

FR-CFG-002 — Infrastructure as Code

[Go to Crucible Configuration Management Requirements](#)

Statement

Crucible SHALL manage each **Infrastructure Configuration** as a uniquely identified, machine-readable, **version-controlled Artifact** that authorized workflows can validate, compare, review, approve, and apply to an **Infrastructure Environment**.

Source Statement

The system SHALL support Infrastructure-as-Code methodologies.

Source

Crucible System Requirements Specification, Version 1.1 Draft, Functional Requirements, Configuration Management, FR-CFG-002.

Assessment

The source statement expresses the approved functional intent but does not provide all information required for deterministic verification.

The following Specification Discipline and Authoring findings apply:

- **The system** does not use the defined system name
- **Support** is a weak verb that does not identify the behavior **Crucible** performs
- **Infrastructure-as-Code** does not identify the infrastructure information managed as code
- **Methodologies** does not identify the required operations, controls, records, or outcomes
- The source statement does not identify whether the infrastructure representation must be machine-readable
- The source statement does not require unique identification or **Version Control**
- The source statement does not identify the validation, comparison, review, or approval activities applied to the configuration
- The source statement does not identify how an approved configuration relates to the resulting Infrastructure Environment
- The source statement does not identify the records required to preserve **Provenance**, **Auditability**, and **Traceability**

The normalized Statement replaces the weak verb with **manage**, identifies the Infrastructure Configuration as the controlled Artifact, and defines the operations required to treat infrastructure configuration as code.

Rationale

Infrastructure as Code applies software configuration management disciplines to infrastructure definitions.

An Infrastructure Configuration that exists only as an informal document, manually executed procedure, or untracked script cannot provide reliable review, comparison, approval, reuse, or reconstruction.

Crucible manages Infrastructure Configurations as machine-readable Artifacts that authorized workflows can:

- Create
- Validate
- Compare
- Review
- Approve
- Version
- Retrieve
- Apply
- Reuse
- Retire

The managed configuration records:

- A stable identifier
- A revision identifier
- The configuration content
- The applicable schema or validation rules
- The source repository
- The author or automated process
- The review and approval status
- The relationship to preceding and succeeding revisions
- The applicable **Baseline** references
- The Infrastructure Environments created or modified using the configuration

This treatment allows organizations to apply familiar software engineering controls to infrastructure changes before those changes affect an operational environment.

Infrastructure as Code supports:

- Automated validation
- Peer review
- Change approval
- Configuration comparison
- Controlled reuse
- Rollback
- Drift detection
- **Reproducibility**
- **Provenance**
- **Auditability**

- [Traceability](#)

Applies To

This requirement applies to:

- [Crucible](#)
- [Infrastructure as Code](#)
- [Infrastructure Configurations](#)
- [Declarative Configuration](#)
- [Crucible Descriptions](#)
- [Infrastructure Environments](#)
- [Infrastructure Baselines](#)
- [Baselines](#)
- [Artifacts](#)
- [Version Control](#)
- [Revisions](#)
- [Revision History](#)
- Configuration validation
- Configuration comparison
- Configuration review
- Configuration approval
- Configuration application
- Configuration rollback
- Configuration retirement
- [Configuration Management](#)
- [Provenance](#)
- [Auditability](#)
- [Reproducibility](#)
- [Traceability](#)
- [CI/CD Pipelines](#)
- [Operational Lifecycle](#)

Verification

1. Verification SHALL confirm that each tested Infrastructure Configuration has a unique identifier
2. Verification SHALL confirm that each tested Infrastructure Configuration has a revision identifier
3. Verification SHALL confirm that each tested Infrastructure Configuration uses a machine-readable representation
4. Verification SHALL confirm that the applicable Version Control system preserves the content associated with each configuration revision
5. Verification SHALL confirm that the Version Control system preserves the relationship between successive configuration revisions
6. Verification SHALL confirm that the Version Control system records the source and author of each tested revision
7. Verification SHALL confirm that Crucible validates an Infrastructure Configuration before applying it
8. Verification SHALL confirm that Crucible rejects an Infrastructure Configuration that violates an

applicable schema, constraint, or validation rule

9. Verification SHALL confirm that an authorized workflow can compare two Infrastructure Configuration revisions
10. Verification SHALL confirm that the configuration record identifies the applicable review and approval status
11. Verification SHALL confirm that Crucible applies only an authorized Infrastructure Configuration revision to the target Infrastructure Environment
12. Verification SHALL confirm that the resulting environment record identifies the Infrastructure Configuration identifier and revision used
13. Verification SHALL confirm that a change to approved configuration content creates a separately identified revision
14. Verification SHALL confirm that an authorized user can retrieve a prior Infrastructure Configuration revision
15. Verification SHALL confirm that a prior authorized revision can be selected for rollback when the applicable rollback criteria are satisfied
16. Verification SHALL confirm that the configuration and environment records preserve the applicable Provenance, Evidence, Auditability, and Traceability

Verification may include:

- Configuration identifier inspection
- Revision identifier inspection
- Machine-readable format inspection
- Repository history inspection
- Configuration schema validation
- Constraint validation
- Invalid configuration tests
- Configuration comparison
- Change review inspection
- Approval record inspection
- Unauthorized revision tests
- Configuration application tests
- Prior revision retrieval tests
- Rollback tests
- Environment record inspection
- End-to-end [CI/CD Pipeline](#) tests

The verification record SHALL identify:

1. The tested Infrastructure Configuration
2. The configuration identifier
3. The configuration revision identifier
4. The machine-readable representation
5. The configuration source location
6. The configuration author or automated process
7. The preceding revision
8. The succeeding revision, when applicable
9. The applicable schema and validation rules
10. The validation results
11. The review status
12. The approval status

13. The target Infrastructure Environment
14. The applied configuration revision
15. The resulting environment record
16. The rollback result, when applicable
17. The observed results
18. The generated [Evidence](#)

Outgoing Traceability

This requirement refines:

- [MO-001 — Repeatable, Compliant, and Secure Infrastructure Environments](#)
- [MO-002 — Reduced Platform Deployment Timelines](#)
- [MO-003 — Platform Independent Deployment](#)
- [MO-005 — Reusable and Version-Controlled Infrastructure](#)
- [MO-006 — Infrastructure Lifecycle Management](#)

This requirement operates within the conditions established by:

- [OR-001 — Operational Roles](#)
- [OR-002 — Deployment Environments](#)
- [OR-003 — Classified and Unclassified Environments](#)
- [OR-004 — Concurrent Environment Management](#)
- [OR-005 — Distributed Environment Management](#)

This requirement also relates to:

- [5. Descriptions, Composition, and Baselines](#)
- [7. Infrastructure and Deployment](#)
- [10. Reproducibility, Provenance, and Traceability](#)

Referenced By

The wiki Backlinks function provides the current list of pages that reference FR-CFG-002.

Incoming traceability should be derived dynamically from backlinks rather than maintained as a duplicate manual list.

Backlinks identify incoming references but do not define the semantics of each relationship. Referencing pages should identify whether the relationship represents realization, refinement, verification, dependency, or another defined traceability relationship.

ConOps Relationship

The Crucible Concept of Operations describes Phase 1 as a command-line utility invoked through [CI/CD Pipeline](#) steps.

The Phase 1 workflow uses version-controlled, machine-readable infrastructure configuration to:

- Select the applicable configuration revision
- Validate the configuration
- Review configuration changes
- Compose selected Baselines
- Build [Machine Images](#)
- Prepare deployment inputs
- Perform [Infrastructure Deployment](#)
- Record the configuration revision used by the deployment
- Preserve deployment results and Evidence

This workflow applies software engineering controls to infrastructure configuration before Crucible applies the configuration to an Infrastructure Environment.

Delivery Phase

Phase 1 and subsequent phases

Implementation Status

Not Assessed

Implementation status requires verification that Crucible manages Infrastructure Configurations as uniquely identified, machine-readable, version-controlled Artifacts that authorized workflows can validate, compare, review, approve, and apply.

Requirement Status

Draft

The source System Requirements Specification identifies Version 1.1 as a draft.

Notes for Editors

This requirement page should retain the stable requirement identifier FR-CFG-002.

Changes to the Statement SHALL preserve the approved intent of the source requirement.

Material changes should receive review and should update the related outbound traceability, verification criteria, acceptance criteria, configuration records, and source records.

The Source Statement should preserve the original wording from the controlling System

Requirements Specification.

Incoming traceability should use the wiki Backlinks function rather than a manually maintained list.

To reference this requirement Statement from another wiki page, insert:

```
{{section>dido:02-crusible:99-annexes:annex-c-requirements:03-functional-requirements:03-01-configuration-management:fr-cfg-002#Statement&noheader&nofooter&noeditbtn}}
```

Do not rename this page after an external citation 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/dido/02-crusible/99-annexes/annex-c-requirements/03-functional-requirements/03-01-configuration-management/fr-cfg-002?rev=1784214305>

Last update: 2026/07/16 08:05

