# Module XX – Capsule Mock Composer & Developer Kickstart Framework

Version: 4.1 | Classification: CTO Mandatory (Engineering Support)

Scope: Local capsule creation support, CLI examples, mock injector logic

## 0. Purpose & Developer Entry Logic

This module provides a simple schema-driven capsule mock composer for engineering teams and prototype developers. It supports structured capsule creation, replay simulation, trust violation triggers, and fork path generation – independent of full runtime deployment. Used for testing, learning, and simulation preparation.

## 1. Capsule Mock Structure

Each mock capsule must define:  
`{ mock\_id, capsule\_type, trust\_scope, payload\_stub[], parent\_hash, anchor\_sim, issue\_flag[] }`  
Supported capsule\_type values:  
- MANIFEST, AUDIT, EXECUTION, REPLAY, ALERT, GRACE, FORK, FEDERATION  
- Can simulate valid/invalid, sandboxed, or signed-only state

## 2. Composer Fields & Flag Triggers

Flag options:  
- `simulate\_replay`: triggers hash collision path  
- `trust\_tier\_override`: forces escalation rejection  
- `fork\_path`: alternate capsule hash lineage  
- `grace\_active`: enters GC + GRCC scenario  
Composer may export to `.mocap` format for CLI ingestion or schema preview.

## 3. CLI Example (YAML + JSON)

Example YAML:  
capsule:  
 capsule\_type: 'MANIFEST'  
 trust\_scope: 'TIER\_3'  
 payload\_stub:  
 origin: 'AddOnAlpha'  
 hash\_seed: 'A0B1C2'  
 issue\_flag:  
 - simulate\_replay  
 - trust\_tier\_override  
 export\_as: 'example\_replay\_01.mocap'

## 4. Use Case Scenarios

- Generate replay attack test capsules for Module 14  
- Compose downgraded fork capsules for Module 18  
- Prepare mock simulation sets for Module 17 validation  
- Build onboarding material for developer training  
- Preview alert chain emissions via CLI path replay

## 5. Limitations & Certification Disclaimer

This tool does not replace full capsule validation or runtime enforcement. It is a developer-side aid to compose, visualize, and debug capsule structures. Certification relies on full runtime replay (Module 17) and not on mock previews.