

At the top of the Example 5-7 code, the constants `USER_ERROR_LOG` and `ERROR_LOG` have been created to pinpoint the name and location of the log files. Locating constants that might be subject to change (such as a tax rate) at the top of the code provides easy access for quick changes by a programmer who is charged with supporting the application. As stated previously, the location of the log file must be in a folder that allows application write access. It is recommended that log files be centrally located in a common folder, with other log files, for easy access by data center personnel (or systems analysts).

The other code changes are located in the catch blocks. The `setException` catch block returns the error message generated by the `setException` class to the users. This message lets the users know what properties (Name, Breed, Color, and Weight) were not updated. Errors that caused this exception could have come from the user, or by corruption when the information was transmitted from the client machine to the server. These messages only provides information about the requirements of the properties, which the user already should have known. The catch block also writes a similar message to the user error log. A user error is not an urgent error that needs to be addressed by the analyst. However, tracking trends of user problems can provide an indication of possible changes needed to ensure the user has the best experience possible with the application.

The `Exception` and `Error` catch blocks captures all non-user generated exceptions. The messages caused by these exceptions might reveal information that would break the security of the application. Therefore a generic message (such as "The system is currently unavailable. Please try again later.") should be displayed to the user. Detailed information about the exception (error message, file location, coding line that raised the exception) should be placed in the error log for further analysis. Most exceptions caught by these catch blocks will keep the application from running. Therefore, it is important that personnel be informed of the problems occurring. This catch blocks are a good location to send an e-mail to the support personnel to alert them of any problems.

Now that you have built-in exception handling and error handling into the program, you could edit the `php.ini` file to turn off error reporting to the user. However, you should wait to do this until all development and testing has been completed. Locate the line `display_errors = On` in the `php.ini` file. If you change this setting to `display_errors = Off`, most error messages will not be displayed to the user. This change will not affect any messages sent back by the program to the user via the `echo` or `print` methods (including in any catch blocks). This change will give the developer greater control over the type of messages displayed to the users when there are system problems.

Do It

1. Download the code for this section. Create or use an existing HTML page that does not check for user input errors. Run the program entering values for the name, breed, weight, and color, which should cause user errors. Stop the program and open the contents of the user error log file. Did the errors appear in the file? If not, check the security of the folder that contains the log file to make sure that it allows write access to the log file. Once you are satisfied that it has caught user errors, try to cause other system errors to occur. Hint: Change the file names to nonexistent names in the dog application XML file. Check the error log to determine if the errors have been written to the file. Were you able to cause any errors that are not captured by one of the log files? If so, is there a way to capture those errors?