

# Documentation de co\_depart

co\_depart est un programme qui, partant d'un fichier CSV d'inscriptions, génère des listes de départ pour les courses d'orientation à étapes.

## Fonctionnalités

Ses caractéristiques principales sont :

- Travaille directement depuis le CSV d'inscription venant de la FFCO
- Génère des fichiers CSV pouvant être directement importés dans le logiciel de GEC
- Regroupe les membres d'un même club dans une même tranche horaire
- Effectue une permutation circulaire des tranches horaires d'un jour sur l'autre
- Gère les demandes particulières, par exemple les rattachements d'une personne à un autre club ou des départs tôt/tardifs
- Gère un nombre quelconque d'étapes et de tranches horaires
- Gère un nombre quelconque de circuits
- Gère plusieurs catégories par circuit
- Permet de définir pour chaque circuit
  - L'heure du premier départ (autorisant ainsi, par exemple, les départs sur minutes paires et impaires)
  - L'intervalle entre départs pour un même circuit
  - La méthode de détermination des heures de départ :
    - \* aléatoire
    - \* suivant un ranking (IOF ou CN)
    - \* au boîtier
    - \* libre
    - \* aucune gestion
- utilise directement les fichiers CSV de l'IOF et de la FFCO pour les circuits avec ranking
- s'assure qu'il n'y a pas deux personnes du même club sur le même circuit partant consécutivement
- minimise la durée des départs
- lisse la charge de l'atelier départ
- génère le fichier CSV des dossards
- génère les étiquettes (par exemple pour les enveloppes ou les sacs de récupération de carte)
- génère les listes récapitulative par club à insérer dans les enveloppes
- génère un fichier de rapport au format *markdown* pouvant être transformé en pdf à l'aide de *Pandoc*
- est configurable à l'aide d'un fichier texte au format *JSON*
- est un programme *Python* open source sous licence GNU GPL pouvant être modifié à volonté
- fonctionne dans tous les environnements (Windows, Linux, Mac,...)

## Installation

### Interpréteur Python

*co\_depart* étant un fichier source *Python*, il est nécessaire de disposer d'un interpréteur *Python* installé sur sa machine. Si vous travaillez sous Linux, c'est déjà probablement le cas, sinon vous pouvez vous en assurer en ouvrant une fenêtre de commande et en tapant *python*

Si aucun interpréteur n'est installé, vous devez en télécharger un, typiquement à partir de <https://www.python.org/downloads/>

Il est recommandé de télécharger le plus récent. Le programme a été testé avec *Python 3.7.3* sous *Windows*. Veuillez suivre les informations du site de téléchargement pour les consignes d'installation de l'interpréteur *Python*.

### Editeur de texte

Vous devez également disposer d'un éditeur de texte, type notepad. Si vous n'en avez pas, quelques bons éditeurs gratuits sont :

- PSPAD (<http://www.pspad.com/fr/>)
- Notepad++ (<https://notepad-plus-plus.org/download/v7.6.6.html>)
- Vim (<https://www.vim.org/download.php>)
- Visual Studio Code (<https://code.visualstudio.com/>)

### Compilation des fichiers LaTeX

Les étiquettes et les listes récapitulatives par club sont générées au format LaTeX. Pour obtenir le fichier PDF correspondant il est nécessaire d'avoir un compilateur LaTeX.

Par exemple sous Windows vous pouvez utiliser MikTeX (<https://miktex.org/download>)

Il peut être utile d'avoir un environnement de développement LaTeX, soit en chargeant le module correspondant dans votre éditeur, soit en utilisant TeXnicCenter (<http://www.tenncenter.org/download/>)

### Mise en forme du rapport

Lors de l'exécution un rapport au format *Markdown* est généré. Ce fichier est plus agréable à lire s'il est transformé en fichier pdf à l'aide de *Pandoc*. Vous pouvez télécharger *Pandoc* ici : <https://pandoc.org/>

Attention *Pandoc* risque de vous obliger à télécharger un compilateur *LaTeX*, MikTeX par exemple. Et l'installation de tout ceci peut être... complexe ! Si vous utilisez déjà *LaTeX* n'hésitez pas, sinon une solution plus simple peut être

d'ajouter une extension à votre navigateur qui sait mettre en forme les fichiers Markdown.

On en trouve facilement, au moins pour Firefox et Chrome, par exemple *Markdown Viewer Webext*. Procédez comme vous le faite habituellement pour ajouter des extensions à votre navigateur.

## Drapeaux

Si vous générez les listes par club ou les étiquettes il est nécessaire de disposer des fichiers png correspondant aux différents pays.

Vous pouvez, par exemple, récupérer ces derniers ici (sélectionner “résolution classique”): <http://www.drapeauxdespays.fr/telecharger>

## Installation

Créez un répertoire de travail et copiez-y le programme python obtenu sur <https://github.com/jmonclard>

Nous vous conseillons également d’y copier les fichiers JSON fournis dans le sous répertoire d'exemple afin de disposer d’une structure de départ complète.

Même s’il est possible de définir des chemins complets dans le fichier JSON, il est souvent plus simple d’également copier dans ce répertoire les fichiers CSV contenant les inscriptions et éventuellement les rankings IOF et CN (FFCO).

Enfin il est nécessaire de créer un sous-répertoire et y mettre les fichiers png des drapeaux des différents pays.

## Configuration

L’essentiel de la configuration du programme s’effectue à l’aide du fichier JSON. Par défaut le fichier JSON recherché est *ofrance2019.json*. Vous pouvez chnager ceci soit en éditant le fichier Python (constante `DEFAULT_JSON_CONFIG` ligne 25, en début de fichier) soit en précisant le nom du fichier en ligne de commande lors de l’exécution.

Les détails des différents champs du fichier JSON sont définis dans le chapitre *Syntaxe du fichier JSON de paramétrage* ci-après.

## Exécution

### Traitement des données

Une fois le fichier JSON adapté, l’exécution du programme s’effectue simplement en exécutant le script *Python*.

Celà se fait :

- soit dans une fenêtre de commande, entrer alors comme commande :

```
python co_depart.py
```

ou

```
python co_depart.py -c <nom_du_fichier_JSON_de_configuration>
```

Il est également possible d'augmenter la verbosité avec l'option `-v` en ligne de commande (il est possible d'en mettre plusieurs pour augmenter la quantité de texte affiché). Exemple :

```
python co_depart.py -v -v -v
```

- soit à partir d'un environnement de développement (IDLE ou votre éditeur par exemple).

Dans IDLE chargez le fichier `co_depart.py` avec *File/Open...* et, dans le fenêtre qui vient de s'ouvrir et contient le fichier python, cliquez sur *Run*

**Génération impossible** Les affectations d'heures de départ dans une tranche sont aléatoires (sauf ranking) mais tiennent compte de la contrainte de ne pas avoir deux coureurs du même club partant successivement.

Le nombre de départs possibles dans une tranche horaire est lié à la catégorie ayant le plus grand nombre de départs. Dans certaines situations l'algorithme peut ne pas trouver de solution avec le tirage aléatoire effectué. Si c'est le cas, vous pouvez relancer l'exécution.

Si après plusieurs tentatives vous ne parvenez pas à obtenir une répartition acceptable, vous pouvez augmenter la valeur du paramètre "*TranchesHoraires*" "*MargeTranches*" dans le fichier JSON. Ceci a pour effet de créer des horaires de départ supplémentaires, facilitant l'affectation d'horaires aux participants au dépend d'une durée totale des départs plus importante.

## Mise en forme du rapport

Le script *Python* génère un rapport d'exécution au format *Markdown*, il s'agit du fichier créé ayant pour suffixe `.md` et que vous pouvez facilement lire avec un éditeur de texte.

Toutefois un rapport plus agréable à lire, au format pdf, peut facilement être généré à l'aide de *pandoc*.

Dans une fenêtre de commande, dans le répertoire de travail entrez :

```
pandoc -t latex -f markdown -o <fichier_à_créer.pdf> <fichier_rapport.md>
```

par exemple :

```
pandoc -t latex -f markdown -o ofrance-out.pdf ofrance-out.md
```

# Syntaxe du fichier JSON de paramétrage

## Généralités sur la syntaxe d'un fichier JSON

Vous pouvez trouver une explication détaillée sur le format JSON ici : <https://www.json.org/json-fr.html> Toutefois, les indications suivantes devraient être suffisantes pour l'utilisation de ce programme.

- Un fichier JSON est un fichier texte standard, utilisez donc un éditeur de texte (et non un traitement de texte !) pour le créer ou l'éditer
- Le contenu du fichier JSON est sensible à la casse. “*Gec*” et “*GEC*” ne sont donc pas la même chose !
- Vous pouvez laisser des lignes vides et indenter les lignes comme vous le souhaitez , mais il n’y a pas de commentaires dans le format JSON
- Un fichier JSON commence par une accolade ouvrante et se termine par une accolade fermante.
- Entre ces deux accolades on va trouver :
  - soit des couples {clef : valeur} où les champs *clef* et *valeur* sont séparés par des virgules
  - soit des listes entre crochets pouvant contenir des nombres ou des couples {clef : valeur} séparées par des virgules ou encore des listes pouvant elles-même contenir. . . .
  - soit des couples {clef : ensemble de couples clef : valeur }. Dans ce cas l'ensemble des couples clef:valeur, comme chaque couple, est entouré d'accolades et les différents couples sont séparés par des virgules
  - il n’y a pas de virgule après le dernier couple {clef : valeur}
  - les clefs sont entre double quotes
  - les valeurs sont entre doubles quotes si ce sont des chaînes de caractères, sans doubles quotes sinon. Un exemple de code légal est :

```
{
  "MaClef": "SaValeur",
  "MonAutreClef": "La Valeur De Cette Autre Clef",
  "UnNombre": 12,
  "UneListeDeNombres": [1,2,3,4],
  "UnObjetComplexe": {
    "SaPremiereClef": "Premier contenu",
    "Une deuxième clef": 2345
  }
}
```

## Mots clefs reconnus

Bien que l'ordre des sections dans un fichier JSON n'ait aucune importance, les mots clefs reconnus sont présentés ci-après dans l'ordre du fichier JSON fourni en exemple.

## En-tête

Clef	Type valeur	Détails
“Nom”	<i>chaîne</i>	Nom de l'évènement. N'est utilisé que dans les rapports.
“Information”	<i>chaîne</i>	Commentaire présent dans le JSON et utile lors de sa consultation mais non utilisé par le script.
“Verbosité”	<i>nombre</i>	Définit le niveau de verbosité du script lors de son exécution. Actuellement non géré. Utilisez l'option -v en ligne de commande ou éditez la valeur par défaut de l'option -v en ligne 69 du programme python.
“FlagsSubdirectory”	<i>chaîne</i>	Sous-répertoire contenant les images des drapeaux. Ceux-ci doivent être des fichiers png dont le nom contient les deux lettres identifiant le pays dans la norme ISO.
“GraineGénérateurAleatoire”	<i>chaîne</i>	Peut contenir soit “None” soit une chaîne de caractères contenant un nombre. Dans le premier cas, chaque exécution conduira à des résultats différents, même avec les mêmes données d'entrée; si un nombre est fourni, il servira de valeur initiale pour le générateur de nombre aléatoire. Dans ce cas chaque exécution - si les données d'entrées sont inchangées - conduiront au même résultat.

## Ranking FFCO

Clef	Type valeur	Détails
“RankingFFCO”	<i>élément complexe</i>	Définit la façon d'accéder et de traiter les données issues du classement national de la FFCO.

Clef	Type valeur	Détails
“RankingFFCO” “FichierCSV”	<i>chaîne</i>	Nom du fichier CSV contenant les données du classement
“RankingFFCO” “Encodage”	<i>chaîne</i>	Encodage utilisé pour le fichier CSV. Typiquement “ansi” ou “utf8”
“RankingFFCO” “SeparateurColonnesCSV”	<i>chaîne</i>	Séparateur de colonnes du fichier CSV. Typiquement une virgule ou un point-virgule
“RankingFFCO” “DebutTitre”	<i>chaîne</i>	Début de la ligne de titre devant être ignorée. Toute ligne commençant par la chaîne indiquée sera ignorée. Si des doubles quotes sont présentes dans le fichier CSV, il est nécessaire de les faire précéder par une barre oblique inverse (backslash). Exemple “\”Place\”;
“RankingFFCO” “Colonnes”	<i>élément complexe</i>	Colonnes du fichier CSV (one based) contenant les différents champs utiles
“RankingFFCO” “Colonnes” “Prenom”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le prénom du licencié (actuellement non utilisé par le script)
“RankingFFCO” “Colonnes” “Nom”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le nom du licencié (actuellement non utilisé par le script)
“RankingFFCO” “Colonnes” “Ranking”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le classement du licencié (actuellement non utilisé par le script)
“RankingFFCO” “Colonnes” “CN”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le nombre de points du licencié
“RankingFFCO” “Colonnes” “NumeroLicenceFFCO”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le numéro de licence FFCO du licencié

Clef	Type valeur	Détails
“RankingFFCO” “AjoutIdentifiant”	<i>liste d’éléments complexes</i>	Cette section permet d’ajouter des numéros de licence FFCO à des participants
“RankingFFCO” “AjoutIdentifiant”[*] “Nom”	<i>chaîne</i>	Nom du participant pour lequel on veut ajouter un numéro de licence FFCO
“RankingFFCO” “AjoutIdentifiant”[*] “Prenom”	<i>chaîne</i>	Prénom du participant pour lequel on veut ajouter un numéro de licence FFCO
“RankingFFCO” “AjoutIdentifiant”[*] “Club”	<i>chaîne</i>	Club du participant pour lequel on veut ajouter un numéro de licence FFCO
“RankingFFCO” “AjoutIdentifiant”[*] “LicenceFFCO”	<i>nombre entier</i>	Numéro de licence FFCO du participant pour lequel on veut ajouter un numéro de licence FFCO

**Remarques** - Cette section n’est nécessaire que si un départ au ranking d’au moins un circuit est envisagé - Le fichier CSV contenant le classement national FFCO peut être téléchargé ici : <http://cn.ffcorientation.fr/classement/>

## Ranking IOF

Clef	Type valeur	Détails
“RankingIOF”	<i>élément complexe</i>	Défini la façon d’accéder et de traiter les données issues des divers classements de l’IOF.
“RankingIOF” “FichierCSV”	<i>élément complexe</i>	Section contenant les noms des fichiers CSV des classements IOF
“RankingIOF” “FichierCSV” “Men”	<i>élément complexe</i>	Section contenant les noms des fichiers CSV des classements IOF pour les hommes



Clef	Type valeur	Détails
“RankingIOF” “FichierCSV” “Men” “PedestreSprint”	<i>chaîne</i>	Nom du fichier CSV contenant le classement IOF des hommes en sprint pédestre
“RankingIOF” “FichierCSV” “Men” “PedestreMDLD”	<i>chaîne</i>	Nom du fichier CSV contenant le classement IOF des hommes en MD et LD pédestre
“RankingIOF” “FichierCSV” “Men” “MTBO”	<i>chaîne</i>	Nom du fichier CSV contenant le classement IOF des hommes VTT’O
“RankingIOF” “FichierCSV” “Men” “SkiO”	<i>chaîne</i>	Nom du fichier CSV contenant le classement IOF des hommes en Ski’O
“RankingIOF” “FichierCSV” “Men” “TrailO”	<i>chaîne</i>	Nom du fichier CSV contenant le classement IOF des hommes en Trail’O
“RankingIOF” “FichierCSV” “Women”	<i>élément complexe</i>	Section contenant les noms des fichiers CSV des classements IOF pour les femmes
“RankingIOF” “FichierCSV” “Women” “PedestreSprint”	<i>chaîne</i>	Nom du fichier CSV contenant le classement IOF des femmes en sprint pédestre
“RankingIOF” “FichierCSV” “Women” “PedestreMDLD”	<i>chaîne</i>	Nom du fichier CSV contenant le classement IOF des femmes en MD et LD pédestre
“RankingIOF” “FichierCSV” “Women” “MTBO”	<i>chaîne</i>	Nom du fichier CSV contenant le classement IOF des femmes VTT’O
“RankingIOF” “FichierCSV” “Women” “SkiO”	<i>chaîne</i>	Nom du fichier CSV contenant le classement IOF des femmes en Ski’O

Clef	Type valeur	Détails
“RankingIOF” “FichierCSV” “Women” “TrailO”	<i>chaîne</i>	Nom du fichier CSV contenant le classement IOF des femmes en Trail’O
“RankingIOF” “Encodage”	<i>chaîne</i>	Encodage utilisé pour le fichier CSV. Typiquement “ansi” ou “utf8”
“RankingIOF” “SeparateurColonnesCSV”	<i>chaîne</i>	Séparateur de colonnes du fichier CSV. Typiquement une virgule ou un point-virgule.
“RankingIOF” “DebutTitre”	<i>chaîne</i>	Début de la ligne de titre devant être ignorée. Toute ligne commençant par la chaîne indiquée sera ignorée.
“RankingIOF” “Colonnes”	<i>élément complexe</i>	Cette section définit dans quelles colonnes du fichier CSV se trouvent les informations devant être utilisées
“RankingIOF” “Colonnes” “Id”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant l’identifiant IOF du concurrent
“RankingIOF” “Colonnes” “Points”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le nombre de points du concurrent au classement IOF
“RankingIOF” “AjoutIdentifiant”	<i>liste d’éléments complexes</i>	Cette section permet d’ajouter des identifiants IOF à des participants
“RankingIOF” “AjoutIdentifiant”[*] “Nom”	<i>chaîne</i>	Nom du participant pour lequel on veut ajouter un identifiant IOF
“RankingIOF” “AjoutIdentifiant”[*] “Prenom”	<i>chaîne</i>	Prénom du participant pour lequel on veut ajouter un identifiant IOF

Clef	Type valeur	Détails
“RankingIOF” “AjoutIdentifiant”[*] “Club”	<i>chaîne</i>	Club du participant pour lequel on veut ajouter un identifiant IOF
“RankingIOF” “AjoutIdentifiant”[*] “IdIoF”	<i>nombre entier</i>	Identifiant IOF du participant pour lequel on veut ajouter un identifiant IOF

**Remarques** - Cette section n’est nécessaire que si un départ au ranking d’au moins un circuit est envisagé - Il n’est pas nécessaire de définir tous les fichiers. Seuls les fichiers concernant le type d’épreuve envisagée comme ayant un départ au ranking sont à renseigner. Les autres peuvent être laissés vides (“”). - Les fichiers CSV des différents rankings IOF peuvent être téléchargés ici : <https://ranking.orienteeering.org/Ranking> - Si des participants n’ont pas renseigné leur identifiant IOF au moment de l’inscription, il est possible de le faire à l’aide de la section “AjoutIdentifiant”

## Inscriptions

Clef	Type valeur	Détails
“Inscriptions”	<i>élément complexe</i>	Cette section détaille la façon de gérer le fichier d’inscription
“Inscriptions” “FichierCSV”	<i>chaîne</i>	Nom du fichier CSV contenant les inscriptions
“Inscriptions” “Encodage”	<i>chaîne</i>	Encodage utilisé pour le fichier CSV. Typiquement “ansi” ou “utf8”
“Inscriptions” “SeparateurColonnesCSV”	<i>chaîne</i>	Séparateur de colonnes du fichier CSV. Typiquement une virgule ou un point-virgule.
“Inscriptions” “LignesDeTitre”	<i>nombre</i>	Nombre de lignes de titre à sauter lors de la lecture du fichier CSV d’inscription
“Inscriptions” “DebutTitre”	<i>chaîne</i>	Début des lignes à sauter
“Inscriptions” “Colonnes”	<i>élément complexe</i>	Définition des colonnes contenant les informations à récupérer

Clef	Type valeur	Détails
“Inscriptions” “Colonnes” “Nom”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le nom du participant
“Inscriptions” “Colonnes” “Prenom”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le prénom du participant
“Inscriptions” “Colonnes” “Sexe”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le sexe du participant
“Inscriptions” “Colonnes” “Annee”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant l’année de naissance du participant
“Inscriptions” “Colonnes” “Pays”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant la nationalité du participant
“Inscriptions” “Colonnes” “NumeroClub”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le numéro de club (par exemple 1307) du participant
“Inscriptions” “Colonnes” “ReferenceClub”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le numéro de club complet (par exemple 1307PZ) du participant
“Inscriptions” “Colonnes” “NomClub”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le nom de club (par exemple ACA Aix-en-Provence) du participant
“Inscriptions” “Colonnes” “NumeroLicenceFFCO”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le numéro de licence du participant
“Inscriptions” “Colonnes” “IOFid”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant l’identifiant IOF du participant

Clef	Type valeur	Détails
“Inscriptions” “Colonnes” “Etapas”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant la liste des étapes auxquelles est inscrit le participant
“Inscriptions” “Colonnes” “Categorie”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant la catégorie du participant
“Inscriptions” “Colonnes” “NumeroPuceSI”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le numéro de puce SI du participant
“Inscriptions” “Colonnes” “ModeleTshirt”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant le modèle de T-shirt souhaité
“Inscriptions” “Colonnes” “TailleTshirt”	<i>nombre</i>	Numéro de la colonne, en comptant à partir de 1, contenant la taille de T-shirt souhaitée
“Inscriptions” “ParticipantsASupprimer”	<i>liste d'éléments complexes</i>	Liste des participants figurant dans le fichier CSV d'inscription mais devant être supprimés ( par exemple à cause d'erreurs d'inscription).
“Inscriptions” “ParticipantsASupprimer”[*] “Nom”	<i>chaîne</i>	Nom du participant devant être supprimé des inscriptions
“Inscriptions” “ParticipantsASupprimer”[*] “Prenom”	<i>chaîne</i>	Prénom du participant devant être supprimé des inscriptions
“Inscriptions” “ParticipantsASupprimer”[*] “Club”	<i>chaîne</i>	Club du participant devant être supprimé des inscriptions

Clef	Type valeur	Détails
“Inscriptions” “Partici- pantsSupple- mentaires” :	<i>liste  d’éléments  complexes</i>	Liste des participants devant être ajoutés ( par exemple à cause d’erreurs d’inscription ou d’inscriptions tardives).
“Inscriptions” “Partici- pantsSupple- mentaires”[*] “Nom”	<i>chaîne</i>	Nom du participant devant être ajouté aux inscriptions
“Inscriptions” “Partici- pantsSupple- mentaires”[*] “Prenom”	<i>chaîne</i>	Prénom du participant devant être ajouté aux inscriptions
“Inscriptions” “Partici- pantsSupple- mentaires”[*] “Sexe”	<i>chaîne</i>	Sexe du participant devant être ajouté aux inscriptions
“Inscriptions” “Partici- pantsSupple- mentaires”[*] “Annee”	<i>chaîne</i>	Année de naissance du participant devant être ajouté aux inscriptions
“Inscriptions” “Partici- pantsSupple- mentaires”[*] “Pays”	<i>chaîne</i>	Nationalité du participant devant être ajouté aux inscriptions
“Inscriptions” “Partici- pantsSupple- mentaires”[*] “NumeroClub”	<i>chaîne</i>	Numéro de club (par exemple “1307”) du participant devant être ajouté aux inscriptions. Peut être laissé vide (“”).

Clef	Type valeur	Détails
“Inscriptions” “ParticipantsSupplementaires”[*] “ReferenceClub”	<i>chaîne</i>	Numéro de club complet (par exemple “1307PZ”) du participant devant être ajouté aux inscriptions. Peut être laissé vide (“”).
“Inscriptions” “ParticipantsSupplementaires”[*] “NomClub”	<i>chaîne</i>	Nom de club du participant devant être ajouté aux inscriptions
“Inscriptions” “ParticipantsSupplementaires”[*] “NumeroLicenceFFCO”	<i>chaîne</i>	Numéro de licence FFCO du participant, sous forme de chaîne de caractères entre doubles quotes, devant être ajouté aux inscriptions (par exemple “12345”). Doit être laissé vide pour les non licenciés FFCO (“”).
“Inscriptions” “ParticipantsSupplementaires”[*] “IOFid”	<i>chaîne</i>	Identifiant IOF du participant devant être ajouté aux inscriptions. Peut être laissé vide (“”).
“Inscriptions” “ParticipantsSupplementaires”[*] “Etapas”	<i>chaîne</i>	Liste des étapes auxquelles le participant doit être inscrit. Les étapes sont indiqués sous forme d’une seule chaîne de caractère, donc entre doubles quotes, contenant les numéros séparés par des virgules (par exemple “1,2,4”).
“Inscriptions” “ParticipantsSupplementaires”[*] “Categorie”	<i>chaîne</i>	Nom de la catégorie du participant devant être ajouté aux inscriptions (par exemple “H50”)
“Inscriptions” “ParticipantsSupplementaires”[*] “NumeroPuceSI”	<i>chaîne</i>	Numéro de puce SI (sous forme de chaîne de caractères) du participant devant être ajouté aux inscriptions (par exemple “123456”).

Clef	Type valeur	Détails
“Inscriptions” “ParticipantsSupplementaires”[*] “ModeleTshirt”	<i>chaîne</i>	Modèle de T-shirt souhaité par le participant devant être ajouté aux inscriptions (par exemple “COL V”).
“Inscriptions” “ParticipantsSupplementaires”[*] “TailleTshirt”	<i>chaîne</i>	Taille du T-shirt (sous forme de chaîne de caractères) souhaitée par le participant devant être ajouté aux inscriptions (par exemple “XXL”).

**Remarques** - L'utilisation des listes “*Inscriptions*” “*ParticipantsASupprimer*” et “*Inscriptions*” “*ParticipantsSupplementaires*” permet de corriger d'éventuelles erreurs d'inscription en faisant figurer un participant dans les deux listes. - Si un participant change de circuit d'un jour à l'autre, on peut le supprimer et l'ajouter deux fois (ou plus), une fois pour chaque circuit. Il figurera autant de fois dans la GEC avec des dossards différents.

### FichiersGeneres

Clef	Type valeur	Détails
“FichiersGeneres”	<i>element complexe</i>	Informations sur le format des fichiers à générer pour la GEC.
“FichiersGeneres” “NomFichierMD”	<i>chaîne</i>	Nom du fichier markdown contenant le rapport d'exécution
“FichiersGeneres” “GEC”	<i>chaîne</i>	Indique le logiciel de GEC qui sera utilisé et adapte le format de sortie. Actuellement seul “MeOS” est reconnu.



Clef	Type valeur	Détails
“FichiersGeneres” “NomFichierCSV”	<i>chaîne</i>	Préfixe des nom de fichiers CSV à générer pour la GEC. Ce préfixe est suivi de _Etape_1, _Etape2 etc. en fonction de l’étape.
“FichiersGeneres” “SeparateurColonnesCSV”	<i>chaîne</i>	Séparateur de colonnes des fichiers CSV prévus pour la GEC. Typiquement une virgule ou un point-virgule.

## Dossards

Clef	Type valeur	Détails
“Dossards”	<i>élément complexe</i>	Informations sur le format du fichier à générer pour l’impression des dossards
“Dossards” “NomFichierCSVDossards”	<i>chaîne</i>	Nom du fichier CSV à générer en vue de l’impression des dossards
“Dossards” “SeparateurColonnesCSV”	<i>chaîne</i>	Séparateur de colonnes du fichier CSV des dossards. Typiquement une virgule ou un point-virgule.
“Dossards” “PremierDossard”	<i>nombre entier</i>	Numéro du premier dossard affecté

Clef	Type valeur	Détails
“Dossards” “DossardsSupplementaires”	<i>nombre entier</i>	Nombre de dossars surnuméraires à prévoir (non affectés à des participants inscrits). Prévu pour les inscriptions ou changements sur place.
“Dossards” “CircuitsSurDossards”	<i>chaîne</i>	Indique si les circuits doivent figurer sur les dossards. Valeurs possibles : ‘oui’ ou ‘non’.

#### Listes détaillées par club pour enveloppes

Clef	Type valeur	Détails
“Enveloppes”	<i>element complexe</i>	Informations sur le format du fichier à générer pour l’impression des listes par club à mettre dans les enveloppes
“Enveloppes” “NomFichierLaTeX”	<i>chaîne</i>	Nom du fichier LaTeX à générer en vue de l’impression des listes
“Enveloppes” “CircuitsSurListe”	<i>chaîne</i>	“oui” si le nom des circuits doit figurer dans le tableau pour chaque étape; “non” dans le cas contraire.
“Enveloppes” “MargeGauche_mm”	<i>nombre</i>	Taille en mm de la marge gauche de la page générée (exemples : 10 ou 10.2)
“Enveloppes” “MargeDroite_mm”	<i>nombre</i>	Taille en mm de la marge droite de la page générée (exemples : 10 ou 10.2)

Clef	Type valeur	Détails
“Enveloppes” “MargeSuperieure_mm”	<i>nombre</i>	Taille en mm de la marge supérieure de la page générée (exemples : 10 ou 10.2)
“Enveloppes” “MargeInferieure_mm”	<i>nombre</i>	Taille en mm de la marge inférieure de la page générée (exemples : 10 ou 10.2)
“Enveloppes” “HauteurDrapeau_mm”	<i>nombre</i>	Hauteur en mm du drapeau correspondant au pays du club (exemples : 10 ou 10.2)

#### **Etiquettes club**

Clef	Type valeur	Détails
“Etiquettes”	<i>element complexe</i>	Informations sur le format du fichier à générer pour l'impression des étiquettes club à mettre sur les enveloppes ou les sacs de récupération des cartes.
“Etiquettes” “NomFichierLaTeX”	<i>chaine</i>	Nom du fichier LaTeX à générer en vue de l'impression des étiquettes
“Etiquettes” “CouleurCadre”	<i>chaine</i>	Couleur (en anglais) du cadre de l'étiquette. Par exemple “black”. Mettre “white” si l'on ne souhaite pas avoir de cadre.

Clef	Type valeur	Détails
“Etiquettes” “NombreDeColonnes”	<i>nombre entier</i>	Nombre de colonnes sur la planche d’étiquettes (exemple : 2)
“Etiquettes” “NombreDeLignes”	<i>nombre entier</i>	Nombre de lignes d’étiquettes sur la planche d’étiquettes (exemple : 5)
“Etiquettes” “LargeurEtiquette_mm”	<i>nombre</i>	Largeur en mm des étiquettes (exemple : 99.1)
“Etiquettes” “HauteurEtiquette_mm”	<i>nombre</i>	Largeur en mm des étiquettes (exemple : 57)
“Etiquettes” “MargeGauche_mm”	<i>nombre</i>	Taille en mm de la marge gauche de la page générée (exemple : 10 ou 10.2)
“Etiquettes” “MargeEntreColonnes_mm”	<i>nombre</i>	Largeur en mm de la marge entre les colonnes d’étiquettes (exemple : 3 ou 3.2)
“Etiquettes” “MargeSuperieure_mm”	<i>nombre</i>	Taille en mm de la marge supérieure de la page générée (exemple : 10 ou 10.2)
“Etiquettes” “MargeEntreLignes_mm”	<i>nombre</i>	Distance en mm entre lignes d’étiquettes (exemple : 3 ou 3.2)
“Etiquettes” “HauteurDrapeau_mm”	<i>nombre</i>	Hauteur en mm du drapeau correspondant au pays du club (exemple : 10)

## TranchesHoraires

Clef	Type valeur	Détails
“TranchesHoraires”	<i>element complexe</i>	Paramètres définissant les tranches horaires qui seront générés
“TranchesHoraires” “ReserveVacantsPourcent”	<i>nombre entier</i>	Pourcentage de vacants à prévoir pour chaque catégorie
“TranchesHoraires” “ReserveVacantsOffset”	<i>nombre entier</i>	Nombre minimal de vacants à prévoir pour chaque catégorie
“TranchesHoraires” “SeuilClub”	<i>nombre entier</i>	Les clubs au-dessus de ce seuil sont considérés comme des <i>gros</i> clubs, et sont affectés en priorité par l'algorithme
“TranchesHoraires” “SeuilPays”	<i>nombre entier</i>	Les participants des pays dont le nombre d'inscrits est au-dessous de ce seuil sont conservés dans la même tranche

horaire même s'ils ne sont pas du même club. | | “TranchesHoraires” “NbTranches” | *nombre entier* | Nombre de tranches horaires. La rotation des tranches s'effectuera sur la base de ce nombre. | | “TranchesHoraires” “MargeTranches” | *nombre entier* | Le nombre de départs possibles dans une tranche horaire est le nombre d'inscrits sur le circuit ayant le plus d'inscrits additionné de ce nombre.

### AliasCategories

Clef	Type valeur	Détails
“AliasCategories”	<i>liste d'éléments complexes</i>	Table de conversion du nom des catégories. Essentiellement une liste de paires de noms et d'identifiants sans ordre particulier.

Clef	Type valeur	Détails
“AliasCategories”[*] “input”	<i>chaîne</i>	Nom de la catégorie telle qu’elle apparait dans le fichier d’inscription (ex H10)
“AliasCategories”[*] “output”	<i>chaîne</i>	Nom de la catégorie telle qu’elle doit apparaître dans le fichier pour la GEC (ex M10)
“AliasCategories”[*] “id”	<i>nombre entier</i>	Numéro de la catégorie à utiliser par le logiciel de GEC

#### AliasSexe

Clef	Type valeur	Détails
“AliasSexes”	liste de {clefs : valeurs}	Table de conversion des identifiants de sexe (par exemple pour passer de français à anglais). Une liste de paires de lettres sans ordre particulier.
“AliasSexes”[*] { , }	{clefs : valeurs}	La clef (par exemple “H”) correspond à l’identifiant dans le CSV d’inscription, la valeur (par exemple “M”) à ce qui doit être fourni à la GEC.

#### AliasClub

Clef	Type valeur	Détails
“AliasClubs”	liste de {clefs : valeurs}	Table de conversion des noms de club (par exemple pour corriger des noms de club). Une liste de paires de noms sans ordre particulier.
“AliasClubs” [*] { , }	{clefs : valeurs}	La clef (par exemple “COLIEGE”) est le nom apparaissant dans le fichier d’inscription, la valeur (par exemple “CO LIEGE”) le nom devant apparaître dans la GEC.

**Remarques** Pour corriger un nom de club ayant des apostrophes et pour lequel une barre oblique inverse apparaît avant l’apostrophe dans le fichier d’inscription, insérez devant les apostrophes une seconde barre oblique inverse. Par exemple utilisez la conversion suivante : {“O\’JURA” : “O’JURA”}

### NationaliteClubs

Clef	Type valeur	Détails
“NationaliteClubs”	liste de {clefs : valeurs}	Table d’affectation d’un pays à un club (si tous les membres n’ont pas la même nationalité). Une liste de paires de noms de clubs et de code pays sur 3 lettres sans ordre particulier.
“NationaliteClubs” [*] { , }	{clefs : valeurs}	La clef (par exemple “IFK GOTEBORG”) est le nom du club apparaissant dans le fichier d’inscription, la valeur (par exemple “SWE”) est le code ISO en 3 lettres du pays.

## AliasPrenoms

Clef	Type valeur	Détails
"AliasPrenoms"	liste de {clefs : valeurs}	Table de conversion de prénoms (par exemple pour accentuer des prénoms). Une liste de paires de prénoms sans ordre particulier.
"AliasPrenoms" [*] { , }	{clefs : valeurs}	La clef (par exemple "Joel") est le prénom apparaissant dans le fichier d'inscription après n'avoir conservé que les capitales en majuscule, la valeur (par exemple "Joël") le nom devant apparaître dans la GEC.

## ClubGroupeHoraireForce

**ATTENTION** : Dans le tableau ci-dessous "CGHF" est à remplacer par "ClubGroupeHoraireForce"

Clef	Type valeur	Détails
"CGHF"	liste de {clefs : valeurs}	Table définissant le groupe horaire auquel doit appartenir le club.
"CGHF" [*] { , }	{clefs : valeurs}	La clef est le nom du club apparaissant dans le fichier d'inscription éventuellement après avoir été remplacé par son alias (voir "AliasClub" ci-dessus), la valeur (nombre entier, par exemple 4) est le numéro du groupe horaire.

**Remarques** - Les groupes horaires vont de 1 au nombre de tranches horaires définies par "TranchesHoraires" "NbTranches". - Le groupe horaire définit la tranche horaire du premier jour. Ainsi : - Le groupe horaire 1 partira successivement dans les tranches horaires 1, 2, 3 etc. selon les jours (la tranche 1 est la plus matinale) - Le groupe horaire 2 partira successivement dans les tranches horaires 2, 3, 4, etc. selon les jours pour terminer par la tranche 1 -



Le groupe horaire 3 partira successivement dans les tranches horaires 3, 4, etc. selon les jours pour terminer par les tranches 1 et 2 - etc.

### ParticipantsTranchesHorairesAutreClub

**ATTENTION** : Dans le tableau ci-dessous “PTHAC” est à remplacer par “ParticipantsTranchesHorairesAutreClub”

Clef	Type valeur	Détails
“PTHAC”	liste d’éléments complexes	Indique qu’un participant doit être dans la même tranche horaire qu’un certain club pour les jours précisés, par exemple parce qu’il loge avec ce club.
“PTHAC”[*] “nom”	<i>chaîne</i>	Nom du participant dont on veut changer la tranche horaire
“PTHAC”[*] “prenom”	<i>chaîne</i>	Prénom du participant dont on veut changer la tranche horaire
“PTHAC”[*] “club”	<i>chaîne</i>	Club du participant dont on veut changer la tranche horaire
“PTHAC”[*] “autreclub”	<i>chaîne</i>	Club auquel le participant veut être rattaché
“PTHAC”[*] “etapes”	<i>liste de nombres entiers</i>	Liste de numéros d’étapes pour lesquelles le rattachement doit être fait (par exemple [3,4,5])

### ParticipantsTranchesHorairesForcees

**ATTENTION** : Dans le tableau ci-dessous “PTHF” est à remplacer par “ParticipantsTranchesHorairesForcees”

Clef	Type valeur	Détails
“PTHF”	liste d’éléments complexes	Indique qu’un participant doit être affecté à une tranche horaire précise pour les étapes précisées. Cette fonctionnalité peut être utilisée pour imposer un départ tôt ou tardif certains jours (garde d’enfants ou trajet long par exemple).
“PTHF”[*] “nom”	<i>chaîne</i>	Nom du participant dont on veut forcer la tranche horaire

Clef	Type valeur	Détails
“PTHF”[*] “prenom”	<i>chaîne</i>	Prénom du participant dont on veut forcer la tranche horaire
“PTHF”[*] “club”	<i>chaîne</i>	Club du participant dont on veut forcer la tranche horaire
“PTHF”[*] “tranches”	<i>liste de nombres entiers</i>	Liste des tranches horaires souhaitées pour les différentes étapes. Il doit y avoir autant de nombre que d’étapes. Chaque nombre doit être compris entre 1 et le nombre de tranches horaires (voir paramètre “TranchesHoraires” “NbTranches”) ou valoir 0 si pour cette étape la tranche horaire ne doit pas être forcée. Dans ce dernier cas l’algorithme normal (groupe du club ou ranking) s’applique pour cette étape.

#### ParticipantsTranchesHorairesIdentiques

**ATTENTION** : Dans le tableau ci-dessous “PTHI” est à remplacer par “ParticipantsTranchesHorairesIdentiques”

Clef	Type valeur	Détails
“PTHI”	liste d’éléments complexes	Indique qu’un participant doit être affecté à la même tranche horaire qu’un second participant (par exemple parce qu’ils covoiturent)
“PTHI”[*] “nom1”	<i>chaîne</i>	Nom du participant dont on veut forcer la tranche horaire
“PTHI”[*] “prenom1”	<i>chaîne</i>	Prénom du participant dont on veut forcer la tranche horaire
“PTHI”[*] “club1”	<i>chaîne</i>	Club du participant dont on veut forcer la tranche horaire
“PTHI”[*] “nom2”	<i>chaîne</i>	Nom du second participant auquel le premier sera rattaché
“PTHI”[*] “prenom2”	<i>chaîne</i>	Prénom du second participant auquel le premier sera rattaché

Clef	Type valeur	Détails
“PTHI”[*] “club2”	<i>chaîne</i>	Club du second participant
“PTHI”[*] “etapes”	<i>liste de nombres entiers</i>	Liste de numéros d’étapes pour lesquelles le rattachement doit être fait (par exemple [3,4,5])

## Etapes

Clef	Type valeur	Détails
“Etapes”	<i>liste d’éléments complexes</i>	Liste, donc entre crochets et séparées par des virgules, des différentes étapes constituant l’évènement
“Etapes”[n] “Nom”	<i>chaîne</i>	Nom de l’étape. Sera utilisé dans le rapport généré.
“Etapes”[n] “Lieu”	<i>chaîne</i>	Lieu de l’étape. Sera utilisé dans le rapport généré.
“Etapes”[n] “ZeroDate”	<i>chaîne</i>	Heure zéro de la GEC. MeOS demande de générer les heures de départ par rapport à cette heure zéro. L’heure doit être rentrée sous un format complet AAAA/MM/JJThh:mm, par exemple “2019/07/07T8:00”
“Etapes”[n] “Information”	<i>chaîne</i>	Commentaire destiné au relecteur du fichier JSON. Non utilisé par le script.
“Etapes”[n] “Format”	<i>chaîne</i>	Un des formats de course (par exemple “PedestreMDLD”) défini dans la section ranking IOF. Cette information n’est utilisée que si un des circuits est au ranking.
“Etapes”[n] “Circuits”	<i>liste d’éléments complexes</i>	Liste, donc entre crochets et séparées par des virgules, des différents circuits de l’étape
“Etapes”[n] “Circuits”[m] “Nom”	<i>chaîne</i>	Nom du circuit tel qu’il apparaîtra dans la GEC

Clef	Type valeur	Détails
“Etapes”[ <i>n</i> ] “Circuits”[ <i>m</i> ] “Depart”	<i>chaîne</i>	Nom du départ correspondant à ce circuit (exemple “red” ou “1”)
“Etapes”[ <i>n</i> ] “Circuits”[ <i>m</i> ] “Horaires”	<i>chaîne</i>	Type de génération d’horaire. Les valeurs reconnues sont définies dans le tableau ci-dessous.
“Etapes”[ <i>n</i> ] “Circuits”[ <i>m</i> ] “HeureDepart”	<i>chaîne</i>	Heure des premiers départ de ce circuit. L’heure doit être rentrée sous un format complet AAAA/MM/JJThh:mm, par exemple “2019/07/07T9:00”. Ce champ peut être utilisé par exemple pour faire partir des circuits sur les minutes paires, et d’autres sur la minute impaire.
“Etapes”[ <i>n</i> ] “Circuits”[ <i>m</i> ] “Ecart”:	<i>chaîne</i>	Temps minimal, exprimé en minute, séparant deux départs consécutifs sur ce circuit
“Etapes”[ <i>n</i> ] “Circuits”[ <i>m</i> ] “Categories”	<i>liste de chaînes</i>	Liste de catégories sur ce circuit entre double quotes et séparées par des virgules. Les catégories doivent être celle qui seront utilisées dans la GEC. Exemple : [“W21A”, “W35”]

#### Valeurs possibles pour le champ *Horaires*

valeur	Signification
“oui”	Les horaires de départ seront attribués automatiquement en tenant compte des groupes horaires des clubs. Les concurrents de ce circuit seront inscrits dans la GEC <b>avec</b> leur heure de départ.
“rank”	Les horaires de départ seront attribués automatiquement en fonction du ranking. Les concurrents de ce circuit seront inscrits dans la GEC <b>avec</b> leur heure de départ.
“man”	Les horaires de départ seront affectés manuellement par la GEC avant l’épreuve. Les concurrents de ce circuit seront inscrits dans la GEC <b>sans</b> heure de départ.

valeur	Signification
“boi”	Les départs de ce circuit s’effectueront au boîtier. Les concurrents de ce circuit seront inscrits dans la GEC <b>sans</b> heure de départ.
“non”	Les concurrents de ce circuit <b>ne seront pas gérés</b> par la GEC

**Remarques** - Il est possible de définir plusieurs fois le même circuit, par exemple si certaines catégories sur ce circuit sont gérées par la GEC et d’autres ne le sont pas (loisirs). Attention toutefois à la gestion des départs pour s’assurer qu’il n’y aura pas de départs simultanés.

## Algorithmes utilisés

### Affectation des groupes horaires aux clubs

Les groupes horaires sont affectés aux clubs de la façon suivante :

1. Le nombre de groupe horaire correspondant au nombre de tranches horaire est créé (voir section “*TranchesHoraires*” “*NbTranches*” dans le fichier JSON)
2. Les clubs faisant parti de la liste des groupes imposés (voir section “*Club-GroupeHoraireForce*” dans le fichier JSON) sont affectés aux groupes horaires spécifiés.
3. Les clubs sont triés par taille décroissante. Tous les *gros* clubs (voir section “*TranchesHoraires*” “*SeuilClub*” dans le fichier JSON) sont successivement affectés au groupe horaire ayant l’effectif minimal
4. Les clubs des *petits* pays (voir section “*TranchesHoraires*” “*SeuilPays*” dans le fichier JSON) sont regroupés par pays et sont tous affectés au même groupe horaire. Les pays sont triés par taille décroissante et chaque pays est successivement affecté au groupe horaire ayant l’effectif minimal
5. Les clubs restants (c’est à dire les *petits* clubs des *grands* pays) sont triés par taille décroissante et sont successivement affectés au groupe horaire ayant l’effectif minimal
6. enfin les participants restants, s’il y en a, sont successivement affectés au groupe horaire ayant l’effectif minimal

En affectant en priorité les clubs au groupe horaire ayant un effectif minimal, cet algorithme conduit à des groupes horaires équilibrés.

### Génération des dossards

Les dossards sont générés par ordre croissant, dans l’ordre alphabétique des noms de club, à partir du numéro défini par la section “*Dossards*” “*PremierDossard*” du fichier JSON.

Pour chaque club, la numérotation commence à la dizaine suivante, tout en assurant un minimum de deux numéros vides afin d’insérer d’éventuelles inscriptions tardives.

Un certain nombre de dossards supplémentaires vides (mais numérotés) sont ensuite ajoutés. Ils sont destinés à être utilisés pour les changements ou les inscriptions tardives sur place. Le nombre de dossards ajouté est défini par “*Dossards*” “*DossardsSupplémentaires*”.

On préférera probablement ne pas utiliser cette dernière fonctionnalité (en mettant la valeur à 0) et imprimer des dossards vierges afin de pouvoir donner un numéro de dossard aux inscriptions tardives qui soit dans la continuité de ceux de leur club.

## Affectation des tranches horaires aux compétiteurs

Les tranches horaires des compétiteurs, pour chacune des étapes est définie de la façon suivante :

1. Si le compétiteur est sur un circuit au ranking, il est affecté automatiquement à la première tranche horaire
2. Sinon, si le compétiteur figure dans la section “ParticipantsTranchesHoraires-Identiques” du fichier JSON pour l’étape considéré, la tranche horaire du participant auquel il doit être rapproché lui est affectée
3. Sinon, si le compétiteur a une tranche horaire imposé par la section “ParticipantsTranchesHorairesForcees” du fichier JSON pour l’étape considéré, cette tranche horaire lui est affectée
4. Sinon, si le compétiteur figure dans la section “ParticipantsTranchesHorairesAutreClub” du fichier JSON pour l’étape considéré, la tranche horaire correspondant au groupe horaire du club auquel il doit être rattaché lui est affectée
5. Sinon, le compétiteur est affecté à la tranche horaire correspondant au groupe horaire de son club

## Détermination des heures de début et de fin des tranches horaires

Pour chacune des étapes on procède de la façon suivante :

### Première tranche

1. L’heure de début de la première tranche est l’heure minimale de toutes les heures de départ déclarées dans la section “*Etapes*”[n] “*Circuits*”[m] “*HeureDepart*” du fichier JSON pour l’étape considérée.

2. Pour chaque circuit, l'heure de fin minimale est déterminée à partir de l'heure de début définie dans *"Etapes"[n] "Circuits"[m] "HeureDepart"*, du nombre de participants sur le circuit et de l'intervalle entre départs définie par *"Etapes"[n] "Circuits"[m] "Ecart"*.
3. A la plus tardive de ces heures minimales, est rajouté une marge définie par la section *"TranchesHoraires" "MargeTranches"* du fichier JSON. Ceci définit l'heure de fin de la tranche horaire.

### Tranches horaires suivantes

1. Pour les autres tranches, l'heure de début est définie comme l'heure de fin de la tranche précédente à laquelle on ajoute 1 minute.
2. L'heure de début d'un circuit tient compte de l'intervalle de départ depuis l'heure de départ déclarée pour le circuit. Ainsi si un circuit a un intervalle de 2 minutes et un premier départ sur une heure impaire, tous les départs, de toutes les tranches, seront sur des heures impaires.
3. Pour chaque circuit, l'heure de fin minimale est déterminée à partir de l'heure de début définie ci-avant, du nombre de participants sur le circuit et de l'intervalle entre départs définie par *"Etapes"[n] "Circuits"[m] "Ecart"*.
4. A la plus tardive de ces heures minimales, est rajouté une marge définie par la section *"TranchesHoraires" "MargeTranches"* du fichier JSON. Ceci définit l'heure de fin de la tranche horaire considérée.

### Détermination des horaires des compétiteurs

Les heures des compétiteurs sont définis de la façon suivantes :

**Circuits non gérés** Les compétiteurs qui sont sur un circuit pour lequel le champ *"Etapes"[n] "Circuits"[m] "Horaires"* du fichier JSON est différent de "oui" et "rank" sont ignorés

**Circuits au ranking** Les compétiteurs qui sont sur un circuit pour lequel le champ *"Etapes"[n] "Circuits"[m] "Horaires"* est "rank" sont affectés successivement à partir de l'heure de départ du circuit déclaré dans la section *"Etapes"[n] "Circuits"[m] "HeureDepart"* en respectant l'intervalle de temps défini par *"Etapes"[n] "Circuits"[m] "Ecart"*. **Il n'y a pas de considération de club ou de pays pour définir l'ordre. Seul le classement compte.** L'ordre d'affectation est le suivant :

1. Les compétiteurs ayant ni CN ni ranking IOF sont affectés dans un ordre aléatoire à partir de l'heure de début
2. Puis les compétiteurs ayant un CN mais pas de ranking IOF sont affectés dans l'ordre des CN croissants (le meilleur CN à la fin)

3. Enfin, les compétiteurs ayant un ranking IOF sont affectés dans l'ordre de leurs points IOF pour le type d'épreuve considéré (les rankings IOF peuvent être différents par exemple pour un sprint et une LD). Le compétiteur ayant le plus grand nombre de points IOF pour le type d'épreuve considéré part en dernier.

**Circuit normaux** Les compétiteurs qui sont sur un circuit pour le quel le champ "*Etapes*"[n] "*Circuits*"[m] "*Horaires*" est "oui" sont affectés de la façon suivante :

1. L'affectation s'effectue dans l'ordre des circuits décroissants en effectifs (on commence donc par le circuit ayant le plus grand nombre de départs dans la tranche horaire considérée)
2. Les compétiteurs du circuit considéré sont dans un premier temps affectés aléatoirement à un horaire, dans leur tranche horaire, compatible avec les personnes déjà affectées, c'est à dire respectant l'heure de début du circuit, l'intervale de temps et en vérifiant qu'il n'y a pas une personne du même club sur le même circuit sur l'horaire de départ précédant ou suivant.
3. Cette répartition purement aléatoire conduit à des pics de charge pour l'atelier départ. L'algorithme essaie alors, pour chaque atelier départ, de lisser la charge en déplaçant les compétiteurs des créneaux horaires les plus chargés vers les moins chargés, tout en respectant les critères précédents.

## Evolutions

### Autres logiciel de GEC

Actuellement *co\_depart* prépare évidemment des fichiers CSV pour *MeOS* (What else ?), mais vous pouvez très facilement adapter le programme pour générer des fichiers CSV - ou autres - adaptés à d'autres logiciels.

Pour cela nous vous conseillons :

- de changer la valeur de la clef "*FichiersGeneres*" / "*GEC*" dans le fichier JSON.
- d'écrire votre code en remplacement de :

```
else: # format pour d'autres logiciels de GEC à écrire ici
```

dans les fonctions *genereCSV* et *genereLigneCSV* aux alentours des lignes 450, en vous inspirant de ce qui a été fait pour MeOS.

## Licence d'utilisation

*co\_depart* est un script *Python* open source distribué sous licence GNU GPL.



Pour plus de détails voir : <https://www.gnu.org/licenses/gpl-3.0.html>

## Auteurs

- Pierre Pardo pour le code initial (04/2019)
- Jérôme Monclard pour la ré-écriture du code, l'ajout de nombreuses fonctionnalités et la documentation (04/2019 - 06/2019)
- Richard Heyriès pour les tests (04/2019 - 06/2019)

## Détails d'implémentation

Ce chapitre donne des détails d'implémentation à l'intention des personnes amenées à faire évoluer ce programme.

Sa lecture n'est pas nécessaire pour l'utilisation du programme.

### Structure de données

#### Clubs :

Champ	Description
nomcomplet	Nom complet du club. Exemple : "A.B.C.O. DIJON 2101BF"
refclub	Code FFCO du club. Exemple "2101BF"
effectif	Nombre de participants pour ce club
groupe	Groupe horaire (correspond à la tranche horaire du premier jour)
flag	Code ISO 3166-1 alpha-3 du pays (déterminé à partir de la nationalité du premier membre rencontré si non forcé dans le JSON)
dossardmin	Preemier numéro de dossard utilisé pour ce club
dossardmax	Dernier numéro de dossard utilisé pour ce club
dossardultramax	Dernier numéro de dossard utilisé pour ce club en prenant en compte la réserve pour inscriptions au dernier moment

#### Competiteurs :

Champ	Description
prenom	Prénom
nom	Nom de famille
sexe	Sexe
an	Années de naissance
flag	Code ISO 3166-1 alpha-3 de nationalité
num_club	Numéro FFCO du club auquel appartient le compétiteur. Exemple “2101”
ref_club	Numéro FFCO complet du club auquel appartient le compétiteur. Exemple “2101BF”
club	Nom du club auquel appartient le compétiteur. Exemple “A.B.C.O. DIJON”
num_lic	Numéro de licence FFCO du compétiteur
etapes	Liste des étapes auxquelles participe le compétiteur. Exemple [1,2,4]
categorie	Catégorie du compétiteur. Exemple : “W21E”
puce_si	Numéro de puce SI
modele_tshirt	Modèle de t-shirt souhaité. Exemples “ZIP” ou “COL V”
taille_tshirt	Taille du T-shirt souhaité. Exemple “XL”
iofid	Identifiant IOF du compétiteur
dossard	Numéro de dossard
heure_dep	Liste d’horaires de départ pour les différentes étapes. Exemple [“10:56”, “13:01”, “13:04”, “09:38”, None]
tranches	Liste de tranches horaires de départ pour les différentes étapes. Exemple [4,1,2,3,None]
posdepart	Liste de nom de départs pour les différentes étapes. Exemple [“red”, “blue”, “blue”, “blue”, “blue”]

Champ	Description
pointscn	Points CN du compétiteur
pointsiofsprint	Points au classement IOF pour le format sprint
pointsiofmdld	Points au classement IOF pour les formats MD et LD
pointsiofmtbo	Points au classement IOF pour le format orientation à VTT
pointsiofskio	Points au classement IOF pour le format orientation à ski
pointsioftrailo	Points au classement IOF pour le format orientation de précision
circuits	Liste de nom de circuits pour les différentes étapes. Exemple ["B","B","C","C","C"]
horaires	Type d'horaires pour les différentes étapes. Exemple ["rank","rank","oui","oui","man"]

### Circuits

Champ	Description
Nom	Nom du circuit (ex. 'A')
Depart	Nom du départ (ex. bleu, rouge)
Horaires	Méthode de génération d'horaire (ex. 'rank')
HeureDepart	Heure des premiers départs (ex. '2019/07/07T09:00')
Ecart	Ecart entre deux départs consécutifs (ex. 2)
Categories	Liste des catégories du circuit (ex. ['M21E'])