

Catching uncatchable fatal errors in PHP

Contributed by Brett Brewer
Thursday, 01 December 2011
Last Updated Thursday, 01 December 2011

Ever been annoyed when your custom error handlers or exceptions don't catch fatal PHP errors and leave you with the dreaded "white screen of death" on your web pages instead of a useful stack trace? Me too. So I was happy to find this helpful article that shows a trick you can use to catch fatal errors such as "out of memory" errors. <?php

```
set_error_handler('myErrorHandler');
register_shutdown_function('fatalErrorShutdownHandler');
function myErrorHandler($code, $message, $file, $line) {
    //handle the fatal error however you want here.
}
function fatalErrorShutdownHandler() {
    $last_error = error_get_last();
    if ($last_error['type'] === E_ERROR) {
        // fatal error
        myErrorHandler(E_ERROR, $last_error['message'], $last_error['file'], $last_error['line']);
    }
}
?>
```

You may also need to put this in your script to get fatal errors onto the error stack...<?php
error_reporting(0);
?>

Also keep in mind, if you have any other functions registered as shutdown functions and one of them throws an error, your custom fatalErrorShutdownHandler will not work.