

cxcodex Manual (Operator Playbook)

Last updated: 2026-02-22

Purpose

This playbook is a set of reproducible procedures. It is intentionally low-prose and high-signal.

1 P0: Confirm Runtime Path

```
./bin/cx version  
./bin/cx where version
```

2 P1: Confirm Schemas Exist

```
./bin/cx schema list  
ls .codex/schemas
```

3 P2: Validate Structured Output (Schema Commands)

Commit JSON

```
./bin/cx commitjson | jq .
```

Diff Summary

```
./bin/cx diffsum-staged | jq .
```

Next Commands

```
./bin/cx next git status | jq .
```

4 P3: Budget Stress Test

```
CX_CONTEXT_BUDGET_CHARS=2000 CX_CONTEXT_BUDGET_LINES=40 ./bin/cx cxo git
status
./bin/cx budget
./bin/cx trace
```

5 P4: Quarantine and Replay (Failure Procedure)

```
./bin/cx quarantine list
./bin/cx quarantine show <id>
./bin/cx replay <id>
```

6 P5: Safety Gate (Before Any Execution)

```
./bin/cx policy show
./bin/cx policy check "<candidate command>"
```

7 P6: Task Graph (Orchestration)

Create and Execute

```
./bin/cx task add "Objective: harden capture clipping" --role implementer
./bin/cx task list --status pending
./bin/cx task run-all --status pending
```

Fan-out

```
./bin/cx task fanout "Objective: refactor logging" --from staged-diff
./bin/cx task list --status pending
```

8 P7: Telemetry Contract Validation

```
./bin/cx logs validate --fix=false
tail -n 1 .codex/cxlogs/runs.jsonl | jq .
```

9 P8: Self-Optimization

```
./bin/cx optimize 200
./bin/cx optimize 200 --json | jq .
```