**Contest Log Analyzer - User Guide**

**Version 0.15.0-Beta**

**1. Introduction: What is the Contest Log Analyzer?**

The Contest Log Analyzer is a powerful command-line tool designed for amateur radio contesters who want to perform deep, data-driven analysis of their operating performance. It goes beyond the simple score summary provided by most logging software, allowing you to:

* **Process raw Cabrillo log files** into a clean, standardized format.
* **Automatically classify** every QSO as "Run," "Search & Pounce," or "Unknown" to analyze your operating strategy.
* **Generate detailed reports and charts** that compare your log against one or more others.
* **Analyze performance on a band-by-band basis** to identify strengths and weaknesses.
* **Calculate contest-specific QSO points** for supported contests.

The ultimate goal of this program is to help you understand your contest operation in minute detail, identify missed opportunities, and improve your strategy for the next event.

**2. What You Need to Get Started**

Before running the analyzer, you will need a few files:

* **Your Cabrillo Log File(s):** These are the standard log files generated by your contest logging software (e.g., k3lr.log, kc1xx.log). You can analyze a single log or compare multiple logs at once.
* **Country Files:** The program needs data files to determine the country and CQ Zone for each callsign.
  + **cty.dat:** This is the standard, universal country file used for most lookups.
  + **cqww.cty:** For the CQ WW contest, this specific country file is required.
  + *Note:* Both of these files must be placed in a data subdirectory within your main project folder.
* **An Environment Variable:** You must tell the program where to find your country files by setting an environment variable named CTY\_DAT\_PATH.

**3. How to Run the Analyzer**

The program is run from your command prompt using main\_cli.py. The basic command structure is:

python main\_cli.py --report <ReportID> <LogFile1> [<LogFile2>...] [options]

**Key Components:**

* **--report <ReportID>:** This is the most important part. It tells the program which analysis to run. You can use a specific ReportID (like rate\_sheet) or use all to generate every available report.
* **<LogFile1>:** You must provide the path to at least one Cabrillo log file. You can list as many as you want to compare them.

**Optional Flags:**

* **--include-dupes:** By default, all reports exclude duplicate QSOs from their calculations. Use this flag to force them to be included.
* **--mult-name <name>:** For reports that analyze multipliers (like the "Missed Multipliers" report), you must use this flag to specify which multiplier you want to analyze (e.g., --mult-name Countries or --mult-name Zones).
* **--metric <qsos|points>:** For the "Cumulative Difference Plot," this flag lets you choose whether to plot the difference in QSO counts or Point totals.

**Examples:**

* **Generate all reports for two logs:** python main\_cli.py --report all k3lr.log kc1xx.log
* **Generate just the Rate Sheet for a single log:** python main\_cli.py --report rate\_sheet k3lr.log
* **Generate the Missed Countries report for two logs:** python main\_cli.py --report missed\_multipliers --mult-name Countries k3lr.log kc1xx.log

**4. What the Program Creates**

When you run the analyzer, it creates two main types of output:

1. **Processed CSV Files:** For each Cabrillo log you provide, the program creates a new, detailed CSV file (e.g., k3lr\_processed.csv). This file is a "super log" that contains all the original QSO data plus all the new information the program has generated, including:
   * Run/S&P/Unknown classification for each QSO.
   * DXCC Country, Continent, and CQ/ITU Zone information.
   * Contest-specific QSO points.
   * Contest-specific multiplier information.
2. **Report Output Directory:** The program creates a structured directory named reports\_output to store all the generated reports. The structure is organized by year and contest name, with subdirectories for each type of report:
   * reports\_output/YYYY/ContestName/text/ (for text-based reports)
   * reports\_output/YYYY/ContestName/plots/ (for line graphs)
   * reports\_output/YYYY/ContestName/charts/ (for bar charts and Venn diagrams)

**5. Description of Available Reports**

**Text Reports**

* **QSO Summary (summary):** A high-level overview for each log, showing the total number of QSOs and the breakdown of Run, S&P, and Unknown contacts.
* **Rate Sheet (rate\_sheet):** A detailed, hour-by-hour breakdown of QSO rates for a single log, showing the number of contacts made on each band.
* **Comparative Rate Sheet (rate\_sheet\_comparison):** A side-by-side, hour-by-hour comparison of the QSO rates for two or more logs.
* **QSO Comparison Summary (qso\_comparison):** A detailed pairwise comparison of two logs, showing the breakdown of Total, Run, S&P, Unique, and Common QSOs on a per-band and overall basis.
* **Missed Multipliers (missed\_multipliers):** A powerful comparative report that shows which multipliers were worked by one station but not the other on a band-by-band basis. This report requires the --mult-name flag.

**Plots and Charts**

* **QSO Rate Plots (qso\_rate\_plots):** A set of line graphs showing the cumulative QSO total over the course of the contest. It generates one plot for all bands combined and separate plots for each individual band.
* **Point Rate Plots (point\_rate\_plots):** Similar to the QSO Rate Plot, but it shows the cumulative *point* total over time, providing insight into the quality of the contacts being made.
* **QSO Breakdown Chart (qso\_breakdown\_chart):** A comparative bar chart that visually breaks down the unique QSOs for two stations on each band by their Run, S&P, and Unknown status, with a central bar showing the number of common callsigns.
* **Cumulative Difference Plots (cumulative\_difference\_plots):** A three-part plot that shows the running difference in QSOs or Points between two logs. The main plot shows the overall difference, while two sub-plots show the difference in Run and S&P contacts, making it easy to see *how* one station pulled ahead.
* **QSO Overlap Venn Diagrams (venn\_diagrams):** A set of area-proportional Venn diagrams that visually represent the number of common and unique callsigns worked by two stations on each band. The unique areas are color-coded to show the Run/S&P breakdown.