Généralités & important

MasterStretcher est une solution médicale avancée pour FiveM qui remplace les brancards statiques par des entités physiques réalistes. Le script gère la gravité dans les pentes, offre une synchronisation réseau parfaite (OneSync), permet l'ancrage automatique dans les ambulances configurées et inclut un module unique pour les nacelles de grandes échelles.

- Présentation du projet
- Installation
- Guide d'utilisation
- Configuration Détaillée
- Configuration des Véhicules (Ambulances/Paniers)
- Journaux des changements
- Exports et événements

Présentation du projet

MasterStretcher remplace les brancards « statiques » par de vrais brancards physiques pour FiveM : on les pousse, ils suivent le terrain, se rangent dans les ambulances configurées et se synchronisent entre tous les joueurs. Le script est **gratuit**, distribué par MasterMods.

> Version documentée : **3.6.4**. Le détail des changements est dans Journaux des changements.

Fonctionnalités

- **Brancard physique** : il roule, s'incline en pente, bute sur les obstacles et ouvre les portes devant lui. Un brancard sans frein dévale une pente.
- **Auto-armement** : chaque ambulance configurée embarque automatiquement son brancard dès qu'un joueur monte à bord.
- **Rangement dans les véhicules** : le brancard se replie et se fixe à l'arrière, patient compris, puis ressort directement en main.
- **Patients** : un joueur s'installe lui-même ou accepte une invitation (10 s pour répondre) ; un PNJ (valide, blessé au sol ou mort) est installé directement, puis peut être confié à l'hôpital. Sept positions : sur le dos, mort, assis à droite / à gauche / de face, RCP, PLS.
- **Relais** : passer le brancard à un collègue sans le poser, avec accord des deux joueurs.
- **Grande échelle** : fixation du brancard dans la nacelle d'une échelle (EPAN).
- **Accessoires** : dossier, garde-corps et matériel selon le modèle (tablette médicale, brancard cuillère, oxygène, plan dur, couverture de survie, MID).
- **Son des roues** spatialisé, entendu par les joueurs proches.
- **Outils de calibration** (licence Premium) : `/configBrancard` pour placer le brancard dans un véhicule, `/configStretcher` pour calibrer un nouveau modèle.

Compatibilité

Domaine	Pris en charge
Framework	ESX, QBCore, QBox, ox_core, standalone
Interaction	ox_target (recommandé), qb-target, ou clavier seul

Domaine	Pris en charge
Inventaire	inventaire natif du framework, ou ox_inventory
Notifications	ox_lib, ESX, QBCore / QBox, okokNotify, natives GTA
Langues	français, anglais, allemand, espagnol (repli automatique sur le français)

Sécurité et stabilité

Le serveur a le dernier mot sur toutes les actions sensibles : création des brancards, permissions, distances, relais, invitations, positions du patient. Un client modifié ne peut ni faire apparaître de brancards à volonté, ni déplacer ou supprimer une entité qui ne lui appartient pas. L'état des brancards est écrit uniquement par le serveur, ce qui rend le script compatible avec `sv_stateBagStrictMode`.

Le script nettoie aussi derrière lui : brancards abandonnés, brancards des ambulances supprimées, places libérées à la déconnexion.

Guide pour les joueurs

L'archive contient **`Formation-Brancard-MasterStretcher.pdf`**, un guide de 13 pages à destination des joueurs (sortir, pousser, installer un patient, ranger, relais, messages et solutions, fiche réflexe). Il est aussi joint à chaque release. Le résumé en ligne est dans Guide d'utilisation.

Liens

- Discord de support : https://discord.mastermods.dev
- Site MasterMods : https://mastermods.dev
- Créer sa MasterKey : https://ressources.mastermods.dev/CreerMasterKey.php

Installation

Prérequis

Élément	Détail
Compte Cfx.re	Pour récupérer la ressource dans Keymaster
Artifacts FiveM	Version récente (la dernière version *recommandée*) ; Lua 5.4 requis
OneSync	**Obligatoire** (les brancards sont créés et synchronisés par le serveur)
ox_lib	**Obligatoire**, démarré avant MasterStretcher
ox_target ou qb-target	Recommandé (sinon : interactions au clavier, `Config.TargetSystem = "none"`)
ox_inventory	Facultatif ; version récente si vous l'utilisez (voir Configuration Détaillée)

1. Récupérer la ressource

1. Sur la boutique Tebex MasterMods, ajoutez le paquet gratuit **MasterStretcher** et validez la commande à 0,00 €. Elle est liée à votre compte Cfx.re.
2. Sur Keymaster, onglet **Granted Assets**, téléchargez l'archive **MasterStretcher**.

L'archive contient :

- le dossier `MasterStretcher/` : la ressource ;
- `Formation-Brancard-MasterStretcher.pdf` : le guide joueurs, à partager à vos secouristes (il ne va pas dans `resources`).

2. Installer sur le serveur

1. Copiez le dossier `MasterStretcher` dans `resources/`. Gardez exactement ce nom : il sert à l'export ox_inventory (`MasterStretcher.stretchermod_use_item`).
2. Dans `server.cfg`, démarrez-le **après** ses dépendances :

```
ensure ox_lib
ensure ox_target        # ou qb-target
ensure ox_inventory     # si utilisé
ensure MasterStretcher
```

3. MasterKey (facultative)

Depuis la 3.6.0, le script fonctionne **sans MasterKey** : le serveur est reconnu automatiquement par son adresse cfx.re. Au démarrage, la console affiche `[MASTERMODS] Serveur identifié : xxxx.users.cfx.re`.

La MasterKey rattache le serveur à votre compte MasterMods (portail, support, outils Premium). Pour l'ajouter :

1. Créez votre MasterKey gratuite : https://ressources.mastermods.dev/CreerMasterKey.php
2. **Recommandé** : ajoutez-la dans `server.cfg` :

```
set mastermods_key "MM-XXXXX"
```

Utilisez bien `set` (et non `setr`) : la clé reste côté serveur. Elle est prioritaire sur la config.

Autre possibilité : `Config.MasterModsKey = "MM-XXXXX"` dans `config/config.lua`. Ce fichier est téléchargé par les joueurs ; `server.cfg` est donc préférable.

Au démarrage, la console doit afficher : `[MASTERMODS] Licence validée pour : <nom du serveur>`. Si elle affiche « Aucune MasterKey (facultative) » alors que vous en avez mis une, la clé n'est pas lue (voir Dépannage - F.A.Q).

4. Adapter la configuration

Les fichiers de `config/` restent modifiables (ils ne sont pas chiffrés) :

Fichier	Contenu	Page
`config/config.lua`	Licence, langue, framework, permissions, gameplay, son, items, sécurité	Configuration Détaillée
`config/stretchers.lua`	Modèles de brancards, items, accessoires, positions du patient	Configuration Détaillée

Fichier	Contenu	Page
`config/vehicles.lua`	Ambulances et nacelles	Configuration des Véhicules

Mettre à jour

1. Sauvegardez votre dossier `config/`.
2. Remplacez le dossier `MasterStretcher` par celui de la nouvelle archive.
3. Remettez votre `config/` et comparez-le avec la nouvelle version : les nouvelles options sont listées dans les Journaux des changements. Une option absente de votre ancienne config prend sa valeur par défaut.
4. Redémarrez le serveur (ou `ensure MasterStretcher`).

> **Mise à jour depuis la 3.4.0** : rien n'est obligatoire. Pensez à déplacer votre clé dans `server.cfg` et regardez les deux nouveaux réglages `Config.MaxStretchersPerPlayer` et `Config.AbandonedStretcherMinutes`.

Ne pas modifier

Le script vérifie au démarrage que tous ses fichiers sont présents. S'il en manque un, il refuse de démarrer et l'indique en console : réinstallez alors la ressource depuis une archive propre.

Guide d'utilisation

Toutes les actions passent par l'**œil d'interaction** (`Alt` maintenu, puis viser le brancard, l'ambulance, la nacelle ou la personne). La seule touche directe est celle qui **lâche le brancard** : `E` par défaut, modifiable par chaque joueur dans *Paramètres > Raccourcis > FiveM > Lâcher le brancard*.

> Le guide complet pour les joueurs, avec illustrations et fiche réflexe, est le PDF **Formation-Brancard-MasterStretcher.pdf** fourni dans l'archive.

Sortir un brancard

Méthode	Comment
Depuis l'ambulance	Ambulance à moins de 5 km/h, portes arrière ouvertes : `Alt` sur l'ambulance → **Sortir le brancard**. Il arrive directement en main.
Depuis l'inventaire	Utiliser l'item brancard. L'item n'est retiré que si le brancard apparaît vraiment.
Commande	`/brancard` (ou `/brancard 2` pour le 2e modèle de `Config.ItemsVeh`).

Chaque ambulance configurée embarque son brancard dès qu'un joueur monte à bord (auto-armement). Un joueur peut avoir au plus `Config.MaxStretchersPerPlayer` brancards en service (2 par défaut) ; ceux des ambulances ne comptent pas.

Pousser et lâcher

- **Prendre** : `Alt` sur le brancard → **Pousser/Lâcher**. Le personnage range son arme, les freins se desserrent.
- **Pendant la poussée** : marche ralentie, pas de course, saut, accroupissement ni tir. Un obstacle devant bloque l'avance ; une porte devant le brancard s'ouvre toute seule. En pente, le brancard s'incline.
- **Lâcher** : touche `E`, puis au choix :

Choix	Usage	Freins

Debout (pieds sortis)	Pause pendant le transport	Toujours serrés
Couché (au sol)	Installer un patient, soins, fin d'intervention	Serrés par défaut (`Config.AutoBrakeOnRelease`)

- **Freins** : sur un brancard couché, `Alt` → **Bloquer/Débloquer Freins**. Un brancard sans frein peut rouler dans une pente.

Installer le patient

Ouvrir le menu : `Alt` sur le brancard → **Menu Gestion**.

Qui	Comment
Soi-même (tout le monde)	**S'installer** → position. La place est réservée par le serveur avant de s'allonger. **Se lever** pour descendre.
Un autre joueur	Brancard libre à moins de 4 m du blessé : `Alt` sur lui → **Inviter la personne à aller sur le brancard** → position. Il a 10 s pour accepter.
Un PNJ	Brancard libre à moins de 4 m : `Alt` sur le PNJ → **Mettre sur le brancard** → position. Fonctionne aussi sur un PNJ mort ou blessé au sol (positions allongées uniquement). **Sortir le patient** pour le descendre : un PNJ ramassé mort retombe au sol.

Un secouriste peut changer la position d'un patient : **Position Patient (Joueur)** ou **Position Patient (PNJ)**.

À l'hôpital, **Confier le patient à l'hôpital** (PNJ uniquement) : le personnel le prend en charge et le brancard est libéré. L'option n'apparaît que dans les zones hôpital configurées.

Réglages et accessoires

Menu Gestion → Accessoires & Réglages : dossier (debout / couché), garde-corps (levés / baissés) et matériel selon le modèle. Tous les joueurs voient les mêmes réglages.

L'ambulance

Action	Conditions

Rentrer dans l'ambulance	En poussant le brancard, portes requises ouvertes, à moins de 6 m. Le patient peut rester allongé.
Sortir le brancard	Ambulance arrêtée, portes ouvertes. Fonctionne même si quelqu'un est resté assis dans l'ambulance.
Menu Intérieur	Brancard rangé : même menu que *Menu Gestion* (patient, réglages) pendant le transport.

L'option de rangement reste toujours proposée : si c'est impossible (ambulance pleine, portes fermées), un message explique pourquoi. Une ambulance ne contient qu'un brancard.

Relais entre secouristes

`Alt` sur un collègue :

- **Donner le brancard** : vous poussez, lui non.
- **Prendre le relais** : il pousse, vous non.

L'autre joueur doit accepter dans les 15 secondes ; les deux doivent rester proches du brancard.

Nacelle de grande échelle

`Alt` sur la nacelle en poussant : **Fixer à la nacelle**. Pour le récupérer : **Sortir le brancard** (sur la nacelle, en main) ou **Détacher de la nacelle** (sur le brancard, reste sur place). **Menu Intérieur** donne accès au patient.

Supprimer un brancard

Brancard **couché et freiné** : `Alt` → **Supprimer le brancard** (selon le grade). Un patient encore allongé est relevé automatiquement.

Le serveur nettoie aussi seul : brancard vide d'un joueur déconnecté, brancard rangé dans une ambulance supprimée, brancard oublié au sol depuis `Config.AbandonedStretcherMinutes` (30 min par défaut, aucun joueur à moins de 50 m).

Touches

Touche	Action

`Alt` maintenu	Œil d'interaction (ox_target / qb-target)
`E`	Lâcher le brancard (choix Debout / Couché)

Sans système d'interaction (`Config.TargetSystem = "none"`), la touche `E` sert aussi à prendre le brancard le plus proche.

Configuration Détaillée

Fichiers modifiables : `config/config.lua` (réglages généraux) et `config/stretchers.lua` (modèles de brancards). Les véhicules sont décrits dans Configuration des Véhicules. Après une modification : `ensure MasterStretcher` (ou redémarrage).

1. MasterKey et langue

Option	Défaut	Rôle
`set mastermods_key "MM-XXXXX"` (server.cfg)	—	**Facultative** depuis la 3.6.0 (sans clé, le serveur est reconnu par son adresse cfx.re). Rattache le serveur à votre compte MasterMods. Prioritaire sur la config et jamais envoyée aux joueurs.
`Config.MasterModsKey`	`"VOTRE_MASTERKEY_ICI"`	Clé si elle n'est pas dans `server.cfg`.
`Config.Locale`	`"fr"`	`fr`, `en`, `de`, `es` (`en-us`, `en-gb`… acceptés). Une langue ou un texte manquant retombe sur le français.

2. Framework, interaction, notifications

Option	Défaut	Valeurs
`Config.FrameworkMetier`	`"standalone"`	`"esx"`, `"qb"`, `"qbx"`, `"ox"` (ox_core), `"standalone"`
`Config.InventorySystem`	`"native"`	`"native"` (items du framework : ESX, QBCore, QBox) ou `"ox"` (ox_inventory). Indépendant du framework.
`Config.TargetSystem`	`"ox"`	`"ox"` (ox_target), `"qb"` (qb-target), `"none"` (clavier seul)
`Config.NotifySystem`	`"ox"`	`"ox"`, `"esx"`, `"qb"`, `"qbx"`, `"okok"`, `"gta"`

3. Permissions par métier

Option	Défaut	Rôle
`Config.JobRestriction`	`false`	`true` : seules les personnes listées dans `Config.AuthorizedJobs` peuvent agir. Sans effet en `standalone`.
`Config.RequireOnDuty`	`true`	Avec `JobRestriction` : il faut aussi être **en service** (QB / QBX, ESX Legacy récent). Sans effet si le framework ne gère pas le service. Absente d'une config d'avant la 3.6.4 : service non exigé.
`Config.AuthorizedJobs`	`ambulance`, `pompier`	Grade minimum par action et par métier.

```
Config.AuthorizedJobs = {
    ['ambulance'] = { push = 0, patient = 0, stow = 1, spawn = 2, extras = 2, delete = 3 },
    ['pompier']   = { push = 1, patient = 1, stow = 1, spawn = 2, extras = 2, delete = 4 },
}
```

Action	Ce qu'elle autorise
`push`	Pousser, freins, relais, installer ou repositionner un patient (joueur ou PNJ), inviter un joueur
`patient`	Réservé (non utilisé actuellement)
`stow`	Ranger / sortir un brancard d'un véhicule ou d'une nacelle
`spawn`	Sortir un brancard depuis l'item ou `/brancard`
`extras`	Accessoires et réglages, menu intérieur
`delete`	Supprimer un brancard

Un métier absent du tableau, ou une action absente pour ce métier, est refusé. **S'allonger soi-même et se lever restent ouverts à tout le monde.** Le contrôle est fait côté serveur : les options sont aussi masquées côté client par confort.

4. Gameplay et physique

Option	Défaut	Rôle
`Config.DefaultKey`	`'E'`	Touche par défaut si `Config.TargetSystem = "none"` (prendre et lâcher)

Option	Défaut	Rôle
`Config.ReleaseKey`	`'E'`	Touche par défaut pour lâcher avec ox_target / qb-target. Chaque joueur peut la changer dans ses raccourcis FiveM.
`Config.EnableSlopePhysics`	`true`	Suivi de pente ; un brancard sans frein dévale
`Config.EnableSlopePitch` / `EnableSlopeRoll`	`true`	Inclinaison avant/arrière et latérale
`Config.SlowDownWhenPushing`	`true`	Marche ralentie en poussant
`Config.TerrainPhysics`	`false`	Expérimental : encore plus lent sur herbe, terre, sable, neige
`Config.EnableAutoArming`	`true`	Brancard automatique dans chaque ambulance configurée
`Config.AutoBrakeOnRelease`	`true`	Freins serrés en posant couché (debout : toujours serrés)
`Config.EnablePickupAnimation`	`false`	Expérimental : le joueur marche jusqu'au brancard et s'accroupit pour le prendre
`Config.HolsterWeaponBeforePush`	`true`	Range l'arme avant de pousser

`Config.PhysicsRefreshRate` est présent dans la config mais n'est plus utilisé.

5. Audio

Option	Défaut	Rôle
`Config.EnableWheelSound`	`true`	Son des roues
`Config.WheelSoundFile`	`"wheels.mp3"`	Fichier dans `html/sounds/`
`Config.WheelSoundVolume`	`0.2`	Volume maximum (0.0 à 1.0)
`Config.WheelFadeTime`	`500`	Fondu d'arrêt (ms)
`Config.WheelSoundDistance`	`15.0`	Distance d'écoute (m)

6. Items, limites et nettoyage

Option	Défaut	Rôle

`Config.RemoveItemOnUse`	`true`	Inventaire natif : retire l'item **après** l'apparition réussie du brancard
`Config.MaxStretchersPerPlayer`	`2`	Brancards en service par joueur (item et `/brancard`), `0` = illimité. Les brancards auto-armés ne comptent pas.
`Config.AbandonedStretcherMinutes`	`30`	Supprime un brancard au sol, vide, sans joueur à moins de 50 m depuis ce délai. `0` = jamais.

Inventaire natif (ESX, QBCore, QBox)

Les items de `Config.ItemsVeh` doivent exister dans votre base / vos items. Sous QBox, préférez ox_inventory.

ox_inventory

Avec `Config.InventorySystem = "ox"`, déclarez chaque item dans `ox_inventory/data/items.lua` :

```
['stretcher'] = {
    label = 'Brancard',
    weight = 5000,
    consume = 1,
    client = { export = 'MasterStretcher.stretchermod_use_item' },
},
```

- Remplacez `MasterStretcher` par le nom exact du dossier de la ressource.
- Le retrait est géré par `consume` (pas par `Config.RemoveItemOnUse`).
- Le serveur valide l'usage par le hook `usingItem` d'ox_inventory : l'item n'est pas consommé si le joueur n'a pas la permission ou a atteint sa limite. Avec une version d'ox_inventory trop ancienne, la console affiche « mode dégradé » : mettez ox_inventory à jour.

PNJ (`Config.NPC`)

Option	Défaut	Rôle
`AllowDown`	`true`	Autorise les PNJ morts ou blessés au sol
`DownPositions`	dos, mort, RCP, PLS	Positions proposées pour un PNJ mort ou blessé (labels des animations)

Option	Défaut	Rôle
`KeepDeadOnRemove`	`true`	Un PNJ ramassé mort redevient un cadavre quand on le sort du brancard
`HospitalDelivery`	`true`	Active *Confier le patient à l'hôpital*
`HospitalZones`	5 hôpitaux d'origine	`{ label, coords = vector3(...), radius }` : à adapter à vos MLO

Section facultative : un `config.lua` sans `Config.NPC` utilise ces valeurs par défaut.

7. Sécurité réseau

Option	Défaut	Rôle
`Config.MaxInteractionDistance`	`6.0`	Distance maximale (m) entre un joueur et le brancard / véhicule qu'il manipule. Refus serveur au-delà.
`Config.InviteToStretcherDistance`	`4.0`	Distance maximale (m) entre le blessé et le brancard pour l'inviter
`Config.Debug`	`false`	Messages de diagnostic en console (F8 et serveur). À laisser à `false` en production.

8. Modèles de brancards (`config/stretchers.lua`)

Items et commande

```
Config.ItemsVeh = {
    { hash = `ferno-f2`,   item = 'stretcher',  label = 'Ferno F2' },
    { hash = `fernof3`,    item = 'stretcher2', label = 'Ferno F3' },
    { hash = `stryker_pp`, item = 'stryk',      label = 'Apex Lifter X1' },
}
```

L'ordre donne le numéro de `/brancard <n>`. Seuls les modèles listés ici peuvent apparaître par item ou commande.

Un modèle de brancard

Un modèle doit figurer **à la fois** dans `Config.Brancards` (réglages) et dans `Config.Lits` (positions du patient) :

```
Config.Brancards[`ferno-f2`] = {
    label            = "Ferno F2",
    extrasPieds      = { haut = 1, bas = 2 },   -- pieds dépliés / repliés
    extrasDossier    = { haut = 3, bas = 4 },
    extrasGardeCorps = { haut = 6, bas = 5 },
    defaultExtras    = { 4, 5 },               -- extras visibles à l'apparition (jamais les
pieds)
    attach = {                                 -- position tenue en main
        pos = { x = -0.05, y = 1.43, z = -0.34 },
        rot = { x = 180.0, y = 180.0, z = 180.0 },
        pivotLength = 0.9,                     -- compensation en pente
    },
    groundOffset = { haut = 0.0, bas = 0.0 },  -- facultatif : seul `bas` (posé couché) est
utilisé
    extras = {                                 -- accessoires du menu
        { id = 7, label = "extra_tablet" },
        { id = 9, label = "extra_oxygen" },
    },
}

Config.Lits = {
    { lit = `ferno-f2`, anims = Config.AnimsFernoF2 },
}
```

- Les `label` des extras sont des clés de langue (`extra_tablet`, `extra_scoop`, `extra_oxygen`, `extra_backboard`, `extra_blanket`, `extra_mid`) ou un texte libre.
- Les positions du patient (`Config.AnimsFernoF2`…) donnent pour chaque animation `dict`, `anim` et le décalage `x`, `y`, `z`, `r` par rapport au brancard.
- `/configStretcher <modèle>` (licence Premium) calibre `attach` et `groundOffset` en jeu et copie le résultat dans le presse-papiers.

9. Commandes

Commande	Qui	Rôle

`/brancard [n]`	Joueurs avec la permission `spawn`	Sortir le brancard n°n de `Config.ItemsVeh`
`/configBrancard`	Licence Premium	Placer le brancard dans le véhicule le plus proche, copie la config
`/configStretcher <modèle>`	Licence Premium	Calibrer la prise en main et la hauteur au sol d'un modèle
`/brancardDoors`	`Config.Debug = true`	Affiche l'état des portes du véhicule proche (trouver `requiredDoors`)
`brancardHeartbeat`	Console serveur, `Config.Debug = true`	Force l'envoi licence / télémétrie / carte

Configuration des Véhicules (Ambulances/Paniers)

Fichier : `config/vehicles.lua`. Un brancard ne peut être rangé que dans un véhicule déclaré ici.

1. Structure d'une ambulance

```
Config.AmbulanceConfig = {
    {
        model = "Master III 2017",          -- nom affiché (informatif)
        hash = `rmaster317_vsav-bspp`,      -- nom de spawn, entre accents graves
        auteur = "Anto",                    -- crédit (informatif)
        AutoBrancardName = "ferno-f2",      -- brancard créé par l'auto-armement
        requiredDoors = {2, 3},             -- portes à ouvrir pour ranger / sortir
        detection = 2.4,                    -- obligatoire, sans effet (voir plus bas)
        alignementLateral = -0.25,          -- X : gauche (-) / droite (+)
        profondeur = -1.25,                 -- Y : arrière (-) / avant (+)
        hauteur = -0.25,                    -- Z : bas (-) / haut (+)
        rotation = 0.0,                     -- 0.0 = pieds vers l'avant
        offsetsByModel = {                  -- position propre à chaque modèle de brancard
            [`ferno-f2`]   = { alignementLateral = -0.25, profondeur = -1.25, hauteur = -0.25,
rotation = 0.0 },
            [`stryker_pp`] = { alignementLateral = -0.35, profondeur = -1.60, hauteur = 0.95,
rotation = 0.0 },
        },
    },
}
```

Champ	Obligatoire	Rôle
`model`	oui	Nom pour vous repérer
`hash`	oui	Modèle du véhicule (`` `nom_de_spawn` ``)
`auteur`	oui	Crédit du modèle

Champ	Obligatoire	Rôle
`AutoBrancardName`	oui	Nom du modèle de brancard auto-armé (doit être dans `Config.Lits`)
`requiredDoors`	oui	Portes qui doivent être ouvertes ; `{}` = aucune
`detection`	oui	Conservé pour compatibilité, **sans effet**
`alignementLateral`, `profondeur`, `hauteur`, `rotation`	oui	Position par défaut du brancard dans le véhicule
`offsetsByModel`	non	Position par modèle de brancard ; prioritaire sur les valeurs par défaut

Une entrée à laquelle il manque un champ obligatoire est **ignorée**, avec un message `ERREUR DANS config/vehicles.lua` en console qui indique le champ manquant.

Distance d'interaction

Les options de l'ambulance apparaissent à moins de `Config.MaxInteractionDistance` (6 m) **du centre du véhicule**. Pour un véhicule très long, augmentez cette valeur dans `config/config.lua` (le champ `detection` n'est plus utilisé).

Plusieurs modèles de brancard

Chaque modèle a ses propres dimensions. Sans entrée dans `offsetsByModel`, un autre modèle que celui calibré sera mal placé : ajoutez une ligne par modèle utilisé dans ce véhicule.

2. Trouver les bonnes valeurs

Avec l'assistant (licence Premium)

1. Placez-vous près du véhicule, portes ouvertes, puis tapez `/configBrancard`.
2. Renseignez le nom de spawn, le nom d'affichage, l'auteur, le brancard automatique et les portes.
3. Déplacez le brancard fantôme : flèches (X/Y), Page préc./suiv. (Z), `Z`/`S` `Q`/`D` `A`/`E` (rotations), `Maj` pour la précision.
4. `Entrée` copie le bloc dans le presse-papiers ; `Retour arrière` annule.

Si le véhicule est déjà configuré, l'assistant ne génère que la ligne `offsetsByModel` du modèle choisi, à coller dans l'entrée existante.

À la main

Ajustez les valeurs par pas de 0,05 puis `ensure MasterStretcher` :

- **X (`alignementLateral`)** : 0.0 = centré, négatif vers le côté conducteur.
- **Y (`profondeur`)** : négatif vers l'arrière. Trop grand : le brancard entre dans les sièges avant.
- **Z (`hauteur`)** : les roues doivent toucher le plancher sans le traverser.

3. Portes requises (`requiredDoors`)

Index	Porte
0	Avant gauche (conducteur)
1	Avant droite (passager)
2	Arrière gauche
3	Arrière droite
4	Capot
5	Coffre / hayon
6	Soute (hélicoptères)

Certains véhicules modifiés ne suivent pas cette numérotation. Avec `Config.Debug = true`, la commande `/brancardDoors` affiche l'état de chaque porte du véhicule le plus proche : ouvrez la porte voulue et relevez son index.

4. Hélicoptères

- Soute arrière : `requiredDoors = {6}`. Sans porte ou toujours ouverte : `requiredDoors = {}`.
- Les patins empêchent souvent de s'approcher : augmentez `Config.MaxInteractionDistance` si l'option n'apparaît pas.

5. Nacelles de grande échelle

```
Config.LadderBasketConfig = {
    {
        label = "Nacelle EPAN",
        hash = -84503767,                    -- modèle de la NACELLE (pas du camion)
        offset = { x = 0.44, y = -0.43, z = 0.15 },
        rot = { x = 90.0, y = 0.0, z = 0.0 },
    },
}
```

Champ	Rôle
`label`	Nom pour vous repérer
`hash`	Modèle de la nacelle elle-même (objet attaché au camion)
`offset`	Position du brancard dans la nacelle
`rot`	Orientation ; une rotation X d'environ 90° le plaque contre la paroi

En jeu : `Alt` sur la nacelle → **Fixer à la nacelle**, **Sortir le brancard** ou **Menu Intérieur** ; `Alt` sur le brancard → **Détacher de la nacelle**. Une nacelle ne porte qu'un brancard.

Journaux des changements

□□ Changelog (Mise à jour du 30/12 - 31/12) - 3.1.0

Nouvelles Fonctionnalités

- **Système PNJ Complet** : Placement de PNJ vivants sur le brancard avec un menu de position dédié pour modifier l'animation après l'installation.
- **Support Nacelle** : Support de l'EPAN d'Anto avec détection automatique et masquage visuel des roues.
- **Alignement Dynamique (Pitch & Roll)** : Le brancard s'incline automatiquement selon l'inclinaison du terrain (avant/arrière et latéral).

Correctifs & Patches

- **Remaniement** : Simplification du fichier de configuration
- **Système de Relais** : Transfert de contrôle fluide entre deux collègues via une demande et une confirmation synchronisées.
- **Persistance d'Animation** : Système de vérification pour empêcher que l'animation du patient ne soit coupée par des scripts tiers.
- **Optimisation des Collisions** : Désactivation des collisions entre le patient et le brancard pendant le transport pour supprimer les mouvements erratiques.
- **Patch Technique** : Correction de la fonction native de débarquement, correction d'un bug où les joueurs-non-métier ne pouvaient pas se coucher sur le brancard.
- **Changement de système** : Plus besoin d'être sur Discord SAUF pour garder les fonctions 'Premium' : Support Nacelle, alignement dynamique, restriction métier
- **Optimisation des Extras** : Regroupement des dossiers et garde-corps en interrupteurs uniques (ON/OFF) avec logique d'exclusion mutuelle pour une gestion simplifiée.

□□ Changelog - MasterStretcher (Mise à jour du 01/01) - 3.1.1

> Attention : changement du fichier config.lua nécessaire !

- **Correction du bug des nacelles** : Le brancard ne *devrait* plus léviter et ne plus boguer lorsqu'il est attaché à une échelle ou un panier. Il reste désormais parfaitement fixé pour tout le monde, même lors des mouvements.

- **Alerte de mise à jour** : Si une nouvelle version du script est disponible, un gros message d'alerte très visible (en rouge et jaune) s'affichera directement dans la console de votre serveur au démarrage.
- **Affichage des véhicules amélioré** : Les véhicules s'affichent désormais par leur **nom** et leur **auteur** (ex: *Master III [Anto]*), au lieu de simples numéros techniques pour une compréhension plus simple de la config.
- **Meilleure stabilité** : Le système de vérification est plus robuste et ne risque plus de bloquer le démarrage, même si vous avez une liste de véhicules configurés extrêmement longue.
- **Gestion des pieds** : Modification afin de pouvoir intégrer plus facilement d'autres brancards.

□□ Changelog - MasterStretcher (Mise à jour du 02/01) - 3.1.4

> Attention : changement du fichier config.lua nécessaire !

□□ Configuration & Identification des Véhicules

- **Labels des Ambulances** : Les véhicules sont désormais identifiés par leurs noms configurés (ex: "Master III 2012") au lieu de simples identifiants techniques.
- **Affichage des Auteurs** : L'auteur du modèle (si renseigné dans la configuration) apparaît désormais entre crochets à côté du nom du véhicule (ex: [Anto]).
- **AutoBrancardName (Liaison automatique)** :
- Ce paramètre définit le modèle de brancard nativement associé à une ambulance.
- Le rapport automatique traduit désormais l'ID technique en label lisible, vous permettant de valider instantanément que la liaison entre votre ambulance et son brancard est opérationnelle.
- **requiredDoors (Gestion des accès)** :
- Ce paramètre liste les indices des portes (ex: 2, 3) qui doivent être impérativement ouvertes pour charger ou décharger un brancard.
- Le rapport technique répertorie ces portes pour chaque véhicule, garantissant un contrôle précis de vos réglages de réalisme.

□□ Changelog - MasterStretcher (Mise à jour du 02/01) - 3.1.5

- **Test** : essais nacelle
- **Rotation** : possibilité de tourner le brancard pour l'attachement dans le véhicule

□□ Changelog - MasterStretcher (Mise à jour du 03/01) - 3.1.6

- Correction d'un bug sur les portes des hélicoptères mal détectées

□□ Changelog - MasterStretcher (Mise à jour du 25/01) - 3.1.7

- Correction d'un bug sur le fait que l'index métier (grade) n'était pas vérifié. Changement de sa logique générale.

□□ Changelog - MasterStretcher (Mise à jour du 25/01) - 3.1.8

- Allègement de code
- Ajout de trois fichiers de langage pour un changement rapide

□□ Changelog - MasterStretcher (Mise à jour du 15/02) - 3.2.0

> Attention : changement du fichier config.lua nécessaire !

- Uniformisation du matériel : Lors de la sortie automatique de l'ambulance, le brancard apparaît avec un équipement propre et défini (via la configuration), supprimant l'apparition d'accessoires aléatoires.
- Nouvelle option de configuration : Ajout d'un paramètre permettant de régler finement la réactivité de la physique selon les besoins du serveur.
- Refonte de l'optimisation : La consommation du script (Resmon) a été drastiquement réduite. Il est désormais ultra-léger (autour de 0.03ms) tout en conservant une physique de pente et d'inclinaison en temps réel.

□□ Changelog - MasterStretcher (Mise à jour du 17/02) - 3.2.1

- Ajout de la langue Espagnole.
- Correction d'un bug où le son de roulement ne pouvait pas être désactivé.

□□ Changelog - MasterStretcher (Mise à jour du 18/03) - 3.2.2

- **Nouvelle API Complète (Exports) :** Intégration totale avec vos menus personnalisés (Context Menu, Radial, etc.) via une nouvelle liste d'exports (pousser, ranger, ouvrir les menus, lire les états).
- **Support "No-Target" :** Possibilité de définir `Config.TargetSystem = "none"` pour désactiver totalement les cibles (ox/qb) sans générer d'erreurs dans la console.
- **Nouveaux Événements (Hooks) :** Ajout d'événements clients et serveurs exploitables pour déclencher vos propres scripts (ex: allumer les gyrophares au déploiement du brancard).
- **Mémorisation du Dossier :** Correction d'un problème agaçant où le dossier du brancard se recouchait tout seul lorsqu'on le posait au sol.
- **Accessoires au Spawn :** Les brancards (qu'ils soient sortis manuellement ou via l'auto-arm de l'ambulance) apparaissent désormais bien avec les accessoires par défaut définis dans votre configuration.
- **Lâcher Réaliste (Ragdoll) :** Si un joueur trébuche, tombe d'un toit ou se fait renverser, il lâche désormais instantanément le brancard, évitant les bugs de collision destructeurs.
- **Blindage des Actions Externes :** Les actions déclenchées via l'API (menus custom) vérifient désormais rigoureusement les permissions des métiers et l'ouverture des portes des ambulances. Fini les vols de brancards par les civils !
- **Alerte de Désynchronisation :** Ajout d'une notification visible pour le joueur s'il tente d'interagir avec un brancard qui ne répond pas au réseau (prévention des bugs de clics dans le vide).

□□ Changelog - MasterStretcher (Mise à jour du 01/04) - 3.2.3

> Attention : changement du fichier config.lua nécessaire !

- **Système UI Caméléon :** Les notifications et textes à l'écran s'adaptent désormais 100% automatiquement à votre serveur (ox, qb, esx, okok ou natif).
- **Synchronisation & Stabilité :** Résolution des animations "dans le vide" après un redémarrage, du spam lors du rangement, et des conflits réseaux sur la gestion des accessoires.

□□ Changelog - MasterStretcher (Mise à jour du 02/04) - 3.2.4

> Attention : changement du fichier config.lua nécessaire !

- **Physique des sols réaliste :** Le script détecte désormais la matière sous vos pieds (sable, boue, neige) et ralentit drastiquement la poussée. Désactivable dans la config.
- **Outil Dev** `/configBrancard` **:** Un assistant visuel 3D directement en jeu pour aligner les brancards dans vos véhicules moddés et copier la configuration générée en un clic.

□□ Changelog - MasterStretcher (Mise à jour du 03/04) - 3.2.5

- **Bouclier Anti-Crash :** Si une erreur ou un oubli se glisse dans le config.lua, le script ne crashera plus en jeu. Il ignore le véhicule défectueux en silence pour les joueurs, et affiche une alerte claire et précise uniquement dans la console du serveur.
- **Sélection des Portes :** L'assistant /configBrancard s'enrichit d'un menu à cases à cocher (multi-select) pour paramétrer les portes requises. Il suffit de cocher "Coffre" ou "Passager" et l'outil génère le code parfait.
- Dépréciation des versions antérieures à 3.2.0 - Mise à jour vers 3.2.5 conseillée

□□ Changelog - MasterStretcher (Mise à jour du 13/05) - 3.2.6

- **Essais :** tentative de correction de désynchronisations.

□□ Changelog - MasterStretcher (Mise à jour du 31/08) - 3.3.4

> Attention : changement du fichier config.lua nécessaire !

- **Framework :** Prise en charge de QBox
- **Compatibilité** `ox_inventory` ajoutée pour l'item brancard, en plus des inventaires natifs

□□ Changelog - MasterStretcher (Mise à jour du 31/08) - 3.3.5 - 3.3.10

- **Le brancard ne spawnait pas avec** `ox_inventory` malgré aucune erreur visible — corrigé
- **Les accessoires du brancard (dossier, garde-corps, options) restaient bloqués** — corrigé.
- **Sur certains serveurs, les accessoires ne pouvaient jamais être changés du tout**, peu importe la situation — corrigé
- Correction d'un cas où la position de spawn du brancard pouvait être rejetée à tort par le serveur.
- Renforcement général de la fiabilité réseau : plusieurs actions (spawn, rangement, freins, patient sur brancard) passent désormais systématiquement par une validation serveur

- **Extras configurables au spawn** : possibilité de définir quels accessoires (dossier, garde-corps, équipements) apparaissent par défaut quand un brancard est créé, via la configuration
- Nombreux messages de diagnostic ajoutés en mode debug pour un support plus rapide en cas de souci.

□□ Changelog - MasterStretcher (Mise à jour du 07/09) 3.3.12

- **Brancard mal positionné après rangement** : dans de rares cas, un brancard rangé dans un véhicule pouvait apparaître visuellement décalé/flottant, alors que le rangement semblait pourtant réussi. Une vérification supplémentaire est désormais faite après chaque rangement, avec une nouvelle tentative automatique si un problème est détecté.
- Renforcement de la fiabilité réseau lors du rangement, pour éviter les cas où l'opération semble aboutir sans l'être réellement.
- Corrections mineures de robustesse interne.

□□ Changelog - MasterStretcher (Mise à jour du 11/09) 3.3.13 - 3.4.0

- **Décalages de véhicules incohérents selon le modèle de brancard** : les offsets de rangement (position dans l'ambulance) étaient calibrés pour un seul modèle de référence et ne convenaient pas forcément aux autres. Chaque véhicule accepte désormais un réglage dédié par modèle de brancard, appliqué aussi bien au rangement manuel qu'à l'auto-armement.
- `permission_denied` **remonté en boucle dans le dashboard** (action=patient) : les options du menu patient (s'installer, changer la position, placer/retirer un PNJ) n'étaient pas filtrées selon la permission du joueur avant affichage — le bouton apparaissait puis se faisait rejeter côté serveur.
- `distance_too_far` **remonté dans le dashboard** (action=stow, jusqu'à 49m constatés) : les options de rangement restaient sélectionnables bien au-delà de la portée autorisée. Un contrôle de distance côté client fait maintenant disparaître le bouton dès que le joueur s'éloigne.
- **Brancard bloqué "occupé" après déconnexion du patient** : si un joueur se déconnectait en étant allongé sur un brancard, celui-ci restait marqué occupé indéfiniment, empêchant quiconque de s'y installer ou de le libérer. Nettoyage automatique ajouté à la déconnexion.
- **Commande** `/configStretcher <modèle>` : calibration complète d'un modèle de brancard en jeu (position portée, position haute et basse au sol), avec animation de poussée réelle jouée pendant le réglage. Génère directement le bloc de configuration à coller.
- **Invitation à s'allonger avec consentement** : un secouriste peut désormais viser un autre joueur et lui proposer de s'allonger sur un brancard à proximité. Le joueur ciblé dispose de 10 secondes pour accepter ou refuser — plus aucun placement forcé d'un vrai joueur sur un brancard.

- **Permissions du menu patient revues** : s'installer soi-même, changer sa propre position et se relever sont désormais libres pour tout le monde. Agir sur un autre joueur ou un PNJ reste réservé aux secouristes.
- Réorganisation de `config.lua` pour plus de clarté (sections Audio et Objets & Inventaire séparées, réglages mal classés remis à leur place).
- Ajout de la configuration pour le stryker en vente ici
- **Nouvelle version du Ferno F2 incluse** : changement de quelques textures (Textures du matelas, Textures du M.I.D. et de la couverture de survie retravaillées pour un meilleur rendu en jeu).

□□ Changelog - MasterStretcher (Mise à jour du 21/10) - 3.4.1

⚠ Aucune modification de config obligatoire. Deux nouveaux réglages facultatifs dans `config/config.lua` (voir plus bas), et la clé peut passer dans `server.cfg`.

Nouveautés

- **Clé dans server.cfg :** `set mastermods_key "MM-XXXXX"` est prioritaire sur `Config.MasterModsKey` et n'est jamais envoyée aux joueurs.
- **Guide de formation joueurs :** `Formation-Brancard-MasterStretcher.pdf` fourni dans l'archive (13 pages, fiche réflexe incluse).
- **Messages clairs :** l'option « Rentrer dans l'ambulance » reste affichée et explique au clic pourquoi c'est impossible (ambulance pleine, portes fermées).
- **Traductions complètes :** tous les textes du script sont traduits en français, anglais, allemand et espagnol ; une langue mal écrite (`en-us`…) ne casse plus rien.
- **Nouveaux réglages :** `Config.MaxStretchersPerPlayer = 2` (brancards en service par joueur) et `Config.AbandonedStretcherMinutes = 30` (nettoyage des brancards oubliés).

Correctifs

- **Rangement à plusieurs joueurs :** l'ambulance n'est plus bloquée « pleine » une fois sur trois. Le brancard auto-armé est fixé par le bon joueur et vérifié par le serveur.
- **Sortir le brancard d'une ambulance occupée :** fonctionne même si quelqu'un est resté assis dedans.
- **Brancard visible chez un seul joueur :** la prise en main exige désormais le contrôle réseau du brancard.
- **Avertissement F8** `GetNetworkObject: no object by ID` **:** corrigé.
- **PNJ sur brancard :** installer et retirer un PNJ fonctionne de nouveau.
- **Brancard figé en l'air** après un rangement refusé : corrigé.
- **Recherche du brancard le plus proche :** ne vise plus une voiture.
- **Son des roues** qui se coupait en pleine poussée : corrigé.
- **Item perdu :** il n'est retiré qu'après l'apparition réussie du brancard, et le grade est vérifié.
- **Deux joueurs sur la même place :** impossible désormais.

- **Brancards orphelins :** supprimés avec leur ambulance ou après 30 minutes d'abandon.

Sécurité

- Le serveur valide toutes les actions sensibles : apparition des brancards (autorisation, distance, limite), relais, invitations, positions du patient, auto-armement (plus de duplication).
- Un joueur malveillant ne peut plus supprimer une entité arbitraire, voler un brancard ou imposer une animation à un autre joueur.
- La désactivation de licence est appliquée côté serveur.

□□ Changelog - MasterStretcher (Mise à jour du 22/10) - 3.4.2

⚠ Aucune modification de config nécessaire.

- **Ambulance qui part dans tous les sens avec un patient à bord :** corrigé. Le brancard rangé et son patient n'entrent plus en collision avec le véhicule.
- **Position du patient vue par les autres joueurs :** plus d'attache approximative en double, seule la position choisie par le patient est utilisée.

□□ Changelog - MasterStretcher (Mise à jour du 22/10) - 3.4.3

⚠ Aucune modification de config nécessaire.

- **Impossible de sortir le brancard après avoir roulé** (ou d'ouvrir le *Menu Intérieur*) : corrigé. Le serveur mesure maintenant la distance jusqu'à l'ambulance qui porte le brancard.

□□ Changelog - MasterStretcher (Mise à jour du 24/10) - 3.5.0

⚠ Aucune modification de config obligatoire. Nouvelle section facultative `Config.NPC` dans `config/config.lua` (zones hôpital à adapter à vos MLO).

Nouveautés

- **PNJ morts et blessés au sol :** ils s'installent sur le brancard (positions allongées). Pendant le transport, le PNJ reste immobile et invulnérable, et n'est plus supprimé par le jeu. Ramassé mort, il redevient un cadavre quand on le sort.
- **Confier le patient à l'hôpital :** dans les zones hôpital, le PNJ est pris en charge et le brancard libéré.

- **Serveurs 100 % PNJ :** exports et événements pour piloter les patients PNJ depuis vos scripts de missions (voir *Exports et événements*).

Correctifs

- **PNJ qui se relève ou tombe du brancard** quand un autre joueur en prend le contrôle réseau : corrigé.
- **qb-target :** l'option « Mettre sur le brancard » sur un PNJ fonctionne.
- **Brancard bloqué « occupé »** par un PNJ supprimé par un autre script : corrigé.
- **« Ouvrez d'abord les portes » alors qu'elles ont été arrachées :** corrigé, une porte arrachée compte comme ouverte.
- **PNJ sur brancard qui réagit aux explosions et coups de feu :** corrigé, il reste dans sa position.
- **Ambulance explosée :** le brancard rangé dedans a brûlé avec elle. Il apparaît calciné et n'est plus utilisable. Un brancard détruit peut toujours être supprimé.

□□ Changelog - MasterStretcher (Mise à jour du 24/10) - 3.5.1

⚠ Aucune modification de config nécessaire.

- **Brancard rangé supprimé par un autre script** (`/dv`, nettoyage des véhicules de txAdmin, garage) : l'ambulance est libérée tout de suite et peut être réarmée, au lieu de rester « pleine » jusqu'à une vérification.

□□ Changelog - MasterStretcher (Mise à jour du 27/10) - 3.5.2

⚠ Aucune modification de config nécessaire.

- **Options du brancard invisibles jusqu'à 30 min après une connexion ou un redémarrage :** corrigé.
- **Ambulance parfois sans brancard après l'auto-armement** (erreur `autoarm_attach_failed`) : le serveur n'attendait pas assez longtemps la confirmation d'un joueur à la connexion lente. Corrigé, avec un essai d'attache de plus.
- **Joueur bloqué en mode « pousser »** quand le brancard est supprimé pendant qu'il le tient : corrigé.
- **Mort ou mis dans un véhicule en poussant :** le brancard est reposé au sol et freiné.
- **Arme rangée avant l'animation de poussée** (bras mal placés) et **nom du collègue** affiché dans la demande de relais.

□□ Changelog - MasterStretcher (Mise à jour du 30/10) - 3.6.0

⚠ Aucune modification de config nécessaire.

- **MasterKey facultative :** le script fonctionne sans clé, le serveur est reconnu automatiquement par son adresse cfx.re. La clé reste utile pour rattacher le serveur à votre compte MasterMods (portail, support, outils Premium).
- **Console plus claire :** l'identité du serveur est affichée au démarrage, avec le lien cfx.re/join.

□□ Changelog - MasterStretcher (Mise à jour du 01/11) - 3.6.1

⚠ Aucune modification de config nécessaire.

- **Ambulance soudain sans brancard après une sortie ratée :** le brancard restait coincé, invisible, dans le véhicule. Il est maintenant remis en place automatiquement.
- **Diagnostic :** les erreurs du script et des joueurs sont remontées à MasterMods même sans le mode debug, pour corriger plus vite les problèmes rencontrés sur vos serveurs.

□□ Changelog - MasterStretcher (Mise à jour du 01/11) - 3.6.2

⚠ Aucune modification de config nécessaire.

- **Suivi des prises en charge :** le portail MasterMods affiche, pour chaque brancard occupé, s'il s'agit d'un joueur ou d'un PNJ, son état, sa position et depuis combien de temps il est installé. Aucune donnée personnelle n'est transmise.

□□ Changelog - MasterStretcher (Mise à jour du 01/11) - 3.6.3

⚠ Aucune modification de config nécessaire.

- **Brancard invisible après une invitation :** inviter un joueur à s'allonger pendant qu'on pousse le brancard le rendait invisible pour ce joueur, et impossible à reprendre. Le brancard est maintenant posé automatiquement (freins serrés) avant l'installation du patient, puis peut être repris normalement.
- **Brancard brûlé avec un patient dessus :** le patient peut maintenant en descendre (et un PNJ peut être retiré), le brancard restant inutilisable pour le reste.
- **Patient allongé :** il ne peut plus prendre le relais ni recevoir d'invitation tant qu'il est sur un brancard.

□□ Changelog - MasterStretcher (Mise à jour du 01/11) - 3.6.4

⚠ Nouvelle option facultative : `Config.RequireOnDuty` (voir Configuration détaillée). Sans elle, rien ne change.

- **Ambulance sans brancard à l'apparition :** sur les machines de joueurs peu puissantes, le brancard automatique pouvait ne jamais apparaître. Corrigé.
- **Service :** avec la restriction métier, le script peut exiger d'être en service (QB, QBX, ESX récent).
- **Redémarrage du script en cours de partie :** les options du brancard ne disparaissent plus pour les métiers autorisés.
- **Fluidité :** les vérifications de permission sont mises en cache.

Exports et événements

Pour intégrer MasterStretcher à vos propres scripts (menu radial, script ambulance, etc.). Remplacez `MasterStretcher` par le nom du dossier de la ressource.

Exports client

Toutes les actions renvoient `true` en cas de succès, ou `false, raison`. Les mêmes contrôles qu'en jeu s'appliquent (licence, permission, distance), et le serveur revalide chaque action.

Export	Paramètres	Retour
`IsPushingStretcher()`	—	`boolean` : le joueur pousse un brancard
`GetAttachedStretcher()`	—	Entité du brancard poussé, ou `nil`
`IsStretcherOccupied(entity?)`	brancard (défaut : le plus proche, 4 m)	`boolean`
`TogglePush(entity?)`	brancard (défaut : le plus proche)	`true, "pushed"` / `true, "dropped"`
`OpenMenu(entity?)`	brancard	Ouvre le *Menu Gestion*
`StowStretcher(ambulance?)`	ambulance (défaut : la plus proche, 5 m)	Range le brancard poussé
`UnstowStretcher(ambulance?)`	ambulance	Sort le brancard rangé
`OpenAmbulanceMenu(ambulance?)`	ambulance	Ouvre le *Menu Intérieur*
`ToggleBrakes(entity?)`	brancard couché	Serre ou desserre les freins
`GetStretcherPatient(entity)`	brancard	`{ type = 'player', serverId }`, `{ type = 'npc', ped, pedNetId, state, position }` ou `nil`

Raisons possibles : `not_authorized`, `no_permission`, `invalid_stretcher`, `holding_stretcher`, `no_stretcher_held`, `no_ambulance_found`, `doors_closed`, `no_stretcher_in_ambulance`.

```
local ok, reason = exports.MasterStretcher:StowStretcher()
if not ok then print('Rangement impossible : ' .. reason) end
```

Événements client

Déclenchés localement (`AddEventHandler`), pour le joueur qui fait l'action :

Événement	Arguments
`stretchermod:onPushStart`	`stretcher`
`stretchermod:onPushStop`	`stretcher`, `position` (`'debout'` ou `'couche'`)
`stretchermod:onStretcherStowed`	`vehicle`, `stretcher`
`stretchermod:onStretcherUnstowed`	`vehicle`, `stretcher`

```
AddEventHandler('stretchermod:onStretcherStowed', function(vehicle, stretcher)
    -- ex. : afficher le moniteur du véhicule
end)
```

PNJ : exports et événements serveur

Pour les serveurs qui génèrent leurs patients PNJ (scripts de missions, ERS, callouts). Tous les identifiants sont des **netId**. Ces exports ne vérifient ni permission ni distance : c'est votre script qui décide.

Export	Paramètres	Retour
`PlaceNPCOnStretcher(pedNetId, stretcherNetId, position?, state?)`	`position` : index ou label (`'anim_dead'`...), défaut : première position autorisée. `state` : `'dead'`, `'injured'` ou `nil`	`true` ou `false, raison`
`RemoveNPCFromStretcher(stretcherNetId, deletePed?)`	`deletePed = true` supprime le PNJ	`boolean`
`GetStretcherPatient(stretcherNetId)`	—	`{ type = 'player', serverId }`, `{ type = 'npc', pedNetId, state, position }` ou `nil`
`GetNPCStretcher(pedNetId)`	—	netId du brancard, ou `nil`

Raisons possibles : `disabled`, `invalid_stretcher`, `invalid_position`, `invalid_ped`, `is_player`, `occupied`, `ped_on_other_stretcher`, `down_not_allowed`, `position_not_allowed`.

Événements serveur (`AddEventHandler`), avec une table `data = { pedNetId, stretcherNetId, state, position, source, reason, zone }` :

Événement	Quand
`masterstretcher:npcPlaced`	PNJ installé sur un brancard

Événement	Quand
`masterstretcher:npcRemoved`	PNJ retiré. `reason` : `'removed'`, `'stretcher_deleted'`, `'despawned'` ou `'api'`
`masterstretcher:npcDelivered`	PNJ confié à l'hôpital (`zone` = label de la zone) ; le PNJ est supprimé

```
-- Le patient de la mission est arrivé à l'hôpital
AddEventHandler('masterstretcher:npcDelivered', function(data)
    print(('PNJ %s confié à %s'):format(data.pedNetId, data.zone))
end)
```

State bags (lecture seule)

L'état est écrit uniquement par le serveur. Vous pouvez le lire :

Clé	Sur	Valeur
`stretcherPos`	brancard	`'haut'`, `'bas'` ou `'basket'`
`stretcherBrakes`	brancard	`boolean`
`attachedPlayer`	brancard	serverId du joueur allongé, ou `nil`
`attachedPedNetId`	brancard	netId du PNJ allongé, ou `nil`
`attachedPedState`	brancard	`'dead'` (ramassé mort), `'injured'` (blessé au sol) ou `nil`
`attachedStretcher`	véhicule	netId du brancard rangé, ou `nil`
`isPushingStretcher`	joueur	netId du brancard poussé, ou `nil`

```
local patient = Entity(stretcher).state.attachedPlayer
```

> Les événements réseau internes (`stretchermod:server:*`, `stretchermod:client:*`) ne font pas partie de l'API : leurs paramètres peuvent changer d'une version à l'autre et le serveur refuse les appels qui ne viennent pas du script.