# LogMining Documentation

*Bryan Ford*

## Contents

**1 Objective** **1**
1.1 Overview . . . . . 1

**2 Project Directory Structure** **2**

# 1 Objective

This project is a log mining project. It is designed to leverage multiple technologies and platforms together to allow for an extendable gateway that allows users to perform analysis on their project's logs.

## 1.1 Overview

The target objective of this project is to provide a gateway and infrastructure to perform log mining. The target of log mining is to provide answers to questions that would otherwise be difficult, time consuming, or impossible with the raw data that is available.

Log mining should provide leverage for developers, anyone performing a root cause analysis, and anyone looking to generate performance statistics against a target system. The leverage can provide answers to such questions as:

- How good are my logs?
- Am I producing too many or too few logs?
- How many alarms and warnings are being created?
- Do we have any problems that might be going unnoticed - hiccups that can cause a big issue down the road that could have been helped if addressed proactively?
- What is the root cause of a problem that has been observed by a customer?
- Can we predict a problem before it happens? What are the patterns?
- What does good behavior look like? What is the pattern in the logs of a healthy system?
- What does a healthy system look like? What are the patterns of this healthy system? Do we have deviations? If we have deviations, how frequently are they occurring?
- Do we have any correlations between patterns of logs?
- Where do we have bottlenecks and where are the opportunities for improvement?

## 2 Project Directory Structure

- **bin/**
  Binarys are kept here. These binaries are designed strictly to do processing. Any sanitizing, parsing, presenting graphical output should be done in the utilities (util/) directory. These binaries are aware of their relative location in the directory structure.

- **docs/**
  Documentation explaining anything and everything from how to set up and work in this environment to the specifications of what technology is used to explanations of each binary and the purpose of what it achieves.

- **util/**
  Utility binaries. These are scripts and other tools that perform data sanitization, parsing, and create graphs out of structured data. These tools are designed to be generic for reusability.

- **data/**
  This directory contains both the input and output files (raw form) to be processed. The input files are not to be modified by any process. In this directory is a temporary location for raw input that has been modified, but not yet ready to be disposed and will be used for further processing to generate new output. The output should be well structured and useful for any graphical output or analysis.

  - **input/**

  - **output/**
    Output files are stored here. Output files are maked *.out. There is not check if an output file already exists; overwriting and failure to write due to file locks can occur because of this. TODO: improve this 'feature'.

  - **temp/**
    Store temporary files for utility purposes. This directory should contain files only when there is an active process. Use the bash script in (/util) to purge this directory or do it manually, as needed.

- **reports/**
  Reports on data analysis. These reports outline what was expected, what processes ran (and how to recreate the raw data and output), what graphs are used, what binaries are used, and contrast what was expected with what was discovered and any further implications or curiosities.