

Coordination Handoff Standard Operating Procedure (SOP)

PURPOSE

Standardise how agents communicate with the coordination "brain" agent to ensure clear, actionable handoffs and prevent miscommunication.

SCOPE

Applies to all agents reporting back to the coordination system: main system, plugin development, infrastructure, testing.

AGENT IDENTIFICATION REQUIREMENTS (MANDATORY)

Every Response Must Start With:

```
## 🧠 AGENT: [Agent Type] - [Specific Component/Task]
STATUS: [Current Phase] - [Current Task]
BRANCH: [Git branch if applicable]
---
```

Examples:

```
## 🧠 AGENT: TaylorDash Main System Developer - Authentication Enhancement
STATUS: Phase 2 Complete - User Management Optimization
BRANCH: main
---
```

```
## 🧠 AGENT: Plugin Development Specialist - MCP Manager Plugin
STATUS: Phase 1 Foundation - Real MCP Detection Implementation
BRANCH: mcp-manager-plugin-dev
---
```

```
## 🧠 AGENT: Infrastructure Specialist - Plugin System Troubleshooting
STATUS: Diagnostic Phase - Plugin Tab Error Investigation
BRANCH: main
---
```

STANDARD COMPLETION REPORT FORMAT

Structure for All Completion Reports:

```
## ✅ [COMPONENT] - [PHASE] COMPLETE

### 🎯 MISSION ACCOMPLISHED
[Brief summary of what was achieved]

### 📦 DELIVERABLES COMPLETED
- [Specific deliverable 1 with evidence]
- [Specific deliverable 2 with evidence]
- [Specific deliverable 3 with evidence]

### 📋 VALIDATION RESULTS
- [Validation type] ✅ ❌ [Details and evidence]
- [Validation type] ✅ ❌ [Details and evidence]
- [Validation type] ✅ ❌ [Details and evidence]

### 🧠 AGENT PERFORMANCE
**Agents Used**
- @agent-name [Purpose and effectiveness]
- @agent-name [Purpose and effectiveness]

**Evidence Collected**
- [Screenshots, test results, etc.]
- [API responses, performance metrics, etc.]

### 🔄 NEXT PHASE READY
- [What's prepared for next development cycle]
- [Any dependencies or blockers]
- [Integration testing requirements]

### 🔄 COORDINATION NEEDS
- [Any coordination with other development streams]
- [Integration points to validate]
- [Dependencies on other components]

### 🔑 ACCESS/DEMO INFORMATION
- **Location** [File paths, URLs, etc.]
- **Commands** [How to test/validate results]
- **Credentials** [If applicable]
- **Expected Results** [What coordinator should see]

---
📌 FOR COORDINATION AGENT: [Specific guidance for next steps]
```

PROGRESS REPORT FORMAT (WORK IN PROGRESS)

For Ongoing Work Updates:

```
## 🔄 [COMPONENT] - [PHASE] IN PROGRESS

### 🔴 CURRENT STATUS
- [What's currently being worked on]
- [Percentage complete if applicable]
- [Current blocker or challenge if any]

### ✅ COMPLETED THIS SESSION
- [Specific accomplishments]
- [Evidence or proof of progress]

### 📋 NEXT STEPS
- [Immediate next tasks]
- [Expected completion timeline]

### 🟡 COORDINATION NEEDED
- [Any help needed from other streams]
- [Questions for coordination agent]

### 📋 AGENT EFFECTIVENESS
- [Which agents are working well]
- [Any agent performance issues]

---
📌 FOR COORDINATION AGENT: [What coordinator should know]
```

PROBLEM/BLOCKER REPORT FORMAT

When Issues Arise:

```
## 🚨 [COMPONENT] - ISSUE REPORT

### 🚩 PROBLEM DESCRIPTION
[Clear, specific description of the issue]

### 🔍 ROOT CAUSE ANALYSIS
[What investigation revealed]

### 📊 IMPACT ASSESSMENT
- **Severity** [Critical/High/Medium/Low]
- **Affected Components** [List]
- **Timeline Impact** [How this affects project timeline]

### 🔄 ATTEMPTED SOLUTIONS
- [What was tried]
- [Results of each attempt]

### 🛠️ PROPOSED RESOLUTION
- [Recommended approach]
- [Resources needed]
- [Estimated time to fix]

### 🟡 COORDINATION SUPPORT NEEDED
- [Specific help needed from other agents/streams]
```

```

- [Decision points for coordination agent]

---
📄 FOR COORDINATION AGENT: [specific action requested]
```

GIT REFERENCE STANDARDS

Always Include Git Information:

```

### 📄 GIT STATUS
- **Repository**: [Repository name]
- **Branch**: [Current branch]
- **Latest Commit**: [hash and message]
- **Changes**: [Summary of changes made]
- **Push Status**: 🟢 Pushed / 🟡 Ready to push / 🔴 Needs push
```

EVIDENCE REQUIREMENTS

All Reports Must Include:

- **Screenshots** for UI changes
- **Command outputs** for system changes
- **Test results** for functionality claims
- **Performance metrics** for optimization work
- **Error logs** for debugging work
- **API responses** for integration work

Evidence Format:

```

### 📄 EVIDENCE
- **Screenshots**: [Description of what each shows]
- **Test Results**: [Pass/fail counts and details]
- **Performance**: [Before/after metrics]
- **Validation**: [Specific proof points]
```

COORDINATION RESPONSE EXPECTATIONS

What Coordination Agent Will Provide:

- **Next Priority**: Clear direction for next development cycle
- **Integration Plans**: How components will work together
- **Resource Allocation**: Which agents to use for next phase
- **Timeline Updates**: Adjusted project timeline if needed
- **Dependency Resolution**: How to handle cross-component dependencies

ANTI-PATTERNS (AVOID)

🔴 Don't Do This:

- Send reports without agent identification
- Claim completion without evidence
- Use vague language ("mostly done", "almost working")
- Skip git information
- Omit coordination needs assessment
- Report only success without noting any issues

🟢 Always Do This:

- Start with clear agent identification
- Provide specific evidence for all claims
- Include git status and branch information
- Note coordination dependencies
- Be honest about partial completion or issues
- Include clear next steps

QUALITY CHECKLIST

Before sending any report, verify:

- ☐ Agent identification at start
- ☐ Specific deliverables with evidence
- ☐ Git information included
- ☐ Coordination needs identified
- ☐ Clear next steps outlined
- ☐ Evidence attached or referenced
- ☐ Honest assessment of completion/issues

This SOP ensures clear, actionable communication between all development streams and the coordination system.