

Intégrations (Pour les moddeurs)

Ce document liste tous les exports et événements disponibles pour intégrer le StretcherMod avec vos propres scripts (menus radiaux, context menus, dispatch, MDT, etc.).

Si vous utilisez vos propres menus via ces exports, n'oubliez pas de désactiver le système de ciblage intégré en mettant `Config.TargetSystem = "none"` dans votre `config.lua` pour éviter les conflits et les erreurs dans la console !

Les exemples ci-dessous utilisent `MasterStretcher` comme nom de ressource — c'est le nom recommandé après avoir renommé le dossier téléchargé (voir chapitre Installation). Si vous avez gardé un autre nom de dossier, remplacez `MasterStretcher` par celui-ci dans chaque appel `exports[...]`.

☐ EXPORTS CLIENT

Vous pouvez appeler ces fonctions depuis n'importe quel script client.

La plupart de nos exports retournent deux valeurs : `success` (boolean) et `message` (string ou boolean selon le cas) pour vous permettre de gérer vos erreurs proprement (ex: si le joueur n'a pas le bon métier).

Note : Si vous ne précisez pas d'entité en paramètre, le script ciblera automatiquement le brancard ou l'ambulance la plus proche du joueur.

☐ Actions

1. Pousser / Lâcher un brancard

```
local success, action = exports['MasterStretcher']:TogglePush(entity)
-- action retourne "pushed" ou "dropped"
```

2. Ouvrir le menu de gestion du brancard

```
local success, reason = exports['MasterStretcher']:OpenMenu(entity)
```

3. Ranger le brancard dans l'ambulance *(Vérifie automatiquement que les portes requises de l'ambulance sont ouvertes)*

```
local success, reason = exports['MasterStretcher']:StowStretcher(ambulanceVehicle)
```

4. Sortir le brancard de l'ambulance

```
local success, reason = exports['MasterStretcher']:UnstowStretcher(ambulanceVehicle)
```

5. Ouvrir le menu du brancard depuis l'intérieur de l'ambulance

```
local success, reason = exports['MasterStretcher']:OpenAmbulanceMenu(ambulanceVehicle)
```

6. Bloquer / Débloquer les freins

```
local success, isBraked = exports['MasterStretcher']:ToggleBrakes(entity)
-- isBraked retourne le nouvel état des freins (true = bloqué)
```

☐☐ Lecture d'informations (Getters)

Savoir si le joueur pousse actuellement un brancard :

```
local isPushing = exports['MasterStretcher']:IsPushingStretcher() -- Retourne true/false
```

Récupérer l'entité du brancard tenu par le joueur :

```
local stretcherEntity = exports['MasterStretcher']:GetAttachedStretcher() -- Retourne l'entité ou nil
```

Savoir si un brancard est occupé (Joueur ou PNJ) :

```
local isOccupied = exports['MasterStretcher']:IsStretcherOccupied(entity) -- Retourne true/false
```

☐ ÉVÉNEMENTS CLIENT (Hooks)

Le script diffuse ces événements en local. Vous pouvez les écouter dans vos autres scripts pour déclencher actions automatiques (ex: allumer les gyrophares quand on sort le brancard, envoyer un message radio, etc.).

- **stretchermod:onPushStart**(entity) : Se déclenche quand le joueur attrape le brancard.
- **stretchermod:onPushStop**(entity, position) : Se déclenche quand le joueur lâche le brancard (position = 'debout' ou 'couche').
- **stretchermod:onStretcherStowed**(ambulanceEntity, stretcherEntity) : Se déclenche quand un brancard est rangé dans un véhicule.
- **stretchermod:onStretcherUnstowed**(ambulanceEntity, stretcherEntity) : Se déclenche quand un brancard est sorti d'un véhicule.

Exemple d'utilisation :

```
AddEventHandler('stretchermod:onStretcherStowed', function(ambulance, stretcher)
    print("Un brancard vient d'être chargé dans l'ambulance !")
    -- Insérez ici votre code pour fermer les portes arrière automatiquement, par exemple.
end)
```

Revision #1

Created 2026-09-07 19:27:53 UTC by Raph

Updated 2026-09-07 19:28:02 UTC by Raph