

# **MaxOneOpen - Twin Lifecycle Control**

*Document ID: v3.4-CTRL-001*

Document ID	v3.4-CTRL-001
Title	Twin Lifecycle Control – MaxOneOpen
Version	1.0
Date	2025-03-31
Author	MaxOne Documentation Unit (GPT-Validated)
Document Type	Operational Governance Layer

### FOUNDATION – Purpose of Lifecycle Control

This document defines the lifecycle phases of any MaxOneOpen twin instance. It governs how twins are created, verified, executed, monitored, forked, archived, and eventually deleted – in compliance with structural and cryptographic governance rules.

### EXECUTION – Twin Lifecycle Phases

Phase	Trigger	System Behavior
1. Creation	Manifest + code submitted	Signature & hash generation, readiness check
2. Activation	Execution request + node init	Runtime boot, resource allocation, log start
3. Runtime	Twin active	Execution loop, periodic verification, peer sync
4. Forking	New twin declared from origin	Manifest duplication + fork signature
5. Suspension	Runtime paused manually or by rule	Twin enters frozen state with last-known hash
6. Archiving	Marked non-active + export flagged	Archived as immutable ZIP with metadata
7. Deletion	Final deactivation approved	Secure twin wipe incl. manifest hash log

### STACK – Lifecycle Governance Rules

- Twins must not bypass lifecycle phases without recorded state transition.
- Forked twins inherit hash lineage but must declare a new twin\_id.
- Only suspended or archived twins are eligible for deletion.
- Each transition must be logged in local twin-log and peer-signed (if mesh-enabled).
- Lifecycle transitions must validate against manifest version and ZKP structure.

### FINAL – CTO-Relevant Conclusion

Twin lifecycle governance ensures architectural and operational consistency. Every MaxOneOpen twin follows the same cryptographically verifiable pattern from creation to deletion. This document enables teams and validators to enforce traceable runtime boundaries – structurally, legally, and operationally.

Status: Final lifecycle governance reference – GPT-certified

