



php 相关扩展及库

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科[PEAR](#)词条，查看内容请访问网站。

php相关扩展及库.....	1
Php APC.....	2
APC的安装.....	2
用法.....	3
利用APC缓冲PHP脚本来提高服务器性能	3
Php PECL.....	4
PEAR.....	5
PEAR的特性	5
PEAR的编码规则	5
PEAR的使用	11
PEAR相关资源	11
PDO.....	11
PDO的目标.....	12
PDO的特点.....	12
安装PDO.....	12
PDO使用指南.....	13
PDO drivers	13
GD库.....	14
配置GD库.....	14
相关条目.....	15
附录.....	16

美国主机侦探(www.idcspy.com) 国内最大的美国主机交流论坛，选购美国主机的最佳去处，十大美国主机任你选 推荐优质的产品 提供一流的服务 Q: 800013301

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条，查看内容请访问网站。

推荐内容: [WordPress免费主题](#) | [WordPress免费插件](#)

Php APC

APC(Alternative PHP Cache)是一个免费强劲的[开源](#)框架, 用来缓存和优化 [php](#)中间码. 它是一种php的缓存解决方案, 目前以pecl方式发布, 有消息说将会出现在 php 6 版本的内核.

APC的安装

linux 类平台

- 从<http://pecl.php.net/package/apc> 下载相应版本
- 解压
- 进入源码目录
- 执行 php 安装目录下的 bin/phpize
- `./configure --enable-apc --enable-apc-mmap`

`--with-apxs=path-to-apache/bin/apxs`

`--with-php-config=path-to-php/bin/php-config`

- `make && make install`
- 将生成的 `apc.so` 加载到 `php.ini` (`extension=apc.so`, 注意 `extension_dir` 的设置)

一般地, 编译生成的 `.so` 会在 php 安装路径的 `lib/php/extensions` 下

- 重启, apache

windows 平台

- 只要到<http://snaps.php.net/> 的相应分支下下载 `php_apc.dll`, 再在 `php.ini` 中加载即可.

- ✓ 出处: 站长百科
- ✓ 原文地址: <http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条, 查看内容请访问网站。

用法

apc 的用法比较简单, 只有几个函数, 列举如下:

`apc_clear_cache()` 清除 apc 缓存内容. 默认(无参数)时, 只清除系统缓存, 要清除用户缓存, 需用 'user' 参数

`apc_define_constants (string key, array constants [, bool case_sensitive])` 将数组 constants 以常量加入缓存

`apc_load_constants (string Key)` 取出常量缓存

`apc_store (string key, mixed var [, int ttl])` 在缓存中保存数据

`apc_fetch (string key)` 获得 apc_store 保存的缓存内容

`apc_delete (string key)` 删除 apc_store 保存的内容

利用APC缓冲PHP脚本来提高服务器性能

在一个有[PEAR](#)支持的PHP5 系统中, APC的安装和它的执行一样简单, 在根目录下执行: `#pecl install APC`

一些分发版本或许会提供APC[二进制](#)的程序包, 所以您也许可以利用 `urpmi php-apc` 或 `apt-get` 来安装 `php-apc`。

一旦 APC 被安装之后, 编辑 `/etc/php.ini` 加入以下内容:

```
extension=apc.so[apc]apc.enabled = 1apc.shm_segments =  
1apc.shm_size = 30apc.optimization = 0apc.ttl = 7200apc.user_ttl =  
7200apc.num_files_hint = 1000apc.mmap_file_mask = /tmp/apc.XXXXXX
```

- ✓ 出处: 站长百科
- ✓ 原文地址: <http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条, 查看内容请访问网站。

这样做是为了激活 APC 缓冲，而不是启动优化器（当前处于实验阶段）。除了激活 APC 之外，应该注意的重要选项还有 `apc.ttl` 和 `apc.user_ttl`，它们定义了脚本保持在缓存中的时

间，以秒计。这一点对于处理提供很多文件的[服务器](#)来说真的很重要；它将防止缓存因被陈旧的条目装满而无法为新的条目提供缓冲的情况出现。

您也可以调整 `apc.num_files_hint`，它为 APC 提供了一个近似值，表示了被请求的或包括在您系统中 PHP 源文件的数量。其缺省值是 1000，但是如果您知道要处理更多文件，可以将值

增加到一个最接近的估计值。同样地，如果文件会少很多的话，也可以相应地减少。

最后，在 APC 源程序包中，有一个叫 `apc.php` 的脚本，它用来从 APC 缓存中获取详细的数据，使得您能够进一步调整 APC 的设置。要使用这个脚本的话，您就需要编辑脚本和改变

`ADMIN_PASSWORD` 选项。通过调整选项，并利用这个脚本确定这种变化所产生的效果，您就可以真正提高 PHP 脚本的性能了。

Php PECL

PECL的全称是The PHP Extension Community Library，是一个开放的并通过[PEAR](#)(PHP Extension and Application Repository，[PHP](#)扩展和应用仓库)打包格式来打包安装的PHP扩展库仓库。

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条，查看内容请访问网站。

PEAR

PEAR是[PHP](#)扩展与应用库 (the PHP Extension and Application Repository) 的缩写。它是一个PHP扩展及应用的一个代码仓库，简单地说，PEAR就是PHP的CPAN。

PEAR的特性

1. PEAR 按照一定的分类来管理 PEAR 应用代码库，你的 PEAR 代码可以组织到其中适当的目录中，其他的人可以方便地检索并分享到你的成果。

2. PEAR 不仅仅是一个代码仓库，它同时也是一个标准，使用这个标准来书写你的 PHP 代码，将会增强你的程序的可读性，复用性，减少出错的几率。

3. PEAR通过提供 2 个类为你搭建了一个[框架](#)，实现了诸如析构函数，错误捕获功能，你通过继承就可以使用这些功能。

PEAR的编码规则

PEAR的编码规则包括：缩进规则，控制结构，函数调用，函数定义，注释，包含代码，PHP标记，文件头的注释块，CVS标记，[URL](#)样例，常量的命名这 11 方面。下面简要地介绍一下：

- 缩进规则

PEAR 中需要使用 4 个空格来缩排代码，并且不使用 TAB。如果你使用 VIM，将下列设置放入你的`~/.vimrc`中：

```
set expandtab
```

```
set shiftwidth=4
```

```
set tabstop=4
```

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条，查看内容请访问网站。

如果，你使用 Emacs/XEmacs, 需要把 `indent-tabs-mode` 设置成 `nil`。

不过你象我一样喜欢用(X)Emacs 编辑 PHP 文件，我强烈推荐你安装 PHP-MODE，这样当你编写 PEAR 代码的时候，它会自动调整你的缩排风格，当然 PHP-MODE 还有许多很优秀的特性，你可以从资源列表中的地方下载最新版的 PHP-MODE。

- 控制结构

这里所说的控制结构包括: `if for while switch` 等。对于控制结构，在关键字（如 `if for ..`）后面要空一个格，然后再跟控制的圆括号，这样，不至于和函数调用混淆，此外，你应该尽量完整的使用花括号 `{}`，即使从语法上来说是可选的。这样可以防止你以后需添加新的代码行时产生逻辑上的疑惑或者错误。这里是一个样例：

```
if ((条件 1) && (条件 2)) {  
  
    语句 1;  
  
}elseif ((条件 3) || (条件 4)) {  
  
    语句 2;  
  
}else {  
  
    语句 3;  
  
}
```

- 函数调用

对于函数调用，函数名和左括号之间不应该有空格，对于函数参数，在分隔的逗号和下一个参数之间要有相同的空格分离，最后一个参数和右括号之间不能有空格。下面是一个标准的函数调用；

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条，查看内容请访问网站。

```
$result = foo($param1, $param2, $param3);
```

不规范的写法:

```
$result=foo ($param1,$param2,$param3);
```

```
$result=foo( $param1,$param2, $param3 );
```

此外, 如果要将函数的返回结果赋值, 那么在等号和所赋值的变量之间要有空格, 同时, 如果是一系列相关的赋值语句, 你添加适当的空格, 使它们对齐, 就象这样:

```
$result1 = $foo($param1, $param2, $param3);
```

```
$var2     = $foo($param3);
```

```
$var3     = $foo($param4, $param5);
```

- 函数定义

函数定义遵循“one true brace”习俗:

```
function connect(&$dsn, $persistent = false)
{
    if (is_array($dsn)) {
        $dsninfo = &&dsn;
    } else {
        $dsninfo = DB::parseDSN($dsn);
    }
}
```

- ✓ 出处: 站长百科
- ✓ 原文地址: <http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条, 查看内容请访问网站。

```

        if (!$dsninfo || !$dsninfo['phptype']) {

            return $this->raiseError();

        }

        return true;

    }

```

如上所示,可选参数要在参数表的末端,并且总是尽量返回有意义的函数值。

- 关于注释

对于类的在线文档,应该能够被PHPDoc转换,就象JavaDoc那样。PHPDoc也是一个PEAR的应用程序,更详细的介绍你可以去 <http://www.phpdoc.de/> 查看。除了类的在线文档,建议你应该使用非文档性质的注释来诠释你的代码,当你看到一段代码时想:哦,我想不需要在文档里去仔细描述它吧。那么你最好给这段代码作一个简单的注释,这样防止你会忘记它们是如何工作的。对于注释的形式,C的 `/* */`和C++的`//`都不错,不过,不要使用Perl或者shell的`#`注释方式。

- 包含代码

无论什么时候,当你需要无条件包含进一个 class 文件,你必须使用 `require_once`;当你需要条件包含进一个 class 文件,你必须使用 `include_once`;这样可以保证你要包含的文件只会包含一次,并且这 2 个语句共用同一个文件列表,所以你无须担心二者会混淆,一旦 `require_once` 包含了一个文件,`include_once` 不会再重复包含相同的文件,反之亦然。

- PHP 代码标记

任何时候都要使用`<?php ?>`定义你的 php 代码,而不要简单地使用`<? ?>`,这样可以保证 PEAR 的兼容性,也利于跨平台的移植。

- ✓ 出处: 站长百科
- ✓ 原文地址: <http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条, 查看内容请访问网站。

- 文件头的注释声明:

所有需要包含在 PEAR 核心发布的 PHP 代码文件, 在文件开始的时候, 你必须加入以下的注释声明:

```
/* vim: set expandtab tabstop=4 shiftwidth=4: */

//
+-----+
--+

// | PHP version 4.0
|

//
+-----+
--+

// | Copyright (c) 1997, 1998, 1999, 2000, 2001 The PHP Group
|

//
+-----+
--+

// | This source file is subject to version 2.0 of the PHP license,
|

// | that is bundled with this package in the file LICENSE, and is
|

// | available at through the world-wide-web at
|
```

- ✓ 出处: 站长百科
- ✓ 原文地址: <http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条, 查看内容请访问网站。

```
// | http://www.php.net/license/2_02.txt.
|
// | If you did not receive a copy of the PHP license and are unable
to |
// | obtain it through the world-wide-web, please send a note to
|
// | license@php.net so we can mail you a copy immediately.
|
//
+-----+
--+
// | Authors: Original Author |
// | Your Name |
//
+-----+
--+
//
// $Id$
```

对于不在 PEAR 核心代码库中的文件，建议你也在文件的开始处有这样一个类似的注释块，标明版权，协议，作者等等。同时也在第一行加入 VIM 的 MODELINE，这样在 VIM 中能够保持 PEAR 的代码风格。

- CVS 标记

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条，查看内容请访问网站。

如上面所展示那样，在每个文件中加入 CVS 的 ID 标记，如果你编辑或修改的文件中没有这个标记，那么请加入，或者是替换原文件中相类似的表现形式(如“Last modified”等等)

- URL 样本

你可以参照 RFC 2606, 使用“www.example.com”作为所有的 URL 样本。

- 常量命名

常量应该尽量使用大写，为了便于理解，使用下划线分割每个单词。同时，你应该常量所在的包名或者是类名作为前缀。比如，对于 Bug 类中常量应该以 Bug_ 开始。以上是 PEAR 的编码规则，详细的编码规则可以参考 PEAR 中的 CODING_STANDARD 文件的说明。为了更好地理解这些编码规则，你也可以参考一下现有 PEAR 核心模块的代码。

PEAR的使用

PEAR相关资源

[官方网站](#) | [PEAR官方下载](#)

PDO

PDO是[PHP5](#) 新加入的一个重大功能，因为在PHP 5 以前的php4/php3 都是一堆的[数据库](#)扩展来跟各个数据库的连接和处理，什么 php_mysql.dll、php_pgsql.dll、php_mssql.dll、php_sqlite.dll等等扩展来连接[MySQL](#)、[PostgreSQL](#)、MS [SQL Server](#)、[SQLite](#)，同样的，我们必须借助ADODB、PEAR::DB、PHPlib::DB之类的数据库抽象类来帮助我们，无比烦琐和低效.PHP6 中将默认使用PDO连接数据库，所有非PDO扩展将会在PHP6 被从扩展中移除。

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条，查看内容请访问网站。

PDO的目标

- 提供一种轻型、清晰、方便的[API](#)
- 统一各种不同 RDBMS 库的共有特性，但不排除更高级的特性。
- 通过 PHP 脚本提供可选的较大程度的抽象/兼容性。

PDO的特点

- 性能

PDO 从一开始就吸取了现有数据库扩展成功和失败的经验教训。因为 PDO 的代码是全新的，所以我们有机会重新开始设计性能，以利用 PHP 5 的最新特性、能力。PDO 旨在将常见的数据库功能作为基础提供，同时提供对于 RDBMS 独特功能的方便访问。

- 简单

PDO 旨在使您能够轻松使用数据库。API不会强行介入您的代码，同时会清楚地表明每个函数调用的过程。运行时可扩展。PDO扩展是模块化的，使您能够在运行时为您的数据库后端加载驱动程序，而不必重新编译或重新安装整个PHP程序。例如，PDO_OCI扩展会替代PDO扩展实现[Oracle](#)数据库API。还有一些用于MySQL、PostgreSQL、[ODBC](#)和[Firebird](#)的驱动程序，更多的驱动程序尚在开发。

安装PDO

- windows 平台

我的是 PHP 5.2，已经自带有了 php_pdo.dll 的扩展，不过需要稍微设置一下才能使用。打开 php 配置文件 php.ini ，找到下面这行：

extension_dir。这个就是我们扩展存在的目录，我的 PHP 5 扩展是在：
C:\php5\ext，那么我就把这行改成：extension_dir = "C:/php5/ext"

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条，查看内容请访问网站。

然后再往 php.ini 下面找到:

```
;;;;;;;;;;;;;;;;;;;;;
; Dynamic Extensions ;
;;;;;;;;;;;;;;;;;;;;;
```

下面有一堆类似 ;extension=php_mbstring.dll 的东西, 这里就是 PHP 扩展加载的配置了, 我们再最后面添加上我们 PDO 的扩展:

```
extension=php_pdo.dll

extension=php_pdo_mysql.dll

extension=php_pdo_pgsql.dll

extension=php_pdo_sqlite.dll

extension=php_pdo_mssql.dll

extension=php_pdo_odbc.dll

extension=php_pdo_firebird.dll

;extension=php_pdo_oci8.dll
```

然后重启我们的 Web 服务器就可以了, 你可以使用 phpinfo () 来检测下是否安装成功了。

PDO使用指南

- [PDO使用指南](#) [PHP中的PDO函数库](#)

PDO drivers

- ✓ 出处: 站长百科
- ✓ 原文地址: <http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条, 查看内容请访问网站。

PDO_DBLIB FreeTDS / Microsoft SQL Server / Sybase

PDO_FIREBIRD Firebird/Interbase 6

PDO_INFORMIX IBM Informix Dynamic Server

PDO_MYSQL MySQL 3. x/4. x

PDO_OCI Oracle Call Interface

PDO_ODBC ODBC v3 (IBM DB2, unixODBC and win32 ODBC)

PDO_PGSQL PostgreSQL

PDO_SQLITE SQLite 3 and SQLite 2

GD库

GD库，是[php](#)处理图形的扩展库，GD库提供了一系列用来处理图片的[API](#)，使用GD库可以处理图片，或者生成图片。在[网站](#)上GD库通常用来生成缩略图或者用来对图片加水印或者对网站数据生成报表。

在PHP处理图像，可使用GD库，而GD库开始时是支持[GIF](#)的，但由于GIF使用了有版权争议的LZW算法，会引起法律问题，于是从GD-1.6开始，GD库不再支持GIF，改为支持更好的，无版权争议的[PNG](#)。php(gd库)输出中文图像的转换函数

配置GD库

1. 用 dos 命令(也可以手动操作, 拷贝 dlls 文件夹里所有 dll 文件到 system32 目录下) copy c:\php\dlls*.dll c:\windows\system32\
2. 打开 php.ini 设置 extension_dir = "c:/php/extensions/";

- ✓ 出处：站长百科
- ✓ 原文地址：<http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条，查看内容请访问网站。

3. extension=php_gd2.dll;把 extension 前面的逗号去掉, 如果没有 php_gd2.dll, php_gd.dll 也一样, 保证确实存在这一文件
c:/php/extensions/php_gd2.dll

4. 运行下面[程序](#)进行测试

```
<?php

ob_end_flush();

//注意, 在此之前不能向浏览器输出任何信息, 要注意是否设置了
auto_prepend_file.

header ("Content-type: image/png");

$im = @imagecreate (200, 100)

    or die ("无法创建图像");

$background_color = imagecolorallocate ($im, 0, 0, 0);

$text_color = imagecolorallocate ($im, 230, 140,
150);

imagestring ($im, 3, 30, 50, "A Simple Text
String", $text_color);

imagepng ($im);

?>
```

相关条目

- [MySQL](#)

- ✓ 出处: 站长百科
- ✓ 原文地址: <http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条, 查看内容请访问网站。

- [PHP](#)

附录

PHP 编程起步自学教程:

<http://down.zzbaike.com/ebook/phpbcqbzx-750.html>

PHP 经典实例(第二版):

<http://down.zzbaike.com/ebook/phpjdshl-996.html>

PHP 与 MYSQL 彻底研究:

<http://down.zzbaike.com/ebook/phpmysqlcdyj-828.html>

更多软件、教程下载: <http://down.zzbaike.com/>

更多PHP信息: <http://www.zzbaike.com/wiki/PHP>

Apache2.2 中文文档电子书: <http://bbs.zzbaike.com/thread-9955-1-1.html>

Godaddy 主机及域名使用视频教程 (在线观看及下载)

<http://bbs.zzbaike.com/thread-50005-1-1.html>

如果您有站长类电子书, 请到这里与我们分享:

<http://bbs.zzbaike.com/forum-69-1.html>

详情见: <http://bbs.zzbaike.com/thread-23156-1-1.html>

站长百科感谢您下载阅读, 多谢支持 !

- ✓ 出处: 站长百科
- ✓ 原文地址: <http://www.zzbaike.com/wiki/PEAR>
- ✓ 本电子书整理自站长百科 [PEAR](#) 词条, 查看内容请访问网站。