

# 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 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 applicable [Transfer Authorization](#), [Transfer Bundle Integrity](#), [Provenance](#), and [Traceability](#).

## Contents

- [FR-DEPC-001](#)
- [FR-DEPC-002](#)
- [FR-DEPC-003](#)
- [FR-DEPC-004](#)
- [FR-DEPC-005](#)

## Notes for Editors

The individual [requirement](#) pages should retain the stable identifiers FR-DEPC-001 through FR-DEPC-005.

Individual requirement pages are leaf nodes and should not include a trailing `:start` in their namespaces.

The Source Statement on each requirement page should preserve the original wording from the controlling System Requirements Specification.

Normalized Statements should use direct, testable behavior and should identify, as applicable:

- The captured 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 should identify the applicable [Transfer](#)

**Authorization** and the controls used to preserve **Transfer Bundle Integrity**.

Requirements governing a **Disconnected Environment** or **Air-Gapped Environment** should not assume access to external repositories or network services.

The dependency-capture contract should remain independent of the internal representation or implementation of the dependency store unless a lower-level requirement explicitly establishes that constraint.

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

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

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

```
{{section><namespace>#Statement&noheader&nofooter&noeditbtn}}
```

Replace <namespace> with the namespace of the requirement leaf page.

The section name following # SHALL match the section heading verbatim on the requirement page.

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=1784298723>

Last update: 2026/07/17 07:32

