PHP 5 Directory 函数 PHP 5 Filesystem 函数

PHP Error 和 Logging 函数

PHP Error 和 Logging 简介

Error 和 Logging 函数允许您对错误进行处理和记录。

Error 函数允许用户定义错误处理规则，并修改记录错误的方式。

Logging 函数允许用户对应用程序进行日志记录，并把日志消息发送到电子邮件、系统日志或其他的机器。

执行配置

error 函数受 php.ini 配置文件影响。

错误和日志配置选项：

参数	默认值	描述	可修改范围
error_reporting	NULL	设置 PHP 的报错级别并返回当前级别(数字或常量)。	PHP_INI_ALL
display_errors	"1"	该选项设置是否将错误信息作为输出的一部分显示到屏幕，或者对用户隐藏而不显示。 **注意：** 该特性不要在上线生产环境中使用 (在开发测试过程中使用)	PHP_INI_ALL
display_startup_errors	"0"	即使 display_errors 设置为开启, PHP 启动过程中的错误信息也不会被显示。强烈建议除了调试目的以外，将 display_startup_errors 设置为关闭。	PHP_INI_ALL
log_errors	"0"	设置是否将脚本运行的错误信息记录到服务器错误日志或者error_log之中。注意，这是与服务器相关的特定配置项。	PHP_INI_ALL
log_errors_max_len	"1024"	设置 log_errors 的最大字节数. 在 error_log 会添加有关错误源的信息。默认值为1024，如果设置为0表示不限长度。该长度设置对记录的错误，显示的错误，以及$php_errormsg都会有限制作用。	PHP_INI_ALL
ignore_repeated_errors	"0"	不记录重复的信息。重复的错误必须出现在同一个文件中的同一行代码上，除非 ignore_repeated_source 设置为true。	PHP_INI_ALL
ignore_repeated_source	"0"	忽略重复消息时，也忽略消息的来源。当该设置开启时，重复信息将不会记录它是由不同的文件还是不同的源代码	PHP_INI_ALL

		行产生的。	
report_memleaks	"1"	如果这个参数设置为Off，则内存泄露信息不会显示 (在 stdout 或者日志中)。	PHP_INI_ALL
track_errors	"0"	如果开启，最后的一个错误将永远存在于变量 $php_errormsg 中。	PHP_INI_ALL
html_errors	"1"	在错误信息中关闭HTML标签。	PHP_INI_ALL PHP_INI_SYSTEM in PHP <= 4.2.3.
xmlrpc_errors	"0"	关闭正常的错误报告，并将错误的格式设置为XML-RPC错误信息的格式。	PHP_INI_SYSTEM
xmlrpc_error_number	"0"	用作 XML-RPC faultCode 元素的值。	PHP_INI_ALL
docref_root	""	新的错误信息格式包含了对应的参考页面，该页面对错误进行具体描述，或者描述了导致该错误发生的函数。 为了提供手册的页面，你可以在PHP官方站点下载对应语言的手册，并在ini中设置网址到本地对应的地址。 如果你的本地手册拷贝可以使用"/manual/" 访问，你就可以简单的设置 docref_root=/manual/。 另外你还需要设置 docref_ext 匹配你本地文件的后缀名 docref_ext=.html。当然也可以设置一个外部的参考地址。 例如你可以设置 docref_root=http://manual/en/ 或者 docref_root="http://landonize.it/?how=url&theme=classic&filter=Landon&url=http%3A%2F%2Fwww.php.net%2F"	PHP_INI_ALL
docref_ext	""	参见 docref_root.	PHP_INI_ALL
error_prepend_string	NULL	错误信息之前输出的内容。	PHP_INI_ALL
error_append_string	NULL	错误信息之后输出的内容。	PHP_INI_ALL
error_log	NULL	设置脚本错误将被记录到的文件。该文件必须是web服务器用户可写的。	PHP_INI_ALL

安装

Error 和 Logging 函数是 PHP 核心的组成部分。无需安装即可使用这些函数。

PHP Error 和 Logging 函数

PHP：指示支持该函数的最早的 PHP 版本。

函数	描述	PHP
debug_backtrace()	生成 backtrace。	4
debug_print_backtrace()	打印 backtrace。	5
error_get_last()	获得最后发生的错误。	5
error_log()	向服务器错误记录、文件或远程目标发送一个错误。	4
error_reporting()	规定报告哪个错误。	4
restore_error_handler()	恢复之前的错误处理程序。	4
restore_exception_handler()	恢复之前的异常处理程序。	5
set_error_handler()	设置用户自定义的错误处理函数。	4
set_exception_handler()	设置用户自定义的异常处理函数。	5
trigger_error()	创建用户自定义的错误消息。	4
user_error()	trigger_error() 的别名。	4

PHP Error 和 Logging 常量

PHP：指示支持该常量的最早的 PHP 版本。

值	常量	描述	PHP
1	E_ERROR	运行时致命的错误。不能修复的错误。停止执行脚本。	
2	E_WARNING	运行时非致命的错误。没有停止执行脚本。	
4	E_PARSE	编译时的解析错误。解析错误应该只由解析器生成。	
8	E_NOTICE	运行时的通知。脚本发现可能是一个错误，但也可能在正常运行脚本时发生。	
16	E_CORE_ERROR	PHP 启动时的致命错误。这就如同 PHP 核心的 E_ERROR。	4
32	E_CORE_WARNING	PHP 启动时的非致命错误。这就如同 PHP 核心的 E_WARNING。	4
64	E_COMPILE_ERROR	编译时致命的错误。这就如同由 Zend 脚本引擎生成的	4

		E_ERROR。	
128	E_COMPILE_WARNING	编译时非致命的错误。这就如同由 Zend 脚本引擎生成的 E_WARNING。	4
256	E_USER_ERROR	用户生成的致命错误。这就如同由程序员使用 PHP 函数 trigger_error() 生成的 E_ERROR。	4
512	E_USER_WARNING	用户生成的非致命错误。这就如同由程序员使用 PHP 函数 trigger_error() 生成的 E_WARNING。	4
1024	E_USER_NOTICE	用户生成的通知。这就如同由程序员使用 PHP 函数 trigger_error() 生成的 E_NOTICE。	4
2048	E_STRICT	运行时的通知。PHP 建议您改变代码，以提高代码的互用性和兼容性。	5
4096	E_RECOVERABLE_ERROR	可捕获的致命错误。这就如同一个可以由用户定义的句柄捕获的 E_ERROR（见 set_error_handler()）。	5
6143	E_ALL	所有的错误和警告的级别，除了 E_STRICT（自 PHP 6.0 起，E_STRICT 将作为 E_ALL的一部分）。	5

← PHP 5 Directory 函数 PHP 5 Filesystem 函数 →