

日志文件优化PHP扩展函数

出于提升性能目的，基于libfastcommon封装的php扩展，提供了函数 `fastcommon_error_log` 来替代PHP原生的 `error_log`，使用 `fastcommon_file_put_contents` 替换PHP原生的 `file_put_contents`。

原理很简单，就是日志文件打开后将其文件描述符（或文件句柄）持久化，避免每次调用 `error_log` 或 `file_put_contents` 时都执行 `open` 和 `close` 等文件操作。

在短字符串的场景下，通过实测 `fastcommon_file_put_contents` 比 `file_put_contents` 性能提升2倍以上。`fastcommon_error_log` 比 `error_log` 性能提升50%以上。

两个扩展函数的用法和PHP原生函数一致。在可以优化的场景下，由fastcommon扩展接管处理，否则透传给PHP原生函数处理。函数简要说明如下：

```
bool fastcommon_error_log (string $message [, int $message_type = 0, string $destination, string $extra_headers] )
```

接管（优化处理）条件：`$message_type`为3，且指定了`$destination`（即日志文件名）在接管的情况下，`$extra_headers`可以为下列常量之一：

`FASTCOMMON_LOG_TIME_PRECISION_NONE`：日志行首不输出日期时间字符串（默认值）

`FASTCOMMON_LOG_TIME_PRECISION_SECOND`：日志行首输出的时间精度到秒

`FASTCOMMON_LOG_TIME_PRECISION_MSECOND`：日志行首输出的时间精度到毫秒

`FASTCOMMON_LOG_TIME_PRECISION_USECOND`：日志行首输出的时间精度到微秒

注：如果`$message`最后没有换行符，会自动增加。这和`error_log`的行为不一致。

```
int fastcommon_file_put_contents (string $filename , mixed $data [, int $flags = 0, resource $context ])
```

接管（优化处理）条件，需满足如下3个条件：

- * `$data`为字符串

- * `$flags` 为 `FILE_APPEND` 或 `(FILE_APPEND | LOCK_EX)`

- * `$context`为null，即没有指定`$context`