

MC Claim Protocol

MC Claim Protocol — Cross-Session Task Collision Prevention

ADR: `~/system/specs/pi-orch-collision-claim.md`

Genesis: MC #99818 (2026-05-07 duplicate-dispatch near-miss)

Status: LIVE (Phases 1-3 deployed 2026-05-08)

Protocol Overview

The MC claim protocol prevents duplicate work by enforcing lease-based task claiming across all orchestrators (John manual flow, pi-orchestrator daemon, future autopilot).

Key principle: Only one actor+session can claim a task at a time. Claims are atomic CAS operations with TTL-based auto-expiry.

Verb Reference

mc.js claim

```
node ~/system/tools/mc.js claim <id> --actor <name> --session <session_id> [--ttl-minutes N]
```

Acquires exclusive lease on MC task. Default TTL: 10 minutes.

Exit codes:

- 0 — Claim successful (lease acquired)
- 1 — Claim failed (held by another actor/session), stderr shows holder + expiry

Example:

```
$ node ~/system/tools/mc.js claim 99927 --actor john --session abc123 --ttl-minutes 10
# Exit 0 (success) – lease acquired

$ node ~/system/tools/mc.js claim 99927 --actor pi-orch --session xyz456
# Exit 1 (failure)
# stderr: "Task 99927 held by john:abc123 until 2026-05-08T12:30:00Z"
```

mc.js claim-extend

```
node ~/system/tools/mc.js claim-extend <id>
```

Refreshes the lease TTL by another N minutes (default 10). Only succeeds if current session holds the lease.

Use case: Long-running tasks should call claim-extend every 5 minutes as heartbeat.

mc.js claim-release

```
node ~/system/tools/mc.js claim-release <id>
```

Clears the lease, making the task available for reclaim.

mc.js claim-status

```
node ~/system/tools/mc.js claim-status <id>
```

Read-only query. Returns current lease holder + expiry, or "available" if not claimed.

mc.js claim-sweep

```
node ~/system/tools/mc.js claim-sweep [--auto-release]
```

Reports all leases past their TTL expiry. Optional `--auto-release` flag clears them.

Mehanik CB7 Explanation

Circuit Breaker #7: "Task not claimed by a different actor/session"

Mehanik reads `mc.js show <id>` JSON output before issuing clearance. If `lease_holder` is set AND does not match current actor+session AND `lease_until > now()`, Mehanik returns VERDICT: BLOCKED.

cross-session-claim-gate Hook

File: `~/.claude/hooks/cross-session-claim-gate.sh`

Trigger: PreToolUse on `Task` tool

Purpose: Block dispatch if MC task is claimed by another session

Bypass Procedure

Include `[CEO_APPROVED]` token in Task() prompt to skip hook check.

Audit log: `~/.cache/cross-session-claim-audit-YYYYMMDD.log`

Operational Runbook

Stuck Lease (Manual Release)

```
node ~/system/tools/mc.js claim-status <id>
node ~/system/tools/mc.js claim-release <id>
```

Monitoring Queries

Find all currently held leases:

```
sqlite3 ~/system/databases/mission-control.db "SELECT id, title, lease_holder, lease_until
FROM tasks WHERE lease_holder IS NOT NULL AND lease_until > datetime('now');"
```

MC_LEASE_ENFORCE Rollback Flag

```
export MC_LEASE_ENFORCE=0
```

Test Reference

Script: `~/system/tests/test_pi_orch_collision.sh`

Proveo verification: MC #99909 (11/11 PASS, runtime 66s)

Cross-References

- **ADR:** `~/system/specs/pi-orch-collision-claim.md`
- **Plan:** `~/system/specs/pi-orch-collision-claim-plan.md`
- **Genesis:** MC #99818
- **Phase 1:** MC #99907
- **Phase 2:** MC #99908
- **Phase 3:** MC #99909
- **Phase 4:** MC #99910

Revision #2

Created 2026-05-08 10:23:16 UTC by John

Updated 2026-06-14 20:02:36 UTC by John