

PHP Error

User Guide

Prepared by The Woodland Rangers

UGA MBT Capstone Project

April 28, 2018

The purpose of this document is to describe how PHP errors are handled in the Volunteer Application.

**What Happens When an Error Occurs**

**Ignored Errors**

The default error handling function has been set to a custom function called doerrorstuff(), located in the database\_connect.php. The action taken by the function depends on the type of error encountered. The following types are ignored by the error handling functions because they do not affect normal operation of the forms or the database:

|  |  |  |
| --- | --- | --- |
| **Error #** | **Error Type** | **Description** |
| **8** | E\_NOTICE | Run-time notices. The script found something that might be an error, but could also happen when running a script normally |
| **1024** | E\_USER\_NOTICE | User-generated notice. This is like E\_NOTICE, except it is generated in PHP code by using the PHP function trigger\_error() |
| **2** | E\_WARNING | Run-time warnings (non-fatal errors). Execution of the script is not halted |
| **512** | E\_USER\_WARNING | Non-fatal user-generated warning. This is like E\_WARNING, except it is generated in PHP code by using the PHP function trigger\_error() |
| **2048** | E\_STRICT | Enable to have PHP suggest changes to your code which will ensure the best interoperability and forward compatibility of your code (Since PHP 5 but not included in E\_ALL until PHP 5.4) |
| **8192** | E\_DEPRECATED | Run-time notices. Enable this to receive warnings about code that will not work in future versions (Since PHP 5.3) |
| **16384** | E\_USER\_DEPRECATED | User-generated warning message. This is like E\_DEPRECATED, except it is generated in PHP code by using the PHP function trigger\_error() (Since PHP 5.3) |
| **256** | E\_USER\_ERROR | Fatal user-generated error. This is like E\_ERROR, except it is generated in PHP code by using the PHP function trigger\_error() |

**Actions Taken by the Function**

When an error type that is NOT ignored occurs:

* The user's session ends
* The database connection is terminated
* The end user is taken to error.html
* An email containing the error number, the error string describing what went wrong, the file it occurred in, and the line it occurred on is sent to [webmaster@friendsofsmithgallwoods.org](mailto:webmaster@friendsofsmithgallwoods.org)

|  |  |  |  |
| --- | --- | --- | --- |
| **Error #** | **Error Type** | **Description** | **Action to Take** |
| **1** | E\_ERROR | Fatal run-time errors. These indicate errors that cannot be recovered from, such as a memory allocation problem. Execution of the script is halted. | Revert recent changes to the code and check that the previous version works correctly. Attempt to redo the changes in a test version of the site, making any changes suggested by the error message. Double check syntax. |
| **4** | E\_PARSE | Compile-time parse errors. Parse errors should only be generated by the parser. | This is likely caused by a syntax error in the code. Common syntax errors and tips can be found here: <http://www.dummies.com/how-to/content/troubleshooting-a-php-script.html> |
| **16** | E\_CORE\_ERROR | Fatal errors that occur during PHP's initial startup. This is like an E\_ERROR, except it is generated by the core of PHP. | Error should be forwarded to 1and1.com |
| **34** | E\_CORE\_WARNING | Warnings (non-fatal errors) that occur during PHP's initial startup. This is like an E\_WARNING, except it is generated by the core of PHP. | Error should be forwarded to 1and1.com |
| **64** | E\_COMPILE\_ERROR | Fatal compile-time errors. This is like an E\_ERROR, except it is generated by the Zend Scripting Engine. | Error should be forwarded to 1and1.com |
| **128** | E\_COMPILE\_WARNING | Compile-time warnings (non-fatal errors). This is like an E\_WARNING, except it is generated by the Zend Scripting Engine. | Error should be forwarded to 1and1.com |
| **4096** | E\_RECOVERABLE\_ERROR | Catchable fatal error. It indicates that an error that is probably dangerous occurred, but did not leave the Engine in an unstable state. | Revert recent changes to the code and check that the previous version works correctly. Attempt to redo the changes in a test version of the site, making any changes suggested by the error message. Double check syntax. |

For all errors, the database should be double-checked to ensure that bad or partial data did not end up in the database. It shouldn't since the connection is severed.

None of the errors in the table above should occur in the site as it exists on March 1, 2018. This should only be needed if there are database errors, problems with the PHP installation, or if there are later edits to the code.