7.1 Capture and Preserve Dependencies

Go to 7. Disconnected and Air-Gapped Operations

Before a lifecycle activity moves into a Disconnected Environment or Air-Gapped Environment, Crucible captures the Dependencies required to perform that activity without direct access to their original connected sources.

Dependency capture identifies the content required by the planned disconnected activity, retrieves that content from authorized sources, and records sufficient information to distinguish the captured content from another version or source.

Captured Dependencies are preserved in a Dependency Store so that subsequent transfer, import, repository population, construction, assessment, and deployment activities can use the identified dependency set.

Identify Required Dependencies

Crucible identifies the Dependencies required by the planned disconnected or air-gapped lifecycle activity.

The required dependency set can include:

- Software packages
- Package metadata
- Build tools
- Runtime libraries
- Base Images
- Container build inputs
- Image Layers
- Infrastructure Baselines
- Configuration content
- Compliance content
- Provider tools and plugins
- Scripts
- Direct Dependencies
- Transitive Dependencies

The applicable Baseline, build definition, assessment definition, deployment description, or other controlled lifecycle input determines which Dependencies are required.

Capture Dependencies

Crucible captures the identified Dependencies while the authorized source remains accessible.

For each captured Dependency, Crucible can record:

- The Dependency identifier
- The Dependency type
- The selected version or revision
- The source
- The retrieval location
- Available integrity information
- The relationship to the lifecycle activity requiring the Dependency
- The relationship to other captured Dependencies
- The capture result

Crucible captures the identified version or revision rather than relying on an unspecified or changing source reference.

When Crucible cannot capture a required Dependency, the capture result identifies the missing or unresolved Dependency.

Capture Direct and Transitive Dependencies

A direct Dependency is explicitly referenced by the selected lifecycle input.

A transitive Dependency is required by another Dependency in the captured set.

Crucible captures both direct and transitive Dependencies when the planned disconnected activity requires them.

Capturing only the directly referenced content can produce an incomplete dependency set when the activity also depends on packages, tools, libraries, metadata, or other indirectly required resources.

Preserve Captured Dependencies

Crucible preserves the captured dependency set in a Dependency Store.

The preserved representation maintains the information required to identify:

- Each captured Dependency
- Its selected version or revision
- Its source
- Its integrity information
- Its relationship to the planned lifecycle activity
- Its relationship to other Dependencies
- Its preservation location
- Its capture and preservation status

Preservation makes the captured content available for subsequent preparation of the Transfer Bundle.

Preservation does not independently authorize transfer across a Security Domain or other controlled boundary.

Extensible Dependency Representation

The Dependency Store represents different dependency types without limiting Crucible to one package manager, repository type, operating system, build tool, or artifact format.

An extensible representation allows Crucible to preserve the identifying information and relationships needed for additional dependency types as Crucible supports them.

The representation does not require every Dependency to use the same native format. It provides a consistent means to identify and relate the preserved content to the lifecycle activity that requires it.

Verify the Captured Set

Before producing the Transfer Bundle, Crucible determines whether the captured dependency set contains the Dependencies identified for the planned disconnected activity.

The verification can identify:

- Successfully captured Dependencies
- Missing Dependencies
- Unresolved transitive Dependencies
- Unavailable versions
- Integrity mismatches
- Unauthorized sources or content
- Dependencies that cannot be represented or preserved

An incomplete dependency set can prevent the planned activity from completing after transfer into the destination environment.

Capture and Preservation Result

The capture and preservation result identifies:

- The planned disconnected lifecycle activity
- The required dependency set
- The captured direct Dependencies
- The captured transitive Dependencies
- The selected versions or revisions
- The source of each Dependency
- The integrity information
- The Dependency Store representation
- The preservation location
- The capture and preservation status
- Any missing or unresolved Dependency

The preserved dependency set becomes an input to 7.2 Produce the Transfer Bundle.

Requirements Addressed

Requirement	Statement
FR-DEPC-001 — Capture Build Dependencies	Crucible SHALL perform Dependency Capture for Build Dependencies.
FR-DEPC-002 — Preserve Dependencies in a Dependency Store	Crucible SHALL preserve captured Build Dependencies in a managed Dependency Store.
FR-DEPC-005 — Dependency Store Implementation Independence	Crucible SHALL permit the implementation of a Dependency Store to change without changing the Dependency Capture Contract.

The linked leaf requirement pages remain the canonical sources.
