Storage & Persistence

Nord Doorlock supports persistence without requiring a database.

JSON Mode

Default:

```
Config.Storage = 'json'
```

Despite the name, the authoritative store is FiveM Resource KVP.

KVP key:

```
nord_doorlock:doors:v2
```

The resource also mirrors data to:

```
data/doors.json
data/doors.backup.json
```

This design allows doors to survive resource/server restarts even when the resource directory is not writable.

Source Selection on Load

When using JSON mode, Nord Doorlock can read:

- KVP
- `data/doors.json`
- `data/doors.backup.json`

The newest valid source is selected.

Legacy JSON arrays are accepted and migrated to the newer persisted wrapper.

JSON Payload Shape

Current mirror structure:

```
{
  "version": 2,
  "savedAt": 1700000000,
  "doors": []
}
```

MySQL Mode

Configure:

```
Config.Storage = 'mysql'
```

oxmysql must be started.

The table is auto-created.

Optional SQL:

```
CREATE TABLE IF NOT EXISTS `nord_doorlock` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `data` LONGTEXT NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Each database row stores the door payload in the `data` column.

Auto Mode

```
Config.Storage = 'auto'
```

If oxmysql is running, MySQL is used.

Otherwise Nord Doorlock falls back to KVP/JSON.

Persistence Safety

For authoritative changes, the server saves before broadcasting success.

If persistence fails, state-changing API operations attempt to roll back the in-memory change rather than pretending the operation succeeded.

Storage Diagnostics

Console or administrator command:

```
doors_storage
```

Typical fields:

```
mode
ready
doors
error
```

Developer export:

```
local health = exports.nord_doorlock:GetStorageHealth()
```

Returned data includes values such as:

```
{
    mode = 'json',
    configuredMode = 'json',
    lastError = nil,
    kvpKey = 'nord_doorlock:doors:v2',
    ready = true,
    serverError = nil,
    doorCount = 10
}
```

Reloading From Persistence

Server export:

```
local ok, count, removedDuplicates =
    exports.nord_doorlock:ReloadDoors()
```

This reloads definitions, performs duplicate cleanup and broadcasts a full synchronization.

Revision #1
Created 2026-09-08 13:05:56 UTC by Riguita
Updated 2026-09-08 13:05:56 UTC by Riguita