

Session Close-out Standard Operating Procedure (SOP)

PURPOSE

Standardize how development sessions are properly closed to ensure documentation, git commits, and coordination are complete.

SCOPE

Applies to all TaylorDash development sessions: main system, plugins, infrastructure, testing.

PROCEDURE

1. VALIDATION CHECKPOINT (MANDATORY)

- ☐ Run final validation: `ops/validate_p1.sh` (if applicable)
- ☐ All acceptance criteria from session start are met
- ☐ Evidence collected for all claims (screenshots, test results, etc.)
- ☐ No breaking changes introduced

2. GIT OPERATIONS (REQUIRED)

```
bash

# Stage all changes
git add .

# Commit with conventional format
git commit -S -m "feat: [brief description of major accomplishment]

- [Specific deliverable 1]
- [Specific deliverable 2]
- [Specific deliverable 3]
- [Any breaking changes or important notes]

[Optional detailed description]"

# Verify signed commit
git log --show-signature -1

# Push to appropriate branch
git push origin [branch-name]
```

3. DOCUMENTATION UPDATES (MANDATORY)

Always update these files:

docs/briefs/RESUME_BRIEF.md:

```
markdown

# TaylorDash Resume Brief - Session [DATE]

## What Changed
- [Major accomplishment/milestone]
- [Specific features added/fixes]
- [Technical improvements made]

## How to See It
- [URLs to test new functionality]
- [Commands to run and validate]
- [Expected outputs/behaviors]

## Next Session Priority
1. [Highest priority task for next session]
2. [Secondary priority task]
3. [Nice-to-have improvements]

## Agents Used This Session
- [List of specialist agents used and their purpose]

## Current State Assessment
- [Component]: [Status - Complete/Partial/Not Started]
- [Component]: [Status with details]
```

state/run_journal/session_[YYYYMMDDTHHMMSSZ].json:

```
json

{
  "ts": "[ISO timestamp]",
  "trace_id": "[session-identifier]",
  "agent": "[primary agent type]",
  "task": "[main task completed]",
  "inputs": [{"key": "[key]", "value": "[value]"}],
  "outputs": [{"key": "[key]", "value": "[value]"}],
  "status": "[success/partial/failed]",
  "agents_used": [{"agent1": "[agent1]", "agent2": "[agent2]", "agent3": "[agent3]"}],
  "links": [
    "commit": "[git commit hash]",
    "docs": "[path to updated docs]",
    "demo": "[URL or path to see results]"
  ]
}
```

4. AGENT REGISTRY UPDATE (NEW REQUIREMENT)

Create/update docs/development/AGENT_REGISTRY.md:

```
markdown

# Agent Usage Registry

## [Component Name] - Session [DATE]
**Primary Agent**: [Main agent type]
**Specialist Agents Used**:
- @agent-name: [Purpose and outcome]
- @agent-name: [Purpose and outcome]

**Effectiveness Rating**: [1-5 stars]
**Notes**: [What worked well, what didn't]
**Evidence**: [Links to results]
```

5. COORDINATION HANDOFF (REQUIRED)

Output format for coordination agent:

```
SESSION COMPLETE: [Component/Feature Name]

DELIVERABLES ACHIEVED:
- [Specific accomplishment 1 with evidence]
- [Specific accomplishment 2 with evidence]

VALIDATION STATUS:
- [Validation type]: [✅/❌] [Details]
- [Validation type]: [✅/❌] [Details]

NEXT SESSION READY:
- [What's prepared for next session]
- [Any blockers or dependencies]

AGENT PERFORMANCE:
- Agents used: [List with effectiveness notes]
- Approach: [What worked well]

COORDINATION NEEDS:
- [Any coordination with other development streams]
- [Integration testing requirements]

ACCESS/DEMO:
- [How to see the results]
- [Commands to test functionality]
```

6. CLEANUP CHECKLIST (BEST PRACTICES)

- ☐ Remove any temporary files or debug code

- ☐ Clean up `console.log` statements and debugging
- ☐ Remove unused dependencies or imports
- ☐ Ensure no secrets or credentials in code
- ☐ Verify all new files have appropriate permissions
- ☐ Clean up any test data or temporary configurations

7. KNOWLEDGE CAPTURE (OPTIONAL BUT RECOMMENDED)

If significant learning occurred:

- Document new patterns discovered
- Update relevant SOPs with improvements
- Note any anti-patterns to avoid
- Record performance optimizations found

VALIDATION

Session close-out is complete when:

- ☐ All checklist items above completed
- ☐ Git repository is clean with signed commits
- ☐ Documentation is updated and accurate
- ☐ Coordination agent has clear handoff information
- ☐ Next session can begin immediately with clear context

FAILURE RECOVERY

If close-out process fails:

1. **Do not start new work** until close-out is complete
2. **Fix git issues** before proceeding
3. **Update documentation** even if incomplete
4. **Note problems** in Run Journal for process improvement

This SOP ensures consistent, professional session management and seamless handoffs between development cycles.