NoiseDoseLab Level 1 CLI

User Manual | schema_version: 2.0 | generated: 2026-06-27 (UTC)

Overview

NoiseDoseLab Level 1 CLI is a scientific command-line prototype for deterministic screening of daily occupational noise exposure from a local CSV file. It computes energy-based normalized 8-hour sound exposure per worker, compares each worker’s exposure to a user-supplied reference level, and produces a deterministic text or JSON report.

In practice, you prepare a baseline CSV file describing noise-exposure segments for workers, provide a numeric reference noise level in dB, and run a single analyze command. The tool validates the CSV structure and reference level, computes per-segment and per-worker exposure metrics using energy-domain decibel math, aggregates results into a global summary, optionally applies scenario adjustments from a separate CSV, and then renders a stable text or JSON report that you can review or feed into downstream tools.

The CLI is invoked as a Python module (`python -m app.cli`) and exposes one main subcommand, `analyze`. With no arguments or with `--help`, it prints help that describes the analyze command and the core options required to run a baseline analysis.

Getting started

Start with the discovery commands below to inspect the available CLI surface.

Use top-level help, command-group help, and action-level help to identify the documented commands, options, and actions.

End-to-end workflows are described only when the available documentary material supports them.

Starter commands

```
$ python -m app.cli
$ python -m app.cli --help
$ python -m app.cli analyze --help
$ python -m app.cli analyze --csv <baseline.csv>
    --reference-db <reference-db>
```

How to use the prototype

Discover available commands and options

Use the built-in help to understand the CLI surface before running any analysis. Top-level help shows the available subcommands, while subcommand help documents required and optional options for the analyze command.

Running the CLI with no arguments is equivalent to requesting top-level help: it prints usage information, lists the `analyze` subcommand, and explains which core options are needed to run a baseline analysis.

Commands

```
$ python -m app.cli
$ python -m app.cli --help
$ python -m app.cli analyze --help
```

Notes

- Help commands exit with return code 0.

- Top-level help exposes the `analyze` command and indicates that `--csv` and `--reference-db` are needed for a baseline run.

- Subcommand help for `analyze` documents all required and optional options, including `--csv`, `--reference-db`, `--scenario-csv`, `--format`, and `--alert-margin-db`.

Run a baseline noise exposure analysis

To perform a baseline analysis, you must provide a local CSV file describing occupational noise exposure segments and a numeric, strictly positive reference LEX,8h level in dB. The CLI validates that the CSV exists, is readable, has the required columns, and that the reference level is valid.

During analysis, the tool parses each baseline row containing a worker, task, equivalent continuous sound level in dB, duration in hours, and hearing protection attenuation in dB. It ignores invalid rows while counting them, computes energy-based exposure metrics per segment and per worker, and produces a deterministic report.

Commands

```
$ python -m app.cli analyze --csv <baseline.csv>
     --reference-db <reference-db>
$ python -m app.cli analyze --csv baseline.csv --reference-db 85
     --format text
$ python -m app.cli analyze --csv baseline.csv --reference-db 85
     --format json
```

Notes

- `--csv` is required and must point to the baseline noise exposure CSV.

- `--reference-db` is required and must be numeric and strictly positive.

- If `--format` is omitted, a default output format is used (the specific default is not documented).

- The analysis computes, for each valid segment, effective dB level, sound-energy contribution, normalized 8-hour energy, optional segment LEX,8h value, energy ratio to reference, and a deterministic status.

- Per-worker aggregates include valid-segment counts, task names, total duration, total sound energy, normalized 8-hour energy, LEX,8h value, energy ratio to reference, and status.

- A global summary includes maximum worker LEX,8h, maximum worker energy ratio, counts of workers above reference, and segments above reference.

Analyze scenarios with adjustments and JSON output

You can supply an optional scenario CSV that defines explicit dB, duration, and protection adjustments applied to the validated baseline segments. For each scenario, the tool recomputes exposure using the same energy-based methods and produces deterministic scenario summaries comparable to the baseline.

JSON output is useful when you want to integrate the results with other tools or process them programmatically. The JSON report preserves a canonical payload structure with input metadata, counts, baseline records, scenarios, and verdict reasons.

Commands

```
$ python -m app.cli analyze --csv <baseline.csv> --reference-db <reference-db>
     --scenario-csv <scenario.csv>
$ python -m app.cli analyze --csv baseline.csv --reference-db 85
     --scenario-csv scenarios.csv --format json
$ python -m app.cli analyze --csv baseline.csv --reference-db 85
     --scenario-csv scenarios.csv --format json --alert-margin-db 3
```

Notes

- `--scenario-csv` is optional and must point to a scenario adjustment CSV when provided.

- The scenario CSV is validated independently; invalid scenario data causes a typed validation error rather than partial silent use.

- `--format json` selects machine-readable JSON output; `--format text` selects human-readable text output.

- `--alert-margin-db` sets an alert margin in dB below the reference, used for alert-status comparisons in scenario exposure classifications.

- Scenario exposure summaries are computed deterministically from the baseline segments and adjustment specifications.

Command reference

Documented global options

Global options
--help
--csv
--reference-db

Group: analyze

Field	Value
Purpose	Analyze a baseline exposure CSV and produce a report.

Field	Value
Notes	Usage: `python -m app.cli analyze [-h] --csv CSV --reference-db REFERENCE_DB [--scenario-csv SCENARIO_CSV] [--format {text,json}] [--alert-margin-db ALERT_MARGIN_DB]`. • `--csv` specifies the path to the baseline noise exposure CSV. • `--reference-db` specifies the reference LEX,8h level in dB and must be numeric and strictly positive. • `--scenario-csv` specifies an optional scenario adjustment CSV path. • `--format` selects the output format: `text` (human-readable) or `json` (machine-readable). • `--alert-margin-db` specifies an alert margin in dB below the reference for alert status comparisons.

Commands

```
$ python -m app.cli analyze --help
$ python -m app.cli analyze --csv <csv> --reference-db <reference-db>
```

Inputs and outputs

The primary input is a local baseline CSV file describing occupational noise exposure segments. Each baseline row must contain a worker identifier, a task name, an equivalent continuous sound level in dB, a duration in hours, and a hearing protection attenuation value in dB. The CLI validates that this CSV exists, is readable, and includes the required columns. Invalid rows are ignored for calculations but are counted in invalid row statistics.

You must also provide a numeric, strictly positive reference LEX,8h level in dB. This reference is used to compute energy ratios and to classify per-segment and per-worker exposure status. Optionally, you can provide a scenario CSV that contains adjustment specifications (dB, duration, and protection adjustments) to be applied to the validated baseline segments.

The main outputs are deterministic reports in either human-readable text or machine-readable JSON. Reports include input metadata, row counts, per-segment metrics, per-worker aggregates, global exposure summaries, scenario summaries when applicable, and a deterministic verdict with reason codes.

Type	Value
Required input	Baseline noise exposure CSV file path for `--csv`.
Required input	Reference LEX,8h level in dB for `--reference-db`.
Produced output	Deterministic human-readable text exposure report
Produced output	Deterministic JSON exposure report
Produced output	Per-segment exposure metrics
Produced output	Per-worker exposure summaries
Produced output	Global exposure summary metrics
Produced output	Scenario exposure summaries
Produced output	Deterministic verdict with reason codes

Configuration

Configuration for this prototype is passed entirely through command-line options. There is no separate configuration file or persistent settings layer documented. You control behavior such as output format and alert margins directly via options on the analyze command.

Output format and alert margin are the primary configurable aspects beyond the required inputs. The optional scenario CSV also acts as a configuration-like input that turns on scenario analysis when provided.

Configuration principles
Use `--format {text,json}` on the `analyze` command to select text or JSON output.
Use `--alert-margin-db ALERT_MARGIN_DB` to configure the alert margin in dB below the reference level for alert status comparisons.
Use `--scenario-csv SCENARIO_CSV` to enable scenario-based exposure calculations with adjustments defined in a separate CSV.

Current limitations

This prototype focuses on deterministic analysis of local CSV data for occupational noise exposure and does not document support for other data formats or remote data sources. It is intentionally limited to a single analyze command with a fixed report structure.

Baseline and scenario analyses rely on specific required columns and energy-based calculations tailored to normalized 8-hour exposure; other exposure models or custom column layouts are not documented as supported.

Documented limitations
Only local CSV files are supported as inputs.
The CLI exposes a single main subcommand (`analyze`) for running analyses.
Baseline CSVs must contain the required columns for worker, task, sound level in dB, duration in hours, and hearing protection attenuation in dB.
Scenario adjustments are only available via an optional scenario CSV; interactive or inline scenario definitions are not documented.
The tool is designed around normalized 8-hour energy-based exposure calculations; alternative exposure models are not documented.

Project map

Application layout

```
app/
|_ cli.py
|_ orchestrator.py
|_ baseline_validator.py
|_ csv_reader.py
|_ noise_calculations.py
|_ renderers.py
|_ report_builder.py
|_ scenario_analysis.py
|_ scenario_loader.py
|_ verdict_logic.py
```