TROUBLE SHOOTING GUIDE

diveR – By Capstone Group 11

Trouble Shooting Guide

This document outlines the errors known by the developers, how they are handled, and what to do in the event of an unhandled error.

Contents

Getting started

Sometimes errors can occur during the downloading and installation process so a great place to start when faced with program breaking errors is to simply re-download and re-install the program. To do this please refer to your version of the Installation and How To guides on how to correctly download and install the program. In the event that your issue persists please refer to the details of this document.

Reporting Errors

If the user encounters any program breaking errors, or any other inconsistencies in the program. It is recommended that the user contacts the owner of this program so that they may either contact the original developers, or a developer to fix the underlying issues.

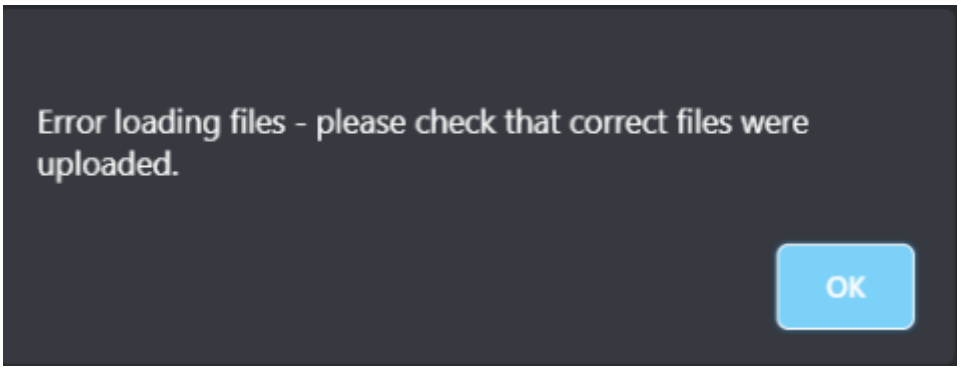
Trouble Shooting

The diveR application has been built to handle most errors a user may encounter while using the program. These range from simple input/output errors (I/O errors) to more complex Runtime errors. In the event of such errors, a dialog box informing you of what has happened will appear and notify you what to do. Please see below for what these errors mean and what causes them to appear.

UI Errors (Front-End)

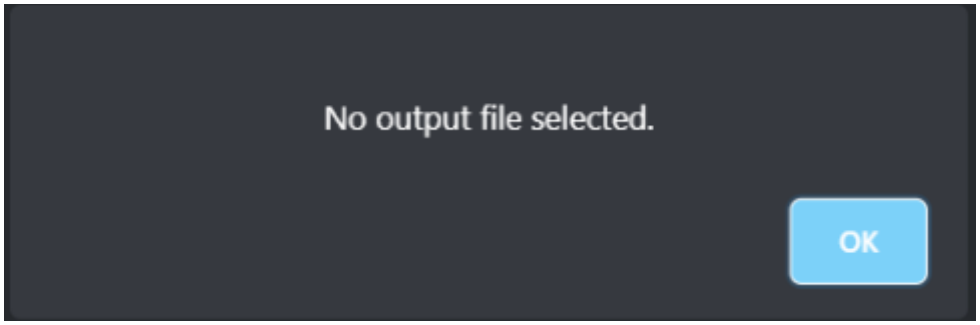
File Loading Error

This will appear when the user has selected the ‘RUN’ option before uploading any valid files.



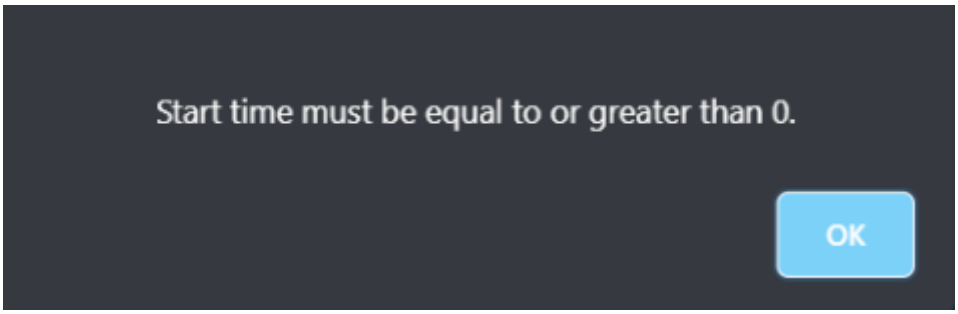
File Output Error

This will appear when the user has selected the ‘RUN’ option before designating an output file using the ‘SAVE AS’ option.



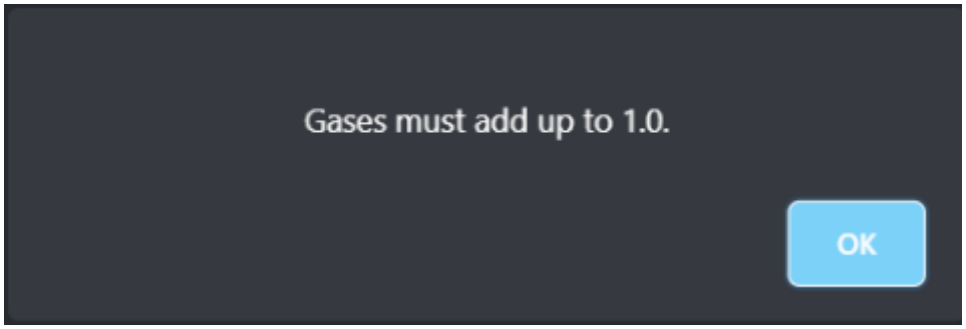
Gas Time Error

This will appear when the user has not entered a start time (in minutes) which is equal to or greater than 0.



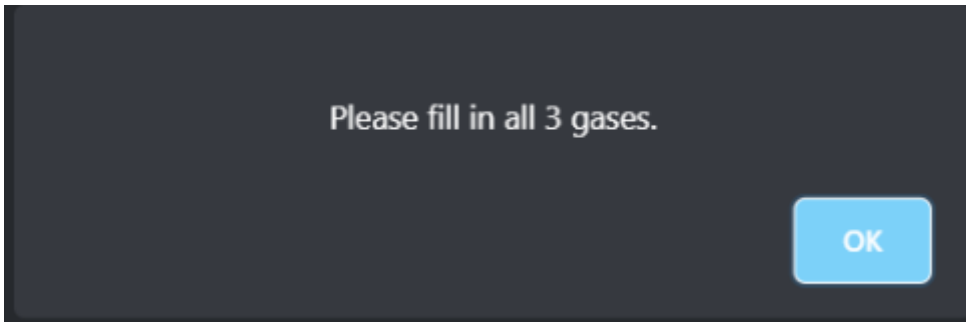
Gas Amount Error

This will appear when the user has tried to set a custom gas and the numerical values for Oxygen, Helium and Nitrogen do not equate to 1.0.



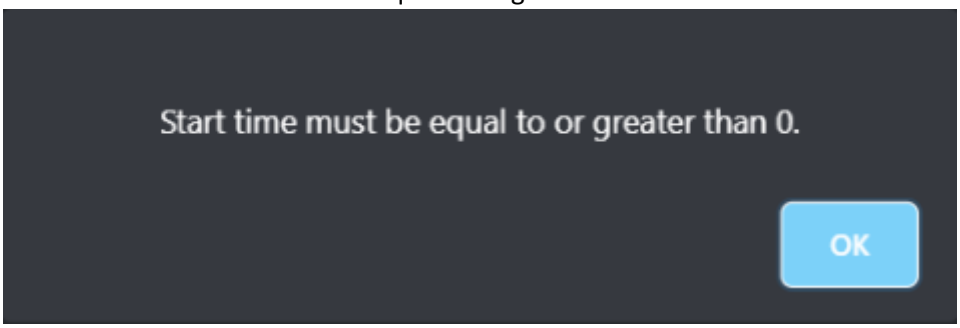
Gas Form Fill Error

This will appear when the user has tried to set a custom gas and has not filled in the entire form



Oxygen Amount Error

This will appear when the user has tried to set a custom gas but has not entered an oxygen amount which is equal to or greater than 0.



Processing Errors (Back-End)

This section will cover how to diagnose any potential processing error made by the back-end component of the program.

It should also be quickly noted that whenever processing errors occur, the program will simply filter out the files it was unable to handle correctly.

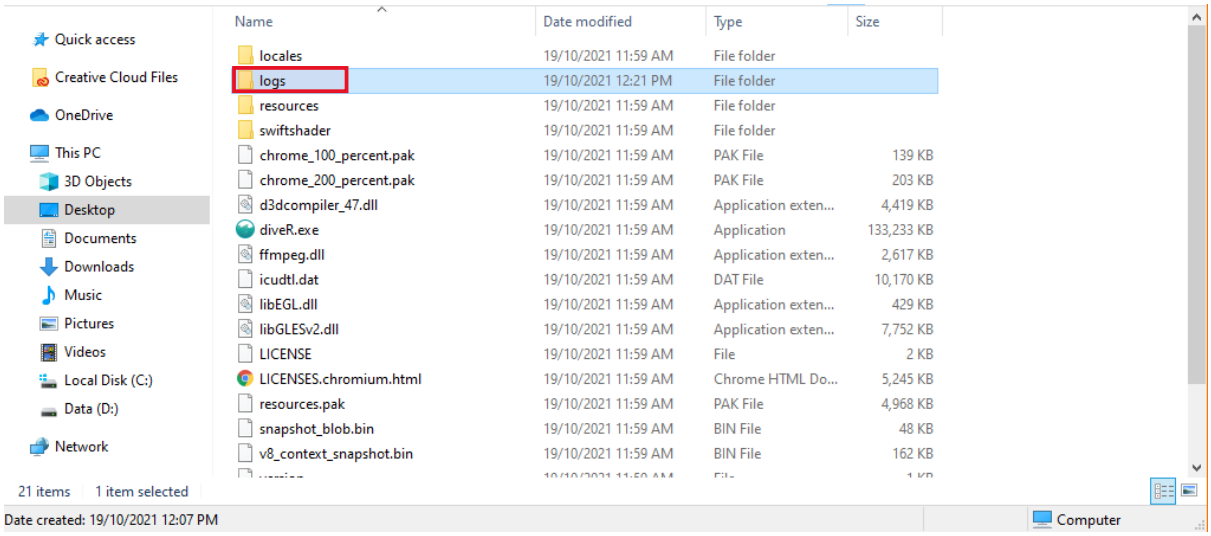
Log files

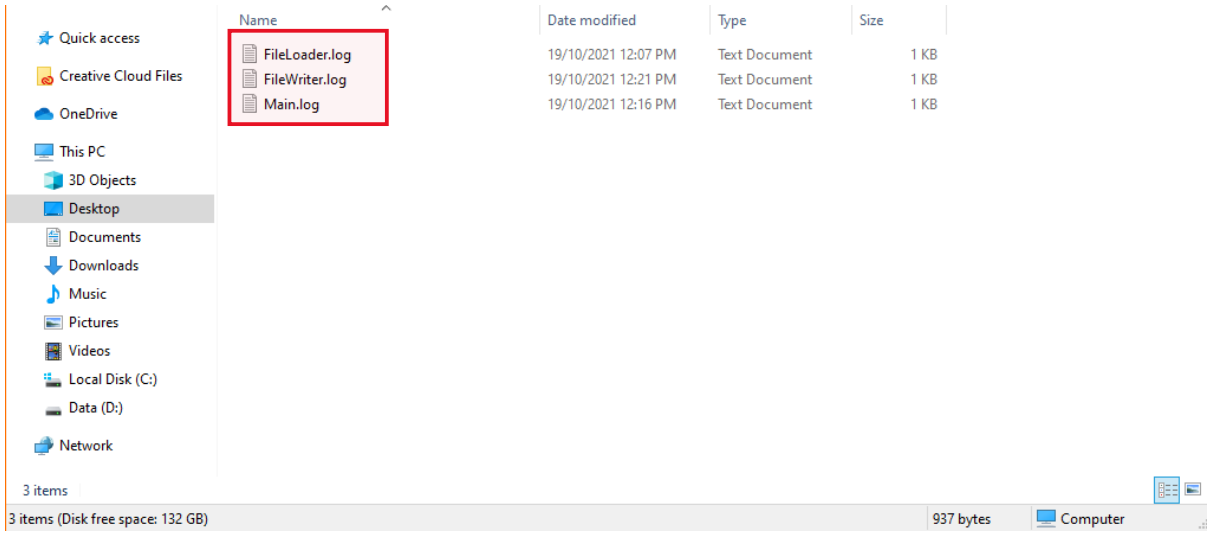
On execution of the program, known issues are written to log-files. To locate these files the user will have to navigate to the following directory: **diveR-win32-x64\logs**. (see below)

<u>If this file DOES NOT exist, it means that no known issues have been encountered.</u>

If this file DOES exist, you will potentially find 1,2 or even all 3 of the following log files.

Note: These log files will only provide a summary of errors. Without knowledge on how to edit the source code, these errors will persist for the files in which they occur. These issues should be addressed to the owner of the program so they can contact the original developers (or current developers) to solve the underlying issues.





FileLoader.log

This log file consists of errors recognised by the file loading component of the back end.

This file will be written to when the following conditions are met:

- The user has uploaded a bad file
    - .csv files are filtered by checking the extension is .csv and if the first line is Time,Depth.
    - .ans files are filtered by checking the extension is .ans or .txt and if the third line is 0.0 (dive starts at a time of 0).

The format of the file is:

<YYYY-MM-DD> <HH-MM-SS> - <filename>: <reason>. Check-firstline(s): <lines>.

The purpose of this log file is to diagnose which files were filtered at what time as well as determining the reason of the failure.

The files picked up by this logfile will likely have some error pertaining to the format of the file.

To recreate an error which writes to the log file, try uploading a known bad file such as an empty text file or a shopping list to see how it is handled by the program.

FileWriter.log

This log file consists of errors recognised by the file writing component of the back end.

This file will be written to when the following condition are met:

- There was an issue while writing the output file
    - The file was already open.
    - There was no data to write.

Note: The latter should not occur under standard consumer use.

The format of the file is:

<YYYY-MM-DD> <HH-MM-SS> - <filepath>: <reason>

The purpose of this log file is to diagnose the reason why a file may not be generated upon completion of the dive analysis made by the diveR program.

To avoid the issues picked up by this log file, do not have the output file open as the program is running.

To recreate an error which writes to the log file, try writing to an output file that is already open to see how it is handled by the program.

Main.log

This log file consists of errors recognised by the main component of the back end.

This file will be written to when the following condition are met:

- There was an RRuntime error which caused the program to not be able to process a particular dive profile
    - Likely due to errors caused when switching gas tanks.
- An index error occurred at some point in the program
    - Likely due to an error in one of the following functions:
        - nIPP
        - heIPP

The format of the file is:

<YYYY-MM-DD> <HH-MM-SS> - <filepath>: <reason>

The purpose of this log file is to diagnose the reason why a file may not have been processed correctly.

The errors picked up by this file will be caused by some underlying problem with the input file specifically to do with the functions which relate to gas tank switching.

In most cases, RRuntime errors will occur when gas tank switches are incorrect in .ans files or when input files are completely empty.

To recreate an error which writes to the log file, try processing a dive profile that contains only the headers “time, depth” to see how it is handled by the program.