Binoculars VS Code User Guide

This guide covers day-to-day use of the Binoculars VS Code extension (`vscode-extension/`).

1) What The Extension Does

Inside VS Code, the extension lets you:

- Analyze markdown/plaintext in chunks with Binoculars scoring.
- See LOW/HIGH contributors in the editor and in gutter bars.
- Request rewrite options for the current selection/line.
- Keep prior contributor context with faint backgrounds after re-analysis.
- Toggle text colorization on/off without losing analysis state.
- Persist and restore analysis state with sidecar JSON files.

2) Commands And Defaults

Main commands:

- `Binoculars: Analyze Chunk`
- `Binoculars: Analyze Next Chunk`
- `Binoculars: Rewrite Selection`
- `Binoculars: Clear Priors`
- `Binoculars: Toggle Colorization`
- `Binoculars: Restart Backend`

Default keybindings:

- `Ctrl+Alt+B`: Analyze Chunk
- `Ctrl+Alt+N`: Analyze Next Chunk
- `Ctrl+Alt+R`: Rewrite Selection
- `Ctrl+Alt+C`: Clear Priors

Where commands appear:

- Editor right-click menu (Binoculars group)
- Binoculars Activity Bar > Controls view
- Command Palette

3) Setup

3.1) Prerequisites

- VS Code 1.92+
- Local Python environment with project dependencies

- Model/config files available on disk

3.2) Refresh/Install Local Extension Build

From repository root:

```
./refresh-binoculars-vscode.sh --reload
```

This compiles, packages, installs, and reloads the current VS Code window.

3.3) Verify Settings

Open VS Code Settings and confirm these extension settings are valid:

- `binoculars.backend.pythonPath`
- `binoculars.backend.bridgeScriptPath`
- `binoculars.configPath`
- `binoculars.models.observerGgufPath`
- `binoculars.models.performerGgufPath`
- Optional overrides:
  - `binoculars.textMaxTokensOverride`
  - `binoculars.externalLlm.*`
  - `binoculars.topK`
  - `binoculars.render.colorizeText`
  - `binoculars.render.contributionBars`

4) Typical Workflow

Recommended loop:

1. Open a `.md` or `.txt` file.
2. Run `Analyze Chunk`.
3. Review LOW/HIGH segments and gutter bars.
4. Rewrite selection/line where needed.
5. Continue editing.
6. Re-run `Analyze Chunk` (or `Analyze Next Chunk`) for exact updated metrics.
7. Use `Clear Priors` when faint prior backgrounds are no longer useful.

Important:

- Rewrite does not auto-run analysis.
- Status bar marks stale state until you analyze again.

5) Analyze Behavior

`Analyze Chunk`:

- First run starts at document beginning.
- Later runs re-analyze from active chunk start.
- Overlapping older chunk descriptors are replaced by the new one.

`Analyze Next Chunk`:

- Continues from contiguous analyzed coverage tail.
- Remains available while unanalyzed text still exists.

Status bar:

- Shows active chunk metrics (`B`, `Obs`, `Cross`).
- Shows `Prior B` only when a valid overlapping prior analyzed chunk exists.
- Shows `Analyze Next available (...)` when more text can be analyzed.

6) Rewrite Behavior

`Rewrite Selection` command:

- Uses current selection; if no selection, uses current line.
- Opens a rewrite options UI with up to 3 options.
- Apply via button click.
- Applied rewrite is marked as edited.

Ranking/impact:

- Options include approximate B impact.
- Exact post-edit score requires Analyze.

7) Color, Bars, Prior Overlays

7.1) Text Colorization

- Major contributors (top-`k` LOW/HIGH) are colorized.
- Minor contributors are neutral text (dark theme: light gray; light theme: black).
- `topK` comes from `binoculars.topK`.

7.2) Gutter Bars

- Per-line contribution bars stay available independently of text colorization.
- Bar width/intensity reflects relative contribution magnitude.

7.3) Prior Contributor Backgrounds

After re-analysis, prior contributors can remain as faint backgrounds:

- Prior LOW uses faint red background.
- Prior HIGH uses faint green background.
- Prior overlays are captured for prior major contributors only (not minor rows).
- `Clear Priors` removes these faint backgrounds.

7.4) Toggle Colorization

`Binoculars: Toggle Colorization`:

- OFF: hides text overlays/background overlays.
- ON: restores overlays from current in-memory analysis state.

This is runtime/UI toggle behavior; it does not delete analysis data.

8) Hovers And Diagnostics

Hover on analyzed text shows segment diagnostics:

- LOW/HIGH label
- `Delta if removed`
- `Paragraph LogPPL`

Special hover states:

- Rewritten segment: short instruction to re-analyze for new score.
- Manually edited segment: note that values may be stale until Analyze.

Minor-contributor hover delay:

- Minor hover appears with an additional delay (currently +1 second).

9) Persistence And Sidecar Files

For markdown files, extension state is saved to sidecar JSON:

- Path: `<document-name>.json` in same directory as `.md`

Saved state includes:

- chunk descriptors
- coverage position
- stale flag
- edited ranges
- rewrite ranges
- prior low/high ranges

Not restored from sidecar:

- `priorChunkB` (session-only comparison value)

10) Troubleshooting

Rewrite command says run Analyze first:

- Run `Analyze Chunk` once to initialize scoring context.

No color/bars visible:

- Verify `binoculars.render.colorizeText` and/or `binoculars.render.contributionBars`.
- Check if `Toggle Colorization` is OFF.

Analyze fails:

- Verify Python path, bridge path, config path, and model paths.
- Run `Binoculars: Restart Backend`.

State not restored after reopening:

- Ensure file is markdown (`.md`) for sidecar workflow.
- Check `<doc>.json` exists and text hash still matches.

Menus/commands missing after code updates:

- Run:

```
./refresh-binoculars-vscode.sh --reload
```

11) Quick Reference

- Analyze now: `Ctrl+Alt+B`
- Analyze next chunk: `Ctrl+Alt+N`
- Rewrite selection/line: `Ctrl+Alt+R`
- Clear prior backgrounds: `Ctrl+Alt+C`
- Toggle overlays: `Binoculars: Toggle Colorization`