

```
<?php
error_log("error message");
?>
```

Enter this code and save it in a test file. Test it in your environment. If your settings are correct, PHP will create the error log at the location specified in the `error_log` parameter. Don't create the file yourself. PHP will not log information to a log file that it did not create. The format of the message sent to your log file should be similar to the following:

```
[25-Jun-2015 17:01:12 America/New_York] error message
```

With only the simple one line of code, PHP created the text-based file in the location specified and placed the message in the file.

If you do not have access to these files, you can specify a specific location in the PHP application to send your messages. This ability also allows you to set up multiple application log files. It is common for an application to have informational log files, authentication (login) log files, error log files, and security log files. By separating each type of message, it's easier to scan for a specific type of message in a log file.

Let's assume you want to log user errors in one file and other errors in a different file.

```
<?php
const USER_ERROR_LOG = 'User_Errors.log';
const ERROR_LOG = 'Errors.log';
// sending a user error
error_log("A user error",3,USER_ERROR_LOG);
// sending all other errors
error_log("A general error",3,ERROR_LOG);
<?php
```

This code will use the **constants** (`USER_ERROR_LOG` and `ERROR_LOG`) to direct the error messages to the correct location. Notice that a second parameter of 3 is used to let the `error_log` method know that a different location will be used to log the error. A standard format should be used for sending messages to your log(s). The format should include the time/date (if not already included by the environment as mentioned previously), the type of message (if there is more than one message type in the file), the error message, and any other pertinent information. By default, the message size is limited to 120 characters. However this can be changed in the `php.ini` file.

```
$date = date('m.d.Y h:i:s');
// For more info on data time format go to: http://php.net/manual/en/function.date.php
$errormessage = "This is the error";
$message = $date . " | User Error | " . $errormessage . "\n";
error_log($message,3,USER_ERROR_LOG);
The above code would produce
06.06.2015 03:00:55 | User Error | This is the error
```

A standard text editor (Notepad++ or Notepad) or log-monitoring software (you will create a log reader program later in this chapter) can be used to view the contents of the file.