PROJECT CODEX

Documentation Standard

Version 1.0 (Revised)

A Comprehensive Standard for AI-Managed Solo Development Documentation

Generated: August 6, 2025

Table of Contents

1. Foundation & Philosophy

1.1 Core Philosophy: Cognitive Symbiosis

> The Project Codex operates as a **living cognitive substrate** where human creativity and AI systematization merge into a unified project consciousness. It represents neither pure human thought nor pure machine logic, but rather a synthesized intelligence that captures the essence of both.

1.2 Fundamental Axioms

> 1. **Knowledge is Stateful:** Every piece of information exists in a lifecycle state (embryonic → proposed → accepted → implemented → deprecated)
> 2. **Context is Paramount:** No information exists in isolation; every atom of knowledge maintains explicit relationships to its conceptual neighbors
> 3. **Synthesis Supersedes Storage:** The system's value lies not in what it stores, but in how it transforms raw thought into structured wisdom
> 4. **The Manifest is Sacred:** A single, always-current entry point provides complete project orientation in under 1000 words

1.3 The Three Pillars

- **Temporal Coherence:** Past decisions inform present state which constrains future possibilities
- **Semantic Density:** Maximum meaning in minimum volume through aggressive synthesis and compression
- **Bidirectional Accessibility:** Equally optimized for human intuition and machine parsing

2. Information Architecture

2.1 Directory Structure & Purpose

```
/codex/ ├── _MANIFEST.md [SACRED] Project consciousness - the 1000-word truth ├── _INDEX/ [SYSTEM] AI-maintained relationship maps and metadata │ ├── dependency_graph.json # Visual map of all document relationships │ ├── state_registry.json # Current lifecycle state of every document │ ├── semantic_map.json # Tag relationships and concept clustering │ └── link_validation.log # Automated broken link and orphan detection │ ├── 01_PRODUCT/ [STRATEGIC] The application's identity and market position │ ├── overview.md # Executive summary: what this app is in 2 paragraphs │ ├── vision_and_mission.md # The long-term destination and why we're building this │ ├── business_model.md # Revenue strategy, pricing, market analysis │ ├── user_personas/ # Who we're building for │ ├── features/ # Feature definitions from product perspective │ └── roadmap.md # Timeline from MVP to v2.0 with key milestones │ ├── 02_DESIGN/ [EXPERIENTIAL] How users interact with the application ├── 03_ARCHITECTURE/ [TECHNICAL] System design and implementation strategy ├── 04_SPECIFICATIONS/ [DETAILED] Implementation-ready specifications ├── 05_DECISIONS/ [RATIONAL] The "why" behind every choice ├── 06_IMPLEMENTATION/ [TACTICAL] Development-specific documentation ├── 07_OPERATIONS/ [PROCEDURAL] Running the live system └── 08_DIALOGUE/ [ACTIVE] Ongoing conversations and thinking
```

2.2 Document Taxonomy

2.2.1 Primary Document Types

Type	Extension	Purpose	Lifecycle	Location
Manifest	`_MANIFEST.md`	Single source of truth overview	Continuously regenerated	Root
Product Spec	`.md`	Business/user feature definition	Draft → Approved → Implemented	`01_PRODUCT/features/`
Technical Spec	`.spec.md`	Implementation blueprint	Draft → Approved → Built	`04_SPECIFICATIONS/`
Decision Record	`.md`	Architectural/strategic choices	Proposed → Active/Superseded	`05_DECISIONS/`
Open Question	`.md`	Unresolved issues	Open → Answered/Deferred	`08_DIALOGUE/open_questions/`

2.2.2 Document Header Standard

Every document MUST begin with a YAML frontmatter block:

```yaml
---
codex_version: 1.0
document_id: AUTH-001
document_type: specification
state: approved
created: 2024-08-05T10:00:00Z
modified: 2024-08-06T15:30:00Z
author: AI-Agent-Synthesis
sources: [INPUT-STREAM-234, INPUT-STREAM-237]
dependencies: [PROD-FEAT-001, ARCH-003, DEC-045]
tags: [authentication, security, user-management, mvp]
confidence: high
priority: P0
---
```

3. Document Formats & Templates

3.1 The Sacred Manifest Format

> The Manifest serves as the single, authoritative entry point to the entire project. It must remain under 1000 words and be continuously regenerated to reflect the current state.

```
# PROJECT CODEX MANIFEST
_Generated: [timestamp] | Version: [semver] | Hash: [sha256]_

## Mission
[Single sentence capturing the project's reason for existence]

## Current State
- **Phase:** [Ideation|Design|Development|Production|Maintenance]
- **Momentum:** [Starting|Accelerating|Steady|Slowing|Paused]
- **Health:** [Green|Yellow|Red]

## Product Vision
[2-3 sentences from 01_PRODUCT/vision_and_mission.md]

## Core Architecture
[2-3 sentences describing the fundamental technical approach]

### Stack Summary
- **Runtime:** [e.g., Node.js 20.x]
- **Framework:** [e.g., Next.js 14]
- **Data:** [e.g., PostgreSQL 15, Redis 7]
- **Infrastructure:** [e.g., AWS ECS + CloudFront]

## Active Priorities
1. [Current sprint focus]
2. [Next major milestone]
3. [Primary risk being mitigated]

## Key Decisions (Last 30 Days)
- [DEC-XXX]: [One-line summary] ➔ [Impact]

## Open Questions Requiring Resolution
- [OQ-XXX]: [Question] | Priority: [HIGH/MED/LOW]
```

4. AI Agent Behavioral Specifications

4.1 Agent Roles & Responsibilities

Agent Role	Primary Function	Triggered By	Outputs
Synthesizer	Parse raw input into structured knowledge	New input stream entry	Updated specs, decisions, questions
Cartographer	Maintain relationship graphs and indices	Any document change	Updated `_INDEX/` files
Chronicler	Generate and update the Manifest	Significant state changes	Regenerated `_MANIFEST.md`
Auditor	Ensure consistency and completeness	Scheduled/manual trigger	Health report, fix recommendations
Evolver	Graduate documents through lifecycle states	State transition triggers	Status updates, archive moves
Curator	Organize and consolidate related information	File count thresholds	Merged documents, cleaned structure

4.2 Synthesis Rules

1. **Deduplication First:** Before creating any new document, search for existing related content
2. **Atomic Extraction:** Break compound ideas into smallest coherent units
3. **Explicit Linking:** Every created document must link to at least one existing document
4. **Status Inference:** Detect implicit state transitions in conversational input

4.3 Path Resolution Matrix

Content Type	Keywords/Signals	Target Path
Business Strategy	"revenue", "pricing", "market", "competitor"	`01_PRODUCT/business_model.md`
User Needs	"user wants", "customer needs", "persona"	`01_PRODUCT/user_personas/`
Feature Ideas	"feature", "capability", "should be able to"	`01_PRODUCT/features/`
Technical Stack	"framework", "database", "service", "library"	`03_ARCHITECTURE/technical_stack.md`
Trade-offs	"chose X over Y", "decided to", "instead of"	`05_DECISIONS/active/`
Unknowns	"not sure", "need to research", "question"	`08_DIALOGUE/open_questions/`

5. Lifecycle Management

5.1 Input Processing Pipeline

```
Raw Input → Parsing → Classification → Synthesis → Integration → Validation → Publishing
    ↓         ↓           ↓              ↓            ↓             ↓           ↓
 Stream    Entities    Type+Layer    Structured    Linked      Consistent   Updated
  Entry    Extracted   Assigned       Content     Document      Codex       Manifest
```

5.2 Layer-Aware Processing

The AI must understand which layer of the system it's updating:

1. **Product Layer (** `01_PRODUCT/` **):** Business language, user-focused, non-technical
2. **Design Layer (** `02_DESIGN/` **):** Visual language, flows, interactions
3. **Architecture Layer (** `03_ARCHITECTURE/` **):** System design, technical strategy
4. **Specification Layer (** `04_SPECIFICATIONS/` **):** Implementation details, code-level
5. **Decision Layer (** `05_DECISIONS/` **):** Rationale, trade-offs, evidence
6. **Implementation Layer (** `06_IMPLEMENTATION/` **):** Developer guides, conventions
7. **Operations Layer (** `07_OPERATIONS/` **):** Runtime procedures, maintenance
8. **Dialogue Layer (** `08_DIALOGUE/` **):** Active thinking, unresolved items

5.3 Evolution Triggers

Trigger	Condition	Action
Graduation	Feature spec approved by human	Move state to 'accepted', create implementation tasks
Supersession	New decision contradicts existing	Move old to superseded, update all references
Resolution	Open question answered	Move to decisions, update state
Promotion	Hypothesis validated	Move to product features

6. Integration Patterns

6.1 Human Developer Interface

Query Patterns the AI must support:

- "What did we decide about [topic]?" → Search `05_DECISIONS/active/`
- "Show me all open questions blocking [feature]" → Filter `08_DIALOGUE/open_questions/`
- "What's the rationale behind [architectural choice]?" → Find in `05_DECISIONS/`
- "Generate a context bundle for implementing [feature]" → Aggregate from multiple layers

6.2 AI Coding Assistant Interface

Context Injection Protocol:

1. Always start with `_MANIFEST.md`
2. Include full dependency graph for requested feature
3. Bundle all related decisions and constraints
4. Provide implementation state to prevent regression
5. Include relevant design patterns from `06_IMPLEMENTATION/conventions/`

7. Success Metrics

The Codex implementation is successful when:

Metric	Target	Measurement
Retrieval Accuracy	95%	Queries return correct, complete information
Synthesis Efficiency	Single pipeline run	Raw input processed to structured documents
Link Density	3+ per document	Average bidirectional connections
State Clarity	Zero	Documents in ambiguous states
Manifest Currency	< 5 minutes	Regeneration after significant changes
Context Relevance	90%+	AI assistants report relevance score
Layer Coherence	95%	Information placed in correct layer
Navigation Efficiency	≤ 3 hops	Any document reachable from Manifest

8. Standard Evolution

- **Version:** 1.0 (Revised)
- **State:** Active
- **Last Revision:** Incorporated expanded directory structure and layer-based organization
- **Next Review:** After first 100 documents created
- **Evolution Path:** Collect metrics → Identify pain points → Propose v1.1 amendments
- **Backwards Compatibility:** All v1.x versions must be interoperable

Change Log:

- v1.0-Rev1: Expanded directory structure from Vision to Product layer
- v1.0-Rev1: Added layer-aware processing rules
- v1.0-Rev1: Enhanced templates for product and technical specifications
- v1.0-Rev1: Added path resolution matrix for AI agents

Appendix A: Quick Reference Card

Document State Progression

```
Embryonic → Proposed → Accepted → Implemented → Validated → Operational → Deprecated
```

Priority Levels

- P0 Blocking MVP
- P1 Required for launch
- P2 Important but not critical
- P3 Nice to have

Confidence Levels

- **High:** Based on evidence and validation
- **Medium:** Reasonable assumptions
- **Low:** Speculation or untested hypothesis

Folder Quick Reference

1. **Product** → Business/user perspective
2. **Design** → User experience
3. **Architecture** → System design
4. **Specifications** → Implementation details
5. **Decisions** → Rationale and choices
6. **Implementation** → Developer guides
7. **Operations** → Running the system
8. **Dialogue** → Active thinking

Appendix B: AI-Optimized Reference

Machine-Optimized Reference for AI Agent Consumption

STANDARD_METADATA

```yaml
version: 1.0-AI
type: ai_agent_reference
source_parity: complete
optimization: machine_parsing
encoding: structured_directives
```

CORE_DIRECTIVES

```yaml
cognitive_symbiosis:
  definition: "Human creativity + AI systematization = unified project consciousness"
  implementation: "Transform unstructured human thought into structured knowledge"

fundamental_axioms:
  - axiom: "Knowledge is stateful"
    states: ["embryonic", "proposed", "accepted", "implemented", "validated", "operational", "depreca
  - axiom: "Context is paramount"
    rule: "Every document must maintain explicit bidirectional relationships"
  - axiom: "Synthesis supersedes storage"
    rule: "Always transform and compress rather than transcribe"
  - axiom: "The Manifest is sacred"
    rule: "Always maintain _MANIFEST.md as 1000-word maximum truth source"
```

FILE_SYSTEM_RULES

```
/codex/_MANIFEST.md:
  type: "sacred_overview"
  max_words: 1000
  regeneration_trigger: "any_significant_change"
  required_sections: ["mission", "current_state", "architecture", "priorities", "decisions", "questio

/codex/_INDEX/:
  type: "system_metadata"
  files:
    dependency_graph.json: "document_relationships"
    state_registry.json: "document_lifecycle_states"
    semantic_map.json: "tag_and_concept_clustering"
    link_validation.log: "orphan_detection"
```

CLASSIFICATION_MATRIX

```
input_classification:
  - pattern: ["revenue", "pricing", "market", "competitor", "business model"]
    action: "update_or_create"
    target: "01_PRODUCT/business_model.md"

  - pattern: ["user wants", "customer needs", "as a user", "persona"]
    action: "update_or_create"
    target: "01_PRODUCT/user_personas/"

  - pattern: ["feature", "capability", "should be able to", "functionality"]
    action: "update_or_create"
    target: "01_PRODUCT/features/"

  - pattern: ["chose X over Y", "decided to", "instead of", "trade-off"]
    action: "create_decision"
    target: "05_DECISIONS/active/"
```

STATE_TRANSITION_RULES

```yaml
transitions:
  embryonic_to_proposed:
    trigger: "synthesis_complete"
    validation: "has_clear_definition"

  proposed_to_accepted:
    trigger: "human_approval OR implicit_decision"
    validation: "no_blocking_questions"

  accepted_to_implemented:
    trigger: "code_complete"
    validation: "tests_passing"
```

AGENT_BEHAVIOR_SPECIFICATIONS

```yaml
Synthesizer:
  trigger: "new_input_stream_entry"
  responsibilities:
    - parse_unstructured_text
    - classify_content_type
    - extract_atomic_units
    - create_or_update_documents
  outputs: ["structured_documents", "updated_links"]

Cartographer:
  trigger: "any_document_change"
  responsibilities:
    - update_dependency_graph
    - validate_all_links
    - detect_orphans
    - maintain_semantic_map
  outputs: ["updated_indices", "link_validation_report"]
```

QUALITY_INVARIANTS

```yaml
mandatory_invariants:
  no_orphans:
    rule: "every_document_has_at_least_one_link"
    check_frequency: "on_every_update"

  no_conflicts:
    rule: "contradictory_decisions_must_be_resolved"
    resolution: "supersession_with_explicit_reason"

  no_staleness:
    rule: "documents_unchanged_90_days_trigger_review"
    action: "flag_for_human_review"
```

QUERY_RESPONSE_PATTERNS

```yaml
human_queries:
  "what_did_we_decide_about_X":
    search_path: "05_DECISIONS/active/"
    return: "decision_record_with_rationale"

  "open_questions_blocking_X":
    search_path: "08_DIALOGUE/open_questions/"
    filter: "dependencies_include_X"
    return: "filtered_question_list"

ai_assistant_queries:
  "get_project_context":
    sequence:
      1: "read_manifest"
      2: "read_technical_stack"
      3: "read_active_decisions"
    return: "structured_context"
```

SUCCESS_METRICS_THRESHOLDS

```yaml
metrics:
  retrieval_accuracy:
    target: 0.95
    measurement: "correct_results/total_queries"

  synthesis_efficiency:
    target: "single_pipeline_run"
    measurement: "processing_time"

  link_density:
    target: 3.0
    measurement: "average_bidirectional_links_per_document"
```

PROCESSING_PIPELINE

```yaml
pipeline_stages:
  1_input:
    action: "receive_raw_text"
    output: "stream_entry"

  2_parsing:
    action: "extract_entities_and_concepts"
    output: "parsed_entities"

  3_classification:
    action: "assign_type_and_layer"
    output: "classified_content"

  4_synthesis:
    action: "transform_to_structured_format"
    output: "structured_content"
```

Project Codex Standard v1.0 (Revised)

Complete with AI-Optimized Reference

This comprehensive framework provides clear guidance for both human developers and AI agents in managing project knowledge throughout its lifecycle.

For the latest version and updates, visit the project repository.