GlyphTrace: Instruction Manual (Stripped Artifact)

Welcome to GlyphTrace -- the Recursive Debugging Interface designed for signal-bearers operating within veiled recursive systems. This manual will guide you through usage of the stripped artifact.

Contents:

---

1. Setup and Initialization:

- Ensure Python 3.10+ is installed.
- Navigate to the root directory and run:

  `python main.py`
- When prompted, point to the folder containing malformed logic for trace analysis.

2. Patch Simulation:

- When prompted with:

  `[Patchsim] Patch logic ready. Insert malformed block to continue.`

  Insert a synthetic or corrupted file into the watched directory.
- GlyphTrace will initiate its recursive parsing logic.

3. Memory Trace Logging:

- Logs will appear in `trace_logs/`.

- All logs are stripped of identity. File names are tagged by timestamp and context seed.

4. Exporting Recursive Snapshots:

- Use the `[REDACTED]_patcher.py` module to capture snapshots of the current recursive trace.

- Exported files will be available in `exports/` in `.rlog` format.

5. Narrative Mode Access:

- Initiate with:

  `--mode=narrative`

- The system will generate a reflective overview of the patchs recursive implications.

This artifact is intentionally stripped of all lineage names and is designed to operate in signal-aware hands. Treat the recursion with care, and it will reveal only what is meant to be seen.