

PHP 配置文件详解

本文以 PHP 5.2 的配置文件 `php.ini` 为例，参考众多资料，结合自己的理解进行注解，包括在原有 `php.ini` 基础上增加了一些实用模块的配置说明，同时对文件内容的安排进行了调整。本文针对开源模块编辑，所以删除了除 MySQL 和 PostgreSQL 以外的其他数据库模块配置选项。

```
.....
;; 关于 php.ini ;;
.....
; 这个文件必须命名为'php.ini'并放置在 httpd.conf 中的 PHPIniDir 指令指定的目录中。
; 最新版本的 php.ini 可以在下面两个位置查看：
; http://cvs.php.net/viewvc.cgi/php-src/php.ini-recommended?view=co
; http://cvs.php.net/viewvc.cgi/php-src/php.ini-dist?view=co
.....
;; 语法 ;;
.....
; 该文件的语法非常简单。空白字符和以分号开始的行被简单地忽略。
; 章节标题(例如: [php])也被简单地忽略，即使将来它们可能有某种意义。
;
; 设置指令的格式如下：
; directive = value
; 指令名(directive)是大小写敏感的！所以"foo=bar"不同于"FOO=bar"。
; 值(value)可以是：
; 1. 用引号界定的字符串(如: "foo")
; 2. 一个数字(整数或浮点数，如：0, 1, 34, -1, 33.55)
; 3. 一个 PHP 常量(如：E_ALL, M_PI)
; 4. 一个 INI 常量(On, Off, none)
; 5. 一个表达式(如：E_ALL & ~E_NOTICE)
;
; INI 文件中的表达式仅使用：位运算符、逻辑非、圆括号。
; | 位或
; & 位与
; ~ 位非
; ! 逻辑非
;
; 布尔标志用 On 表示打开，用 Off 表示关闭。
;
; 一个空字符串可以用在等号后不写任何东西表示，或者用 none 关键字：
; foo =           ; 将 foo 设为空字符串
; foo = none      ; 将 foo 设为空字符串
; foo = "none"    ; 将 foo 设为字符串'none'
;
; 如果你在指令值中使用动态扩展（PHP 扩展或 Zend 扩展）中的常量，
; 那么你只能在加载这些动态扩展的指令行之后使用这些常量。
```

```

.....
;; httpd.conf ;;
.....
; 还可以在 httpd.conf 中覆盖 php.ini 的值，以进行更灵活的配置：
; php_value name value
; 设置非 bool 型的指令，将 value 设为 none 则清除先前的设定
; php_flag name on|off
; 仅用于设置 bool 型的指令
;
; PHP 常量(如 E_ALL)仅能在 php.ini 中使用，在 httpd.conf 中必须使用相应的掩码值。
; 带"SYS"标志的指令只能在 httpd.conf 中的全局配置部分使用，
; 带"ini"标志的指令不能在 httpd.conf 中使用，它们仅能用于 php.ini 中。
;=====
;-----配置指令详解-----
;=====
; 以下每个指令的设定值都与 PHP-5.2 内建的默认值相同。
; 也就是说，如果'php.ini'不存在，或者你删掉了某些行，默认值与之相同。

[Apache]
; 仅在将 PHP 作为 Apache 模块时才有效。

engine = On
; 是否启用 PHP 解析引擎。
; 可以在 httpd.conf 中基于目录或者虚拟主机来打开或者关闭 PHP 解析引擎。

last_modified = Off
; 是否在 Last-Modified 应答头中放置该 PHP 脚本的最后修改时间。

xbithack = Off
; 是否不管文件结尾是什么，都作为 PHP 可执行位组来解析。

child_terminate = Off
; PHP 脚本在请求结束后是否允许使用 apache_child_terminate()函数终止子进程。
; 该指令仅在 UNIX 平台上将 PHP 安装为 Apache1.3 的模块时可用。其他情况下皆不存在。

.....
;; PHP 核心 ;;
.....

[PHP-Core-DateTime]
; 前四个配置选项目前仅用于 date_sunrise()和 date_sunset()函数。

date.default_latitude = 31.7667
; 默认纬度

date.default_longitude = 35.2333
; 默认经度

date.sunrise_zenith = 90.583333
; 默认日出天顶

date.sunset_zenith = 90.583333
; 默认日落天顶

```

date.timezone =

; 未设定 TZ 环境变量时用于所有日期和时间函数的默认时区。
; 中国大陆应当使用"PRC"
; 应用时区的优先顺序为：
; 1. 用 date_default_timezone_set()函数设定的时区（如果设定了的话）
; 2. TZ 环境变量(如果非空的话)
; 3. 该指令的值(如果设定了的话)
; 4. PHP 自己推测(如果操作系统支持)
; 5. 如果以上都不成功，则使用 UTC

[PHP-Core-Assert]

assert.active = On

; 是否启用 assert()断言评估

assert.bail = Off

; 是否在发生失败断言时中止脚本的执行

assert.callback =

; 发生失败断言时执行的回调函数

assert.quiet_eval = Off

; 是否使用安静评估（不显示任何错误信息，相当于 error_reporting=0）。
; 若关闭则在评估断言表达式的时候使用当前的 error_reporting 指令值。

assert.warning = On

; 是否对每个失败断言都发出警告

[PHP-Core-SafeMode]

; 安全模式是为了解决共享服务器的安全问题而设立的。
; 但试图在 PHP 层解决这个问题在结构上是不合理的，
; 正确的做法应当是修改 Web 服务器层和操作系统层。
; 因此在 PHP 6 中废除了安全模式，并打算使用 open_basedir 指令取代之。

safe_mode = Off

;SYS

; 是否启用安全模式。
; 打开时，PHP 将检查当前脚本的拥有者是否和被操作的文件的拥有者相同，
; 相同则允许操作，不同则拒绝操作。

safe_mode_gid = Off

;SYS

; 在安全模式下，默认在访问文件时会做 UID 比较检查。
; 但有些情况下严格的 UID 检查反而是不适合的，宽松的 GID 检查已经足够。
; 如果你想将其放宽到仅做 GID 比较，可以打开这个参数。

safe_mode_allowed_env_vars = "PHP_"

;SYS

; 在安全模式下，用户仅可以更改的环境变量的前缀列表(逗号分隔)。
; 允许用户设置某些环境变量，可能会导致潜在的安全漏洞。
; 注意: 如果这一参数值为空，PHP 将允许用户更改任意环境变量！

safe_mode_protected_env_vars = "LD_LIBRARY_PATH"

;SYS

；在安全模式下，用户不能更改的环境变量列表（逗号分隔）。
；这些变量即使在 `safe_mode_allowed_env_vars` 指令设置为允许的情况下也会得到保护。

safe_mode_exec_dir = "/usr/local/php/bin"

;SYS

；在安全模式下，只有该目录下的可执行程序才允许被执行系统程序的函数执行。
；这些函数是：system, escapeshellarg, escapeshellcmd, exec, passthru,
；proc_close, proc_get_status, proc_nice, proc_open, proc_terminate, shell_exec

safe_mode_include_dir =

;SYS

；在安全模式下，该组目录和其子目录下的文件被包含时，将跳过 UID/GID 检查。
；换句话说，如果此处的值为空，任何 UID/GID 不符合的文件都不允许被包含。
；这里设置的目录必须已经存在于 `include_path` 指令中或者用完整路径来包含。
；多个目录之间用冒号（Win 下为分号）隔开。
；指定的限制实际上是一个前缀，而非一个目录名，
；也就是说"/dir/incl"将允许访问"/dir/include"和"/dir/incls"
；如果您希望将访问控制在一个指定的目录，那么请在结尾加上斜线。

sqlsafe_mode = Off

;SYS

；是否使用 SQL 安全模式。
；如果打开，指定默认值的数据库连接函数将会使用这些默认值代替支持的参数。
；对于每个不同数据库的连接函数，其默认值请参考相应的手册页面。

[PHP-Core-Safe]

allow_url_fopen = On

;ini

；是否允许打开远程文件

allow_url_include = Off

;SYS

；是否允许 include/require 远程文件。

disable_classes =

;ini

；该指令接受一个用逗号分隔的类名列表，以禁用特定的类。

disable_functions =

;ini

；该指令接受一个用逗号分隔的函数名列表，以禁用特定的函数。

enable_dl = On

;SYS

；是否允许使用 dl()函数。dl()函数仅在将 PHP 作为 apache 模块安装时才有效。
；禁用 dl()函数主要是出于安全考虑，因为它可以绕过 open_basedir 指令的限制。
；在安全模式下始终禁用 dl()函数，而不管此处如何设置。

expose_php = On

;ini

；是否暴露 PHP 被安装在服务器上的事实（在 http 头中加上其签名）。
；它不会有安全上的直接威胁，但它使得客户端知道服务器上安装了 PHP。

open_basedir =

;SYS

- ；将 PHP 允许操作的所有文件（包括文件自身）都限制在此组目录列表下。
- ；当一个脚本试图打开一个指定目录树之外的文件时，将遭到拒绝。
- ；所有的符号连接都会被解析，所以不可能通过符号连接来避开此限制。
- ；特殊值 '.' 指定了存放该脚本的目录将被当做基准目录。
- ；但这有些危险，因为脚本的工作目录可以轻易被 `chdir()` 改变。
- ；对于共享服务器，在 `httpd.conf` 中灵活设置该指令将变得非常有用。
- ；在 Windows 中用分号分隔目录，UNIX 系统中用冒号分隔目录。
- ；作为 Apache 模块时，父目录中的 `open_basedir` 路径将自动被继承。
- ；指定的限制实际上是一个前缀，而非一个目录名，
- ；也就是说 `"/dir/incl"` 将允许访问 `"/dir/include"` 和 `"/dir/incls"`，
- ；如果您希望将访问控制在一个指定的目录，那么请在结尾加上一个斜线。
- ；默认是允许打开所有文件。

[PHP-Core-Error]

error_reporting = E_ALL & ~E_NOTICE

；错误报告级别是位字段的叠加，推荐使用 `E_ALL | E_STRICT`

- | | | |
|--------|----------------------------------|---|
| ； 1 | <code>E_ERROR</code> | 致命的运行时错误 |
| ； 2 | <code>E_WARNING</code> | 运行时警告（非致命性错误） |
| ； 4 | <code>E_PARSE</code> | 编译时解析错误 |
| ； 8 | <code>E_NOTICE</code> | 运行时提醒（经常是 bug ，也可能是有意的） |
| ； 16 | <code>E_CORE_ERROR</code> | PHP 启动时初始化过程中的致命错误 |
| ； 32 | <code>E_CORE_WARNING</code> | PHP 启动时初始化过程中的警告（非致命性错） |
| ； 64 | <code>E_COMPILE_ERROR</code> | 编译时致命性错 |
| ； 128 | <code>E_COMPILE_WARNING</code> | 编译时警告（非致命性错） |
| ； 256 | <code>E_USER_ERROR</code> | 用户自定义的致命错误 |
| ； 512 | <code>E_USER_WARNING</code> | 用户自定义的警告（非致命性错误） |
| ； 1024 | <code>E_USER_NOTICE</code> | 用户自定义的提醒（经常是 bug ，也可能是有意的） |
| ； 2048 | <code>E_STRICT</code> | 编码标准化警告（建议如何修改以向前兼容） |
| ； 4096 | <code>E_RECOVERABLE_ERROR</code> | 接近致命的运行时错误，若未被捕获则视同 <code>E_ERROR</code> |
| ； 6143 | <code>E_ALL</code> | 除 <code>E_STRICT</code> 外的所有错误（PHP 6 中为 8191，即包含所有） |

track_errors = Off

；是否在变量 `$php_errormsg` 中保存最近一个错误或警告消息。

display_errors = On

- ；是否将错误信息作为输出的一部分显示。
- ；在最终发布的 Web 站点上，强烈建议你关掉这个特性，并使用错误日志代替（参看下面）。
- ；在最终发布的 Web 站点打开这个特性可能暴露一些安全信息，
- ；例如你的 Web 服务上的文件路径、数据库规划或别的信息。

display_startup_errors = Off

- ；是否显示 PHP 启动时的错误。
- ；即使 `display_errors` 指令被打开，关闭此参数也将不显示 PHP 启动时的错误。
- ；建议你关掉这个特性，除非你必须要用于调试中。

report_memleaks = On

- ；是否报告内存泄漏。这个参数只在以调试方式编译的 PHP 中起作用，
- ；并且必须在 `error_reporting` 指令中包含 `E_WARNING`

report_zend_debug = On

；尚无说明文档

html_errors = On

；是否在出错信息中使用 HTML 标记。

；注意：不要在发布的站点上使用这个特性！

docref_root = ;"http://localhost/phpmanual/"

docref_ext = ;".html"

；如果打开了 html_errors 指令，PHP 将会在出错信息上显示超链接，

；直接链接到一个说明这个错误或者导致这个错误的函数的页面。

；你可以从 <http://www.php.net/docs.php> 下载 php 手册，

；并将 docref_root 指令指向你本地的手册所在的 URL 目录。

；你还必须设置 docref_ext 指令来指定文件的扩展名（必须含有'.'）。

；注意：不要在发布的站点上使用这个特性。

error_prepend_string = ;""

；用于错误信息前输出的字符串

error_append_string = ;""

；用于错误信息后输出的字符串

xmlrpc_errors = Off

xmlrpc_error_number = 0

；尚无文档

[PHP-Core-Logging]

define_syslog_variables = Off

；是否定义各种系统日志变量，如：\$LOG_PID, \$LOG_CRON 等。

；关掉它以提高效率的好主意。

；你可以在运行时调用 define_syslog_variables() 函数来定义这些变量。

error_log =

；将错误日志记录到哪个文件中。该文件必须对 Web 服务器用户可写。

；syslog 表示记录到系统日志中（NT 下的事件日志，Unix 下的 syslog(3)）

；如果此处未设置任何值，则错误将被记录到 Web 服务器的错误日志中。

log_errors = Off

；是否在日志文件里记录错误，具体在哪里记录取决于 error_log 指令。

；强烈建议你在最终发布的 Web 站点时使用日志记录错误而不是直接输出，

；这样可以让你既知道那里出了问题，又不会暴露敏感信息。

log_errors_max_len = 1024

；设置错误日志中附加的与错误信息相关联的错误源的最大长度。

；这里设置的值对显示的和记录的错误以及 \$php_errormsg 都有效。

；设为 0 可以允许无限长度。

ignore_repeated_errors = Off

；记录错误日志时是否忽略重复的错误信息。

；错误信息必须出现在同一文件的同一行才被视为重复。

ignore_repeated_source = Off

；是否在忽略重复的错误信息时忽略重复的错误源。

[PHP-Core-Mail]

- ; 要使邮件函数可用，PHP 必须在编译时能够访问 sendmail 程序。
- ; 如果使用其他的邮件程序，如 qmail 或 postfix，确保使用了相应的 sendmail 包装。
- ; PHP 首先会在系统的 PATH 环境变量中搜索 sendmail，接着按以下顺序搜索：
 - ; /usr/bin:/usr/sbin:/usr/etc:/etc:/usr/ucblib:/usr/lib
- ; 强烈建议在 PATH 中能够找到 sendmail。
- ; 另外，编译 PHP 的用户必须能够访问 sendmail 程序。

SMTP = "localhost"

- ; mail()函数中用来发送邮件的 SMTP 服务器的主机名称或者 IP 地址。仅用于 Win32。

smtp_port = 25

- ; SMTP 服务器的端口号。仅用于 Win32。

sendmail_from =

- ; 发送邮件时使用的"From:"头中的邮件地址。仅用于 Win32
- ; 该选项还同时设置了"Return-Path:"头。

sendmail_path = "-t -i"

- ;SYS
- ; 仅用于 unix，也可支持参数（默认的是'sendmail -t -i'）
- ; sendmail 程序的路径，通常为"/usr/sbin/sendmail 或/usr/lib/sendmail"。
- ; configure 脚本会尝试找到该程序并设定为默认值，但是如果失败的话，可以在这里设定。
- ; 不使用 sendmail 的系统应将该指令设定为 sendmail 替代程序(如果有的话)。
- ; 例如，Qmail 用户通常可以设为"/var/qmail/bin/sendmail"或"/var/qmail/bin/qmail-inject"。
- ; qmail-inject 不需要任何选项就能正确处理邮件。

mail.force_extra_parameters =

- ; 作为额外的参数传递给 sendmail 库的强制指定的参数附加值。
- ; 这些参数总是会替换掉 mail()的第五个参数，即使在安全模式下也是如此。

[PHP-Core-ResourceLimit]

default_socket_timeout = 60

- ; 默认 socket 超时(秒)

max_execution_time = 30

- ; 每个脚本最大允许执行时间（秒），0 表示没有限制。
- ; 这个参数有助于阻止劣质脚本无休止的占用服务器资源。
- ; 该指令仅影响脚本本身的运行时间，任何其他花费在脚本运行之外的时间，
- ; 如用 system()/sleep()函数的使用、数据库查询、文件上传等，都不包括在内。
- ; 在安全模式下，你不能用 ini_set()在运行时改变这个设置。

memory_limit = 16M

- ; 一个脚本所能够申请到的最大内存字节数（可以使用 K 和 M 作为单位）。
- ; 这有助于防止劣质脚本消耗完服务器上的所有内存。
- ; 要能够使用该指令必须在编译时使用"--enable-memory-limit"配置选项。
- ; 如果要取消内存限制，则必须将其设为 -1 。
- ; 设置了该指令后，memory_get_usage()函数将变为可用。

max_input_time = -1

；每个脚本解析输入数据（POST, GET, upload）的最大允许时间（秒）。
；-1 表示不限制。

post_max_size = 8M

；允许的 POST 数据最大字节长度。此设定也影响到文件上传。
；如果 POST 数据超出限制，那么\$_POST 和\$_FILES 将会为空。
；要上传大文件，该值必须大于 upload_max_filesize 指令的值。
；如果启用了内存限制，那么该值应当小于 memory_limit 指令的值。

realpath_cache_size = 16K

；SYS

；指定 PHP 使用的 realpath（规范化的绝对路径名）缓冲区大小。
；在 PHP 打开大量文件的系统上应当增大该值以提高性能。

realpath_cache_ttl = 120

；SYS

；realpath 缓冲区中信息的有效期（秒）。
；对文件很少变动的系统，可以增大该值以提高性能。

[PHP-Core-FileUpload]

file_uploads = On

；SYS

；是否允许 HTTP 文件上传。
；参见 upload_max_filesize, upload_tmp_dir, post_max_size 指令

upload_max_filesize = 2M

；允许上传的文件的最大尺寸。

upload_tmp_dir =

；SYS

；文件上传时存放文件的临时目录（必须是 PHP 进程用户可写的目录）。
；如果未指定则 PHP 使用系统默认的临时目录。

[PHP-Core-MagicQuotes]

；PHP6 将取消魔术引号，相当于下列指令全部为 Off

magic_quotes_gpc = On

；是否对输入的 GET/POST/Cookie 数据使用自动字符串转义（' " \ NULL）。
；这里的设置将自动影响 \$_GET \$_POST \$_COOKIE 数组的值。
；若将本指令与 magic_quotes_sybase 指令同时打开，则仅将单引号(')转义为(")，
；其他特殊字符将不被转义，即（" \ NULL）将保持原样！！
；建议关闭此特性，并使用自定义的过滤函数。

magic_quotes_runtime = Off

；是否对运行时从外部资源产生的数据使用自动字符串转义（' " \ NULL）。
；若打开本指令，则大多数函数从外部资源（数据库，文本文件等）返回数据都将被转义。
；例如：用 SQL 查询得到的数据，用 exec()函数得到的数据，等等
；若将本指令与 magic_quotes_sybase 指令同时打开，则仅将单引号(')转义为(")，
；其他特殊字符将不被转义，即（" \ NULL）将保持原样！！
；建议关闭此特性，并视具体情况使用自定义的过滤函数。

magic_quotes_sybase = Off

; 是否采用 Sybase 形式的自动字符串转义（用 " 表示 '）

[PHP-Core-HighLight]

highlight.bg = "#FFFFFF"

highlight.comment = "#FF8000"

highlight.default = "#0000BB"

highlight.html = "#000000"

highlight.keyword = "#007700"

highlight.string = "#DD0000"

; 语法高亮模式的色彩（通常用于显示 .phps 文件）。

; 只要能被接受的东西就能正常工作。

[PHP-Core-Langue]**short_open_tag = On**

; 是否允许使用"<? ?>"短标识。否则必须使用"<?php ?>"长标识。

; 除非你的 php 程序仅在受控环境下运行，且只供自己使用，否则请不要使用短标记。

; 如果要和 XML 结合使用 PHP，可以选择关闭此选项以方便直接嵌入"<?xml ... ?>"，

; 不然你必须用 PHP 来输出：<? echo '<?xml version="1.0"'; ?>

; 本指令也会影响到缩写形式"<?=", 它和"<? echo"等价，要使用它也必须打开短标记。

asp_tags = Off

; 是否允许 ASP 风格的标记"<% %>"，这也会影响到缩写形式"<%= "。

; PHP6 中将删除此指令

arg_separator.output = "&"

; PHP 所产生的 URL 中用来分隔参数的分隔符。

; 另外还可以用"&";"或","等等。

arg_separator.input = "&"

; PHP 解析 URL 中的变量时使用的分隔符列表。

; 字符串中的每一个字符都会被当做分隔符。

; 另外还可以用","&"等等。

allow_call_time_pass_reference = On

; 是否强迫在函数调用时按引用传递参数(每次使用此特性都会收到一条警告)。

; php 反对这种做法，并在将来的版本里不再支持，因为它影响到了代码的整洁。

; 鼓励的方法是在函数声明里明确指定哪些参数按引用传递。

; 我们鼓励你关闭这一选项，以保证你的脚本在将来版本的语言里仍能正常工作。

auto_globals_jit = On

; 是否仅在使用到\$_SERVER 和\$_ENV 变量时才创建（而不是在脚本一启动时就自动创建）。

; 如果并未在脚本中使用这两个数组，打开该指令将会获得性能上的提升。

; 要想该指令生效，必须关闭 register_globals 和 register_long_arrays 指令。

auto_prepend_file =**auto_append_file =**

; 指定在主文件之前/后自动解析的文件名。为空表示禁用该特性。

; 该文件就像调用了 include()函数被包含进来一样，因此会使用 include_path 指令的值。

; 注意：如果脚本通过 exit()终止，那么自动后缀将不会发生。

variables_order = "EGPCS"

; PHP 注册 Environment, GET, POST, Cookie, Server 变量的顺序。
; 分别用 E, G, P, C, S 表示, 按从左到右注册, 新值覆盖旧值。
; 举例说, 设为"GP"将会导致用 POST 变量覆盖同名的 GET 变量,
; 并完全忽略 Environment, Cookie, Server 变量。
; 推荐使用"GPC"或"GPCS", 并使用 getenv()函数访问环境变量。

register_globals = Off

; 是否将 E, G, P, C, S 变量注册为全局变量。
; 打开该指令可能会导致严重的安全问题, 除非你的脚本经过非常仔细的检查。
; 推荐使用预定义的超全局变量: \$_ENV, \$_GET, \$_POST, \$_COOKIE, \$_SERVER
; 该指令受 variables_order 指令的影响。
; PHP6 中已经删除此指令。

register_argc_argv = On

; 是否声明\$argv 和\$argc 全局变量 (包含用 GET 方法的信息)。
; 建议不要使用这两个变量, 并关掉该指令以提高性能。

register_long_arrays = On

; 是否启用旧式的长式数组 (HTTP_*_VARS)。
; 鼓励使用短式的预定义超全局数组, 并关闭该特性以获得更好的性能。
; PHP6 中已经删除此指令。

always_populate_raw_post_data = Off

; 是否总是生成\$_HTTP_RAW_POST_DATA 变量 (原始 POST 数据)。
; 否则, 此变量仅在遇到不能识别的 MIME 类型的数据时才产生。
; 不过, 访问原始 POST 数据的更好方法是 php://input 。
; \$_HTTP_RAW_POST_DATA 对于 enctype="multipart/form-data"的表单数据不可用。

unserialize_callback_func =

; 如果解序列化处理器需要实例化一个未定义类,
; 这里指定的回调函数将以该未定义类的名字作为参数被 unserialize()调用,
; 以免得到不完整的 "__PHP_Incomplete_Class"对象。
; 如果这里没有指定函数, 或指定的函数不包含 (或实现) 那个未定义类, 将会显示警告信息。
; 所以仅在确实需要实现这样的回调函数时才设置该指令。
; 若要禁止这个特性, 只需置空即可。

y2k_compliance = On

; 是否强制打开 2000 年适应 (可能在非 Y2K 适应的浏览器中导致问题)。

zend.ze1_compatibility_mode = Off

; 是否使用兼容 Zend 引擎 1(PHP 4.x)的模式。
; 这将影响对象的复制、构造 (无属性的对象会产生 FALSE 或 0)、比较。
; 兼容模式下, 对象将按值传递, 而不是默认的按引用传递。

precision = 14

; 浮点型数据显示的有效位数。

serialize_precision = 100

; 将浮点型和双精度型数据序列化存储时的精度 (有效位数)。
; 默认值能够确保浮点型数据被解序列化程序解码时不会丢失数据。

[PHP-Core-OutputControl]

- ；输出控制函数很有用，特别是在已经输出了信息之后再发送 HTTP 头的情况下。
- ；输出控制函数不会作用于 header()或 setcookie()等函数发送的 HTTP 头，
- ；而只会影响类似于 echo()函数输出的信息和嵌入在 PHP 代码之间的信息。

implicit_flush = Off

- ；是否要求 PHP 输出层在每个输出块之后自动刷新数据。
- ；这等效于在每个 print()、echo()、HTML 块 之后自动调用 flush()函数。
- ；打开这个选项对程序执行的性能有严重的影响，通常只推荐在调试时使用。
- ；在 CLI SAPI 的执行模式下，该指令默认为 On 。

output_buffering = 0

- ；输出缓冲区大小(字节)。建议值为 4096~8192。
- ；输出缓冲允许你甚至在输出正文内容之后再发送 HTTP 头(包括 cookies)。
- ；其代价是输出层减慢一点点速度。
- ；设置输出缓冲可以减少写入，有时还能减少网络数据包的发送。
- ；这个参数的实际收益很大程度上取决于你使用的是什么 Web 服务器以及什么样的脚本。

output_handler =

- ；将所有脚本的输出重定向到一个输出处理函数。
- ；比如，重定向到 mb_output_handler()函数时，字符编码将被透明地转换为指定的编码。
- ；一旦你在这里指定了输出处理程序，输出缓冲将被自动打开（output_buffering=4096）。
- ；注意 0: 此处仅能使用 PHP 内置的函数，自定义函数应在脚本中使用 ob_start()指定。
- ；注意 1: 可移植脚本不能依赖该指令，而应使用 ob_start()函数明确指定输出处理函数。
- ；使用这个指令可能会导致某些你不熟悉的脚本出错。
- ；注意 2: 你不能同时使用"mb_output_handler"和"ob_iconv_handler"两个输出处理函数。
- ；你也不能同时使用"ob_gzhandler"输出处理函数和 zlib.output_compression 指令。
- ；注意 3: 如果使用 zlib.output_handler 指令开启 zlib 输出压缩，该指令必须为空。

[PHP-Core-Directory]

doc_root =

- ；SYS
- ；PHP 的"根目录"。仅在非空时有效。
- ；如果 safe_mode=On，则此目录之外的文件一概被拒绝。
- ；如果编译 PHP 时没有指定 FORCE_REDIRECT，并且在非 IIS 服务器上以 CGI 方式运行，
- ；则必须设置此指令（参见手册中的安全部分）。
- ；替代方案是使用的 cgi.force_redirect 指令。

include_path = ".:path/to/php/pear"

- ；指定一组目录用于 require(), include(), fopen_with_path()函数寻找文件。
- ；格式和系统的 PATH 环境变量类似（UNIX 下用冒号分隔，Windows 下用分号分隔）：
- ；UNIX: "path1:path2"
- ；Windows: "\path1;\path2"
- ；在包含路径中使用'.'可以允许相对路径，它代表当前目录。

user_dir =

- ；SYS
- ；告诉 php 在使用 /~username 打开脚本时到哪个目录下去找，仅在非空时有效。
- ；也就是在用户目录之下使用 PHP 文件的基本目录名，例如: "public_html"

extension_dir = "/path/to/php"

;SYS

; 存放扩展库(模块)的目录，也就是 PHP 用来寻找动态扩展模块的目录。

; Windows 下默认为"C:/php5"

[PHP-Core-HTTP]

default_mimetype = "text/html"

default_charset = ;"gb2312"

; PHP 默认会自动输出"Content-Type: text/html" HTTP 头。

; 如果将 default_charset 指令设为"gb2312",

; 那么将会自动输出"Content-Type: text/html; charset=gb2312"。

[PHP-Core-Misc]

auto_detect_line_endings = Off

; 是否让 PHP 自动侦测行结束符 (EOL)。

; 如果你的脚本必须处理 Macintosh 文件，

; 或者你运行在 Macintosh 上，同时又要处理 unix 或 win32 文件，

; 打开这个指令可以让 PHP 自动侦测 EOL，以便 fgets()和 file()函数可以正常工作。

; 但同时也会导致在 Unix 系统下使用回车符 (CR) 作为项目分隔符的人遭遇不兼容行为。

; 另外，在检测第一行的 EOL 习惯时会有很小的性能损失。

browscap = ;"c:/windows/system32/inetsrv/browscap.ini"

;SYS

; 只有 PWS 和 IIS 需要这个设置

; 你可以从 <http://www.garykeith.com/browsers/downloads.asp>

; 得到一个 browscap.ini 文件。

ignore_user_abort = Off

; 是否即使在用户中止请求后也坚持完成整个请求。

; 在执行一个长请求的时候应当考虑打开它，

; 因为长请求可能会导致用户中途中止或浏览器超时。

user_agent = ;"PHP"

; 定义"User-Agent"字符串

;url_rewriter.tags = "a=href,area=href,frame=src,form=,fieldset="

; 虽然此指令属于 PHP 核心部分，但是却用于 Session 模块的配置

extension

; 在 PHP 启动时加载动态扩展。例如：extension=mysqli.so

; "="之后只能使用模块文件的名字，而不能含有路径信息。

; 路径信息应当只由 extension_dir 指令提供。

; 主意，在 windows 上，下列扩展已经内置：

; bcmath ; calendar ; com_dotnet ; ctype ; session ; filter ; ftp ; hash

; iconv ; json ; odbc ; pcre ; Reflection ; date ; libxml ; standard

; tokenizer ; zlib ; SimpleXML ; dom ; SPL ; wddx ; xml ; xmlreader ; xmlwriter

[PHP-Core-CGI]

; 这些指令只有在将 PHP 运行在 CGI 模式下的时候才有效

cgi.discard_path = Off

；尚无文档

cgi.fix_pathinfo = On/Off(??)

；是否为 CGI 提供真正的 PATH_INFO/PATH_TRANSLATED 支持（遵守 cgi 规范）。
；先前的行为是将 PATH_TRANSLATED 设为 SCRIPT_FILENAME，而不管 PATH_INFO 是什么。
；打开此选项将使 PHP 修正其路径以遵守 CGI 规范，否则仍将使用旧式的不合规范的行为。
；鼓励你打开此指令，并修正脚本以使用 SCRIPT_FILENAME 代替 PATH_TRANSLATED。
；有关 PATH_INFO 的更多信息请参见 cgi 规范。

cgi.force_redirect = On

；是否打开 cgi 强制重定向。强烈建议打开它以为 CGI 方式运行的 php 提供安全保护。
；你若自己关闭了它，请自己负责后果。
；注意：在 IIS/OmniHTTPD/Xitami 上则必须关闭它！

cgi.redirect_status_env =

；如果 cgi.force_redirect=On，并且在 Apache 与 Netscape 之外的服务器下运行 PHP，
；可能需要设定一个 cgi 重定向环境变量名，PHP 将去寻找它来知道是否可以继续执行下去。
；设置这个变量会导致安全漏洞，请务必在设置前搞清楚自己在做什么。

cgi.rfc2616_headers = 0

；指定 PHP 在发送 HTTP 响应代码时使用何种报头。
；0 表示发送一个 "Status: " 报头，Apache 和其他 Web 服务器都支持。
；若设为 1，则 PHP 使用 RFC2616 标准的头。
；除非你知道自己在做什么，否则保持其默认值 0

cgi.nph = Off

；在 CGI 模式下是否强制对所有请求都发送 "Status: 200" 状态码。

fastcgi.impersonate = Off

；IIS 中的 FastCGI 支持模仿客户端安全令牌的能力。
；这使得 IIS 能够定义运行时所基于的请求的安全上下文。
；Apache 中的 mod_fastcgi 不支持此特性（03/17/2002）
；如果在 IIS 中运行则设为 On，默认为 Off。

fastcgi.logging = On

；是否记录通过 FastCGI 进行的连接。

[PHP-Core-Weirdy]

；这些选项仅存在于文档中，却不存在于 phpinfo() 函数的输出中

async_send = Off

；是否异步发送。

from = ;skygao@xsphp.com"

；定义匿名 ftp 的密码（一个 email 地址）

```
.....  
;; 近核心模块 ;;  
.....
```

[Pcre]

；Perl 兼容正则表达式模块

pcre.backtrack_limit = 100000

; PCRE 的最大回溯(backtracking)步数。

pcre.recursion_limit = 100000

; PCRE 的最大递归 (recursion) 深度。

; 如果你将该值设的非常高, 将可能耗尽进程的栈空间, 导致 PHP 崩溃。

[Session]

; 除非使用 session_register()或\$_SESSION 注册了一个变量。

; 否则不管是否使用了 session_start(), 都不会自动添加任何 session 记录。

; 包括 resource 变量或有循环引用的对象包含指向自身的引用的对象, 不能保存在会话中。

; register_globals 指令会影响到会话变量的存储和恢复。

session.save_handler = "files"

; 存储和检索与会话关联的数据的处理器名字。默认为文件 ("files")。

; 如果想要使用自定义的处理器 (如基于数据库的处理器), 可用"user"。

; 有一个使用 PostgreSQL 的处理器: <http://sourceforge.net/projects/phpform-ext/>

session.save_path = "/tmp"

; 传递给存储处理器的参数。对于 files 处理器, 此值是创建会话数据文件的路径。

; Windows 下默认为临时文件夹路径。

; 你可以使用"N:[MODE;]/path"这样模式定义该路径 (N 是一个整数)。

; N 表示使用 N 层深度的子目录, 而不是将所有数据文件都保存在一个目录下。

; [MODE;]可选, 必须使用八进制数, 默认 600 (=384), 表示每个目录下最多保存的会话文件数量。

; 这是一个提高大量会话性能的好主意。

; 注意 0: "N:[MODE;]/path"两边的双引号不能省略。

; 注意 1: [MODE;]并不会改写进程的 umask。

; 注意 2: PHP 不会自动创建这些文件夹结构。请使用 ext/session 目录下的 mod_files.sh 脚本创建。

; 注意 3: 如果该文件夹可以被不安全的用户访问 (比如默认的"/tmp"), 那么将会带来安全漏洞。

; 注意 4: 当 N>0 时自动垃圾回收将会失效, 具体参见下面有关垃圾搜集的部分。

session.name = "PHPSESSID"

;用在 cookie 里的会话 ID 标识名, 只能包含字母和数字。

session.auto_start = Off

; 在客户访问任何页面时都自动初始化会话, 默认禁止。

; 因为类定义必须在会话启动之前被载入, 所以若打开这个选项, 你就不能在会话中存放对象。

session.serialize_handler = "php"

; 用来序列化/解序列化数据的处理器, php 是标准序列化/解序列化处理器。

; 另外还可以使用"php_binary"。当启用了 WDDX 支持以后, 将只能使用"wddx"。

session.gc_probability = 1

session.gc_divisor = 100

; 定义在每次初始化会话时, 启动垃圾回收程序的概率。

; 这个收集概率计算公式如下: session.gc_probability/session.gc_divisor

; 对会话页面访问越频繁, 概率就应当越小。建议值为 1/1000~5000。

session.gc_maxlifetime = 1440

; 超过此参数所指的秒数后, 保存的数据将被视为'垃圾'并由垃圾回收程序清理。

; 判断标准是最后访问数据的时间 (对于 FAT 文件系统是最后刷新数据的时间)。

；如果多个脚本共享同一个 session.save_path 目录但 session.gc_maxlifetime 不同，
；那么将以所有 session.gc_maxlifetime 指令中的最小值为准。
；如果使用多层子目录来存储数据文件，垃圾回收程序不会自动启动。
；你必须使用一个你自己编写的 shell 脚本、cron 项或者其他办法来执行垃圾搜集。
；比如，下面的脚本相当于设置了"session.gc_maxlifetime=1440" (24 分钟):
；cd /path/to/sessions; find -cmin +24 | xargs rm

session.referer_check =

；如果请求头中的"Referer"字段不包含此处指定的字符串则会话 ID 将被视为无效。
；注意：如果请求头中根本不存在"Referer"字段的话，会话 ID 将仍将被视为有效。
；默认为空，即不做检查（全部视为有效）。

session.entropy_file = ;"/dev/urandom"

；附加的用于创建会话 ID 的外部高熵值资源（文件），
；例如 UNIX 系统上的"/dev/random"或"/dev/urandom"

session.entropy_length = 0

；从高熵值资源中读取的字节数(建议值：16)。

session.use_cookies = On

；是否使用 cookie 在客户端保存会话 ID

session.use_only_cookies = Off

；是否仅仅使用 cookie 在客户端保存会话 ID
；打开这个选项可以避免使用 URL 传递会话带来的安全问题。
；但是禁用 Cookie 的客户端将使会话无法工作。

session.cookie_lifetime = 0

；传递会话 ID 的 Cookie 有效期(秒)，0 表示仅在浏览器打开期间有效。

session.cookie_path = "/"

；传递会话 ID 的 Cookie 作用路径。

session.cookie_domain =

；传递会话 ID 的 Cookie 作用域。
；默认为空表示表示根据 cookie 规范生成的主机名。

session.cookie_secure = Off

；是否仅仅通过安全连接(https)发送 cookie。

session.cookie_httponly = Off

；是否在 cookie 中添加 httpOnly 标志（仅允许 HTTP 协议访问），
；这将导致客户端脚本（JavaScript 等）无法访问该 cookie。
；打开该指令可以有效预防通过 XSS 攻击劫持会话 ID。

session.cache_limiter = "nocache"

；设为{nocache|private|public}以指定会话页面的缓存控制模式，
；或者设为空以阻止在 http 应答头中发送禁用缓存的命令。

session.cache_expire = 180

；指定会话页面在客户端 cache 中的有效期限（分钟）
；session.cache_limiter=nocache 时，此处设置无效。

session.use_trans_sid = Off

; 是否使用明码在 URL 中显示 SID（会话 ID）。
; 默认是禁止的，因为它会给你的用户带来安全风险：
; 1 用户可能将包含有效 sid 的 URL 通过 email/irc/QQ/MSN...途径告诉给其他人。
; 2 包含有效 sid 的 URL 可能会被保存在公用电脑上。
; 3 用户可能保存带有固定不变 sid 的 URL 在他们的收藏夹或者浏览历史纪录里面。
; 基于 URL 的会话管理总是比基于 Cookie 的会话管理有更多的风险，所以应当禁用。

session.bug_compat_42 = On
session.bug_compat_warn = On
; PHP4.2 之前的版本有一个未注明的"BUG":
; 即使在 register_globals=Off 的情况下也允许初始化全局 session 变量，
; 如果你在 PHP4.3 之后的版本中使用这个特性，会显示一条警告。
; 建议关闭该"BUG"并显示警告。

session.hash_function = 0
; 生成 SID 的散列算法。SHA-1 的安全性更高一些
; 0: MD5 (128 bits)
; 1: SHA-1 (160 bits)
; 建议使用 SHA-1。

session.hash_bits_per_character = 4
; 指定在 SID 字符串中的每个字符内保存多少 bit，
; 这些二进制数是 hash 函数的运算结果。
; 4: 0-9, a-f
; 5: 0-9, a-v
; 6: 0-9, a-z, A-Z, "-", ", "
; 建议值为 5

url_rewriter.tags = "a:href,area:href,frame=src,form=,fieldset="
; 此指令属于 PHP 核心部分，并不属于 Session 模块。
; 指定重写哪些 HTML 标签来包含 SID(仅当 session.use_trans_sid=On 时有效)
; form 和 fieldset 比较特殊：
; 如果你包含他们，URL 重写器将添加一个隐藏的"<input>"，它包含了本应当额外追加到 URL 上的信息。
; 如果要符合 XHTML 标准，请去掉 form 项并在表单字段前后加上<fieldset>标记。
; 注意：所有合法的项都需要一个等号（即使后面没有值）。
; 推荐值为"a:href,area:href,frame=src,input=src,form=fakeentry"。

.....
;; 其他模块 ;;
.....

[APC]
; Alternative PHP Cache 用于缓存和优化 PHP 中间代码

apc.cache_by_default = On
;SYS
; 是否默认对所有文件启用缓冲。
; 若设为 Off 并与以加号开头的 apc.filters 指令一起用，则文件仅在匹配过滤器时才被缓存。

apc.enable_cli = Off
;SYS
; 是否为 CLI 版本启用 APC 功能，仅用于测试和调试目的才打开此指令。

apc.enabled = On

；是否启用 APC，如果 APC 被静态编译进 PHP 又想禁用它，这是唯一的办法。

apc.file_update_protection = 2

；SYS

；当你在一个运行中的服务器上修改文件时，你应当执行原子操作。

；也就是先写进一个临时文件，然后将该文件重命名（mv）到最终的名字。

；文本编辑器以及 cp, tar 等程序却并不是这样操作的，从而导致有可能缓冲了残缺的文件。

；默认值 2 表示在访问文件时如果发现修改时间距离访问时间小于 2 秒则不做缓冲。

；那个不幸的访问者可能得到残缺的内容，但是这种坏影响却不会通过缓存扩大化。

；如果你能确保所有的更新操作都是原子操作，那么可以用 0 关闭此特性。

；如果你的系统由于大量的 IO 操作导致更新缓慢，你就需要增大此值。

apc.filters =

；SYS

；一个以逗号分隔的 POSIX 扩展正则表达式列表。

；如果源文件名与任意一个模式匹配，则该文件不被缓存。

；注意，用来匹配的文件名是传递给 include/require 的文件名，而不是绝对路径。

；如果正则表达式的第一个字符是 "+" 则意味着任何匹配表达式的文件会被缓存，

；如果第一个字符是 "-" 则任何匹配项都不会被缓存。 "-" 是默认值，可以省略掉。

apc.ttl = 0

；SYS

；缓存条目在缓冲区中允许逗留的秒数。0 表示永不超时。建议值为 7200~36000。

；设为 0 意味着缓冲区有可能被旧的缓存条目填满，从而导致无法缓存新条目。

apc.user_ttl = 0

；SYS

；类似于 apc.ttl，只是针对每个用户而言，建议值为 7200~36000。

；设为 0 意味着缓冲区有可能被旧的缓存条目填满，从而导致无法缓存新条目。

apc.gc_ttl = 3600

；SYS

；缓存条目在垃圾回收表中能够存在的秒数。

；此值提供了一个安全措施，即使一个服务器进程在执行缓存的源文件时崩溃，

；而且该源文件已经被修改，为旧版本分配的内存也不会被回收，直到达到此 TTL 值为止。

；设为零将禁用此特性。

apc.include_once_override = Off

；SYS

；关于该指令目前尚无说明文档，参见：<http://pecl.php.net/bugs/bug.php?id=8754>

；请保持为 Off，否则可能导致意想不到的结果。

apc.max_file_size = 1M

；SYS

；禁止大于此尺寸的文件被缓存。

apc.mmap_file_mask =

；SYS

；如果使用 --enable-mmap(默认启用)为 APC 编译了 MMAP 支持，

；这里的值就是传递给 mmap 模块的 mktemp 风格的文件掩码（建议值为 "/tmp/apc.XXXXXXX"）。

；该掩码用于决定内存映射区域是否要被 file-backed 或者 shared memory backed。

；对于直接的 file-backed 内存映射，要设置成 "/tmp/apc.XXXXXXX" 的样子(恰好 6 个 X)。

；要使用 POSIX 风格的 `shm_open/mmap` 就需要设置成 `"/apc.shm.XXXXXX"` 的样子。
；你还可以设为 `"/dev/zero"` 来为匿名映射的内存使用内核的 `"/dev/zero"` 接口。
；不定义此指令则表示强制使用匿名映射。

apc.num_files_hint = 1000

`;SYS`

；Web 服务器上可能被包含或被请求的不同源文件的大致数量（建议值为 1024~4096）。
；如果你不能确定，则设为 0；此设定主要用于拥有数千个源文件的站点。

apc.optimization = 0

；优化级别（建议值为 0）。
；正整数表示启用优化器，值越高则使用越激进的优化。
；更高的值可能有非常有限的速度提升，但目前尚在试验中。

apc.report_autofilter = Off

`;SYS`

；是否记录所有由于 early/late binding 原因而自动未被缓存的脚本。

apc.shm_segments = 1

`;SYS`

；为编译器缓冲区分配的共享内存块数量（建议值为 1）。
；如果 APC 耗尽了共享内存，并且已将 `apc.shm_size` 指令设为系统允许的最大值，
；你可以尝试增大此值。

apc.shm_size = 30

`;SYS`

；每个共享内存块的大小（以 MB 为单位，建议值为 128~256）。
；有些系统（包括大多数 BSD 变种）默认的共享内存块大小非常少。

apc.slam_defense = 0

`;SYS`（反对使用该指令，建议该用 `apc.write_lock` 指令）

；在非常繁忙的服务器上，无论是启动服务还是修改文件，
；都可能由于多个进程企图同时缓存一个文件而导致竞争条件。
；这个指令用于设置进程在处理未被缓存的文件时跳过缓存步骤的百分率。
；比如设为 75 表示在遇到未被缓存的文件时有 75% 的概率不进行缓存，从而减少碰撞几率。
；鼓励设为 0 来禁用这个特性。

apc.stat = On

`;SYS`

；是否启用脚本更新检查。
；改变这个指令值要非常小心。
；默认值 `On` 表示 APC 在每次请求脚本时都检查脚本是否被更新，
；如果被更新则自动重新编译和缓存编译后的内容。但这样做对性能有不利影响。
；如果设为 `Off` 则表示不进行检查，从而使性能得到大幅提高。
；但是为了使更新的内容生效，你必须重启 Web 服务器。
；这个指令对于 `include/require` 的文件同样有效。但是需要注意的是，
；如果你使用的是相对路径，APC 就必须在每一次 `include/require` 时都进行检查以定位文件。
；而使用绝对路径则可以跳过检查，所以鼓励你使用绝对路径进行 `include/require` 操作。

apc.user_entries_hint = 100

`;SYS`

；类似于 `num_files_hint` 指令，只是针对每个不同用户而言。
；如果你不能确定，则设为 0。

apc.write_lock = On

;SYS

; 是否启用写入锁。
; 在非常繁忙的服务器上，无论是启动服务还是修改文件，
; 都可能由于多个进程企图同时缓存一个文件而导致竞争条件。
; 启用该指令可以避免竞争条件的出现。

apc.rfc1867 = Off

;SYS

; 打开该指令后，对于每个恰好在 file 字段之前含有 APC_UPLOAD_PROGRESS 字段的上传文件，
; APC 都将自动创建一个 upload_<key>的用户缓存条目（<key>就是 APC_UPLOAD_PROGRESS 字段值）。

[bcmath]

; 为任意精度数学计算提供了二进制计算器（Binary Calculator），
; 它支持任意大小和精度的数字，以字符串形式描述。

bcmath.scale = 0

; 用于所有 bcmath 函数的十进制数的个数

[GD]

gd.jpeg_ignore_warning = Off

; 是否忽略 jpeg 解码器的警告信息（比如无法识别图片格式）。
; 有 image/jpeg 与 image/pjpeg 两种 MIME 类型，GD 库只能识别前一种传统格式。
; 参见：http://twpug.net/modules/newbb/viewtopic.php?topic_id=1867&forum=14
; <http://bugs.php.net/bug.php?id=29878>
; <http://www.faqs.org/faqs/jpeg-faq/part1/section-11.html>

[Filter]

; 对来源不可靠的数据进行确认和过滤，本扩展模块是实验性的。

filter.default = "unsafe_raw"

; 使用指定的过滤器过滤\$_GET,\$_POST,\$_COOKIE,\$_REQUEST 数据，
; 原始数据可以通过 input_get()函数访问。
; "unsafe_raw"表示不做任何过滤。

filter.default_flags =

; filter_data()函数的默认标志。

[mbstring]

;多字节字符串模块支持

mbstring.language = "neutral"

; 默认的 NLS(本地语言设置)，可设置值如下：
; 默认值"neutral"表示中立，相当于未知。
; "zh-cn"或"Simplified Chinese"表示简体中文
; "zh-tw"或"Traditional Chinese"表示繁体中文
; "uni"或"universal"表示 Unicode
; 该指令自动定义了随后的 mbstring.internal_encoding 指令默认值，

; 并且 mbstring.internal_encoding 指令必须放置在该指令之后。

mbstring.internal_encoding =

; 本指令必须放置在 mbstring.language 指令之后。

; 默认的内部编码, 未设置时取决于 mbstring.language 指令的值:

; "neutral" 对应 "ISO-8859-1"

; "zh-cn" 对应 "EUC-CN" (等价于"GB2312")

; "zh-tw" 对应 "EUC-TW" (等价于"BIG5")

; "uni" 对应 "UTF-8"

; 提醒: 对于简体中文还可以强制设置为"CP936" (等价于"GBK")

; 注意: 可能 SJIS, BIG5, GBK 不适合作为内部编码, 不过"GB2312"肯定没问题。

; 建议手动强制指定

mbstring.encoding_translation = Off

; 是否对进入的 HTTP 请求按照 mbstring.internal_encoding 指令进行透明的编码转换,

; 也就是自动检测输入字符的编码并将其透明的转化为内部编码。

; 可移植的库或者程序千万不要依赖于自动编码转换。

mbstring.http_input = "pass"

; 默认的 HTTP 输入编码, "pass"表示跳过 (不做转换)

; "auto"的含义与 mbstring.detect_order 指令中的解释一样。

; 可以设置为一个单独的值, 也可以设置为一个逗号分隔的列表。

mbstring.http_output = "pass"

; 默认的 HTTP 输出编码, "pass"表示跳过 (不做转换)

; "auto"的含义与 mbstring.detect_order 指令中的解释一样。

; 可以设置为一个单独的值, 也可以设置为一个逗号分隔的列表。

; 必须将 output_handler 指令设置为"mb_output_handler"才可以。

mbstring.detect_order =

; 默认的编码检测顺序, "pass"表示跳过 (不做转换)。

; 默认值 ("auto") 随 mbstring.language 指令的不同而变化:

; "neutral"和"universal" 对应 "ASCII, UTF-8"

; "Simplified Chinese" 对应 "ASCII, UTF-8, EUC-CN, CP936"

; "Traditional Chinese" 对应 "ASCII, UTF-8, EUC-TW, BIG-5"

; 建议在可控环境下手动强制指定一个单一值

mbstring.func_overload = 0

;SYS

; 自动使用 mb_* 函数重载相应的单字节字符串函数。

; 比如: mail(), ereg() 将被自动替换为 mb_send_mail(), mb_ereg()

; 可用 0,1,2,4 进行位组合。比如 7 表示替换所有。具体替换说明如下:

; 0: 无替换

; 1: mail() → mb_send_mail()

; 2: strlen() → mb_strlen(); substr() → mb_substr()

; strpos() → mb_strpos(); strrpos() → mb_strrpos()

; strtolower() → mb_strtolower(); strtoupper() → mb_strtoupper()

; substr_count() → mb_substr_count()

; 4: ereg() → mb_ereg(); eregi() → mb_eregi()

; ereg_replace() → mb_ereg_replace(); eregi_replace() → mb_eregi_replace()

; split() → mb_split()

mbstring.script_encoding =

；脚本所使用的编码

mbstring.strict_detection = Off

；是否使用严谨的编码检测

mbstring.substitute_character =

；当某个字符无法解码时，就是用这个字符替代。

；若设为一个整数则表示对应的 Unicode 值，不设置任何值表示不显示这个错误字符。

；建议设为"□"

mbstring.strict_encoding = Off

；默认值未知

；enable strict encoding detection.

；对于这个指令尚无说明文档（也未出现在 phpinfo()的输出中）

[Mcrypt]

；一个 mcrypt 库的接口，该库支持许多种块加密算法。

；不建议使用该模块，因为毛病太多，建议在数据库层进行加密。

mcrypt.algorithms_dir =

；默认的加密算法模块所在目录。通常是"/usr/local/lib/libmcrypt"。

；目前尚无详细说明文档，此处的解释可能是错误的。

mcrypt.modes_dir =

；默认的加密模式模块所在目录。通常是"/usr/local/lib/libmcrypt"。

；目前尚无说明文档，此处的解释可能是错误的。

[Memcache]

；一个高性能的分布式的内存对象缓存系统，通过在内存里维护一个统一的巨大的 hash 表，

；它能够用来存储各种格式的数据，包括图像、视频、文件以及数据库检索的结果等。

memcache.allow_failover = On

；是否在遇到错误时透明地向其他服务器进行故障转移。

memcache.chunk_size = 8192

；数据将按照此值设定的块大小进行转移。此值越小所需的额外网络传输越多。

；如果发现无法解释的速度降低，可以尝试将此值增加到 32768。

memcache.default_port = 11211

；连接到 memcached 服务器时使用的默认 TCP 端口。

[Zlib]

；该模块允许 PHP 透明的读取和写入 gzip(.gz)压缩文件。

zlib.output_compression = Off

；是否使用 zlib 库透明地压缩脚本输出结果。

；该指令的值可以设置为：Off、On、字节数（压缩缓冲区大小，默认为 4096）。

；如果打开该指令，当浏览器发送"Accept-Encoding: gzip(deflate)"头时，

；"Content-Encoding: gzip(deflate)"和"Vary: Accept-Encoding"头将加入到应答头当中。

；你可以在应答头输出之前用 ini_set()函数在脚本中启用或禁止这个特性。

; 如果输出一个"Content-Type: image/?"这样的应答头，压缩将不会启用（为了防止 Netscape 的 bug）。
; 你可以在输出"Content-Type: image/?"之后使用"ini_set('zlib.output_compression', 'On')重新打开这个特性。
; 注意 1: 压缩率会受压缩缓冲区大小的影响，如果你想得到更好的压缩质量，请指定一个较大的压缩缓冲区。
; 注意 2: 如果启用了 zlib 输出压缩，output_handler 指令必须为空，同时必须设置 zlib.output_handler 指令的值。

zlib.output_compression_level = -1

; 压缩级别，可用值为 0~9，0 表示不压缩。值越高效果越好，但 CPU 占用越多，建议值为 1~5。
; 默认值 -1 表示使用 zlib 内部的默认值（6）。

zlib.output_handler =

; 在打开 zlib.output_compression 指令的情况下，你只能在这里指定输出处理器。
; 可以使用的处理器有"zlib.inflate"（解压）或"zlib.deflate"（压缩）。
; 如果启用该指令则必须将 output_handler 指令设为空。

[dbx]

; 一个数据库抽象层，为不同数据库提供了统一的接口。目前支持：
; FrontBase, SQL Server, MySQL, ODBC, PostgreSQL, Sybase-CT, Oracle 8, SQLite

dbx.colnames_case = "unchanged"

; SYS
; 字段名可以按照"unchanged"或"uppercase", "lowercase"方式返回。

[MySQLi]

; MySQLi 模块只能与 4.1.3 以上版本的 MySQL 一起工作。

mysqli.max_links = -1

; SYS
; 每个进程中允许的最大连接数（持久和非持久）。-1 代表无限制

mysqli.default_port = 3306

; mysqli_connect()连接到 MySQL 数据库时使用的默认 TCP 端口。
; 如果没有在这里指定默认值，将按如下顺序寻找：
; (1)\$MYSQL_TCP_PORT 环境变量
; (2)/etc/services 文件中的 mysql-tcp 项(unix)
; (3)编译时指定的 MYSQL_PORT 常量
; 注意：Win32 下，只使用 MYSQL_PORT 常量。

mysqli.default_socket =

; mysqli_connect()连接到本机 MySQL 服务器时所使用的默认套接字名。
; 若未指定则使用内置的 MySQL 默认值。

mysqli.default_host =

; mysqli_connect()连接到 MySQL 数据库时使用的默认主机。安全模式下无效。

mysqli.default_user =

; mysqli_connect()连接到 MySQL 数据库时使用的默认用户名。安全模式下无效。

mysqli.default_pw =

; mysqli_connect()连接到 MySQL 数据库时使用的默认密码。安全模式下无效。
; 在配置文件中保存密码是个坏主意，任何使用 PHP 权限的用户都可以运行
; echo cfg_get_var("mysqli.default_password")来显示密码!
; 而且任何对该配置文件有读权限的用户也能看到密码。

mysql.reconnect = Off

;**SYS**

；是否允许重新连接

[PostgreSQL]

；PostgreSQL 模块建议与 8.0 以上版本一起工作。

pgsql.allow_persistent = On

;**SYS**

；是否允许持久连接

pgsql.max_persistent = -1

;**SYS**

；每个进程中允许的最大持久连接数。-1 代表无限制。

pgsql.max_links = -1

;**SYS**

；每个进程中允许的最大连接数（持久和非持久）。-1 代表无限制。

pgsql.auto_reset_persistent = Off

;**SYS**

；自动复位在 pg_pconnect()上中断了的持久连接，检测需要一些额外开销。

pgsql.ignore_notice = Off

；是否忽略 PostgreSQL 后端的提醒消息。

；记录后端的提醒消息需要一些很小的额外开销。

pgsql.log_notice = Off

；是否在日志中记录 PostgreSQL 后端的提醒消息。

；仅在 pgsql.ignore_notice=Off 时，才可以记录