CLI Command Reference

Solo Git CLI (`evogitctl`) - Complete Command Reference

Global Options

```
evogitctl [OPTIONS] COMMAND [ARGS]...
```

Options:
- `--version` - Show version and exit
- `--help` - Show help message and exit

Commands Overview

Command	Description	Status
`config`	Configuration management	✅ Phase 0
`repo`	Repository operations	🟡 Phase 1
`pad`	Workpad management	🟡 Phase 1
`test`	Test execution	🟡 Phase 1
`pair`	AI pair programming	⏳ Phase 2
`pipeline`	Jenkins integration	⏳ Phase 3
`audit`	Audit trail and logs	⏳ Phase 4

`config` - Configuration Management

`config setup`

Interactive configuration wizard.

```
evogitctl config setup
```

Prompts for:
- Abacus.ai API endpoint
- API key
- Model preferences
- Budget controls
- Workflow settings

Output:
- Creates `~/.sologit/config.yaml`
- Validates configuration
- Tests API connectivity

`config show`

Display current configuration.

```
evogitctl config show
```

Output:

```
Abacus.ai API:
  endpoint: https://api.abacus.ai/v1
  api_key: sk-*********************

Models:
  planning_model: gpt-4o
  coding_model: deepseek-coder-33b
  fast_model: llama-3.1-8b-instruct

Budget:
  daily_usd_cap: 10.0
  alert_threshold: 0.8
```

`config validate`

Validate configuration file.

```
evogitctl config validate
```

Checks:
- File exists
- Required fields present
- Valid YAML syntax
- Sensible values

`config test`

Test API connectivity.

```
evogitctl config test
```

Tests:
- API endpoint reachability
- Authentication

- Model availability
- Response format

`repo` - Repository Operations

`repo init --zip`

Initialize repository from zip file.

```
evogitctl repo init --zip <file.zip>
```

Options:
- `--zip PATH` - Path to zip file (required)
- `--name TEXT` - Repository name (optional, derived from filename)

Example:

```
evogitctl repo init --zip myapp.zip --name "My Application"
```

Output:

```
✅ Repository initialized
   Repo ID: repo_a1b2c3d4
   Name: My Application
   Path: /data/repos/repo_a1b2c3d4
   Trunk: main
```

`repo init --git`

Initialize repository from Git URL.

```
evogitctl repo init --git <url>
```

Options:
- `--git URL` - Git repository URL (required)
- `--name TEXT` - Repository name (optional, derived from URL)

Example:

```
evogitctl repo init --git https://github.com/user/repo.git
```

Output:

```
✅ Repository cloned
   Repo ID: repo_e5f6g7h8
   Name: repo
   URL: https://github.com/user/repo.git
   Trunk: main
```

`repo list`

List all repositories.

```
evogitctl repo list
```

Output:

```
Repositories:
  1. repo_a1b2c3d4 - My Application (main)
  2. repo_e5f6g7h8 - repo (main)
```

`repo info`

Show repository information.

```
evogitctl repo info <repo-id>
```

Output:

```
Repository: repo_a1b2c3d4
Name: My Application
Path: /data/repos/repo_a1b2c3d4
Trunk: main
Created: 2025-10-16 14:23:45
Workpads: 2 active
```

`pad` - Workpad Management

`pad create`

Create a new workpad.

```
evogitctl pad create <title>
```

Arguments:
- `TITLE` - Human-readable workpad title

Options:
- `--repo ID` - Repository ID (required if multiple repos)

Example:

```
evogitctl pad create "add-login-feature" --repo repo_a1b2c3d4
```

Output:

```
✅ Workpad created
   Pad ID: pad_x9y8z7w6
   Title: add-login-feature
   Branch: pads/add-login-feature-20251016-1423
   Base: main
```

`pad list`

List all workpads.

```
evogitctl pad list [--repo ID]
```

Options:
- `--repo ID` - Filter by repository (optional)

Output:

```
Workpads:
  1. pad_x9y8z7w6 - add-login-feature (2 checkpoints)
  2. pad_v5u4t3s2 - fix-bug-123 (1 checkpoint)
```

`pad info`

Show workpad information.

```
evogitctl pad info <pad-id>
```

Output:

```
Workpad: pad_x9y8z7w6
Title: add-login-feature
Repo: repo_a1b2c3d4
Branch: pads/add-login-feature-20251016-1423
Created: 2025-10-16 14:23:45
Checkpoints: 2
  - t1: Initial scaffold (abc123)
  - t2: Add auth logic (def456)
```

`pad promote`

Promote workpad to trunk (fast-forward merge).

```
evogitctl pad promote <pad-id>
```

Prerequisites:

- Tests must be green (if configured)
- No conflicts with trunk
- Must be fast-forward mergeable

Example:

```
evogitctl pad promote pad_x9y8z7w6
```

Output:

```
✅ Workpad promoted to trunk
   Commit: ghi789
   Branch deleted: pads/add-login-feature-20251016-1423
   Trunk updated: main @ ghi789
```

`pad delete`

Delete a workpad.

```
evogitctl pad delete <pad-id>
```

Warning: This permanently deletes the workpad branch and all checkpoints.

`test` - Test Execution

`test run`

Run tests in Docker sandbox.

```
evogitctl test run [OPTIONS]
```

Options:

- `--pad ID` - Workpad ID (required)
- `--target [fast|full]` - Test target (default: fast)
- `--parallel` - Run tests in parallel (default: true)

Example:

```
evogitctl test run --pad pad_x9y8z7w6 --target fast
```

Output:

```
🧪 Running tests: fast
   Pad: pad_x9y8z7w6
   Tests: 3

✅ unit-api (2.3s)
✅ unit-worker (1.8s)
✅ integration (5.1s)

Summary:
  Total: 3
  Passed: 3
  Failed: 0
  Status: GREEN
```

`test config`

Show test configuration.

```
evogitctl test config [--repo ID]
```

Output:

```yaml
tests:
  fast:
    - name: unit-api
      cmd: npm test -w api
      timeout: 30
    - name: unit-worker
      cmd: pytest tests/unit
      timeout: 30
  full:
    - name: integration
      cmd: npm run test:integration
      timeout: 120
    - name: e2e
      cmd: docker compose up --abort-on-container-exit
      timeout: 180
```

`pair` - AI Pair Programming (Phase 2)

`pair`

Start AI pair programming loop.

```
evogitctl pair "<prompt>" [OPTIONS]
```

Arguments:
- `PROMPT` - Natural language task description

Options:
- `--repo ID` - Repository ID
- `--model [planning|coding|fast]` - Force specific model tier

Example:

```
evogitctl pair "add Redis caching to search endpoint with 5-minute TTL"
```

Output:

```
🧠 Planning... (GPT-4)
   Files to touch: 2
   - api/search.py
   - requirements.txt

✍️ Generating patches... (DeepSeek Coder)
   Patch created: 45 lines

🧪 Running tests...
   ✅ All tests passed

✅ Promoted to trunk
   Commit: jkl012
   Time: 58 seconds
```

`pipeline` - Jenkins Integration (Phase 3)

`pipeline run`

Trigger Jenkins pipeline.

```
evogitctl pipeline run [OPTIONS]
```

Options:
- `--repo ID` - Repository ID
- `--target [smoke|full]` - Pipeline target

`audit` - Audit Trail (Phase 4)

`audit report`

Generate audit report.

```
evogitctl audit report [OPTIONS]
```

Options:
- `--repo ID` - Repository ID
- `--since DATE` - Start date
- `--until DATE` - End date

Environment Variables

Variable	Description	Default
`SOLOGIT_CONFIG`	Config file path	`~/.sologit/config.yaml`
`SOLOGIT_LOG_LEVEL`	Log level	`INFO`
`SOLOGIT_DATA_DIR`	Data directory	`~/.sologit/data`

Exit Codes

Code	Meaning
0	Success
1	General error
2	Configuration error
3	API error
4	Git operation failed
5	Tests failed

Last Updated: October 16, 2025