Code Review

Please review the following code snippet. Assume that all referenced assemblies have been properly included.

The code is used to log different messages throughout an application. We want the ability to be able to log to a text file, the console and/or the database. Messages can be marked as message, warning or error. We also want the ability to selectively be able to choose what gets logged, such as to be able to log only errors or only errors and warnings.

1) If you were to review the following code, what feedback would you give? Please be specific and indicate any errors that would occur as well as other best practices and code refactoring that should be done.
2) Rewrite the code based on the feedback you provided in question 1. Please include unit tests on your code.

```
using System;
using System.Linq;
using System.Text;

public class JobLogger
{

    private static bool _logToFile;
    private static bool _logToConsole;
    private static bool _logMessage;
    private static bool _logWarning;
    private static bool _logError;
    private static bool LogToDatabase;
    private bool _initialized;
    public JobLogger(bool logToFile, bool logToConsole, bool logToDatabase, bool
logMessage, bool logWarning, bool logError)
    {
        _logError = logError;
        _logMessage = logMessage;
        _logWarning = logWarning;
        LogToDatabase = logToDatabase;
        _logToFile = logToFile;
        _logToConsole = logToConsole;
    }

    public static void LogMessage(string message, bool message, bool warning, bool
error)
    {
        message.Trim();
        if (message == null || message.Length == 0)
        {
            return;
```

It's not necessary to be static.

Nomenclature are different. Should

It won't compile if two or more parameters has the same name.

If message is null, it will throw an exception. Validate before.

If message is null, it will throw an exception. Validate before.

```
        }
        if (!_logToConsole && !_logToFile && !LogToDatabase)
        {
            throw new Exception("Invalid configuration");
        }
        if ((!_logError && !_logMessage && !_logWarning) || (!message && !warning
&& !error))
        {
            throw new Exception("Error or Warning or Message must be specified");
        }

        System.Data.SqlClient.SqlConnection connection = new
System.Data.SqlClient.SqlConnection(System.Configuration.ConfigurationManager.AppS
ettings["ConnectionString"]);
        connection.Open();
        int t;
        if (message && _logMessage)
        {
            t = 1;
        }
        if (error && _logError)
        {
            t = 2;
        }
        if (warning && _logWarning)
        {
            t = 3;
        }
        System.Data.SqlClient.SqlCommand command = new
System.Data.SqlClient.SqlCommand("Insert into Log Values('" + message + "', " +
t.ToString() + ")");
        command.ExecuteNonQuery();

        string l;
        if
(!System.IO.File.Exists(System.Configuration.ConfigurationManager.AppSettings["Log
FileDirectory"] + "LogFile" + DateTime.Now.ToShortDateString() + ".txt"))
        {
            l =
System.IO.File.ReadAllText(System.Configuration.ConfigurationManager.AppSettings["
LogFileDirectory"] + "LogFile" + DateTime.Now.ToShortDateString() + ".txt");
        }
        if (error && _logError)
        {
            l = l + DateTime.Now.ToShortDateString() + message;
        }
        if (warning && _logWarning)
        {
            l = l + DateTime.Now.ToShortDateString() + message;
        }
        if (message && _logMessage)
        {
            l = l + DateTime.Now.ToShortDateString() + message;
        }

System.IO.File.WriteAllText(System.Configuration.ConfigurationManager.AppSettings[
"LogFileDirectory"] + "LogFile" + DateTime.Now.ToShortDateString() + ".txt", l);
```

connection is never closed. Enclose with “using” to handle this problem: using(SqlConnection connection = new SqlConnection(“…”)){}

It should validate if it’s configured to log in a database. if(_logToDatabase == true){…}. Also should open connection only if the condition is true.

Code susceptible to SQL injection attack. It’s better use stored procedure.

connection object is not passed to command constructor: command = new SqlCommand(sqlInstruction, connection)

Should validate if the file doesn’t exist then create it.

It should validate if it’s configured to log in a file. if(_logToFile == true){…}. Also should check if a file exist and read a file only if the condition is true.

```
            if (error && _logError)
            {
                Console.ForegroundColor = ConsoleColor.Red;
            }
            if (warning && _logWarning)
            {
                Console.ForegroundColor = ConsoleColor.Yellow;
            }
            if (message && _logMessage)
            {
                Console.ForegroundColor = ConsoleColor.White;
            }
            Console.WriteLine(DateTime.Now.ToShortDateString() + message);
        }
    }
```

Should validate if it's configured to log in console. if(_logToConsole == true){…}