

C.3.10 Dependency Capture and Offline Transfer

[Go to Crucible Functional Requirements](#)

The [Dependency Capture](#) and [Offline Transfer requirements](#) define how [Crucible](#) captures [Build Dependencies](#), preserves those dependencies in a managed [Dependency Store](#), produces a [Transfer Bundle](#), populates [Offline Repositories](#) in a [Disconnected Environment](#), and permits the Dependency Store implementation to evolve without changing the Dependency Capture [Contract](#).

Together, these requirements establish controlled Dependency Capture and Offline Transfer across a [Transfer Boundary](#) while preserving the applicable [Transfer Authorization](#), [Transfer Bundle Integrity](#), [Provenance](#), and [Traceability](#).

Contents

- [FR-DEPC-001 — Capture Build Dependencies](#)
 - [FR-DEPC-002 — Preserve Dependencies in a Dependency Store](#)
 - [FR-DEPC-003 — Produce a Transfer Bundle](#)
 - [FR-DEPC-004 — Populate Offline Repositories](#)
 - [FR-DEPC-005 — Dependency Store Implementation Independence](#)
-
- [FR-DEPC-001 — Capture Build Dependencies](#)
 - [FR-DEPC-002 — Preserve Dependencies in a Dependency Store](#)
 - [FR-DEPC-003 — Produce a Transfer Bundle](#)
 - [FR-DEPC-004 — Populate Offline Repositories](#)
 - [FR-DEPC-005 — Dependency Store Implementation Independence](#)

Requirement Status

- ☐ Review and approve the Dependency Capture and Offline Transfer requirement set.
- ☐ Review and approve FR-DEPC-001 through FR-DEPC-005 as leaf requirements.

Issues

- ☐ Confirm that the requirement set distinguishes Dependency Capture, Dependency Store management, Transfer Bundle creation, Offline Repository population, and Dependency Store implementation independence without overlapping normative behavior.
- ☐ Confirm that requirements governing Offline Transfer identify the source environment, destination environment, Transfer Boundary, and required transfer result when the controlling source establishes

that information.

☐ Confirm that requirements governing transfer across a Transfer Boundary identify the applicable Transfer Authorization and Transfer Bundle Integrity obligations.

Notes for Editors

This page is a non-leaf requirement page and therefore retains a trailing `:start` in its namespace.

The namespace for this page is:

`dido:02-crusible:99-annexes:annex-c-requirements:03-functional-requirements:03-10-dependency-capture-and-offline-transfer:start`

Use the following controlling Terms and Definitions entries:

- [Dependency Capture](#)
- [Offline Transfer](#)
- [Transfer Bundle](#)
- [Build Dependency](#)
- [Offline Repository](#)
- [Dependency Store](#)

The individual requirement pages retain the stable identifiers FR-DEPC-001 through FR-DEPC-005.

Individual requirement pages are leaf nodes nested beneath the Dependency Capture and Offline Transfer namespace and omit a trailing `:start`.

The Derived From section on each leaf requirement page preserves the original wording from the controlling System Requirements Specification.

Normalized Statements use direct, testable behavior and identify the following only when the controlling source establishes that information:

- The captured Build Dependency
- The Dependency Capture event
- The Dependency Store
- The Transfer Bundle
- The source environment
- The destination environment
- The [Transfer Boundary](#)
- The transfer operation
- The required result

Requirements governing transfer across a Transfer Boundary identify the applicable [Transfer Authorization](#) and the controls used to preserve [Transfer Bundle Integrity](#) only when the controlling source establishes those obligations.

Requirements governing a [Disconnected Environment](#) or [Air-Gapped Environment](#) do not assume

access to external repositories or network services.

The Dependency Capture contract remains independent of the internal representation or implementation of the Dependency Store unless a lower-level requirement explicitly establishes that constraint.

Verification criteria test only the behavior stated in the normalized Statement and do not introduce additional normative obligations.

Incoming [Traceability](#) uses the wiki Backlinks function rather than a manually maintained list.

Each leaf requirement page includes its actual namespace in the DokuWiki section-transclusion example.

Do not rename a requirement 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/03-10-dependency-capture-and-offline-transfer/start?rev=1785506399>

Last update: 2026/07/31 06:59

