

Annex C: Requirements

[Go to Annexes](#)

Annex C contains the canonical requirements register for Crucible.

The requirements are organized according to the requirement categories established in the Crucible System Requirements Specification. Each requirement has a stable identifier and resides on a separate page so that architecture pages, implementation artifacts, tests, Evidence, and other records can reference the requirement through a stable URI.

The requirements register distinguishes between:

- The governance status of a requirement
- The delivery phase associated with a requirement
- The implementation status of Crucible relative to a requirement
- The architecture and operational concepts that support a requirement
- The verification activities used to determine whether Crucible satisfies a requirement

The original System Requirements Specification provides the source baseline for this register. Annex C becomes the maintained representation of the Crucible requirements as the register is reviewed, corrected, and expanded.

Requirement Categories

- [C.1 Mission Objectives](#)
- [C.2 Operational Requirements](#)
- [C.3 Functional Requirements](#)
 - [C.3.1 Configuration Management](#)
 - [C.3.2 Image Management](#)
 - [C.3.3 Deployment Orchestration](#)
 - [C.3.4 Multi-Cloud Management](#)
 - [C.3.5 Air-Gapped Operations](#)
 - [C.3.6 Compliance Management](#)
 - [C.3.7 Test Framework Integration](#)
 - [C.3.8 DevSecOps Integration](#)
 - [C.3.9 Baseline Composition and Workspace](#)
 - [C.3.10 Dependency Capture and Offline Transfer](#)
 - [C.3.11 Management Interfaces](#)
- 4. [C.4 Security Requirements](#)
- 5. [C.5 Logging and Auditing Requirements](#)
- 6. [C.6 Reliability Requirements](#)
- 7. [C.7 Performance Requirements](#)
- 8. [C.8 Maintainability Requirements](#)
- 9. [C.9 Data Management Requirements](#)
- 10. [C.10 Interoperability Requirements](#)
- 11. [C.11 Future Capability Requirements](#)
- 12. [C.12 Key Value Proposition Requirements](#)

Requirement Page Structure

Each requirement page uses the following sections where applicable:

- Statement
- Source
- Rationale
- Applies To
- Verification
- Traceability
- ConOps Relationship
- Delivery Phase
- Implementation Status
- Requirement Status

Normative Language

Requirement statements use the following normative terms:

- **SHALL** identifies a mandatory requirement
- **SHALL NOT** identifies a mandatory prohibition
- **MAY** identifies a permitted capability or action
- **SHOULD** identifies a recommendation when advisory language remains appropriate

A requirement page may editorially correct the original source statement for terminology, atomicity, clarity, or normative style without changing its approved intent.

Requirement Status

Requirement Status identifies the governance state of the requirement.

Possible states include:

- Draft
- Proposed
- Approved
- Deprecated
- Superseded
- Withdrawn

Implementation Status

Implementation Status identifies the state of Crucible relative to the requirement.

Possible states include:

- Implemented
- Partially Implemented
- Demonstrated
- Planned
- Phase 2
- Strategic Unscheduled
- Unsupported
- Not Assessed

Requirement Status and Implementation Status remain separate. An approved requirement may remain planned, partially implemented, or unsupported.

Contents

- [C.1 Mission Objectives](#)
- [C.2 Operational Requirements](#)
- [C.3 Functional Requirements](#)

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

From:

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

Permanent link:

<https://wiki.didosolutions.com/dido/02-crusible/99-annexes/annex-c-requirements/start?rev=1784131833>

Last update: **2026/07/15 09:10**

