用 PHP/FI 的 Web 网站超过五万个。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 依靠与 Apache Web 服务器程序紧密结合的特性,加上版本的不断更新及加入新的功能,几乎支持所有主流与非主流的数据库,新的函数库不断加入,使得 PHP 无论在 UNIX/Linux 或是 Win32 的平台上都可以使用。

PHP 的第四代 Zend 核心引擎已经进入实用阶段。整个程序的核心大幅更动,让程序的执行速度得到了很大的提高。优化之后的效率,已较传统 CGI 或者 ASP 等程序有更好的表现,而且还有不断增强的新功能、不断加入的函数库。无论您接不接受,PHP 都将在 WebCGI 的领域上,引起革命性的影响。

2. PHP 程序的运行原理

Internet 上的 Web 站点包含了大量的静态 HTML 页面,这些站点的页面是一些由简单的

文本、图片和其他文档的连接所组成的文档。用 HTML 和 CSS 层叠表可以完成页面的结构和表示,用 JavaScript 或 Java 小程序可以使页面变得更加丰富有趣。

但在更多的情况下,大量的 Web 站点都要用到数据库。根据数据库的内容和用户的动作,动态改变 Web 页面的内容。这就是 PHP 产生的原因。通过在服务器端运行 PHP 程序,我们可以建立非常强大的应用程序,让它从数据库中提取数据从而动态产生所需 Web 页面。

PHP 是一种 HTML 内嵌式的服务器端脚本语言。它所生成的页面与 HTML 页面的区别主要在于服务器怎样处理它们。通常当页面为静态的 HTML 页面时,客户端的浏览器向服务器发送一个请求,服务器读取了这个请求,便在服务器端寻找所请求的页面,然后通过 Internet 或 Intranet 将页面发送给浏览器。

PHP 页面的执行过程增加了一步。和简单地把页面发送给浏览器不一样,Web 服务器将根据 PHP 页面做出一些动作,根据特定的情况产生合适的内容,按照 PHP 提供的指令修改页面,然后将修改过的页面发送给浏览器。它与静态页面的显著不同是静态页面不会被解释执行,而 PHP 可以通过写一些能被服务器执行的代码,做一些在很多其他情况下做不到的事情。

比如一个很简单的例子就是图书馆的书籍查询系统,如果用静态页面就得为每种书籍作一个页面,这显然不现实。用 PHP 做的话,我们可以先将书籍的数据都放到一个数据库中,然后做个页面模板,接收到客户端的请求后,从数据库中读取数据填入模板,再经过优化将页面发送给客户端浏览器,这样我们可以节省很多空间和时间。

下面让我们举例说明 PHP 的运行原理。

【例 1.1】

```
//helloworld.php
<html>
<head>
<title> my first php program</title>
</head>
<body>
<? php
echo "hello, world!!";
? >
</body>
</html>
```

这个程序存在于服务器上。当我们在浏览器地址栏键入这个文件的 URL: `http://servername/helloworld.php` 时,服务器收到请求,在服务器上找到文件,然后调出 PHP 解释器解释执行 HTML 中的 PHP 嵌入语句(粗体),生成 HTML 文档:

【例 1.2】

```
//helloworld.html
<html>
<head>
<title> my first php program</title>
</head>
```

```
< body >
```

```
hello world!!
```

```
< /body >
```

```
< /html >
```

这个文档通过 Internet 或 Intranet 发送到客户端浏览器, 显示如图 1-1:

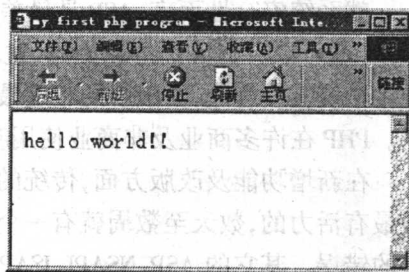


图 1-1 我的第一个 PHP 程序

3. PHP 与 ASP、JSP 等脚本语言的比较

服务器端 CGI 的编写方式有很多种, 而 PHP 只是其中的一种选择罢了。对资深的网络管理员而言, CGI 的写作界面应是随着需求而更动。对系统反应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。在其它场合, 相信使用 PHP 作为 CGI 的界面是游刃有余, 而且是最适合的。表 1-1 给出 PHP 与其它服务器后端处理技术的比较。

表 1-1

开发技术	PHP	ASP	CGI	NSAPI	ISAPI	JSP
作业平台	均可	Win32	均可	均可	Win32	均可
Web 服务器	数种	IIS	均可	NetscapeServer	IIS	多种
执行效率	快	快	慢	极快	极快	快
稳定性	佳	中等	最高	差	差	一般
开发时间	短	短	中等	长	长	较短
修改时间	短	短	中等	长	长	较短
程序语言	PHP	VB	不限	C/C++	C/Delphi	Java
网页结合	佳	佳	差	差	差	差
学习门槛	低	低	高	极高	差	较低
函数支援	多	少	不定	中等	少	较多
系统安全	佳	极差	最佳	佳	尚可	佳
使用站台	超多	多	多	极少	少	多
改进速度	快	慢	无	慢	慢	较慢

其中的 PHP 可用在多数 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 NetscapeEnterpriseServer 或 FastTrackServer)上才可以执行, 但可支援多种操作系统(UNIX 或 Win32); ASP 及 ISAPI 只在 IIS 上才有完整的功能。

在稳定性上, 由于 NSAPI 或 ISAPI 采用的是动态连结的方式, 因此在程序执行过程中若出现问题, 会使得 Web 服务器一起瘫痪。而 ASP 在实际应用时, 隔阵子就会使系统不稳定, 需要重新启动操作系统。PHP 在许多的网站上得到使用, 长期表现良好, 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; 当稳定性居首要位置时, 这后二种技术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 shellscript 不需要编译的过程, 直接就可以执行,

若用 Delphi 或 VC/BCB 甚至用复合语言等都要经过编译才能执行。

与网页结合的能力,PHP 和 ASP 是并驾齐驱的,其它的方式就不能内嵌进 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 的原始码,这真是叫人不敢领教。当然,传统 CGI 的程式,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。

总之,处于 Web 的后端 CGI 程序,就像鱼与熊掌一般,没有既高效率又开发方便的选择。不过相信 PHP 是较接近于开发容易、效率也不错的平衡点的。

4. PHP 技术的发展前景

PHP 的强大功能是它风靡互联网的基本武器。作为一种具有开源(OpenSource)特色的软件,它也具备其他开源软件的特点,那就是良好的可扩充性和异常丰富的软件支持。

- 数据库方面,它就能方便地存取 AdabasD、DBA、dBase、dbm、filePro、Informix、InterBase、mSQL、Microsoft SQL Server、MySQL、Solid、Sybase、ODBC、Oracle8、Postgre SQL 等主流及非主流的数据库,任何支持 ODBC 的数据库它都能操作,这就使得用 PHP 来编写数据库程序变得非常简单。
- 网络通信协议方面,它支持包括与电子邮件相关的 IMAP、POP3、网管系统 SNMP、网络新闻组 NNTP、帐号共享 NIS、全球信息网 HTTP 和 pache 服务器、目录协定 LDAP 以及其它网络的相关协议。
- PHP 程序还具有极佳的可扩充性和开放性。由于它的源代码是公开的,所以你可以根据需要为自己和为别人定制某些功能。因为是开源软件,只要新的需要一出现,在互联网上我们马上就可以找到新需要的解决方案。除此之外,PHP 的所有源码和文档都可以免费复制、编译、打印和分发,用户用 PHP 写的任何程序都属于自己,用户可以无限制地任意发布自己编写 PHP 程序。
- PHP 程序具有良好的可移植性。用 PHP 写出来的 Web 后端 CGI 程序,可以不经过或经过很少的改动移植到不同的作业平台上。例如,先以 Linux 架构的网站,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,这确实是长期规划的最好选择。
- 丰富的软件支持。在加入了其他的模块后,我们还可以得到更多的支持:如英文拼字检查、高精度度计算、PDF 文档格式、Hyper wave 服务器、图形处理、编码与解码功能、WD-DX 功能、qmail 与 vmmailmgr 系统、压缩文件处理以及 XML 解析等。此外,一般语言有的数学运算、时间处理、文件系统、字符串处理、进程处理等功能,它一样都不缺。

总而言之,在互联网迅速发展的今天,Web 网络上的新技术、新语言层出不穷。PHP 技术依靠开放源码运动的支持,使得它能够迅速跟上甚至引领 Web 技术潮流,这是它能够成为当今主流网络技术的重要原因。面对将来信息科技的瞬息万变,我们要尽量减少许可证的使用费用,控制网站的软硬件预算,并且为适应发展的需要,必须具备在不同的服务器和操作系统之间的可移植性。我们也要确保能消除一些在服务器或脚本环境下发现的程序错

误,理解脚本的工作原理,并能够为不同的需要修改我们的服务器和脚本。那么,作为开放源码软件的 PHP 技术将是您最佳的选择,互联网上无数和您一样优秀的程序员将保证 PHP 及其支持软件的先进性。采用 PHP 将大大降低了构建网站的成本,使得 PHP 成为广大中小网站的首选。

5. 本书例程的环境配置

本书的大多数例程都在 Windows98 + Apache - 1.3.19 - Win32 + PHP - 4.0.6 - Win32 + MySQL - 3.23.22 - beta 的环境中测试过。少数例程在 RedHat7.0 + Apache1.3.19 + PHP4.0.2 + MySQL3.23.22 的环境中测试。当然,在实际应用中, 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。而许多成功站点的经验,更是采用这种组合最好的搭配。在本书的写作过程中,为了调试的方便以及读者的易于入门,我们采用了 Win32 + Apache + PHP + Mysql 的配置,这对读者的学习没有任何不良的影响。

下面将 PHP 的概况给大家作一个介绍:

- 适用人群:熟悉 Unix 环境,需要在最少的投入下完成快速的应用开发。
- 适用平台:FreeBSD/Linux/Unix 操作系统、Apache 服务器适用应用:Internet 高访问量、快速数据库开发的应用。
- 推荐开发工具:Home Site/PHPEd
- 推荐开发环境:FreeBSD 4.1/RedHat 7.0 操作系统/Apache 1.3.19/MySQL 3.22/P 200/64M RAM/9G HD
- 推荐应用环境(最少):FreeBSD 3.5/MySQL 3.22/Apache 1.3.19/P 200/128M RAM/9G HD
- 推荐应用环境(最佳):FreeBSD 3.5/Sybase 11/Apache 1.3.19/P III 500/512M RAM/18G HD

1.2 建立 PHP 与 Web 数据库的运行平台

PHP 是跨平台的,基本上所有的操作系统和 Web 服务器都支持它。而它支持的数据库也十分广泛。多数用户使用 Apache + PHP + MySQL。

1.2.1 Win32 平台下 PHP 系统的安装

在 Windows 下安装 PHP,最简单的方法是用预编译的二进制文件,它可以从 <http://www.php.net/downloads.php.php4> 下载。若需要一些外挂模块的支持,我们最好看一下系统是否具有相应的动态链接库。表 1-2 是摘录的动态链接库。

表 1-2

动态链接库文件	说 明
Php_mssql70.dll	连接 MS SQL7.0 数据库的函数
Php_exif.dll	exif 函数
Php_ftp.dll	FTP 函数
Php_imap.dll	IMAP 4 函数
Php_ldap.dll	LDAP 函数
Php_calendar.dll	日历会话函数
Php_zlib.dll	zlib 函数
Php_dbase.dll	xBase(DBF) 函数
Php_interbase.dll	连接 InterBase 数据库的函数

PHP 在 Windows 下的安装,分为两部分:PHP 及其模块的安装和 Web 服务器的配置。PHP 及其模块的安装,包括 PHP 核心文件的安装,相关扩展模块的安装及 php.ini 文件的各种选项的设定。

1. PHP 及其模块的安装

PHP 的安装比较简单。首先下载 PHP 二进制代码文件,并把它解压。接着就要修改 PHP 的配置文件 php.ini。把 PHP 的安装目录下的样本配置文件 php.ini 拷贝到 Windows 目录下(c:\windows 或 c:\winnt)。有几个地方需要修改:

①Extension_dir:扩展模块或外挂动态链接库的路径。可以不修改,而把所有的 DLL 文件复制到 Windows 的 system 目录下,使 PHP 启动时能找到所需的动态链接库文件。

②Mail 函数的相关配置。Unix 和 Linux 下的 PHP 用 sendmail 来发送邮件,在 Windows 下用 SMTP 服务器。设置如下:

```
[mail function]
```

```
SMTP = public.bta.net.cn
```

```
Sendmail_from = webmaster@yourweb.com
```

③浏览器的兼容性设置。把变量 browscap.ini 指向你机器上的文件。

这样,PHP 的安装就算完成了。

2. Web 服务器的安装和配置

在安装 Web 服务器之前,必须确保机器已安装了 TCP/IP 协议。下面我们以 Apache 为例做简单介绍。

首先,安装 Apache for Win32。Apache 的二进制执行文件可以从 <http://www.apache.org> 得到,如同 apache_x.x.x_win32.exe,其中 x.x.x 是 apache 的版本号码。运行这个文件,开始安装。

然后对配置文件 httpd.conf 进行修改,它在安装目录下的 conf 目录中。

必须要做的是定义 ServerName。打开 httpd.conf,找到 ServerName,插入一行:

```
servername localhost
```

如果你希望 index.php 能作为当前目录下的默认文档,可以修改:

```
DirectoryIndex index.html index.htm index.php
```

CGI 方式和模块状方式的安装不同,还需要其他的修改。

对于 CGI 方式,要找到 ScriptAlias 段,在后面插入:

```
scriptalias /php c:/php
```

找到 AddType 段,在后面插入:

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

找到 Action 段,在后面插入:

```
Action application/x-httpd-php /php/php.exe
```

如果是模块方式,那么完全不同。首先,下载符合你的 PHP 版本的模块文件,是 mod_php4_x.x.x_pl2.zip。把它解压,得到三个文件:mod_php4.dll, mod_php4.conf 和 readme.txt。把 mod_php4.dll 复制到 Apache 的 modules 目录,把 mod_php4.conf 复制到 Apache 的 conf 目录下,至于 readme.txt,看一下就够了。

接着,修改 Apache 的 conf 目录下的 srm.conf 文件,在后面加上:

```
include conf/mod_php4.conf
```

这样,安装就算完成了。

1.2.2 在 Unix 和 Linux 系统下 PHP 系统的安装

在 Unix 系统下安装 PHP 非常简单,运行 autocom 脚本,它会自动生成所需的 make 文件,然后执行 make 和 install。

下面以 RedHat6.1(Linux2.2.12)为例来说明 Linux 系统下 PHP 的安装。若某一模块的头文件不在默认的路径下,而我们希望 PHP 能找到它,那么在配置前,可以用 CPPFLAGS 环境变量指定它的路径,命令如下:

```
$ CPPFLAGS = -I/home/php/myinclude; export CPPFLAGS
```

跟上面的类似,要让 PHP 自动搜索安装程序所需要的库,需要设置 LDFLAGS 变量:

```
$ LDFLAGS = -I/home/php/mylibs/; export LDFLAGS
```

首先,下载 PHP 源代码包,保存到 /usr/local/src/ 目录下。然后,解压源代码:

```
$ gunzip php-x.x.x.tar.gz
```

```
$ tar xvf php-x.x.x.tar
```

这样,会产生一个叫 php-x.x.x 的目录(php-x.x.x 代表 PHP 的当前版本),所有的代码都放在这个目录下。

1. 建立 PHP 的 CGI 解释器版本

由于建立 PHP 的 CGI 解释器版本是默认的,所以很容易。

运行 configure 命令,将生成必需的 Make 文件。输入选项,告诉脚本程序模块的构成,同时确认配置的参数。以下是常用的命令:

```
$ ./configure -with -ldap -other -options
```

这里的 -with -ldap 选项是一个例子,告诉我们如何在安装时指定需要的外挂模块。-with -ldap 选项使得配置脚本能支持 LDAP(Light-weight Directory Access Protocol) API。

很多人希望 PHP 能够支持 MySQL,因为它们是天才的搭档。若是 PHP3,要使用下面的命令:

```
$ ./configure.. -with -mysql -other -options
```

但是 PHP4 内置了对 MySQL 的支持,即 MySQL 已成为 PHP4 内核的一部分,在编译时无须指定。

如果配置脚本运行顺利,接下来就是编译源代码:

```
$ make
```

若库的定位不准确,可以检查环境变量:LD_LIBRARY_PATH;如果不存在,在变量中加入此库的路径。比如,库 libxyz.so 位于一个不规则的目录/var/yourlibdir 下,我们需要在 Bash shell 下执行如下命令,再重新运行 make 程序。

```
$ export LD_LIBRARY_PATH = SLD_LIBRARY_PATH:/var/yourlibdir
```

```
$ make
```

当源代码被成功编译和连接之后,就可以安装了。在此之前假定我们有 root 权限。

```
$ su -
```

```
Password:
```

```
# make install
```

如果我们只想安装 PHP 的 CGI 版本,那我们就可以到此为止。

2. 把 PHP 编译为 Apache 模块

如果我们要把 PHP 编译成 Apache 的模块,必须拥有 Apache 的源代码。在 <http://www.apache.org/dist> 下载最新版本,把软件包保存到 /usr/local/src 下,并解压出源代码:

```
$ tar xvzf apache_1.3.x
```

会产生 apache_1.3.x 的目录,切换到这个目录,运行

```
$ cd apache_1.3.x
```

```
$ ./configure --prefix=/usr/local/apache --other-option-for-apache
```

configure 脚本在建立 Apache 模块时能提供许多选项,比如,选项 --with-apache=../apache_1.3.x,表明建立 Apache 模块,同时也表明了 Apache 源代码的位置。

现在开始编译:

```
$ make
```

编译和链接后,就可以进行安装了,同样假设我们有 root 权限。

```
$ su -
```

```
password
```

```
$ make install
```

接下来转到有 apache 源代码的目录下,运行 configure 脚本。

```
$ cd ../apache_1.3.x
```

```
$ ./configure --active=src/modules/php3/libphp3.a --other-apache-options
```

然后编译:

```
$ make
```

接着安装(取得 root 权限,如果正在运行 Apache):

```
$ su -
```

```
password:
```

```
# /usr/local/apache/bin/apachectl atop
```

```
# make install
```

最后对 Apache 的配置文件/etc/httpd/conf/httpd.conf 进行一些编辑,加入下面的配置信息:

```
Addtype application/x-httpd-php.php
```

到此为止,全部装完,可以进行调试了。

3. 把 PHP 安装为 Apache 的可加载模块

有几种方法可以实现,我们只学习最受欢迎,当然也是较简单的方法:使用 apxs 脚本。

首先要确保已能使用 mod_so 选项来编译 Apache。可以用以下命令编译:

```
bash $ cd /usr/web/src/apache_1.3.x
```

```
bash $ ./configure --with-layout=Apache --prefix=/usr/web/server/apache \
```

```
--enable-module=mime --magic --enable-shared=mime --magic \
```

```
--enable-module=headers --enable-shared=headers \
```

```
--enable-module=info --enable-shared=info \
```

```
--enable-module=rewrite --enable-shared=rewrite \
```

```
--enable-module=speling --enable-shared=speling
```

```
bash $ make
```

```
bash $ make install
```

然后我们可以建立 apxs 支持的 PHP, 并进行安装。

```
bash $ cd /usr/web/src/php/php-4.0.x
bash $ export CPPFLAGS = " -I/usr/local/expat/"
bash $ export LDFLAGS = " -L/usr/local/expat/"
bash $ ./configure --with-afx = /usr/web/server/apache/bin/apxs \
    --with-config-file-path = /usr/web/server/php3 \
    --with-mysql = /usr/web/databases/mysql/ \
    --with-gd = /usr/local/gd1.3/ \
    --with-xml \
    --enable-track-vars \
    --enable-magic-quotes \
    --enable-debugger
```

```
bash $ make
```

```
bash $ make install
```

```
bash $ cp php3.ini -dist /usr/web/server/php3
```

无论采用何种方式, 都必须在 Apache 的配置文件(通常为 httpd.conf)中加入几行, 让服务器知道如何处理 PHP 脚本。

加上如下 PHP 扩展名与合适的路径, 如果要编译独立的或可加载的 PHP。

```
Addtype application/x-httpd-php.php3.php
```

```
Addtype application/x-httpd-php3-source.phps
```

加上如下命令使 index.php 能够像 index.html 一样, 在服务器使用它们时可作为默认值。

```
DirectoryIndex index.html index.shtml index.cgi index.php
```

加上如下的命令, 使得在列出目录中文件时可以显示正确的图标。

```
Addition /icons/script.gif.conf.sh.shar.csh.ksh.tcl.php.phps
```

加上如下命令, 启动 PHP 引擎, 并打开显示错误信息的选项。

```
Php_admin_flag engine on
```

```
Php_admin_flag display_errors on
```

现在, 一切都完了, 运行吧。

1.2.3 PHP 配置

下面我们将探讨如何根据特定的安装与需求来配置 PHP。

有三种方法:

- (1) 在建立 PHP 的时候进行配置: 把参数提供给 configure 脚本, 或运行 PHP 的 setup 脚本。
- (2) 修改 PHP 的配置文件 php.ini。
- (3) 利用 Apache 的 PHP 配置命令。

配置的当前值可以用函数 phpinfo() 得到。它将提供所有经过配置的变量值以及 PHP 的列表。要得到某个单独变量的值, 可以调用函数 get_cfg_var()。例如, 采用如下语句可得到变量 include_path 的值:

```
<? php
```

```
echo get_cfg_var("include_path");
```

? >

1. 在编译时进行编译

在编译时配置对于指定所需要安装的外挂模块很有用。比较好的做法是利用编译时配置方法选择所需要安装的外挂模块,再用 Apache 命令或 php.ini 文件来更灵活的设置。

Setup 脚本应在编译之前运行:

```
$ ./setup
```

这一脚本用交互的方式询问用户在编译时需要何种支持,它产生一个叫做 do-conf 的文件,包含了进行配置时所需的选择,我们可用如下命令:

```
$ ./do-conf
```

```
$ ./configure
```

另一种方法就是跳过 setup 和 do-conf 脚本而直接用 configure 脚本进行配置。表 1-3 详细介绍所用到的参数的含义。

表 1-3

特性或附加模块	传给 configure 的参数	说 明
检查内部的重定向	--enable-force-cgi-redirect	检查一个给定的请求是否为服务器的内部请求。如果 PHP 以 CGI 方式运行,这个对于增强安全很有帮助。
配置文件的位置	--with-config-file-path=DIR	指定配置文件 php.in 的位置
可执行文件的目录	--with-exec-dir=DIR	如果 PHP 运行在安全模式下,只有这个目录中的文件允许执行
对外部传来的数据进行自动转义	--enable-magic-quotes	从客户端或数据库传递过来的数据中可能包括引号。打开这个选项,可以对这些数据进行转义
GPC 变量	--enable-track-vars	自动跟踪 GPC (GET, POST, COOKIE) 变量。
LDAP 的支持	--with-ldap=DIR	打开对 LDAP 的支持
Mcrypt 支持	--enable-discard-path	Mcrypt 加密方法支持。Mcrypt 函数库支持很多种商业或开放源代码的基于块的加密方法
CGI 解释器的路径	--enable-discard-path	CGI 解释器可以放在 Web 服务器的根目录之外以提高安全性
正规表达式库	--with-system-regex	用操作系统的正规表达式库代替自带的
远程包含	--enable-url-include	这个选项是 include() 可以通过 HTTP 或 FTP 协议来包含文件
安全模式操作	--enable-safe-mode	使 PHP 运行在安全系统模式下

表 1-3(续表)

特性或附加模块	传给 configure 的参数	说 明
Semaphores (信号量)	--enable-sysvsem	Semaphore 在支持共享内存系统里提供了对共享内存的控制
共享内存	--enable-sysvshm	类 System V 的共享内存只有在 *nix 的系统上才能用。它用来在多个进程共享全局变量
短标记	--disable-short-tags	PHP 脚本可以被包括在 <code><?php?></code> 和 <code><?></code> 中。本选项禁止使用短标记
语法加亮	--disable-syntax-hl	在显示 PHP 脚本在源代码时,禁止使用自动的语法加亮功能
警告	--enable-maintrainer-mode	打开编译器及依赖警告,对于开发 PHP 的人来说很有用
XML 支持	--enable-xml	打开支持 XML 的模块 <code>expect</code>
BC 数学函数库	--enable-bcmath	BC 数学函数库的支持
调试器	--enable-debugger	打开 PHP 内置的调试器
调试特性	--enable-debug	允许跟踪 PHP 自身的问题

你可以用 `configure - help` 的方法得到详细的选项,见表 1-4。

表 1-4

数据库	传给 configure 的参数	说 明
Adabas - D	--with-adas = DIR	增加对 Adabas D 数据库的支持。DIR 是放数据库软件的目录,默认的路径是 <code>/usr/local/adabased</code>
定制 ODBC 库	--with-custom-odbc	支持任何定制的 ODBC 库。默认的位置是 <code>/usr/local</code>
dBase	--with-dbase	支持 dBase 文件,无须其他的库
Filepro	--with-filepro	支持 FilePro(只读),无须其他的库
IODBC	--with-iodbc = DIR	支持 IODBC 驱动。IODBC 是 UNIX 下的开放源代码软件。默认的路径是 <code>/usr/local</code>
MSQL	--with-mysql = [DIR]	支持 MSQL,由 DIR 给定目录。默认的路径是 <code>/usr/local/Hufhes</code>
MySQL	--with-mysql = DIR	支持 MySQL,由 DIR 给定目录,默认的路径是 <code>/usr/local/</code>