Contest Log Analyzer - User Guide

Version: 0.90.16-Beta Date: 2025-10-06

--- Revision History ---

[0.90.16-Beta] - 2025-10-06

Added

- Added the complete list of valid `CONTEST:` tags to Section 4.

- Added a new subsection to Section 3 explaining the `<EventID>` component of the output directory structure.

[0.90.15-Beta] - 2025-10-06

Fixed

- Added the missing `CQ160mults.dat` file to the list of required data files in Section 2.

- Added the missing `--cty` argument to the list of command-line options in Section 3.

[0.88.4-Beta] - 2025-09-21

Fixed

- Synchronized the "Available Reports" list with the current project baseline to add one missing report and correct one report ID.

[0.86.8-Beta] - 2025-09-15

Changed

- Synchronized the "Available Reports" list to include the new WAE reports.

[0.85.13-Beta] - 2025-09-13

Changed

- Added WAE CW and WAE SSB to the list of supported contests.

[0.62.0-Beta] - 2025-09-08

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 and some configuration:

- **Your Cabrillo Log File(s)**: These are the standard log files generated by your contest logging software (e.g., `kd4d.log`, `n0ni.log`). You can analyze a single log or compare multiple logs at once.
- **Data Files**: The program requires specific data files to be placed in a central `data/` directory.
  - `cty.dat`: Required for all contests.
  - `arrl_10_mults.dat`: Required for the ARRL 10 Meter contest.
  - `ARRLDXmults.dat`: Required for the ARRL DX contest.
  - `CQ160mults.dat`: Required for the CQ 160-Meter contest.
  - `NAQPmults.dat`: Required for NAQP and CQ 160-Meter contests.
  - `SweepstakesSections.dat`: Required for ARRL Sweepstakes and ARRL Field Day.
  - `band_allocations.dat`: Required for all contests to perform frequency validation.
  - `iaru_officials.dat`: Required for the IARU HF World Championship contest.
- **Environment Variables**: You must tell the program where to find your input files and where to save your output reports by setting two environment variables:
  - `CONTEST_INPUT_DIR`: This variable must point to the root directory that contains your `Logs` and `data` subdirectories. This can be inside a cloud-synced folder like OneDrive.

  - `CONTEST_REPORTS_DIR`: This variable must point to the directory where the analyzer will create its `reports` output folder. **This must be a local, non-synced path** (e.g., `C:\Users\YourUser\HamRadio\CLA`) to avoid file-locking errors.

---

3. How to Run the Analyzer

The program is run from your command prompt or terminal using `main_cli.py`.

Basic Syntax

```
python main_cli.py --report <ReportID|all|chart|text|plot|animation|html> <LogFile1> [<LogFi
```

Command-Line Options

- `--report <ReportID|all|chart|text|plot|animation|html>`: (Required) Specifies which report to generate. Use a specific `ReportID` (e.g., `score_report`), `all` to generate every available report, or a category keyword like `chart` to run all chart reports.
- `<LogFile1> ...`: (Required) One or more paths to the Cabrillo log files you want to analyze. The path should be relative to the `Logs` subdirectory in your `CONTEST_INPUT_DIR`.
- `--verbose`: (Optional) Enables detailed `INFO`-level status messages for debugging.
- `--include-dupes`: (Optional) By default, duplicate QSOs are ignored. Use this flag to include them in all calculations.
- `--mult-name <name>`: (Optional) For reports that analyze multipliers (like `missed_multipliers`), this specifies which multiplier to use (e.g., 'Countries', 'Zones').
- `--metric <qsos|points>`: (Optional) For the `cumulative_difference_plots` report, this specifies whether to compare QSO counts or Point totals. Defaults to 'qsos'.
- `--debug-data`: (Optional) When used with a visual report (chart, plot, animation), this saves the report's source data to a `.txt` file in a `Debug/` subdirectory.
- `--cty <specifier>`: (Optional) Specify the CTY file: 'before', 'after' (default), or a specific filename (e.g., 'cty-3401.dat').
- `--debug-mults`: (Optional) Save intermediate multiplier lists from text reports for debugging.

Examples

- **Generate all available reports for two logs:**

```
python main_cli.py --report all 2025/cq-160-cw/kd4d.log 2025/cq-160-cw/n0ni.log
```

- **Generate only the text reports for two logs:**

```
python main_cli.py --report text 2025/cq-160-cw/kd4d.log 2025/cq-160-cw/n0ni.log
```

- **Generate a specific report (Score Summary) for a single log:**

```
python main_cli.py --report score_report 2025/cq-160-cw/kd4d.log
```

- **Generate a Missed Multipliers report for CQ WW Zones:**

```
python main_cli.py --report missed_multipliers --mult-name Zones 2024/cq-ww-cw/k3lr.log 2024
```

Understanding the Output Directory

The analyzer organizes reports into a structured directory path: `reports/<Year>/<ContestName>/<EventID>/`

- **`<EventID>`**: For contests that run multiple times a year (like NAQP), this is a short identifier to separate the events. For NAQP, this will be the three-letter month abbreviation (e.g., `JAN`, `FEB`, `AUG`). For other contests, this may be blank.

This structure ensures that reports from different events are kept separate.

4. Supported Contests

The analyzer uses the `CONTEST:` field in your Cabrillo file header to automatically apply the correct rules. Your log file must contain one of the following exact tags:

- `ARRL-10`
- `ARRL-DX-CW`
- `ARRL-DX-SSB`
- `ARRL-FIELD-DAY`
- `ARRL-SS-CW`
- `ARRL-SS-PH`
- `CQ-160-CW`
- `CQ-160-SSB`
- `CQ-WPX-CW`
- `CQ-WPX-SSB`
- `CQ-WW-CW`
- `CQ-WW-SSB`
- `IARU-HF`
- `NAQP-CW`
- `NAQP-PH`
- `NAQP-RTTY`
- `WAE-CW`
- `WAE-SSB`

5. Available Reports

Use the `Report ID` with the `--report` command-line option.

Animation Reports (`animations/`)

- `hourly_animation`: Hourly Rate Animation

HTML Reports (`html/`)

- `html_qso_comparison`: HTML QSO Comparison Report

Chart Reports (`charts/`)

- `chart_point_contribution`: Point Contribution Breakdown (Comparative)
- `chart_point_contribution_single`: Point Contribution Breakdown (Single Log)
- `qso_breakdown_chart`: QSO Breakdown by Run/S&P

Plot Reports (`plots/`)

- `band_activity_heatmap`: Band Activity Heatmap
- `comparative_band_activity`: Comparative Band Activity
- `comparative_band_activity_heatmap`: Comparative Band Activity Heatmap
- `comparative_run_sp_timeline`: Comparative Activity Timeline (Run/S&P)
- `cumulative_difference_plots`: Cumulative Difference Plot
- `point_rate_plots`: Cumulative Point Rate Plot
- `qso_rate_plots`: Cumulative QSO Rate Plot

Text Reports (`text/`)

- `comparative_continent_summary`: Comparative Continent Summary
- `comparative_score_report`: Comparative Score Report
- `continent_breakdown`: Continent Breakdown by Run/S&P
- `continent_summary`: Continent Summary
- `missed_multipliers`: Missed Multipliers
- `multiplier_summary`: Multiplier Summary
- `multipliers_by_hour`: Multipliers by Hour
- `qso_comparison`: QSO Comparison Summary
- `rate_sheet`: Rate Sheet (per hour)
- `rate_sheet_comparison`: Rate Sheet Comparison
- `score_report`: Score Report

- `summary`: QSO Summary by Run/S&P
- `text_wae_comparative_score_report`: WAE Comparative Score Report
- `text_wae_score_report`: WAE Score Summary