

La format du fichier `extensions.cfg`

■ Contenus

- ~ 1. Aperçu général
- ~ 2. Syntaxe
- ~ 3. La section `Cari`
- ~ 4. La section `Coupleri`
- ~ 5. La section `Exterior` (outdated)

■ 1. Aperçu général

Le fichier `extensions.cfg` permet de définir les propriétés des voitures individuelles, comme la longueur, la position des axes et les objets extérieurs.

Le fichier est un fichier texte encodé dans un simple [codage](#) arbitraire, cependant, UTF-8 avec une marque d'ordre d'octet est le choix préféré. Le modèle d'analyse des nombres est **Strict**. Le fichier doit être localisé dans le dossier `train` et doit être nommé `extensions.cfg`. Le dossier est interprété sur la base de ligne par ligne, du début à la fin.

● Index de voiture

Toutes les voitures dans `extensions.cfg` sont énumérées de 0 (la première voiture) à $n-1$ (la dernière voiture), où n est le nombre de voitures que le train a au total, selon le fichier de `train.dat`. Par exemple, sur un train avec 10 voitures, la première voiture a l'indice 0 et la dernière voiture a l'indice 9.

● Index d'attelage

L'attelage dans ce document se réfère à l'espace entre les voitures. Tous les attelages dans `extensions.cfg` sont énumérés de 0 (le premier attelage) à $n-2$ (le dernier attelage), où n est le nombre de voiture que comporte le train, selon le fichier `train.dat`. L'indice d'attelage i correspond à l'attelage entre les voitures avec l'indice i et $i+1$, donc l'attelage entre la première et deuxième voitures (0 et 1) dans le train à l'indice 0. De la même manière, si un train a 10 voitures, l'attelage entre les deux dernière voitures (8 et 9) aurait l'indice 8.

■ 2. Syntaxe

Chaque ligne dans le fichier peut être vide (ou consiste en seulement des espaces blancs) sera ignoré, pouvant marquer le commencement d'une nouvelle section ou contenant des paires de valeur de clé dans une section. Toutes les paires de valeur de clé correspondent à la dernière section ouverte.

Une nouvelle section est ouverte en commençant la ligne par un crochet ouvert (U+005B) et se terminant par un crochet fermé (U+005D). Le texte entre les crochets indique le nom de la section

et est insensible à la casse. Les espaces en blanc au commencement et à la fin de la ligne sont ignorés, de même qu'au début et à la fin du nom de la section. Ainsi, le début de la section a la forme suivante :

`[NameOfTheSection]`

Exemple d'indication d'une nouvelle section:

▶ `[exterior]`

La valeur d'une clé est indiquée en donnant le numéro de la clé, un signe égal (U+003D) et ensuite la valeur. La clé est insensible à la casse. Les espaces de blanc au commencement et la fin de la ligne sont ignorés, de même que devant et après que les signes égaux. Comme alternative exprimé, les espaces blanc qui entourent la clé et la valeur sont ignorés. Ainsi, une paire de clé de valeur a la forme suivante :

`NameOfTheKey = Value`

Exemple pour une double valeur clé:

▶ `0 = train.csv`

Vous pouvez utiliser des commentaires n'importe où, à la fin d'une ligne. Un commentaire commence par un point virgule (U+003B). Les commentaires, si présent, sont ignorés de toutes les lignes avant que celle-ci ne soit traité.

■ 3. La section *Cari*

La section *Cari* permet de définir les propriétés spécifiques de certaine voiture.

`[Cari]`

C'est le début de la section pour la voiture i , qui est un nombre entier entre 0 et $n-1$, où n est le nombre de voitures qui compose le train.

`Object = File`

File : Le nom du fichier relatif à l'objet extérieur utiliser pour cette voiture, relative au dossier train.

Cette valeur clé définit l'objet extérieur pour cette voiture. Dans le fichier objet, les coordonnées (0,0,0) (x, y, z) correspond au centre de la voiture, horizontalement droite gauche (x) et avant arrière (z), alors que $y=0$ correspondent au dessus des rails.

`Length = Value`

Value: Un nombre positif (à point flottant) représentant la longueur de la voiture.

Cette valeur clé définit la longueur pour cette voiture. Annule la longueur défini dans train.dat pour cette voiture particulière. Si inutilisé, cette voiture aura la longueur par défaut défini dans train.dat.

`Axles = Rear, Front`

Rear: Un nombre (point flottant) qui indique la position z de l'essieu arrière mesuré du centre de la

voiture. En général une valeur négative.

Front: Un nombre (point flottant) qui indique la position z de l'essieu avant mesuré du centre de la voiture. En général une valeur positive.

Cette valeur clé définit les positions des axes. Cependant *Rear* et *Front* peuvent prendre n'importe quelles valeurs, la condition $Rear < Front$ doit être respectée.

Reversed = Value

Value: Soit **True (vrai)** ou **False (faux)** pour indiquer si l'on inverse la voiture.

Avec ce paramètre, vous pouvez inverser ce qui constitue la voiture. Notez, S'il vous plaît, que la position des axes est donnée comme si la voiture n'a pas été inversée.

Exemple pour une section Cari :

```
► [Car0]
Object = locomotive.b3d
Length = 16
Axles = -6, 6
Reversed = False
```

■ 4. La section Coupleri

La section *Coupleri* permet de définir des propriétés spécifiques à certains attelages.

[Coupleri]

Début de la section coupler *i*, c'est un nombre entier entre 0 et $n-2$, où n est le nombre de voiture que possède le train.

Distances = Minimum, Maximum

Minimum: Un nombre (point flottant) qui indique la distance minimum permise entre les voitures.

Maximum: Un nombre (point flottant) qui indique la distance maximum permise entre les voitures.

Cette valeur clé définit les distances les plus basses et plus hautes permises entre les voitures, ressemblant à un attelage avec tampon et chaîne. La condition $Minimum \leq Maximum$ doit être respecté.

Exemple pour une section Coupleri :

```
► [Coupler0]
Distances = 0.30, 0.35
```

■ 5. La section Exterior (ancienne)

La section *Exterior* fournit une façon facile d'ajouter les objets extérieurs au train en particulier.

Pour plus de contrôle sur les paramètres de positions individuelles d'axe et de longueurs de voiture, la section *Cari* a été introduite et devrait plutôt être utilisée.

[Exterior]

Début de la section Exterior.

i = File

i: Un nombre entier entre 0 et $n-1$, où n est le nombre de voiture que possède le train. L'index 0 correspond à la première voiture et l'index $n-1$ à la dernière voiture du train.

File: Le nom relatif du fichier de l'objet extérieur utilisé pour car *i*, relatif au dossier train.

Cette double valeur clé peut être utilisée pour mettre en place les objets extérieur pour chaque voiture individuelle. Dans le fichier object, les coordonnées (0,0,0) (x, y, z) correspondent au centre de la voiture, horizontalement droite/gauche (x) et avant/arrière (z), alors que $y=0$ correspond à la partie supérieure des rails.

Exemple pour une section [Exterior] :

► [Exterior]
0 = cars\engine.csv
1 = cars\passenger_mk1.b3d
2 = cars\passenger_mk1.b3d
3 = cars\passenger_bistro.b3d
4 = cars\passenger_mk2.b3d
5 = cars\postal.x