MaxTreaty v1.0 – License Class Embedding and Scope Binding

This document describes how MaxTreaty embeds license classifications and binds their scope to capsules, operators, or deployments. It ensures that legal restrictions and usage boundaries are transparently enforceable across all linked systems.

# 1. License Class Structure

MaxTreaty integrates license class logic from the TBYD classification model. Each treaty may reference a fixed license class:  
- Class A: Critical infrastructure systems (e.g., MaxOneOpen)  
- Class B: Functional modules (e.g., MaxAudit, MaxTune, MaxBridge)  
- Class C: Utility-level extensions and OSS-specific structures  
The class is defined within `treaty.scope.license.class`.

# 2. Embedding into Treaty Capsules

Each MaxTreaty document embeds its license class via:  
- treaty.scope.json: with license.class field  
- capsule.signature.sig: including enforcement scope hash  
- optional anchor.exec.ref to MaxReg class registry

# 3. Scope Binding

Each license class defines boundaries for:

* - Who may use or deploy the capsule
* - In which environments (public, state, enterprise)
* - Whether delegation or sub-licensing is allowed

# 4. Enforcement Interaction

License class mismatches can trigger blocking behavior in MaxDeploy or MaxBridge. Example: A Class A treaty embedded in a Class C capsule would be flagged and rejected. Treaties enforce integrity through class compliance checking.

# 5. Exportability and Audit Binding

Treaties that define license class binding must also include exportable verification components for external audits:  
- license.class.ref.yaml  
- legal.scope.json  
- capsule.audit.trace.sig