CF Log Parser Documentation

written by Adam Presley

Table of Contents

Introduction

CF Log Parser is a Java-based tool for mass processing of ColdFusion/JRun log files. This tool collects all error information found in ColdFusion logs and will format and save them into a single, concise report, making the task of troubleshooting and error analysis a little easier.

Written in Groovy, a JVM-based language, CF Log Parser is fast and efficient and designed to be executed from the command line. This makes it suitable for ad-hoc log processing and scheduled parsing for a more proactive approach.

Note that this document is **FAR** from complete and is considered a work in progress.

Installation

To install CF Log Parser download the ZIP/TAR file and extract the contents to wherever you prefer. There are a number of JAR files that are included, but the primary file to execute is named **cflogparser.jar**.

Because this is a Java application the Java JDK must be installed on your system. Version 1.6.0_20 or higher is required to run CF Log Parser.

Cookbook

Here are a few examples of how one might use CF Log Parser to perform various tasks.

List of Log Files

This example is the most basic. This will parse two log files and save the results into an Excel file named *results.xlsx*.

```
java -jar cflogparser.jar --file-list=/jrun4/logs/web1-out.log,/jrun4/logs/web2-
out.log --output-file=/home/user/results.xlsx
```

Regex Pattern of Log Files

This example will process log files that match a specific regular expression. The results will be saved into an Excel file named *results.xlsx*.

```
java -jar cflogparser.jar --regex-basePath=/jrun4/logs/ --regex-search=web[0-9]+-
out\.log --output-file=/home/user/results.xlsx
```

Regex Pattern of Log Files With Minimum Date

This example will process log files that match a specific regular expression, and have a file modified date of **1/1/2011** or greater. The results will be saved into an Excel file named *results.xlsx*.

```
java -jar cflogparser.jar --regex-basePath=/jrun4/logs/ --regex-search=web[0-9]+-
out\.log --dmgt=2011-01-01 --output-file=/home/user/results.xlsx
```

Sorting Results by Date/Time

To sort results by date/time use the --sortdt argument and pass either **desc** or **asc**.

```
java -jar cflogparser.jar --regex-basePath=/jrun4/logs/ --regex-search=web[0-9]+-
out\.log --sortdt=desc --output-file=/home/user/results.xlsx
```

Console Output

This example will send the output to the console. That by itself is not terribly useful, but

now we can pipe the output to a text file for additional processing using tools like **grep** or Windows Powershell.

```
java -jar cflogparser.jar --file-list=/jrun4/logs/web1-out.log,/jrun4/logs/web2-out.log --output-format=console > out.txt
```

Last N Errors

To get a tail view, or last N records, use the --tail argument.

```
java -jar cflogparser.jar --file-list=/jrun4/logs/web1-out.log,/jrun4/logs/web2-out.log --tail=5

java -jar cflogparser.jar --file-list=/jrun4/logs/web1-out.log,/jrun4/logs/web2-out.log --output-format=console --tail=5 > out.txt
```

Log Types

By default CF Log Parser searches for error logs. You can tell it search for informational logs as well.

```
java -jar cflogparser.jar --file-list=/jrun4/logs/web1-out.log,/jrun4/logs/web2-out.log --log-types=info

java -jar cflogparser.jar --file-list=/jrun4/logs/web1-out.log,/jrun4/logs/web2-out.log --log-types=info,error
```