2 · Configuration

Configure key behaviour, vehicle state, sharing, recovery, theft systems and persistence.

- Core Configuration
- Physical Keys
- Remote Configuration
- Sharing & Job Access
- Persistence & Ownership

Core Configuration

The primary configuration file is:

```
config/main.lua
```

Common options:

```
Config.Debug = false
Config.Locale = 'en'

Config.Keybinds = {
    Lock = 'L',
    Hotwire = 'H'
}
```

Distances are controlled separately:

```
Config.Distances = {
    ToggleLock = 18.0,
    Action = 7.0,
    Share = 7.0,
    Ensure = 35.0
}
```

Vehicle state

```
Config.VehicleState = {
    OwnedVehiclesSpawnLocked = true,
    NPCVehiclesSpawnLockedChance = 65,
    NPCLocksPlayerOnly = true,
    NPCKeysInsideChance = 15,
}
```

`NPCLocksPlayerOnly` keeps ambient NPC traffic usable by GTA AI while still enforcing the logical player-facing lock state.

Physical Keys

Physical-key behaviour is configured under `Config.PhysicalKeys`.

```
Config.PhysicalKeys = {
    Enabled = true,
    Item = 'vehicle_key',
    SyncWithAccess = true,
    ItemGrantsAccess = true,
    RequirePhysicalItemForAccess = true,
    OwnerAccessWithoutPhysicalItem = false,
    StoredAccessWithoutPhysicalItem = false,
    AllowClientOwnerClaim = false,
    IssueForJobAccess = false,
}
```

Strict mode behaviour

With the default strict settings:

- A physical key is authoritative for normal owner/shared access.
- Deleting or losing the item removes that physical access.
- Ownership alone does not silently recreate a missing key.
- A client cannot claim an owner key simply because it is near an owned vehicle.
- Trusted server resources should intentionally call `GiveKeys` when issuing a new key.

This prevents accidental unlimited key regeneration.

Remote Configuration

Using a physical `vehicle_key` can open the built-in vehicle remote.

```
Config.PhysicalKeys.Remote = {
    Enabled = true,
    Range = 25.0,
    RefreshMs = 250,
    AllowTrunk = true,
    CloseOnAction = false,
    TrunkDoors = {},
}
```

The remote supports:

- Lock
- Unlock
- Open/close supported trunk
- Live in-range status
- Plate display

The remote closes with `ESC`. Its NUI is transparent while closed and does not create a fullscreen dark overlay.

`TrunkDoors` can be used to explicitly map unusual/front-trunk vehicles to a door index.

Sharing & Job Access

Key sharing

```
Config.ShareKeys = {
    Enabled = true,
    AllowReshare = false,
    DefaultDurationSeconds = 3600,
    PersistTemporaryKeys = true,
}
```

Player commands:

```
/givekey [server id]
/revokekey [server id]
```

The player must use the current vehicle or a nearby vehicle within the configured share distance.

Job access

Job access is empty by default to avoid unintended fleet access.

Example:

```
Config.JobAccess.police = {
    minGrade = 0,
    requireDuty = true,
    platePrefixes = { 'LSPD', 'POL' },
    models = { 'police', 'police2', 'police3' }
}
```

Job access can stay virtual so shared fleet vehicles do not generate large numbers of physical key items.

Persistence & Ownership

Persistence

```
Config.Persistence = {
    Enabled = true,
    Resource = 'oxmysql',
    AutoInstall = true,
    QueryTimeoutMs = 10000,
    StartupTimeoutMs = 60000,
    RetryIntervalMs = 1000,
    PersistPermanentSharedKeys = true,
    PersistTemporaryKeys = true,
}
```

Ownership tables

Default ownership mappings:

```
Config.Ownership.QB = {
    table = 'player_vehicles',
    ownerColumn = 'citizenid',
    plateColumn = 'plate',
    vehicleColumn = 'vehicle',
}

Config.Ownership.ESX = {
    table = 'owned_vehicles',
    ownerColumn = 'owner',
    plateColumn = 'plate',
    vehicleColumn = 'vehicle',
}
```

QBX uses the same default `player_vehicles` layout as QB.

For a custom garage schema, either change the configured table/columns or enable custom ownership and implement the ownership bridge.