

Part 5: Phase 0 Developer Handbook

[Go to The FX Demo Root](#)



Financial Systems Archetype

Foreign Exchange (FX) Demo

Abstract

Part 5 provides practical guidance for developers on building, running, reviewing, and maintaining the Phase 0 implementation baseline. It describes repository organisation, file naming, script usage, generated artefacts, node implementation patterns, logging, exception handling, container build conventions, acceptance checks, and evidence capture. Part 5 supports repeatable implementation work by giving developers a consistent operational handbook. It complements Part 4 by explaining how developers apply the selected platform-specific model in source code, scripts, configuration files, and local runtime procedures.

Contents

- [Foreword](#)
- [Introduction](#)
- [1. Purpose and Scope](#)
- [2. Relationship to the Architecture Documents](#)
- [3. Phase 0 Baseline](#)
- [4. Repository Layout](#)
- [5. Development Environment](#)
- [6. Prerequisites and Tooling](#)
- [7. Naming and File Conventions](#)
- [8. Generated Artifacts](#)
- [9. DDS / IDL / Type Generation](#)
- [10. Node Implementation Pattern](#)

- [11. Control Plane Topics](#)
- [12. Data Plane Topics](#)
- [13. Build and Run Scripts](#)
- [14. Containerisation](#)
- [15. Logging and Observability](#)
- [16. Exception Handling](#)
- [17. Git and Review Workflow](#)
- [18. Acceptance Criteria](#)
- [19. Phase 0 Non-Goals](#)
- [20. Risks and Open Decisions](#)

Contents

* [Part 5: Phase 0 Developer Handbook](#)

- [Foreword](#)
- [Introduction](#)
- [1. Purpose and Scope](#)
 - [1.1 Purpose of the Handbook](#)
 - [1.2 Scope of Phase 0](#)
 - [1.3 Intended Audience](#)
 - [1.4 How to Use This Handbook](#)
- 4. [2. Relationship to the Architecture Documents](#)
 - [2.1 Parent Architecture Sources](#)
 - [2.2 Relationship to SIP-RA](#)
 - [2.3 Relationship to FDIS-RA](#)
 - [2.4 Relationship to Parts 1, 2, and 3](#)
 - [2.5 Relationship to the Phase 0 Implementation Source Material](#)
 - [2.6 Relationship to Generated Code and Runtime Artefacts](#)
- 5. [3. Phase 0 Baseline](#)
 - [3.1 Purpose of the Phase 0 Baseline](#)
 - [3.2 Logical Architecture Baseline](#)
 - [3.3 FX Demo Logical Profile Baseline](#)
 - [3.4 Implementation Technology Baseline](#)
 - [3.5 Repository Baseline](#)
 - [3.6 Baseline Change Control](#)
- 6. [4. Repository Layout](#)
 - [4.1 Repository Design Principles](#)
 - [4.2 Top-Level Repository Structure](#)
 - [4.3 Source Directories](#)
 - [4.4 Generated Artefact Directories](#)
 - [4.5 Configuration Directories](#)
 - [4.6 Script Directories](#)
 - [4.7 Test Directories](#)
 - [4.8 Documentation Directories](#)
 - [4.9 Git-Controlled and Non-Git-Controlled Content](#)
- 7. [5. Development Environment](#)
 - [5.1 Supported Operating Systems](#)
 - [5.2 VS Code / Code](#)

- 5.3 Python Environment
- 5.4 Java Environment
- 5.5 DDS Tooling
- 5.6 Docker / Container Environment
- 5.7 Git Environment
- 5.8 Environment Verification
- 8. 6. Prerequisites and Tooling
 - 6.1 Required Tools
 - 6.2 Optional Tools
 - 6.3 Version Requirements
 - 6.4 Tool Installation Notes
 - 6.5 Prerequisite Validation
 - 6.6 Troubleshooting Tool Issues
 - 6.7 Tool Baseline Checklist
 - 6.8 Multiple Products, Platforms, and Implementations
 - 6.9 Tool Baseline Checklist Instances
- 9. 7. Naming and File Conventions
 - 7.1 General Naming Principles
 - 7.2 Directory Naming
 - 7.3 Source File Naming
 - 7.4 Generated File Naming
 - 7.5 Script Naming
 - 7.6 Configuration File Naming
 - 7.7 Topic Naming
 - 7.8 Node Naming
 - 7.9 Container Naming
 - 7.10 File Headers and Inline Documentation
 - 7.11 Spelling, Grammar, and Style
- 10. 8. Generated Artefacts
 - 8.1 Purpose of Generated Artefacts
 - 8.2 Generated Versus Handwritten Files
 - 8.3 IDL-Derived Artefacts
 - 8.4 Python-Generated Support Artefacts
 - 8.5 Generated Documentation
 - 8.6 Regeneration Rules
 - 8.7 Protecting Handwritten Code from Overwrite
- 11. 9. DDS / IDL / Type Generation
 - 9.1 Role of IDL in Phase 0
 - 9.2 IDL Source Location
 - 9.3 Control Plane Type Definitions
 - 9.4 Data Plane Type Definitions
 - 9.5 Type Generation Workflow
 - 9.6 Generated Type Locations
 - 9.7 Type Generation Validation
 - 9.8 Type Compatibility Rules
- 12. 10. Node Implementation Pattern
 - 10.1 Purpose of the Node Pattern
 - 10.2 Node Lifecycle
 - 10.3 Node Startup Behaviour
 - 10.4 Node Runtime Behaviour
 - 10.5 Node Shutdown Behaviour

- 10.6 Node Status Reporting
 - 10.7 Node Command Handling
 - 10.8 Node Configuration
 - 10.9 Node Error Handling
 - 10.10 Node Documentation Requirements
13. 11. Control Plane Topics
- 11.1 Purpose of the Control Plane
 - 11.2 Node Status Topic
 - 11.3 Node Command Topic
 - 11.4 Control Plane Message Semantics
 - 11.5 Control Plane QoS Expectations
 - 11.6 Control Plane Validation
 - 11.7 Control Plane Observability
14. 12. Data Plane Topics
- 12.1 Purpose of the Data Plane in Phase 0
 - 12.2 Placeholder Data Plane Topics
 - 12.3 FX Demo Data Topics
 - 12.4 Topic Participation Rules
 - 12.5 Data Plane QoS Expectations
 - 12.6 Data Plane Validation
 - 12.7 Data Plane Non-Goals for Phase 0
15. 13. Build and Run Scripts
- 13.1 Purpose of Build and Run Scripts
 - 13.2 Shell Script Conventions
 - 13.3 Python Utility Script Conventions
 - 13.4 createRepositoryStructure.sh
 - 13.5 checkPrerequisites.sh
 - 13.6 initialisePythonEnvironment.sh
 - 13.7 generateTypes
 - 13.8 buildContainers.sh
 - 13.9 runAll.sh
 - 13.10 runNode.sh
 - 13.11 stopAll.sh
 - 13.12 prepareGitRepository.sh
 - 13.13 Script Exit Codes
 - 13.14 Script Logging
16. 14. Containerisation
- 14.1 Purpose of Containerisation in Phase 0
 - 14.2 Container Design Principles
 - 14.3 Container Build Inputs
 - 14.4 Container Runtime Configuration
 - 14.5 Container Networking
 - 14.6 Container Naming
 - 14.7 Container Logs
 - 14.8 Container Cleanup
17. 15. Logging and Observability
- 15.1 Purpose of Logging
 - 15.2 Log Format
 - 15.3 Log Location
 - 15.4 Node Logs
 - 15.5 Script Logs

- 15.6 Container Logs
- 15.7 Control Plane Observation
- 15.8 Evidence Collection
- 18. 16. Exception Handling
 - 16.1 Purpose of Exception Handling Rules
 - 16.2 Node Exceptions
 - 16.3 Script Failures
 - 16.4 Configuration Errors
 - 16.5 DDS Communication Errors
 - 16.6 Container Errors
 - 16.7 Recoverable and Non-Recoverable Errors
 - 16.8 Error Reporting
- 19. 17. Git and Review Workflow
 - 17.1 Repository Initialisation
 - 17.2 Branching Approach
 - 17.3 Commit Conventions
 - 17.4 Pull Request Expectations
 - 17.5 Review Checklist
 - 17.6 Generated File Review Rules
 - 17.7 Tagging the Phase 0 Baseline
 - 17.8 Review of Documentation and Comments
 - 17.9 Review Outcomes
 - 17.10 Relationship to Issues and Decision Records
- 20. 18. Acceptance Criteria
 - 18.1 Repository Acceptance Criteria
 - 18.2 Environment Acceptance Criteria
 - 18.3 Type Generation Acceptance Criteria
 - 18.4 Node Acceptance Criteria
 - 18.5 Control Plane Acceptance Criteria
 - 18.6 Data Plane Acceptance Criteria
 - 18.7 Container Acceptance Criteria
 - 18.8 Documentation Acceptance Criteria
 - 18.9 Evidence Acceptance Criteria
 - 18.10 Phase 0 Completion Criteria
- 21. 19. Phase 0 Non-Goals
 - 19.1 Production Readiness
 - 19.2 Full ACTUS Processing
 - 19.3 Full Financial Semantics
 - 19.4 Full Security Model
 - 19.5 Full IEF Policy Enforcement
 - 19.6 Full Operational Dashboard
 - 19.7 Performance Optimisation
 - 19.8 Enterprise Deployment
 - 19.9 Full Multi-Vendor Interoperability
 - 19.10 Complete Automation
- 22. 20. Risks and Open Decisions
 - 20.1 Implementation Language Decisions
 - 20.2 DDS Vendor Decisions
 - 20.3 Python Scope Decisions
 - 20.4 Container Runtime Decisions
 - 20.5 Generated Artefact Governance

- [20.6 Repository Ownership](#)
- [20.7 September 2026 Demo Dependencies](#)
- [20.8 Open Issues Register](#)
- [20.9 Decision Record Expectations](#)
- [20.10 Risk Review and Baseline Acceptance](#)

Notes for Editors

This document should be maintained as a set of linked wiki pages rather than a single monolithic page. Each major section should have its own stable URI so it can be referenced from other wiki pages, exported documents, presentations, AI-assisted tools, and semantic models.

Do not rename pages once they have been cited externally unless a redirect or move plan is in place.

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.

From:

<https://wiki.didosolutions.com/> - **Dido Solutions, Inc Wiki**

Permanent link:

<https://wiki.didosolutions.com/fxdemo/05-part/start?rev=1782956839>

Last update: **2026/07/01 18:47**

