

Script MasterLadder

MasterLadder : échelle pivotante EPAN / EPA pour FiveM (MasterMods).

- [Présentation](#)
- [Dépendances](#)
- [Installation](#)
- [Guide d'utilisation](#)
- [Configuration générale](#)
- [Configuration des échelles](#)
- [Permissions & administration](#)
- [En cas d'erreur](#)
- [Journaux des changements](#)
- [Exports et événements](#)

Présentation

Ce chapitre présente MasterLadder, l'échelle pivotante pompiers (EPAN / EPA) de MasterMods pour FiveM.

MasterLadder transforme un véhicule à échelle en véritable engin d'intervention : calage au sol, manœuvre de l'échelle depuis le poste de la tourelle ou depuis la nacelle, montée à pied par l'échelle et deux pupitres de commande au choix. Toute l'échelle est synchronisée pour l'ensemble des joueurs, sans saccades.

Ce que fait le script

- **Déploiement réaliste** : calage des 4 stabilisateurs, contrôle de la pente du terrain, véhicule immobilisé et portes verrouillées pendant l'intervention.
- **Manœuvre** : rotation, inclinaison, longueur et bras de nacelle, avec inertie hydraulique et ralentissement près des butées. Zone interdite à l'arrière et protection de la cabine.
- **Équipage** : un poste de commande sur la tourelle, une nacelle jusqu'à 3 personnes, et jusqu'à 2 personnes sur l'échelle.
- **Montée à pied** : du pied de l'échelle (sur le porteur) jusqu'à la nacelle, et retour.
- **Deux pupitres** : écran tactile moderne ou pupitre années 2000, au choix de chaque joueur (ou imposé par le serveur).
- **Équipements** : projecteur orientable, caméra opérateur, arrêt d'urgence, verrouillage de la nacelle et des portes, coffres latéraux, écran de commande arrière.
- **Interactions** : ox_target / qb-target, ou touches avec aide à l'écran.
- **Sécurité** : le serveur valide chaque action (distances, permissions, état de l'échelle) et refuse les tentatives de triche.

Compatibilité

- **Frameworks** : ESX, QBCore, QBox, ox_core, Standalone
- **Target** : ox_target, qb-target, aucun
- **Notifications** : ox_lib, ESX, QBCore, okok, GTA
- **Langues** : français, anglais

Licence

MasterLadder est **gratuit**. Il nécessite une **MasterKey**, gratuite elle aussi : voir [Installation](#).

Support : discord.mastermods.dev

Dépendances

Ce document liste ce dont MasterLadder a besoin pour fonctionner.

Obligatoire

- **OneSync** activé sur le serveur (`set onesync on` dans `server.cfg`, ou via txAdmin).
- **Un véhicule à échelle et ses props**, streamés par une autre ressource (non inclus dans MasterLadder). Les noms des modèles doivent correspondre à ceux de `config/ladders.lua` (voir [Configuration des échelles](#)).
 - Exemple : le mod **Renault D16 EPAN BSPP** fournit le véhicule et les props `prop_t12_*` utilisés par l'entrée `epabspp`.
- **Une MasterKey** gratuite : ressources.mastermods.dev/CreerMasterKey.php

Optionnel

- **Framework** (pour la restriction par métier) : `es_extended`, `qb-core`, `qbx_core` ou `ox_core`.
- **Target** : `ox_target` ou `qb-target`. Sans target, tout se fait aux touches.
- **Notifications** : `ox_lib` ou `okok` si vous les choisissez dans `Config.NotifySystem`.

⚠ Démarrez le framework, le target et la ressource du véhicule **avant** MasterLadder dans `server.cfg`.

Installation

Ce chapitre décrit la mise en place de MasterLadder sur votre serveur.

1. Fichiers

1. Téléchargez `MasterLadder-vX.Y.Z.zip`.
2. Décompressez le dossier `MasterLadder` dans votre dossier `resources/` (par exemple `resources/[mastermods]/MasterLadder`).

⚠ Ne renommez pas les fichiers à l'intérieur du dossier.

2. server.cfg

```
ensure ox_target          # si vous utilisez un target
ensure votre-vehicule-epan # la ressource qui stream le véhicule et les props
ensure MasterLadder
```

3. MasterKey

Créez votre clé gratuite sur ressources.mastermods.dev/CreerMasterKey.php, puis renseignez-la à **un seul de ces deux endroits** :

- dans `server.cfg` (recommandé, prioritaire) :

```
set mastermods_key "MM-XXXXXXX"
```

- ou dans `config/config.lua` :

```
Config.MasterModsKey = "MM-XXXXXXX"
```

Au démarrage, la console serveur affiche `Licence MasterMods validée pour : <nom du serveur>`. La clé est revérifiée toutes les 30 minutes.

⚠ Une même MasterKey peut servir sur **3 serveurs différents maximum par 24 h**. Au-delà, elle est bloquée 1 jour.

En cas de problème, voir [En cas d'erreur](#).

4. Configuration

- `config/config.lua` : framework, permissions, interface, touches, vitesses, sons... voir [Configuration générale](#).
- `config/ladders.lua` : vos véhicules à échelle (props, butées, extras, places)... voir [Configuration des échelles](#).

5. Administration (optionnel)

Pour autoriser vos administrateurs à utiliser `/mladmin` :

```
add_ace group.admin command.mladmin allow
```

Mise à jour

Remplacez tout le dossier **sauf** `config/config.lua` et `config/ladders.lua`, puis comparez-les avec ceux de la nouvelle version : les nouvelles options sont listées dans les [Journaux des changements](#). Une option absente de votre config prend sa valeur par défaut.

Guide d'utilisation

Ce chapitre explique comment utiliser l'échelle en jeu, du calage au repli. Les touches indiquées sont celles par défaut : chaque joueur peut les changer dans **Paramètres > Raccourcis > FiveM**, et le serveur peut modifier les valeurs par défaut (voir [Configuration générale](#)).

Avec un système de visée (ox_target / qb-target), les actions « à pied » se font à l'**œil (ALT)** sur le véhicule ou sur l'échelle. Sans target, une aide en haut de l'écran indique la touche à utiliser.

1. Mettre l'échelle en station

1. Garez le véhicule, **à l'arrêt**, sur un sol le plus plat possible (pente et dévers : 8° maximum par défaut).
2. **Au volant**, engagez la **prise de force** : touche **K** (`/mlpto`), moteur tournant. Le véhicule ne peut plus rouler tant qu'elle est engagée. En calage manuel, l'engin **étudie aussitôt sa position** : une notification dit si un équilibre est possible ici (de niveau ou en dévers compensé, avec combien de cales sous quels patins), ou ce qui manque (« il manque 40 cm sous l'arrière gauche même avec 3 cales », obstacle). Il n'y a pas de limite en degrés : seule cette étude décide, et le calage est refusé au démarrage si elle échoue.
3. **À l'arrière du véhicule**, allumez l'écran de commande : visée « **Allumer / éteindre l'écran** » ou touche **E**.
4. *Sol meuble (herbe, terre, sable, gravier, neige)* : en calage manuel, les cales se posent sous chaque patin depuis le pupitre (**E / A**, voir plus bas). En calage automatique : visée « **Poser / ramasser les cales** » ou `/mlcales`.
5. Lancez la mise en station : visée « **Calage manuel des stabilisateurs** » ou touche **U** devant l'écran (à pied).

Calage manuel (par défaut)

Debout à l'écran arrière, face au camion, vous calez vous-même, stabilisateur par stabilisateur. Le porteur suit comme un vrai camion : tant qu'un patin ne pousse pas, il reste sur ses roues ; un vérin qui pousse soulève son coin et le camion s'incline (il peut **cambrier**) en pivotant sur ses autres appuis. La tourelle et l'échelle suivent l'inclinaison du châssis.

Touche	Action
Ctrl + ← / →	choisir le stabilisateur (AVG, AVD, ARD, ARG)
Ctrl + ↑ / ↓	tous ensemble / un seul

Touche	Action
← / →	rentrer / sortir le stabilisateur
↓ / ↑	descendre / remonter le vérin
E / A	glisser / retirer une cale sous le patin (patin levé)
U	valider le calage (ou ranger, une fois tout rentré)
X (maintenu 1 s, réglable dans les raccourcis FiveM)	abandonner : tout se range seul, cales ramassées
F	quitter le poste (le calage reste en l'état)

Toutes les touches sont listées sur le pupitre, une par ligne, avec les touches réellement liées.

Le **niveau à bulle** fonctionne comme un vrai : la bulle part du côté **haut** ; descendez les vérins du côté opposé.

Conseils : le pupitre indique en bleu la prochaine manœuvre utile (sortir tel stabilisateur, descendre tel vérin, coin le plus bas et de combien de centimètres, quand caler) et entoure le patin concerné. Les verrouillages s'affichent en orange.

Abandonner : **X** maintenu, l'option de visée « Abandonner le calage » ou `/mlset abandon`. Les vérins remontent, les cales sont ramassées, les stabilisateurs rentrent, puis l'EPAN est rangée.

Verrouillages comme sur une vraie EPA : le vérin ne descend que stabilisateur sorti (au moins la largeur réduite), le stabilisateur ne bouge que vérin remonté, la descente s'arrête si quelqu'un est sous le patin, la sortie s'arrête devant un obstacle (limite rouge sur le pupitre), et le vérin s'arrête en **fin de course** : sur le plat, le châssis ne monte pas de plus de `maxLift` (0,18 m). Quelques centimètres suffisent à décoller les roues.

En forte pente, comme les pompiers :

- **Placez l'engin dans le sens de la pente** plutôt qu'en travers : le dévers est le plus pénalisant.
- **Course fixe du vérin** : le stabilisateur ne quitte jamais son logement. Le vérin descend jusqu'au sol sous son logement, plus `maxLift`. Si le sol est plus bas (bordure de trottoir, dévers) ou si un coin doit monter plus, le pupitre affiche « sol trop bas » : il faut des cales.
- **Calez les patins** : levez le patin, **E** glisse une cale de 15 cm dessous (jusqu'à 3 empilées), **A** en retire une. Chaque cale rehausse l'appui et ajoute 15 cm de levée possible. En pente, caliez surtout les patins aval. Une cale compte aussi comme cale sur sol meuble. Un stabilisateur calé ne bouge plus latéralement.
- **Compensation de dévers** : si le porteur ne peut pas être remis parfaitement de niveau, le calage reste valide jusqu'à 6° d'inclinaison (`slopeCompensation`). La tourelle suit le châssis et l'échelle se remet d'aplomb pendant le dépliage. Sur le pupitre, le niveau indique « compensé » (anneau pointillé). L'abaque est réduit de 3 % par degré au-delà de 1° (`slopeAbaqueLoss`).

Le **pupitre de calage** montre la vue de dessus (patin vert = en appui), la course de chaque vérin, un **niveau à bulle** et l'état : patins en appui, roues décollées, niveau. Le calage est validé avec **4 patins en appui, roues décollées et niveau $\leq 1^\circ$** (`levelTolerance`) ; un stabilisateur sorti partiellement donne un calage réduit de ce côté. La nacelle se déploie ensuite.

Un poste laissé libre peut être repris par un autre joueur (visée « **Prendre les commandes des stabilisateurs** »).

Calage automatique (mode = "auto")

La séquence se déroule comme sur une vraie EPA, avec bips de manœuvre et bruit hydraulique autour du camion :

Étape	Ce qui se passe
Sortie des stabilisateurs	un par un : avant gauche, avant droit, arrière droit, arrière gauche
Descente des vérins	un par un, jusqu'au contact du sol (plus long si le sol est bas)
Mise à niveau du porteur	les vérins soulèvent le camion : roues décollées, pente et dévers corrigés
Dépliage de la nacelle	l'échelle est prête

La progression et l'étape en cours s'affichent sur le pupitre. Le véhicule est immobilisé et ses portes verrouillées.

Interrompre : visée « **Interrompre le calage** » ou **U** pendant la séquence : les stabilisateurs rentrent depuis leur position du moment.

Calage réduit (rue étroite) : visée « **Déployer (calage réduit)** », **Maj + U** ou `/mlset réduit` pour les deux côtés ; `/mlset gauche` ou `/mlset droite` pour un seul côté. Si un **obstacle** (mur, véhicule, poteau) empêche un stabilisateur de sortir complètement, ce côté passe tout seul en calage réduit. Le déport autorisé par l'abaque baisse alors **du côté réduit** : il reste plein au-dessus de l'avant et de l'arrière du camion.

⚠ La mise en station est refusée, avec un message qui précise la cause, si :

- la prise de force n'est pas engagée, le véhicule roule, est dans l'eau ou a le moteur hors service ;
- la pente ou le dévers dépasse le maximum ;
- le sol n'est pas trouvé sous un patin, est trop haut (bordure) ou trop bas (course du vérin dépassée) ;
- quelqu'un se tient sous un patin ou sur sa trajectoire ;
- un obstacle empêche même le calage réduit d'un côté.

Sur **sol meuble sans cales**, la mise en station se fait mais l'abaque est réduit (voyant de calage orange et alarme « Sol meuble »).

2. Prendre place

Place	Comment
Poste de commande (tourelle)	Visée sur la tourelle « Monter au poste de commande », ou F près de la tourelle
Nacelle (jusqu'à 3)	Visée sur la nacelle « Monter dans la nacelle », ou F près de la nacelle
Échelle (jusqu'à 2)	Voir « 4. Monter par l'échelle » ci-dessous

- **F** (ou `/mlenter`) permet aussi de descendre. `/mlenter b2` choisit une place précise (`seat`, `b1`, `b2`, `b3`).
- Pour sortir de la nacelle sur le côté, maintenez **← ou →** en appuyant sur **F**.
- `[]` Par défaut, la **nacelle est verrouillée** à la fin du calage : l'opérateur la déverrouille pendant la manœuvre (**U**). On peut toujours y arriver par l'échelle.

3. Manœuvrer

Depuis le poste ou la nacelle, appuyez sur **R** pour prendre les commandes (une seule personne à la fois). Le pupitre s'affiche.

Touche	Action
Q / D	Rotation gauche / droite
Z / S	Monter / baisser l'échelle (inclinaison)
Maj / Ctrl	Sortir / rentrer l'échelle (longueur)
E / A	Lever / baisser le bras de nacelle
Flèches	Orienter le projecteur
Y	Allumer / éteindre le projecteur
U	Verrouiller / déverrouiller la nacelle
H	Caméra opérateur (molette : distance)
R	Lâcher les commandes

Pupitre à joysticks (souris)

Pendant la manœuvre, un pupitre à joysticks s'affiche au-dessus du pupitre (au-dessous si le pupitre est placé en haut de l'écran), comme sur une vraie EPA. Les touches restent actives.

- **Clic droit** : prendre la souris (le curseur apparaît). **Clic droit** ou **Échap** : la rendre.
- **Joystick gauche** : ← → rotation, ↑ ↓ lever / baisser l'échelle.
- **Joystick droit** : ↑ ↓ sortir / rentrer, ← → bras nacelle bas / haut.
- Attrapez la poignée et poussez-la : **plus on pousse, plus ça va vite**. Lâchez, elle revient au neutre et le mouvement s'arrête.
- Boutons : **arrêt d'urgence** (coup de poing : tout s'arrête, cliquez à nouveau pour réarmer), **vitesse lente / rapide**, verrou nacelle, projecteur, caméra, lâcher. Croix du **projecteur** (maintenir).
- Une touche du clavier passe toujours avant le joystick. Sans souris, les joysticks bougent avec les touches.
- **Shunt anticollision** : si la nacelle reste bloquée par l'anticollision, ce bouton coupe la protection (voyant orange, alarme sur le pupitre) et la manœuvre passe en vitesse lente. Recliquez pour la rétablir ; elle revient d'elle-même en lâchant les commandes.
- Interface années 2000 : le pupitre à joysticks prend l'habillage du pupitre (boîtier gris, film ardoise), et les touches du clavier à membrane et l'arrêt d'urgence sont aussi cliquables.
- On peut masquer ce pupitre dans (« Pupitre à joysticks »).

Au calage (poste de l'écran arrière), les commandes de calage s'affichent de la même façon au-dessus du pupitre de calage : **clic droit** pour prendre la souris, puis :

- choisir le stabilisateur (AVG, AVD, ARG, ARD, **Tous ensemble**), ou cliquer directement sur un patin de la vue de dessus ;
- **joystick** : ← → rentrer / sortir le stabilisateur, ↓ descendre le vérin, ↑ le remonter (proportionnel) ;
- **Cale + / Cale –**, **Valider**, **Abandonner** (maintenir le bouton 1 s), **Quitter**.

Les mouvements démarrent et s'arrêtent progressivement, et ralentissent à l'approche des butées. Le pupitre affiche :

- les valeurs (rotation, inclinaison, longueur, bras), la hauteur, le déport et la portée ;
- la position de l'échelle vue de côté et de dessus, avec les zones interdites ;
- les voyants : calage, verrou nacelle, contact sol, projecteur, butées ;
- le nombre de personnes dans la nacelle.

Moteur hors service : la pompe de secours prend le relais, la manœuvre est lente mais toujours possible (pour ramener la nacelle et replier).

Sécurités automatiques :

- Zone interdite à l'arrière du véhicule et pas d'inclinaison sous l'horizontale au-dessus de la cabine.
- Arrêt avant que la nacelle ne touche le sol.
- Échelle bloquée tant que quelqu'un y grimpe (le projecteur reste libre).

- **Abaque** : déport limité selon la charge (voir ci-dessous).
- **Anémomètre** : vent fort (alarme « Vent fort ») = manœuvre ralentie ; vent trop fort = on ne peut plus que rentrer, descendre et tourner l'échelle.
- **Anticollision** : près d'une façade, d'un poteau ou d'un câble, la nacelle ralentit puis s'arrête (alarme « Anticollision »). S'éloigner reste possible.
- **Nacelle auto-nivelante** : son plancher reste horizontal, même châssis incliné.

Abaque (calculateur de charge)

Comme sur une vraie EPA, la nacelle ne peut pas aller aussi loin avec 3 personnes qu'à vide. Le **déport** (distance horizontale entre la tourelle et la nacelle) est limité selon :

- le **nombre de personnes** dans la nacelle (et sur l'échelle, selon la configuration) ;
- le **calage** : complet ou réduit ;
- la **lance** : quand elle débite, la réaction du jet retire encore quelques mètres.

Sur le pupitre, la vue de côté affiche les **courbes de l'abaque** (une par charge : 0, 1, 2, 3 personnes ; celle de la charge actuelle en trait plein) et la zone de travail autorisée. Le relevé **Déport** indique actuel / maximum.

À l'approche de la limite, la manœuvre ralentit puis s'arrête (voyant **Abaque** et alarme). Seuls les mouvements qui **rapprochent** la nacelle restent possibles : rentrer l'échelle, la relever, relever le bras. Si quelqu'un monte dans la nacelle alors qu'elle est déjà loin, l'alarme **Surcharge** s'affiche : ramenez la nacelle.

4. Monter par l'échelle

1. Échelle déployée, placez-vous **au pied de l'échelle, sur le porteur**.
2. Visée « **Gravir l'échelle** » ou touche **G**.
3. **Z** pour monter, **S** pour descendre.
4. Arrivé en haut, **Z** fait entrer dans la nacelle, même si elle est verrouillée.

Pour **redescendre depuis la nacelle** : visée « **Redescendre par l'échelle** » (au bord de la nacelle, côté échelle) ou touche **G**. En bas, **S** fait descendre de l'échelle.

☐ En cas de blocage, /mlexit détache le joueur (échelle, poste ou nacelle).

5. Lance canon (nacelle)

Une personne **dans la nacelle** (pas l'opérateur aux commandes) peut prendre la lance :

Touche	Action
J (<code>/mlmonitor</code>)	Prendre / lâcher la lance
Caméra (souris)	Viser : la lance suit la caméra, dans ses butées
Clic gauche maintenu	Arroser

- Une seule personne tient la lance à la fois ; elle est rendue automatiquement si son porteur quitte la nacelle ou prend les commandes de l'échelle.
- Le jet éteint les **feux du jeu** (et ceux des scripts qui s'en servent) et les feux **SmartFires** (London Studios).
- Ouvrir la lance réduit l'abaque : si la nacelle est déjà trop loin, le jet ne s'ouvre pas et l'aide à l'écran l'indique (« nacelle trop loin... ») : rentrez ou relevez l'échelle.

6. Coffres, portes et écran

- **Coffres latéraux** : visée « **Coffres gauche / droite** » ou **E** à côté d'un coffre.
- **Portes du véhicule** : visée « **Verrouiller / déverrouiller** » ou `/mllock` (échelle déployée).

7. Replier l'échelle

1. Ramenez l'échelle en position de route : **rotation à 0°, inclinaison à 0°, longueur rentrée, bras de nacelle relevé.**
2. Tout le monde descend (poste, nacelle, échelle).
3. À l'arrière : visée « **Replier l'échelle** » ou **U** devant l'écran (ou **U** depuis la cabine).

En **calage manuel**, la nacelle se replie, puis vous reprenez le poste : remontez les vérins et rentrez les stabilisateurs, puis **U** pour ranger.

En **calage automatique**, le repli rejoue la mise en station à l'envers (nacelle repliée, porteur reposé sur ses roues, vérins remontés, stabilisateurs rentrés). Les cales sont ramassées avec le repli. Il est refusé tant qu'une condition n'est pas remplie, avec un message qui précise quoi corriger (par exemple « L'échelle n'est pas rangée (rot, ext) »).

4. Au volant, désengagez la prise de force (**K**) avant de rouler.

8. Choisir son interface

`/mlui` ouvre les réglages du pupitre :

- **Interface** : moderne (écran tactile) ou années 2000 (pupitre LCD) ;
- **Position** à l'écran, **taille, mode compact** ;
- **Volume** des sons et **aide aux touches**.

Les réglages sont gardés d'une session à l'autre. Le serveur peut imposer une interface pour certaines échelles.

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 (× `slow`). Au-delà de `stop` (12,5 m/s ≈ 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.

Configuration des échelles

Ce chapitre explique `config/ladders.lua` : la liste des véhicules à échelle reconnus par le script, avec leurs props, leurs butées, leurs extras et leurs places. C'est le chapitre le plus technique, mais c'est lui qui permet d'utiliser MasterLadder avec n'importe quel véhicule à échelle.

Principe

- Une entrée par véhicule. **La clé est le nom de spawn du véhicule** (ex. `['epan']`).
- Tout réglage absent d'une entrée reprend celui de `Config.LadderDefaults` (en bas du fichier). Une entrée peut donc se limiter à son nom, ses props et ce qui diffère.
- **Unités** : angles en degrés, distances en mètres, durées en millisecondes.
- **Repère de rotation** : 0° = échelle vers l'avant (au-dessus de la cabine), sens horaire vu de dessus.

```
Config.Ladders = {
  ['epabspp'] = {
    label = "EPA BSPP",
    props = {
      turntable = "prop_tl2_base",
      D          = "prop_tl2_ladder1",
      C          = "prop_tl2_ladder2",
      B          = "prop_tl2_ladder3",
      A          = "prop_tl2_ladder4",
      S          = "prop_tl2_ladder5",
      basket     = "prop_tl2_cage",
      stab       = "prop_tl2_outriggers",
    },
    interface = { theme = nil, forced = false, model = "ML 32", serial = "N° de série 2004-1187" },
    seat = { angle = 143.24, distance = 1.3, height = 0.24 },
    -- le reste reprend Config.LadderDefaults
  },
}
```

Identité et props

- `label` : nom affiché (pupitre, messages).
- `props` : noms des modèles de l'échelle, streamés par la ressource du véhicule. **Les 8 sont obligatoires** ; une entrée incomplète est ignorée, avec un avertissement dans la console.
 - `turntable` : tourelle
 - `D` : plan fixe
 - `C`, `B`, `A` : plans coulissants
 - `S` : bras de nacelle
 - `basket` : nacelle
 - `stab` : stabilisateur (utilisé 4 fois)

Déploiement

- `deployTime` : durée totale de la mise en station et du repli (ms). `0` = durées réalistes de `Config.Stabilization.timing` (environ 9 s) ; une autre valeur ramène toute la séquence à cette durée.
- `maxSlope` (défaut `8.0`) : pente **et** dévers maximaux du véhicule (°) pour caler. Ils sont corrigés par la mise à niveau.

Butées

- `elevation` (défaut `{ min = -15.0, max = 75.0 }`) : inclinaison de l'échelle (°).
- `length` (défaut `{ min = 0.0, max = 23.0 }`) : longueur sortie (m), en plus de la longueur repliée.
- `arm` (défaut `{ min = -90.0, max = 0.0, floor = -35.0 }`) : bras de nacelle (°). `floor` : le bras ne pointe jamais plus bas que `floor` sous l'horizontale, quelle que soit l'inclinaison de l'échelle. Plus bas que -35° environ, la nacelle heurte le bras.
- `light` (défaut `{ left = -90.0, right = 90.0, down = -85.0, up = 85.0 }`) : débattement du projecteur (°).
- `rotation` : zone interdite à l'arrière du véhicule.
 - `stopRight` / `stopLeft` : butées en tournant à droite / à gauche.
 - `resumeRight` / `resumeLeft` : au-delà, la rotation est de nouveau possible (pour ressortir de la zone).
- `cabZone` (défaut `{ from = 324.0, to = 34.0 }`) : secteur au-dessus de la cabine où l'échelle ne peut pas descendre sous l'horizontale.

Extras du véhicule

- **extras** : numéros des extras du modèle.
 - **ladder** (défaut `{ 5, 10 }`) : extras de l'échelle repliée, masqués quand elle est déployée.
 - **left** / **right** (défaut `{ open = 6, closed = 7 }` / `{ open = 12, closed = 11 }`) : coffres gauche / droite.
 - **screen** (défaut `{ on = 9, off = 8 }`) : écran de commande arrière.
- **spawnExtras** : état du véhicule à sa première apparition.
 - **left**, **right** : "open" ou "closed"
 - **screen** : "on" ou "off"
 - **others** : autres extras, [numéro] = true (visible) ou false (masqué). Ex. `{ [1] = true, [3] = false }`.

☐ Les extras sont appliqués sans réparer le véhicule : les dégâts et les portes ouvertes sont conservés.

Places

- **seat** (défaut `{ angle = 131.78, distance = 1.0, height = -0.28 }`) : position du poste de commande autour de la tourelle (angle en °, distance et hauteur en m).
- **basketPlaces** (défaut `3`) : nombre de places dans la nacelle (1 à 3).
- **basketSpots** : position de chaque place autour du centre de la nacelle (**angle** en °, **distance** en m).

Logements des stabilisateurs

- **stabilizers** (défaut `{ front = 1.5, rear = -2.3, side = 0.02, height = -0.2 }`) : emplacement des stabilisateurs rentrés, dans le repère du véhicule (m). C'est leur position au repos (`Config.StowedStabilizers`) et le point de départ du déploiement.
 - **front** / **rear** : position le long du véhicule (+ vers l'avant, - vers l'arrière)
 - **side** : écart latéral rentré
 - **height** : hauteur (négatif = plus bas)

☐ Réglage en direct : `Config.Debug = true`, approchez-vous du véhicule replié et tapez par exemple `/mlstab front 1.9` (valeur absolue), `/mlstab out +0.02` (sortir un peu plus) ou `/mlstab rear -0.05 rel` (relatif). Les stabilisateurs se replacent immédiatement et la console F8 affiche la ligne `stabilizers = { ... }` à copier dans `config/ladders.lua`.

Abaque

- **abaque** : déport maximal de la nacelle (m), c'est-à-dire la distance horizontale entre l'axe de la tourelle et la nacelle.
 - **enabled** (défaut **true**)
 - **persons** (défaut **{ 23.0, 20.0, 17.0, 14.0 }**) : nacelle vide, 1, 2, 3 personnes, en **calage complet**.
 - **reduced** (défaut **{ 15.0, 13.0, 11.0, 9.0 }**) : les mêmes en **calage réduit**.
 - **monitor** (défaut **3.0**) : mètres retirés quand la lance débite.
 - Calage asymétrique : la courbe **reduced** s'applique **par le travers du côté réduit** ; au-dessus de l'avant et de l'arrière du camion, la courbe **persons** reste valable, avec une transition progressive entre les deux.
 - **reducedWidth** (défaut **1.0**) : sortie latérale des stabilisateurs en calage réduit (m ; calage complet : 2,02 m).

Les valeurs doivent aller du plus grand (nacelle vide) au plus petit (3 personnes). Elles sont dessinées sur le pupitre sous forme de courbes.

Lance canon

- **monitor**
 - **enabled** (défaut **true**) : lance présente sur cette échelle.
 - **offset** (défaut **{ side = 0.0, forward = 0.85, height = 1.05 }**) : position de la lance par rapport au centre de la nacelle (m ; **forward** = vers l'avant de l'échelle).
 - **yaw** (défaut **60.0**) : débattement en degrés de part et d'autre de l'axe de l'échelle.
 - **pitch** (défaut **{ down = -60.0, up = 45.0 }**) : débattement vertical (°).

Collisions des props

- **collisions** (défaut **{ sections = false, basket = true, stabs = true }**) : active les collisions des sections, de la nacelle et des stabilisateurs. À n'activer que si les modèles en possèdent.

Interface

- **interface**

- `theme` : `nil` (choix du joueur), `"modern"` ou `"retro"` (interface suggérée pour cette échelle).
- `forced` : `true` impose `theme` à tous les joueurs pour cette échelle.
- `model`, `serial` : textes de la plaque constructeur du pupitre années 2000.

Ajouter un véhicule

1. Vérifiez que la ressource du véhicule stream bien **le véhicule et les 8 props** de l'échelle.
2. Copiez une entrée existante et renommez la clé avec le **nom de spawn** du véhicule.
3. Renseignez `label` et `props`.
4. Ajustez les **numéros d'extras** du modèle (`extras`) et l'état de spawn (`spawnExtras`).
5. Redémarrez la ressource, déployez, puis ajustez en jeu `seat`, `basketSpots` et les butées.

☐ Avec `Config.Debug = true`, la console signale les props introuvables par leur nom, et `/mldebug` affiche au-dessus de l'échelle son état complet.

Permissions & administration

Ce chapitre explique qui peut faire quoi avec l'échelle, et les outils des administrateurs.

Actions contrôlées

Action	Ce qu'elle autorise
<code>deploy</code>	Déployer et replier l'échelle
<code>enter</code>	Monter au poste, dans la nacelle ou sur l'échelle
<code>control</code>	Manœuvrer l'échelle
<code>lock</code>	Verrouiller les portes du véhicule
<code>locker</code>	Ouvrir les coffres, allumer l'écran arrière

Trois façons de gérer les permissions

1. **Tout le monde** : `Config.JobRestriction = false` (par défaut), ou framework `standalone`.
2. **Par métier et par grade** : `Config.JobRestriction = true` et `Config.AuthorizedJobs` (voir [Configuration générale](#)). Avec QBCore / QBox, `Config.RequireOnDuty = true` exige d'être en service.
3. **Votre propre règle** : `Config.CustomPermission`, une fonction qui reçoit le joueur, l'action et l'échelle, et retourne `false` pour refuser. Elle remplace la restriction par métier.

☐ Toutes les vérifications sont faites **côté serveur** : un joueur ne peut pas les contourner en modifiant son client.

Commande d'administration

Autorisation (dans `server.cfg`) :

```
add_ace group.admin command.mladmin allow
```

Commande	Effet
<code>/mladmin list</code>	Liste les échelles déployées et leurs occupants
<code>/mladmin evict <netId cl> [all op seat b1 b2 b3 c1 c2]</code>	Fait descendre les occupants (tous par défaut) ; <code>op</code> retire l'opérateur
<code>/mladmin del <netId cl></code>	Replie instantanément l'échelle et libère le véhicule

`cl` désigne l'échelle la plus proche de l'administrateur. La commande fonctionne aussi depuis la console serveur (avec un `netId`).

Dépannage joueur

`/mlexit` détache le joueur (échelle, poste ou nacelle) s'il reste bloqué.

En cas d'erreur

Ce chapitre regroupe les problèmes les plus courants et l'outil de diagnostic intégré.

Le mode debug

Passez `Config.Debug = true` dans `config/config.lua` et redémarrez la ressource. MasterLadder écrit alors tout ce qu'il fait :

- **console serveur** : vérification de la licence (requête et réponse), chaque action reçue d'un joueur et son résultat, **chaque refus avec sa raison précise** (distance, phase de l'échelle, place occupée, permission, grade...), chaque changement d'état de l'échelle ;
- **console F8** (client) : synchronisation, places, commandes, montée, target, extras, props manquants, interface, performances.

Exemples de lignes :

```
[ML:deny] mladder:enter ← Jean #12 : pied de l'échelle à 6.2 m (max 5.0 m)
[ML:deny] mladder:set ← Jean #12 : déploiement : pente 9.4° (véhicule 9.1°) > 8.0° max
[ML:deny] mladder:control ← Jean #12 : « control » : pompier grade 0 < 1 requis
[ML:state] EPAN 42 tl (rév 18) : occ.b1: 0 → 12
```

Commandes de diagnostic

- `mldebug` (console serveur) : état complet de chaque échelle (phase, opérateur, places, valeurs, propriétaire réseau). `mldebug 42` pour une seule.
- `/mldebug` (F8) : état côté client et **surimpression 3D** au-dessus des échelles. Retapez la commande pour la masquer.

Pour ne garder qu'une partie du journal : `Config.DebugCategories = { "license", "deny", "perm" }`.

⚠ Remettez `Config.Debug = false` une fois le problème trouvé.

Problèmes fréquents

« **Licence MasterMods refusée : script inactif** »

- La console affiche un rapport : clé envoyée (masquée), d'où elle a été lue, et la raison donnée par le serveur.
- « Clé inconnue » : vérifiez la saisie (copier-coller depuis le portail, sans espace).
- « Aucune clé personnelle trouvée » : la clé n'est ni dans `server.cfg` (`set mastermods_key`) ni dans `Config.MasterModsKey`.
- Clé bloquée : elle a été utilisée sur plus de 3 serveurs différents en 24 h ; elle se débloque seule au bout d'un jour.

« API MasterMods injoignable (code 0) »

- Le serveur ne peut pas joindre `ressources.mastermods.dev` (pare-feu, DNS, hébergeur qui bloque les connexions sortantes). Le script reste actif pendant environ 1 h 30 avant de se couper.

Les props de l'échelle n'apparaissent pas

- La console F8 affiche « Props non chargés ». Vérifiez que la ressource du véhicule est démarrée **avant** MasterLadder et que les noms de `props` dans `config/ladders.lua` sont exacts. En mode debug, les modèles manquants sont listés.

Rien ne se passe en visant le véhicule

- `Config.TargetSystem` ne correspond pas à votre target, ou celui-ci démarre après MasterLadder. La console F8 l'indique ; le script passe alors aux touches.

« Pente trop forte » / déploiement refusé

- Déplacez le véhicule sur un sol plus plat, ou augmentez `maxSlope` de l'échelle.

Impossible de replier

- Le message précise ce qui manque : rotation, inclinaison, longueur ou bras pas en position de route, ou quelqu'un encore à bord.

Le joueur est mal placé sur l'échelle (dans les barreaux, flotte, penché en arrière)

- Ajustez `Config.ClimbOffset` (par pas de `0.05`) et `Config.ClimbTilt` (voir [Configuration générale](#)).

Un joueur est bloqué (attaché, figé)

- Le joueur tape `/mlexit`. Un administrateur peut aussi utiliser `/mladmin evict cl`.

Support

Si le problème persiste, passez sur discord.mastermods.dev avec :

- la version de MasterLadder (dans `fxmanifest.lua`) ;
- les lignes de la console avec `Config.Debug = true` au moment du problème.

⚠ Ne partagez jamais votre MasterKey en entier : les journaux l'affichent toujours masquée.

Journaux des changements

Ce document liste les évolutions de MasterLadder, de la plus récente à la plus ancienne.

1.3.0 — 19/10/2026

Pupitre à joysticks, comme sur une vraie EPA

- Pendant la manœuvre, un pupitre à joysticks s'affiche au-dessus du pupitre. **Clic droit** pour prendre la souris (clic droit ou Échap pour la rendre) ; le clavier reste actif.
- Joystick gauche : rotation et inclinaison ; joystick droit : longueur et bras nacelle.
Proportionnels : plus on pousse, plus ça va vite.
- Boutons cliquables : arrêt d'urgence, vitesse lente / rapide, verrou nacelle, projecteur, caméra, lâcher, croix du projecteur. Interface années 2000 : clavier et arrêt d'urgence cliquables.
- **Calage** : commandes cliquables au-dessus du pupitre de calage (choix du stabilisateur, joystick sortie / vérin, cales, valider, abandonner, quitter).
- Masquable par chaque joueur dans `/mlui`, désactivable par le serveur (`Config.Joysticks`).

Corrigé

- Sol meuble avec dévers compensé : un seul pourcentage d'abaque annoncé, celui réellement appliqué.
- Calage manuel sur sol meuble : le message renvoie au pupitre de calage (une cale sous chaque patin).

1.2.0 — 18/10/2026

- **Guide de formation** joueurs en PDF (20 pages), fourni dans l'archive : calage pas à pas, cas difficiles, manœuvre, sécurité, messages et solutions, fiche réflexe.
- Étude du calage : elle ne compte plus que les cales qui tiennent réellement sous le patin ; message clair quand il n'y a plus de place.

1.0.0 — 18/10/2026

Mise en station comme sur une vraie EPA

- **Calage manuel** au pupitre de l'écran arrière : choisissez chaque stabilisateur, sortez-le, descendez son vérin ; le camion se soulève et s'incline selon les patins posés. Cales sous chaque patin, conseils de manœuvre, niveau à bulle, abandon avec rangement automatique. `mode = "auto"` pour la séquence automatique.
- **Prise de force** à engager au volant (**K**) : l'engin étudie aussitôt sa position et dit si un calage est possible ici (et avec combien de cales), ou ce qui manque.
- **Pentes et rues étroites** : sortie partielle (calage réduit d'un côté), cales sous les patins aval, compensation de dévers de l'échelle jusqu'à 6°.

Manœuvre et sécurité

- **Abaque** (calculateur de charge) avec courbes sur les pupitres.
- **Anémomètre** : vent fort, manœuvre ralentie ; vent trop fort, on ne peut plus que ramener l'échelle.
- **Anticollision** de la nacelle (façades, poteaux, câbles).
- **Nacelle auto-nivelante**, tourelle solidaire du châssis.
- **Lance canon** depuis la nacelle (**J**), compatible feux du jeu et SmartFires.

Ambiance et intégration

- **Sons de manœuvre** réalistes entendus autour de l'engin (pompe hydraulique, moteur, bips).
- **Exports et événements** pour les autres scripts : voir [Exports et événements](#).
- **Espagnol** et **allemand**.

0.1.0 — 18/10/2026

Première version publique.

- Échelle EPAN / EPA synchronisée entre tous les joueurs, serveur autoritaire, mouvements lissés.
- Deux pupitres au choix (`/mlui`) : écran tactile moderne et pupitre années 2000.
- Stabilisateurs rentrés visibles sur le véhicule dès son apparition, déployés avec l'échelle.
- Nacelle 3 places, montée à pied par l'échelle et redescende depuis la nacelle.
- Interactions `ox_target` / `qb-target` ou touches.
- Licence MasterKey gratuite, vérifiée au démarrage puis toutes les 30 minutes.
- Mode debug complet et commandes `/mldebug`, `/mlstab`.

Exports et événements

MasterLadder expose des **exports** et des **événements** pour que vos autres scripts (dispatch, HUD, scripts de feu, métiers...) sachent ce qui se passe sur les échelles. Le nom de la ressource doit rester `MasterLadder`.

Unités : angles en **degrés**, distances en **mètres**. Une échelle est identifiée par le **netId** de son véhicule.

Places

Code	Place
<code>seat</code>	Poste de commande (tourelle)
<code>b1</code> , <code>b2</code> , <code>b3</code>	Places de la nacelle
<code>c1</code> , <code>c2</code>	Sur l'échelle (montée à pied)

État d'une échelle (`GetLadderState`)

Les deux côtés renvoient la même table (ou `nil` si l'échelle n'est pas déployée) :

Champ	Contenu
<code>netId</code> , <code>model</code> , <code>label</code>	Véhicule, clé dans <code>Config.Ladders</code> , nom affiché
<code>phase</code>	calage automatique : <code>"deploy"</code> (calage), <code>"ready"</code> (manœuvrable), <code>"pack"</code> (repli) · calage manuel : <code>"setup"</code> (stabilisateurs au pupitre), <code>"unfold"</code> (dépliage), <code>"ready"</code> , <code>"fold"</code> (repli de la nacelle), <code>"abort"</code> (abandon, rangement automatique)
<code>calage</code>	<code>"full"</code> , <code>"reduced"</code> , <code>"left"</code> (gauche réduit) ou <code>"right"</code> (droit réduit)
<code>rotation</code> , <code>elevation</code> , <code>length</code> , <code>arm</code>	Position de l'échelle (°, °, m de longueur totale, °)
<code>basket</code>	Position de la nacelle (<code>vector3</code>)
<code>reach</code> , <code>maxReach</code>	Déport actuel et déport maximal de l'abaque (m)
<code>load</code>	Personnes comptées dans la charge
<code>occupants</code>	<code>{ seat = id, b1 = id, ... }</code> (0 = libre)

Champ	Contenu
<code>operator</code>	ID serveur de l'opérateur aux commandes (0 = personne)
<code>basketLocked</code> , <code>spotlight</code>	Verrou de nacelle, projecteur
<code>monitor</code>	<code>{ holder, flowing, yaw, pitch }</code> (+ <code>impact</code> côté client)

Côté client s'ajoutent `vehicle` (entité) et `height` (hauteur de la nacelle au-dessus du sol).

Exports client

```
local ml = exports['MasterLadder']
```

Export	Retour
<code>ml:GetPlace()</code>	<code>netId</code> , <code>place</code> du joueur, ou <code>nil</code>
<code>ml:IsInLadder()</code>	Le joueur occupe une place
<code>ml:IsInBasket()</code> / <code>ml:IsClimbing()</code> / <code>ml:IsAtSeat()</code>	Nacelle / sur l'échelle / au poste
<code>ml:IsOperator()</code>	Le joueur a les commandes
<code>ml:IsHoldingMonitor()</code>	Le joueur tient la lance
<code>ml:IsLadderVehicle(veh)</code>	Ce véhicule est une échelle configurée
<code>ml:IsDeployed(veh)</code>	Cette échelle est déployée
<code>ml:GetLadderState(netId)</code>	Table d'état (échelles affichées chez ce joueur)
<code>ml:GetBasketCoords(netId)</code>	Position de la nacelle
<code>ml:IsWaterFlowing(netId)</code>	La lance débite
<code>ml:GetWaterImpact(netId)</code>	Point d'impact du jet (<code>vector3</code>)
<code>ml:GetNearestLadder(range)</code>	<code>netId</code> , <code>distance</code> de l'échelle déployée la plus proche (défaut 50 m)
<code>ml:GetVisibleLadders()</code>	Liste des <code>netId</code> des échelles affichées

Exemple : couper un HUD de vitesse dans la nacelle.

```
CreateThread(function()
    while true do
        Wait(500)
        local inBasket = exports['MasterLadder']:IsInBasket()
        -- masquer / afficher votre HUD ici
    end
end
```

```
end)
```

Exports serveur

Export	Retour
<code>IsDeployed(netId)</code>	L'échelle est déployée
<code>GetLadderState(netId)</code>	Table d'état
<code>GetDeployedLadders()</code>	Liste des netId déployés
<code>GetBasketCoords(netId)</code>	Position de la nacelle
<code>GetPlayerPlace(src)</code>	<code>netId</code> , <code>place</code> du joueur, ou <code>nil</code>
<code>GetOccupants(netId)</code>	<code>{ place = src }</code> des places occupées
<code>IsWaterFlowing(netId)</code>	La lance débite
<code>EjectPlayer(src)</code>	Fait descendre le joueur de sa place (<code>true</code> si fait)
<code>EvictLadder(netId)</code>	Vide toutes les places et rend les commandes
<code>StopWater(netId)</code>	Ferme le jet (le porteur peut le rouvrir)

Événements serveur

À écouter avec `AddEventHandler` dans une ressource serveur :

Événement	Paramètres	Quand
<code>masterladder:deployed</code>	<code>netId</code> , <code>src</code> , <code>calage</code>	Déploiement lancé (<code>"full"</code> / <code>"reduced"</code> ; <code>"manual"</code> au début d'un calage manuel)
<code>masterladder:ready</code>	<code>netId</code>	Calage terminé
<code>masterladder:packing</code>	<code>netId</code> , <code>src</code>	Repli lancé
<code>masterladder:packed</code>	<code>netId</code>	Échelle rangée
<code>masterladder:ptoChanged</code>	<code>netId</code> , <code>src</code> , <code>engagée</code>	Prise de force engagée / désengagée
<code>masterladder:placeChanged</code>	<code>netId</code> , <code>src</code> , <code>place</code>	Place prise, ou quittée (<code>place = nil</code>)
<code>masterladder:controlChanged</code>	<code>netId</code> , <code>src</code>	Opérateur aux commandes (0 = personne)
<code>masterladder:monitorChanged</code>	<code>netId</code> , <code>src</code>	Porteur de la lance (0 = personne)
<code>masterladder:waterChanged</code>	<code>netId</code> , <code>src</code> , <code>ouvert</code>	Jet ouvert / fermé

Événement	Paramètres	Quand
<code>masterladder:waterImpact</code>	<code>netId, src, coords</code>	Impact du jet, 2 fois par seconde

Brancher un script de feu

```
-- server.lua de votre script de feu
AddEventHandler('masterladder:waterImpact', function(netId, src, coords)
    -- éteindre / affaiblir vos feux autour de coords (rayon ~3 m)
    MonScriptDeFeu.DamageAt(coords, 3.0, 25.0, src)
end)
```

Événements client

Événement	Paramètres	Quand
<code>masterladder:client:placeChanged</code>	<code>netId, place</code>	Ma place change (<code>nil, nil</code> quand je descends)
<code>masterladder:client:waterImpact</code>	<code>coords, rayon</code>	Impact du jet près de moi (feux éteints localement)