

Configuration générale

Ce chapitre détaille toutes les options de `config/config.lua`. C'est ici que vous adaptez MasterLadder à votre serveur : framework, permissions, interface, touches, vitesses, sons et distances. Les échelles elles-mêmes se règlent dans `config/ladders.lua` (voir [Configuration des échelles](#)).

☐ Toute option absente ou invalide reprend sa valeur par défaut, avec un avertissement dans la console : une faute de frappe ne fait jamais planter le script.

1. Licence

- **Config.MasterModsKey** : votre MasterKey (`"MM-XXXXXXX"`). Une clé mise dans `server.cfg` avec `set mastermods_key "..."` est prioritaire. Voir [Installation](#).

2. Langue

- **Config.Locale** (défaut `"fr"`) : langue des messages et de l'interface.
 - `"fr"` : français
 - `"en"` : anglais
 - `"es"` : espagnol
 - `"de"` : allemand

3. Framework & permissions

- **Config.FrameworkMetier** (défaut `"standalone"`) : framework utilisé pour lire le métier des joueurs.
 - `"esx"`, `"qb"`, `"qbx"`, `"ox"`, `"standalone"`
- **Config.NotifySystem** (défaut `"gta"`) : système de notifications.
 - `"ox"`, `"esx"`, `"qb"`, `"okok"`, `"gta"`
- **Config.TargetSystem** (défaut `"ox"`) : système de visée (œil ALT).
 - `"ox"` : ox_target
 - `"qb"` : qb-target
 - `"none"` : touches et aide à l'écran

- Si la ressource choisie n'est pas démarrée, le script passe automatiquement aux touches.
- **Config.TargetDistance** (défaut `2.0`) : portée des options de visée, en mètres.
- **Config.JobRestriction** (défaut `false`) : active la restriction par métier (ignorée en standalone).
- **Config.RequireOnDuty** (défaut `true`) : le joueur doit être en service (QBCore / QBox uniquement).
- **Config.AuthorizedJobs** : grade minimum par métier et par action. Un métier absent de la liste n'a accès à rien.

```
Config.AuthorizedJobs = {
  ['pompier'] = {
    deploy = 1, -- déployer / replier
    enter  = 0, -- monter au poste ou dans la nacelle
    control = 1, -- manœuvrer l'échelle
    lock    = 0, -- verrouiller les portes
    locker  = 0, -- coffres et écran arrière
  },
}
```

Le chiffre correspond au **grade minimum** requis.

- **Config.CustomPermission** (défaut `nil`) : votre propre fonction de permission, qui remplace la restriction par métier. Elle reçoit `source`, l'action (`deploy`, `enter`, `control`, `lock`, `locker`) et le nom de l'échelle. Retourner `false` refuse l'action.

```
Config.CustomPermission = function(source, action, ladder)
  return IsPlayerAceAllowed(source, 'masterladder.use')
end
```

4. Interface

- **Config.DefaultInterface** (défaut `"modern"`) : pupitre par défaut.
 - `"modern"` : écran tactile
 - `"retro"` : pupitre années 2000
- **Config.AvailableInterfaces** (défaut `{ modern = true, retro = true }`) : interfaces que les joueurs peuvent choisir.
- **Config.AllowPlayerUiSettings** (défaut `true`) : `false` désactive `/mlui`, tout le monde garde les réglages ci-dessous.
- **Config.DefaultUiCorner** (défaut `"br"`) : position du pupitre (`"br"` bas droite, `"bl"` bas gauche, `"tr"` haut droite, `"tl"` haut gauche).

- `Config.DefaultUiScale` (défaut `1.0`) : taille, de `0.6` à `1.6`.
- `Config.DefaultUiCompact` (défaut `false`) : mode compact (relevés et voyants seulement).
- `Config.DefaultUiVolume` (défaut `0.6`) : volume des sons du pupitre, de `0` à `1`.
- `Config.ShowKeyHints` (défaut `true`) : aide aux touches en bas d'écran pendant la manœuvre.
- `Config.UiRefreshRate` (défaut `100`) : millisecondes entre deux mises à jour du pupitre.

Une échelle peut aussi imposer son interface : voir `interface` dans [Configuration des échelles](#).

5. Commandes & touches

- `Config.Commands` : noms des commandes du chat (sans le `/`).

Clé	Défaut	Action
<code>set</code>	<code>mlset</code>	Déployer / replier (verrou nacelle pendant la manœuvre)
<code>enter</code>	<code>mlenter</code>	Monter / descendre / changer de place
<code>control</code>	<code>mlcontrol</code>	Prendre / lâcher les commandes
<code>light</code>	<code>mllight</code>	Projecteur
<code>lock</code>	<code>mllock</code>	Verrouiller les portes
<code>camera</code>	<code>mlcam</code>	Caméra opérateur
<code>ui</code>	<code>mlui</code>	Réglages de l'interface
<code>climb</code>	<code>mlclimb</code>	Gravir / redescendre l'échelle
<code>exit</code>	<code>mlexit</code>	Dépannage : se détacher de l'échelle
<code>admin</code>	<code>mladmin</code>	Administration
<code>monitor</code>	<code>mlmonitor</code>	Prendre / lâcher la lance canon (dans la nacelle)
<code>pto</code>	<code>mlpto</code>	Engager / désengager la prise de force (au volant)
<code>pads</code>	<code>mlcales</code>	Poser / ramasser les cales sous les patins (sol meuble)

- `Config.DefaultKeys` : touches par défaut des commandes (`set = 'U'`, `enter = 'F'`, `control = 'R'`, `light = 'Y'`, `camera = 'H'`, `climb = 'G'`, `monitor = 'J'`, `pto = 'K'`).

Chaque joueur peut les changer dans **Paramètres > Raccourcis > FiveM**.

⚠ FiveM lie les touches au **nom** de la commande : si vous renommez une commande, les touches personnalisées des joueurs reviennent à la valeur par défaut.

- **Config.ManeuverControls** : contrôles GTA utilisés pendant la manœuvre ([liste des contrôles](#)).
- `rotateRight` / `rotateLeft` (défaut `35` / `34`, D / Q)
- `raise` / `lower` (défaut `32` / `33`, Z / S)
- `extend` / `retract` (défaut `131` / `132`, Maj / Ctrl)
- `armUp` / `armDown` (défaut `38` / `44`, E / A)
- `lightRight`, `lightLeft`, `lightUp`, `lightDown` (défaut flèches)
- `exitLeft` / `exitRight` (défaut `174` / `175`) : maintenir pour sortir de la nacelle sur le côté
- `climbUp` / `climbDown` (défaut `32` / `33`, Z / S) : monter / descendre l'échelle
- `interact` (défaut `51`, E) : coffres et écran arrière sans target

6. Gameplay & mouvement

Vitesses maximales

- **Config.RotationSpeed** (défaut `2.25`) : m/s en bout d'échelle (la vitesse de rotation s'adapte à la longueur sortie).
- **Config.ElevationSpeed** (défaut `1.5`) : m/s en bout d'échelle.
- **Config.ExtendSpeed** (défaut `1.5`) : m/s.
- **Config.ArmSpeed** (défaut `19.36`) : °/s, bras de nacelle.
- **Config.LightSpeed** (défaut `5.73`) : °/s, projecteur.

Inertie et butées

- **Config.EnableInertia** (défaut `true`) : démarrage et arrêt progressifs.
- **Config.AccelerationTime** (défaut `0.35`) / **Config.BrakingTime** (défaut `0.25`) : secondes pour atteindre la vitesse maximale / pour s'arrêter.
- **Config.SlowDownNearLimits** (défaut `true`) : ralentit à l'approche des butées.
- **Config.SlowDownZone** : distance avant la butée où le ralentissement commence (° ou m) pour `rotation`, `elevation`, `length`, `arm`, `light`.
- **Config.SlowDownMinSpeed** (défaut `0.15`) : vitesse minimale dans cette zone (fraction de la vitesse max).

Nacelle

- **Config.BasketLockedOnDeploy** (défaut `true`) : nacelle verrouillée à la fin du calage ; l'opérateur la déverrouille pendant la manœuvre (**U**). `false` = accessible dès la fin du calage.
- **Config.GroundClearanceLower** (défaut `0.2`) / **Config.GroundClearanceExtend** (défaut `0.1`) : garde au sol minimale de la nacelle (m) pour descendre / sortir l'échelle.

Montée par l'échelle

- **Config.EnableLadderClimb** (défaut `true`) : active la montée à pied.
- **Config.MaxClimbers** (défaut `2`) : personnes sur l'échelle en même temps (1 ou 2).
- **Config.ClimbSpeed** (défaut `0.6`) : m/s le long de l'échelle.
- **Config.BlockLadderWhileClimbing** (défaut `true`) : l'échelle ne peut pas être manœuvrée quand quelqu'un y grimpe (le projecteur reste libre).
- **Config.ClimbAnimation** (défaut `{ dict = "laddersbase", clip = "climb_up", cycle = 0.6 }`) : animation d'escalade ; `cycle` = mètres parcourus par cycle d'animation. Augmentez-le si les mains « patinent » sur les barreaux.
- **Config.ClimbOffset** (défaut `{ normal = 0.70, height = 0.95 }`) : position du joueur sur l'échelle.
 - `normal` : distance aux barreaux. Joueur dans ou sous l'échelle → augmenter (pas de `0.05`).
 - `height` : décalage le long de l'échelle (position des mains sur les barreaux).
- **Config.ClimbTilt** (défaut `{ axis = "x", invert = false }`) : inclinaison du joueur comme l'échelle. `invert = true` s'il se penche en arrière ; `axis = "y"` s'il part sur le côté.
- **Config.ClimbHelperModel** (défaut `"prop_golf_ball"`) : petit prop invisible servant d'ancrage au joueur.
- **Config.ClimbDownTarget** (défaut `{ distance = 0.6, height = 1.1, reach = 3.0, model = "prop_cs_cardbox_01" }`) : point de visée « Redescendre par l'échelle » dans la nacelle (écart au centre, hauteur, portée, prop invisible qui sert de surface à viser).

Caméra opérateur

- **Config.CameraDistance** (défaut `7.0`), **Config.CameraMinDistance** (`3.0`), **Config.CameraMaxDistance** (`18.0`) : distance de la caméra (m), réglable à la molette.
- **Config.CameraSensitivity** (défaut `4.0`).

Mise en station — `Config.Stabilization`

- `mode` (défaut `"manual"`) : `"manual"` = l'opérateur cale lui-même chaque stabilisateur à l'écran arrière, le porteur suit (soulèvement, cambrage) ; `"auto"` = séquence automatique.
- `levelTolerance` (défaut `1.0`) : calage manuel, écart de niveau maximal (°) pour valider.
- `maxLift` (défaut `0.18`) : calage manuel, levée maximale sur le plat (m). Course d'un vérin = descente jusqu'au sol sous son logement + `maxLift` : le stabilisateur ne quitte jamais son logement ; au-delà, des cales.
- `slopeCompensation` (défaut `6.0`) : compensation de dévers de l'échelle (°, `0` = aucune). Le calage reste valide jusqu'à cet angle ; l'échelle se remet d'aplomb.
- `slopeAbaqueLoss` (défaut `0.03`) : abaque réduit de cette fraction par degré d'inclinaison au-delà de `levelTolerance` (50 % minimum).
- `blocks` : cales sous les patins au pupitre (`enabled`, `height` = 0,15 m, `max` = 3 par patin, `model` = prop utilisé).
- `requirePT0` (défaut `true`) : prise de force obligatoire avant le calage (touche « pto », au volant, véhicule arrêté, moteur tournant). Engagée, le véhicule ne peut plus rouler ; elle ne se désengage qu'échelle rangée.

- `autoLevel` (défaut `true`) : mise à niveau du porteur (châssis remis à l'horizontale). `false` : le porteur garde sa pente.
- `maxStroke` (défaut `2.2`) : course maximale d'un vérin (m). Au-delà (trou, fort dévers), le calage est refusé.
- `wheelLift` (défaut `0.12`) : roues décollées du sol après la mise à niveau (m).
- `maxSpeed` (défaut `0.5`) : vitesse maximale du véhicule (m/s) pour engager la prise de force et caler.
- `timing` (défaut `{ out = 1800, down = 1600, stagger = 400, level = 1800, unfold = 1200, min = 300 }`) : durées en ms (sortie d'un stabilisateur, descente d'un vérin pour 1 m de course, décalage entre deux stabilisateurs, mise à niveau, dépliage de la nacelle, durée minimale d'une descente). Si l'échelle a un `deployTime` supérieur à 0, toute la séquence est ramenée à cette durée.
- `obstacleCheck` (défaut `true`) : obstacle sur la trajectoire d'un stabilisateur → calage réduit de ce côté s'il passe, sinon refus.
- `pedCheck` (défaut `true`), `pedRadius` (défaut `1.2`) : refus si quelqu'un se tient sous un patin ou sur sa trajectoire.
- `softGround` (défaut `{ enabled = true, factor = 0.8, requirePads = false }`) : sol meuble sans cales → abaque multiplié par `factor` ; `requirePads = true` refuse le calage sans cales.
- `padTime` (défaut `4000`) : ms pour poser / ramasser les cales.
- `emergencySpeed` (défaut `0.25`) : moteur hors service, vitesse de la pompe de secours (fraction de la vitesse normale).
- `beeps` (défaut `true`), `beepRange` (défaut `30.0`) : bips de manœuvre et hydraulique audibles autour du véhicule (m).

Abaque — `Config.Abaque`

- `enabled` (défaut `true`) : active le calculateur de charge (courbes par échelle dans `config/ladders.lua`, voir [Configuration des échelles](#)).
- `mode` (défaut `"block"`) : `"block"` = au bord de la courbe, seuls les mouvements qui rapprochent la nacelle restent possibles (serveur et client) ; `"warn"` = alarme seulement.
- `countClimbers` (défaut `true`) : les personnes sur l'échelle comptent dans la charge.
- `slowZone` (défaut `1.5`) : mètres avant la limite où la manœuvre ralentit.
- `reducedKey` (défaut `21`, Maj) : contrôle à maintenir avec la touche « déployer » pour un calage réduit.

Lance canon — `Config.WaterMonitor`

- `enabled` (défaut `true`) : active la lance de la nacelle.
- `range` (défaut `40.0`) : portée du jet (m, à 45° sur terrain plat).
- `radius` (défaut `3.0`) : rayon d'extinction autour de l'impact (m).
- `drawDistance` (défaut `200.0`) : distance d'affichage du jet pour les autres joueurs (m).
- `sendRate` (défaut `100`) : ms entre deux envois de l'orientation de la lance.
- `fireControl` (défaut `24`, clic gauche) : contrôle GTA maintenu pour arroser.
- `native` (défaut `true`) : éteint les feux du jeu chez tous les joueurs proches (ce qui couvre les scripts de feu basés sur les feux natifs, comme ceux de GIMI ou d'Albo1125).

- `smartFires` (défaut `{ enabled = true, resource = "SmartFires", damage = 25.0, radius = 3.0, v1Hits = 6 }`) : intégration London Studios SmartFires. En v2, chaque impact (2 par seconde) retire `damage` points de vie aux feux dans `radius` m, et le joueur est crédité. En v1, un feu touché `v1Hits` fois est éteint.
- `fx` (défaut `{ jet = "water_cannon_jet", spray = "water_cannon_spray", scale = 1.0, rotation = { x = 0.0, y = 0.0, z = 0.0 } }`) : particules du jet (attachée à un support invisible orienté comme la lance) et de l'impact. `rotation` corrige l'orientation de la particule si le jet ne part pas dans l'axe de la lance (réglage en jeu : `Config.Debug = true` puis `/mlwfx x y z`).

☐ Un autre script de feu peut se brancher sur l'impact du jet : voir [Exports et événements](#).

7. Audio

- `Config.EnablePanelSounds` (défaut `true`) : sons du pupitre (moteur hydraulique, bip de butée, déclic du verrou).
- `Config.LockerSoundDistance` (défaut `10.0`) : portée du son des coffres (m).
- `Config.LockerSoundVolume` (défaut `0.45`).
- `Config.InVehicleSoundFactor` (défaut `0.4`) : atténuation quand le joueur est dans un véhicule.
- `Config.ManeuverSounds` `{ enabled, range = 10.0, volume = 1.0 }` : sons de manœuvre entendus autour de l'EPAN (pompe hydraulique, moteur électrique, bips quand les stabilisateurs bougent), du gauche à droite selon la caméra. L'opérateur entend toujours sa machine. Pour des sons réels, déposez `html/sounds/hydraulic.ogg` (boucle de pompe) et `html/sounds/motor.ogg` (boucle de moteur) : ils remplacent la synthèse automatiquement.

7 bis. Sécurité de la manœuvre

- `Config.Wind` `{ enabled, warn = 10.0, stop = 12.5, slow = 0.5 }` : anémomètre (m/s, vent du jeu, plus fort en hauteur). Au-delà de `warn` : alarme et manœuvre ralentie ($\times$ `slow`). Au-delà de `stop` (12,5 m/s $\approx$ 45 km/h) : on ne peut plus que rentrer, descendre et tourner.
- `Config.AntiCollision` `{ enabled, slow = 2.0, stop = 0.6, shunt = true, shuntSpeed = 0.3 }` : capteurs tout autour de la nacelle et au-dessus (façades, poteaux, câbles, véhicules). En approche, la manœuvre ralentit dès `slow` (m) et s'arrête à `stop`. On peut toujours s'éloigner de l'obstacle. `shunt` : bouton « Shunt anticollision » du pupitre à joysticks pour couper la protection et dégager une nacelle coincée, à `shuntSpeed` de la vitesse.
- `Config.Joysticks` `{ enabled = true, showByDefault = true, cursorControl = 25 }` : pupitre à joysticks cliquable pendant la manœuvre (joystick gauche rotation / inclinaison, droit longueur / bras nacelle, proportionnels ; arrêt d'urgence, vitesse lente, verrou, projecteur,

caméra, lâcher). `showByDefault` : affiché pour un joueur qui ne l'a pas masqué dans `/mlui`.
`cursorControl` : contrôle GTA qui prend la souris (25 = clic droit). Les touches restent actives.

8. Rendu & réseau

- `Config.LadderDrawDistance` (défaut `250.0`) : distance d'affichage des échelles déployées (m).
- `Config.PropLodDistance` (défaut `300`) : distance de détail des props (m).
- `Config.StowedStabilizers` (défaut `true`) : les 4 stabilisateurs sont posés en position rentrée sur le véhicule dès son apparition, puis remplacés par ceux de l'échelle au déploiement. `false` = stabilisateurs visibles seulement une fois l'échelle déployée.
- **Projecteur** : `Config.SpotlightDistance` (`150.0`), `Config.SpotlightColor` (`{ 255, 255, 255 }`), `Config.SpotlightBrightness` (`80.0`), `Config.SpotlightRadius` (`25.0`).
- `Config.NetworkRate` (défaut `100`) : ms entre deux envois de position par l'opérateur.
- `Config.InterpolationDelay` (défaut `160`) : ms de retard d'affichage chez les autres joueurs, pour un mouvement parfaitement lisse.
- `Config.PropHierarchy` (défaut `true`) : sections attachées à la tourelle. `false` = placement individuel (solution de repli).
- `Config.LocalAttachOthers` (défaut `true`) : les autres occupants sont attachés localement (supprime leurs tremblements).

9. Distances

Côté joueur

- `Config.PanelDistance` (défaut `2.0`) : distance à l'écran arrière pour déployer.
- `Config.EnterHelpDistance` (`3.0`) / `Config.EnterDistance` (`4.0`) : affichage de l'aide / portée de la commande « monter ».
- `Config.LockerDistance` (`1.0`) / `Config.LockerHelpDistance` (`1.75`) : ouverture / aide des coffres.
- `Config.NearVehicleDistance` (`7.0`) : véhicule considéré à proximité.

Côté serveur (anti-triche) : distances maximales tolérées, un peu plus larges que côté joueur.

- `Config.MaxPanelDistance` (`3.0`), `Config.MaxEnterDistance` (`5.0`),
`Config.MaxInteractionDistance` (`7.0`).

10. Debug

- `Config.Debug` (défaut `false`) : journal complet dans la console serveur et dans F8. Voir [En cas d'erreur](#).
- `Config.DebugCategories` (défaut `nil` = tout) : limite le journal à certaines catégories, par exemple `{ "license", "deny" }`.

⚠ Laissez `Config.Debug = false` en production : le journal est très bavard.

Revision #2

Created 2026-10-19 09:08:48 UTC by Raph

Updated 2026-10-19 09:25:24 UTC by Raph