

# Guide Détaillé des Options gwexport

Date de création : 11 septembre 2025 Outil : gwexport - Sélection générique pour l'export Version analysée : master

---

## Table des matières

1. Vue d'ensemble
  2. Modèle d'options et usages
  3. Options détaillées
  4. Sélection: algorithmes et filtres
  5. Intégrations: gwb2ged, gwu, autres
  6. Exemples
  7. Dépannage
  8. Bonnes pratiques
- 

## 1. Vue d'ensemble

Le module gwexport fournit les options et la logique de sélection génériques pour les outils d'export GeneWeb (par exemple gwb2ged). Il permet de définir un sous-ensemble d'individus et de familles à exporter à partir d'une base, selon des critères tels que ascendance, descendance, paires de parenté, patronymes, censure, etc.

---

## 2. Modèle d'options et usages

Type d'options (simplifié): - asc, desc, ascdesc: profondeurs de sélection autour de racines (clés) - keys: clés de personnes racines (format: "Prenom.occ NOM") - censor: censure basée sur l'âge - surnames: sélection par patronymes - parentship: sélection par parenté entre paires de clés - charset: encodage sortie (valeurs possibles ASCII, ANSEL, ANSI, UTF-8) - mem, no\_notes, no\_picture, picture\_path, source, verbose, oc: champs utilisés par les outils consommateurs (ex: gwb2ged), pas tous appliqués dans gwexport lui-même

Erreur par défaut: errmsg = "Usage: <prog> <BASE> [OPT]"

---

## 3. Options détaillées

- -a <N>: nombre maximum de générations d'ascendants à sélectionner depuis les racines

- **-ad <N>**: nombre maximum de générations d'ascendants, puis sélection des descendants de ces ascendants (mascdesc)
- **-key <KEY>**: ajoute une clé de personne racine (utilisable plusieurs fois); format: `Prenom.occ NOM`
- **-c <NUM>**: censure; personnes nées il y a moins de NUM années (et non Public) ne sont pas exportées; conjoints et descendants de personnes censurées sont aussi censurés
  - Valeur spéciale: **-c -1** active un mode de restriction basé sur la visibilité (voir plus bas)
- **-charset [ASCII|ANSEL|ANSI|UTF-8]**: encodage de sortie (UTF-8 par défaut)
- **-d <N>**: nombre maximum de générations de descendants à sélectionner depuis les racines
- **-mem**: économiser la mémoire (utilisation côté outil; gwexport n'optimise pas directement ici)
- **-nn**: pas de notes de base (utilisation côté outil)
- **-nnn**: pas de notes du tout (implique **-nn**) (utilisation côté outil)
- **-nopicture**: ne pas extraire d'images (utilisation côté outil)
- **-o <FILE>**: fichier de sortie (utilisation côté outil)
- **-parentship**: sélection par parenté à partir de paires de clés fournies via **-key** (descendant d'abord, puis ancêtre). Si plusieurs paires, l'union des personnes est renvoyée
- **-picture-path**: inclure les chemins d'images (utilisation côté outil)
- **-s <SN>**: patronyme; peut être répété; la sélection est l'union des patronymes
- **-source <SRC>**: remplacer sources individuelles/familiales (utilisation côté outil)
- **-v**: verbeux (utilisation côté outil)

---

## 4. Sélection: algorithmes et filtres

### 4.1 Clés des racines

- Les racines explicites sont prises depuis les **ips** passées à **select** et enrichies par les clés **-key**.

### 4.2 Censure

- Si **-c** est non nul:
  - **-c -1**: restreint selon la visibilité de la base: marque censurés les personnes non visibles pour le visiteur et les familles dont tous les membres (parents et enfants) sont censurés.
  - **-c N**: calcule un seuil d'année (`annee_courante - N`). Toute personne née après ce seuil et non Public est censurée, ainsi que certains proches (couples et descendants). Les marqueurs (`per_tab/fam_tab`)

- évitent les duplications.
- Les filtres finaux excluent les éléments marqués censurés.

#### 4.3 Portée asc/desc/ascdesc

- Si `-ad` ou `-d` est fourni, `-c` doit être nul (contrainte dans le code). La sélection est:
  - `-ad <A>`: pour chaque racine, prend ses ascendants jusqu'à `-a` (ou illimité si `-a` absent), puis les descendants de ces ascendants jusqu'à `-ad` (mascdesc).
  - `-d <D>`: descendants jusqu'à D générations (D converti en profondeur négative interne).
  - Si `-ad` omis mais `-a` présent: uniquement ascendants jusqu'à A; familles retenues si père et mère sont dans l'ensemble.
- Les familles sont ensuite ajoutées si l'autre conjoint de la famille est aussi dans l'ensemble des personnes retenues.

#### 4.4 Patronymes (-s)

- Sélectionne les familles où le père ou la mère a le patronyme demandé (normalisé en minuscules sans accents), marque ces familles et leurs parents. Marque les enfants qui portent ce patronyme exact.
- Deux fonctions résultantes: `per_sel` i vrai si la personne est marquée; `fam_sel` i vrai si la famille est marquée.

#### 4.5 Parenté (-parentship)

- Requiert des clés par paires dans `-key`: descendant puis ancêtre.
- Calcule ascendants du descendant et l'ensemble des descendants de l'ancêtre; retourne l'intersection (personnes communes) et les familles où les conjoints sont tous deux dans l'ensemble retenu.
- En cas de multiples paires, calcule l'union des ensembles résultants.

#### 4.6 Cas par défaut

- Si aucune option de portée, patronyme ou parenté n'est fournie: sélectionne tout (filtres retournent true).

---

### 5. Intégrations: gwb2ged, gwu, autres

- **gwb2ged**: étend `speclist` avec `-indexes` et utilise `select` pour filtrer les individus/familles exportés; applique aussi `-charset`, `-nn`, `-nnn`, `-picture-path`, `-source`, `-mem`, `-o`.
- **gwu**: réutilise `speclist` pour la sélection (filtres), puis applique ses propres sorties.

- Autres outils: peuvent combiner `speclist` et `select` pour obtenir la même sémantique de sélection.
- 

## 6. Exemples

### 6.1 Racines et descendants

```
# Exporter 2 générations de descendants depuis une personne racine
<prog> base -key "Jean.0 MARTIN" -d 2 -o out.ext
```

### 6.2 Ascendants puis tous les descendants

```
# 3 générations d'ascendants, puis descendants de ces ascendants
<prog> base -key "Anne.0 DURAND" -a 3 -ad 99 -o out.ext
```

### 6.3 Censure

```
# Censure récente: ne pas inclure personnes nées depuis 100 ans (non Public)
<prog> base -c 100 -o out.ext
```

```
# Restriction par visibilité
<prog> base -c -1 -o out.ext
```

### 6.4 Parenté

```
# Personnes sur les chemins de parenté entre paires descendant -> ancêtre
<prog> base -parentship \
  -key "Descendant.0 NOM" \
  -key "Ancetre.0 NOM" \
  -o out.ext
```

### 6.5 Patronymes

```
# Sélection par patronymes multiples
<prog> base -s "MARTIN" -s "DURAND" -o out.ext
```

---

## 7. Dépannage

- `bad -charset value:` encodage non pris en charge; utiliser ASCII, ANSEL, ANSI ou UTF-8.
- Aucun résultat avec `-parentship`: vérifier l'ordre des clés (`-key` descendant d'abord, puis ancêtre) et la présence des personnes dans la base.
- Résultat vide avec `-c`: vérifier la valeur et le caractère Public des individus; `-c -1` dépend des règles de visibilité de la base.

---

## 8. Bonnes pratiques

- Fournir explicitement des **-key** quand on utilise **-a**, **-d**, **-ad**.
- Préférer UTF-8 pour compatibilité moderne; ANSEL pour anciens logiciels GEDCOM.
- Limiter la portée (générations) pour de meilleures performances.
- Combiner **-s** et **-parentship** pour des exports ciblés (familles et liens précis).