

# Formats d'import/export - Format v1.0



Accéder à la page de [discussions sur le format d'import/export](#) en cliquant sur l'icône  de la barre d'outil à droite.

**Objectif** : proposer un format d'échange de données **simple** (autant que possible), bien **documenté**, **pratique**, **facilitant l'intégration des données sous GeoNature** et respectant le **standard OccTax** du SINP.

## Processus d'importation



Ce document est en cours de travail. Les informations qu'il contient peuvent donc être amenées à changer à tout moment !

## Transmission des fichiers

Les fichiers seront transmis dans un fichier d'archive au format ZIP. Le nom du fichier devra être en minuscule et contenir plusieurs parties séparées par des underscores ("\_"). Les parties du fichier seront les suivantes :

1. date au format [ISO 8601](#) : 2020-08-26
2. sujet : sinp
3. abréviation de la région concernée : paca / aura
4. abréviation de l'organisme fournisseur : cbna / cbnmed / cbnmc / cenpaca ...
5. extension : .zip

Exemple : *2020-08-26\_sinp\_paca\_cbna.zip*

L'archive devra contenir au minimum les fichiers suivants :

- **meta\_archive.ini** : fichier contenant les métadonnées sur le fournisseur de l'archive et la version du format d'échange utilisée.
- **synthese.csv** : les données d'observation à intégrer à la synthèse sont les seules requises.

Les fichiers suivants pourront aussi être fournis sans être obligatoires :

- *additional\_data.csv* : contient la description des champs additionnels présents dans les différents fichiers à importer.
- *source.csv* : permet de décrire la source des données. Correspond à la table "gn\_synthese.t\_sources".
- *dataset.csv* : permet de fournir les informations sur les jeux de données. Correspond à la table "gn\_meta.t\_datasets".
- *acquisition\_framework.csv* : permet de fournir les informations sur les cadres d'acquisition des

jeux de données. Correspond à la table "*gn\_meta.t\_acquisition\_framework*".

- *organism.csv* : permet de fournir les informations sur les organismes liées aux utilisateurs, jeux de données et cadres d'acquisition. Correspond à la table "*utilisateurs.bib\_organismes*".
- *user.csv* : permet de fournir des informations sur les utilisateurs à l'origine des données. Correspond à la table "*utilisateurs.t\_roles*".

## Principe de la procédure d'import initial

- L'import initial peut concerner plusieurs millions de données. Elles seront transmises sous forme de fichiers textes.
  - Nous utiliserons plusieurs fichiers textes respectant tous le même format CSV (voir ci-dessous)
  - Le détail des différents fichiers (*user*, *organism*, *acquisition\_framework*, *dataset*, *source*, *synthese*) est présentés ci-dessous
  - Un seul fichier est obligatoire *synthese* mais les autres sont nécessaire pour éviter une saisie manuelle sur GeoNature des métadonnées et sur UsersHub des utilisateurs et organismes.
  - Dans ces fichiers, nous utiliserons les codes alphanumériques des valeurs pour les champs devant contenir des **identifiant de nomenclatures**. Un script Python se connectant à la base de GeoNature permettra de récupérer l'identifiant numérique spécifique à une instance de base de données.
  - Pour les lien avec les **organismes, cadres d'acquisition, jeux de données et sources**, nous utiliserons le même principe que pour les nomenclatures. Les liens se feront sur les codes ou noms et le script Python permettra de récupérer les identifiants spécifiques à chaque instance de base de données. Les champs sur lesquels les liens sont basés peuvent être amené à évoluer dans la base d'origine. Lors d'une mise à jour nous appuierons aussi sur les champs "*unique\_id*" de ces entités (s'ils sont fournis) pour retrouver la correspondance avec un enregistrement en base de données. Il est donc important que les bases de données sources stockent l'UUID des ces entités où ne permettent pas la modification des champs servant de lien...
- L'import des fichiers se fera à l'aide d'un script Bash exécutant des scripts SQL et Python.
  - Un script Python sera chargé de vérifier les données et de remplacer les différents codes standards par leur identifiant numérique spécifique à la base de données courante.
  - Des scripts SQL se chargeront de désactiver triggers, contraintes... puis de les réactiver et exécuter globalement après la commande de *COPY* d'insertion du fichier lancée.
- De cette façon, nous pouvons espérer intégrer jusqu'à 3 millions d'observations en moins d'une heure dans la table *synthese* de GeoNature.

## Principe de la procédure de mise à jour

- Pour chaque mise à jour, nous devons importer seulement un différentiel :
  - le différentiel sera réalisé en local sur une machine disposant d'assez de mémoire et d'un disque dur de type SSD NVMe de façon à réduire au maximum les temps de traitement. Idéalement, le différentiel doit pouvoir être extrait des bases de données d'origines.
  - le nombre moins important de données nous permettra de réaliser rapidement la mise à jour de la table *synthese* de GeoNature sans gêner son utilisation via les interface web.
  - le différentiel sera fournie au format CSV (voir ci-dessous)
  - le fichier sera importé dans une première table d'import à partir de laquelle nous réaliserons les requêtes d'import dans la table *synthese*

- Nous procéderons pour l'import du différentiel dans la table "gn\_synthese.synthese" à partir de la table d'import en 3 étapes :
  - Ajout des nouvelles observations
  - Modification des observations existantes qui ont été modifiée depuis le dernier import
  - Suppression des observations qui ne sont plus présente dans l'import courant. Dans le cas du différentiel, les lignes supprimées doivent être présentes dans l'import.
- Pour réaliser le différentiel, nous pouvons utiliser :
  - La clé primaire des données d'origine (champ "entity\_source\_pk\_value") pour vérifier si la données existe déjà ou pas dans la table "gn\_synthese.synthese". Idéalement, il faudrait aussi y ajouter le champ "code\_source" et/ou "code\_dataset".
  - Le champ "last\_action" permettra d'identifier les lignes ajoutées depuis le dernier import ("I"), modifiées ("U") ou supprimées ("D").
  - Dans le cas des observations modifiées, le champ "meta\_update\_date" permettra de déterminer la version de la modification. Une date plus récente dans la table d'import indiquera la nécessité de mettre à jour l'enregistrement.
- Pour les champs contenant des **identifiant de nomenclatures**, une fonction Postgresql présente dans la base de GeoNature permettra de récupérer l'identifiant numérique spécifique à chaque instance de base de données et correspondant au code fournie dans le CSV. Les codes de nomenclature à utiliser sont disponibles dans le champ "cd\_nomenclature" de la table "ref\_nomenclatures.t\_nomenclatures".
- Pour les lien avec les **organismes, cadres d'acquisition, jeux de données et sources**, nous utiliserons le même principe que pour les nomenclatures. Les liens se feront sur les codes ou noms et des fonctions permettront de récupérer les identifiants spécifiques à chaque instance de base de données.
- L'ordre d'import des tables devrait être le suivant : organism, user, acquisition\_framework, dataset, source, synthese.

## Format du fichier des métadonnées de l'archive "meta\_archive.ini"

Ce fichier au format INI a pour objectif de fournir les informations sur l'origine des autres fichiers fournis dans l'archive.

Il concerne le créateur de l'archive.

Ce fichier devra:

- être encodé en **UTF-8**
- être nommé (en minuscule) : **meta\_archive.ini**

Les règles à respecter pour ce format INI sont le suivantes :

- une ligne peut contenir soit un commentaire débutant par le caractère **#** ou une entrée **clé / valeur**
- la clé doit être séparé de sa valeur par un **=**
- les clés doivent être en minuscule et utilisé l'underscore ( **\_** ) comme séparateur de mots
- des espaces peuvent encadrer les clés et valeurs (ils seront supprimés).
- Si la valeur contient plusieurs lignes, encadrée là par des guillemets doubles ( **"** ) et indenter les ligne supplémentaires.

Format (en grans les champs obligatoires) :

- **format\_version** [VARCHAR(8)] : version du format d'échange utilisé pour les fichiers à

importer.

- `export_date` [DATE(YYYY-MM-DD HH:MM)] : date et heure de l'export des observations de la synthèse hors de la base d'origine.
- `taxref_version` [VARCHAR(8)] : version de TaxRef utilisée lors de la génération de l'archive.
- `habref_version` [VARCHAR(8)] : version de HabRef utilisée lors de la génération de l'archive.
- **editor** [VARCHAR(100)] : nom de l'organisme créateur de l'archive.
- **contact** [VARCHAR(100)] : infos sur la personne ayant créé l'archive. Format : NOM Prénom <email>.
- `notes` [TEXT] : remarques divers sur les fichiers de l'archive.

Exemple :

```
format_version = 1.0
export_date = 2020-08-27 10:15
editor = Conservatoire Botanique National Alpin
contact = jp.milcent@cbn-alpin.fr
notes = "Données de test.
        À utiliser seulement lors de la phase de conception."
```

## Format des fichiers d'import

Pour importer les données (initialement ou en mise à jour), nous utiliserons des fichiers [CSV](#) associé à la commande *COPY*. Ces fichiers CSV devront :

- être encodée en **UTF-8**
- avoir un nom au singulier, en minuscules et avec des underscores comme séparateur de mots.
- avoir l'extension *.csv*
- avoir un des noms suivant : *synthese.csv*, *source.csv*, *dataset.csv*, *acquisition\_framework.csv*, *organism.csv* ou *user.csv*

Le format CSV (en réalité plutôt [TSV](#)) qu'ils contiendront devra respecter les règles suivantes :

- utiliser une **tabulation** comme caractère de séparation des champs
- posséder une **première ligne d'entête** indiquant les noms des champs
- utiliser les caractères **\N** pour indiquer une valeur nulle (*NULL*) pour un champ
- si nécessaire utiliser le caractère **guillemet** (") pour préfixer et suffixer une valeur de champ
- si nécessaire utiliser **deux guillemets** successifs (") pour échapper le caractère guillemet dans une valeur de champ préfixé et suffixé par des guillemets.

## Champs additionnels

Le standard prévoit la transmission de champs additionnels.

Nous prévoyons la possibilité de fournir des champs additionnels pour toutes les types de ressources de 2 façons :

- si les champs ne sont pas valables pour toutes les lignes, 1 seule possibilité :
  - utiliser le champ *additional\_data* dont les valeurs doivent être [au format JSON](#) et être [valide](#). Le JSON en question devra correspondre à un objet d'un seul niveau avec attribut

et valeur. Ex. :

```
{"attribut1": "valeur1", "attribut2": "valeur2"}
```

- si les champs sont valables pour toutes les lignes du fichier d'import, 2 possibilités :
  - ajouter des colonnes supplémentaires au fichier
  - utiliser le champ *additional\_data* (voir ci-dessus)

Afin de respecter le standard, il est nécessaire de fournir les méta-données de ces champs additionnel en fournissant le fichier *meta\_additional\_data.csv* (voir META\_ADDITIONAL\_DATA). Ce fichier contient la descriptions des champs additionnels.

Pour faciliter l'intégration des données, l'utilisation du champ *additional\_data* est conseillée. Les champs additionnels ne seront pas forcément traités pour toutes les ressources. Seuls les champs de la ressource SYNTHESE seront transmis au niveau national.

## À faire / Améliorations

- Intégrer si nécessaire les nouveaux champs de GeoNature v2.6.0 aux formats (- @jpmilcent 2021-02-21) :
  - ACQUISITION\_FRAMEWORK : opened, id\_digitizer, initial\_closing\_date
  - DATASET : id\_nomenclature\_resource\_type, active, validable, id\_digitizer
  - SYNTHESE : sample\_number\_proof,
- Revoir le format des champs "cor\_actors\_organism" et "cor\_actors\_user" du format ACQUISITION\_FRAMEWORK. La récupération des valeurs de tableaux imbriqués nécessite de passer par du JSON. Il serait donc plus efficace d'utiliser directement le format JSON pour ces champs.- @jpmilcent 2021-02-21.
- Dans le format DATASET, indiquer les valeurs par défaut des nomenclatures à utiliser. Les champs correspondant dans la table GeoNature étant obligatoire ! Les nomenclatures ne sont pas obligatoires pour les cadres d'acquisition et la synthese.

## Format SYNTHESE d'import

- But : Permet de fournir les informations sur les observations.
- Table GeoNature : "*gn\_synthese.synthese*".
- Standard : [OccTax v2](#)
- Correspondance GeoNature/Standard : [gn\\_synthese.synthese](#) et [OccTax](#)

**NOTES** : nous n'utilisons par le champ *meta\_v\_taxref* car la transmission de l'info via le fichier *meta\_archive.ini* suffit.

## Description du format SYNTHESE

Pour chaque ligne : nom\_du\_champ [format du champ] (=nom\_champ\_table\_geonature) : description du champ.. Les champs **en gras** sont obligatoires. Pour les nomenclatures, le nom de la mnémonique du type dans GeoNature est indiqué en italique en fin de description. La correspondance avec le nom de cette nomenclature dans le standard OccTax est indiqué entre parenthèses et un lien pointe vers ses valeurs sur le site [Standards d'échanges du SINP](#). Pour les

nomenclatures, la valeur à transmettre est celle présente dans la colonne "**Code**" du standard qui est équivalente au champ `cd_nomenclature` de la table `ref_nomenclatures.t_nomenclatures` de GeoNature.

- `unique_id_sinp` [UUID] : UUID SINP s'il existe déjà dans les données sources.
- `unique_id_sinp_grp` [UUID] : UUID SINP du relevé s'il existe déjà dans les données sources.
- `source_id` [VARCHAR(25)] (= `entity_source_pk_value`) : code alphanumérique correspondant à l'équivalent d'une clé primaire pour les occurrences des données sources.
- `source_id_grp` [VARCHAR(25)] (Pas dans `synthese`) : code alphanumérique correspondant à l'équivalent d'une clé primaire pour les relevés des données sources.
- **code\_source** [VARCHAR(255)] (= `id_source`) : code alphanumérique permettant d'identifier la source des données. Sert de lien avec la ressource `SOURCE` et son champ "`name`".
- **code\_dataset** [VARCHAR(255)] (= `id_dataset`) : code alphanumérique permettant d'identifier le jeu de donnée de l'observation. Sert de lien avec la ressource `DATASET` et son champ "`shortname`".
- `code_nomenclature_geo_object_nature` [VARCHAR(25)] (= `id_nomenclature_geo_object_nature`) : code alphanumérique de la valeur du type de nomenclature `NAT_OBJ_GEO` (`NatureObjetGeoValue`).
- `code_nomenclature_grp_typ` [VARCHAR(25)] (= `id_nomenclature_grp_typ`) : code alphanumérique de la valeur du type de nomenclature `TYP_GRP` (`TypeRegroupementValue`).
- `grp_method` [VARCHAR(255)] : description de la méthode ayant présidé au regroupement, de façon aussi succincte que possible : champ libre.
- `code_nomenclature_obs_technique` [VARCHAR(25)] (= `id_nomenclature_obs_technique`) : code alphanumérique de la valeur du type de nomenclature `METH_OBS` (`ObservationTechniqueValue`).
- `code_nomenclature_bio_status` [VARCHAR(25)] (= `id_nomenclature_bio_status`) : code alphanumérique de la valeur du type de nomenclature `STATUT_BIO` (`OccurrenceStatutBiologiqueValue`).
- `code_nomenclature_bio_condition` [VARCHAR(25)] (= `id_nomenclature_bio_condition`) : code alphanumérique de la valeur du type de nomenclature `ETA_BIO` (`OccurrenceEtatBiologiqueValue`).
- `code_nomenclature_naturalness` [VARCHAR(25)] (= `id_nomenclature_naturalness`) : code alphanumérique de la valeur du type de nomenclature `NATURALITE` (`OccurrenceNaturaliteValue`).
- `code_nomenclature_exist_proof` [VARCHAR(25)] (= `id_nomenclature_exist_proof`) : code alphanumérique de la valeur du type de nomenclature `PREUVE_EXIST` (`PreuveExistanteValue`).
- `code_nomenclature_valid_status` [VARCHAR(25)] (= `id_nomenclature_valid_status`) : code alphanumérique de la valeur du type de nomenclature `STATUT_VALID` (`NiveauValidationValue`).
- `code_nomenclature_diffusion_level` [VARCHAR(25)] (= `id_nomenclature_diffusion_level`) : code alphanumérique de la valeur du type de nomenclature `NIV_PRECIS` (`NiveauPrecisionValue`).
- `code_nomenclature_life_stage` [VARCHAR(25)] (= `id_nomenclature_life_stage`) : code alphanumérique de la valeur du type de nomenclature `STADE_VIE` (`OccurrenceStadeDeVieValue`).
- `code_nomenclature_sex` [VARCHAR(25)] (= `id_nomenclature_sex`) : code alphanumérique de la valeur du type de nomenclature `SEXE` (`OccurrenceSexeValue`).
- `code_nomenclature_obj_count` [VARCHAR(25)] (= `id_nomenclature_obj_count`) : code alphanumérique de la valeur du type de nomenclature `OBJ_DENBR` (`ObjetDenombrementValue`).
- `code_nomenclature_type_count` [VARCHAR(25)] (= `id_nomenclature_type_count`) : code alphanumérique de la valeur du type de nomenclature `TYP_DENBR` (`TypeDenombrementValue`).
- `code_nomenclature_sensitivity` [VARCHAR(25)] (= `id_nomenclature_sensitivity`) : code alphanumérique de la valeur du type de nomenclature `SENSIBILITE` (`SensibiliteValue`).

- `code_nomenclature_observation_status` [VARCHAR(25)] (=id\_nomenclature\_observation\_status) : code alphanumérique de la valeur du type de nomenclature `STATUT_OBS` ([StatutObservationValue](#)).
- `code_nomenclature_blurring` [VARCHAR(25)] (=id\_nomenclature\_blurring) : code alphanumérique de la valeur du type de nomenclature `DEE_FLOU` ([dEEFloutageValue](#)).
- `code_nomenclature_source_status` [VARCHAR(25)] (=id\_nomenclature\_source\_status) : code alphanumérique de la valeur du type de nomenclature `STATUT_SOURCE` ([StatutSourceValue](#)).
- `code_nomenclature_info_geo_type` [VARCHAR(25)] (=id\_nomenclature\_info\_geo\_type) : code alphanumérique de la valeur du type de nomenclature `TYP_INF_GEO` ([TypeInfoGeoValue](#)).
- `code_nomenclature_behaviour` [VARCHAR(25)] (=id\_nomenclature\_behaviour) : code alphanumérique de la valeur du type de nomenclature `OCC_COMPORTEMENT` ([OccurrenceComportementValue](#)).
- `code_nomenclature_biogeo_status` [VARCHAR(25)] (=id\_nomenclature\_biogeo\_status) : code alphanumérique de la valeur du type de nomenclature `STAT_BIOGEO` ([StatutBiogeographiqueValue](#)).
- `reference_biblio` [VARCHAR(255)] : référence de la source de l'observation lorsque celle-ci est de type « Littérature », au format ISO690. La référence bibliographique doit concerner l'observation même et non uniquement le taxon ou le protocole.
- `count_min` [INT(4)] : nombre minimum d'individus du taxon composant l'observation.
- `count_max` [INT(4)] : nombre maximum d'individus du taxon composant l'observation. Mettre la même valeur que `count_min` si 1 seule valeur de dénombrement.
- `cd_nom` [INT(4)] : code du taxon en vigueur « cd\_nom » de TaxRef référençant au niveau national le taxon. Il peut être trouvé dans la colonne "CD\_NOM" de TaxRef.
- `cd_hab` [INT(4)] : code HABREF de l'habitat où le taxon de l'observation a été identifié. Il peut être trouvé dans la colonne "CD\_HAB" d'HabRef.
- **nom\_cite** [VARCHAR(1000)] : nom du taxon cité à l'origine par l'observateur.
- `digital_proof` [TEXT] : si possible une URL (idéalement [un permalien](#)) vers la preuve en ligne.
- `non_digital_proof` [TEXT] : adresse ou nom de la personne ou de l'organisme qui permettrait de retrouver la preuve non numérique de l'observation.
- `altitude_min` [INT(4)] : altitude minimum de l'observation en mètres.
- `altitude_max` [INT(4)] : altitude maximum de l'observation en mètres.
- `depth_min` [INT(4)] : profondeur Minimum de l'observation en mètres selon le référentiel des profondeurs indiqué dans les métadonnées (système de référence spatiale verticale).
- `depth_max` [INT(4)] : profondeur Maximale de l'observation en mètres selon le référentiel des profondeurs indiqué dans les métadonnées (système de référence spatiale verticale).
- `place_name` [VARCHAR(500)] : Nom du lieu ou de la station où a été effectuée l'observation.  
**ATTENTION** : cet attribut ne pourra pas être flouté !
- `geom` [geometry(Geometry,2154)] (=the\_geom\_local ⇒ the\_geom\_4326, the\_geom\_point) : géométrie détaillée de l'observation transféré au format [WKT](#). Utiliser la syntaxe Postgis pour indiquer le [SRID](#) utilisé au début du WKT (Ex. SRID=2154;POLYGON( (786000 6515000,786500 6515000,786500 6514500,786000 6514500,786000 6515000) ) ou SRID=2154;POINT(959204.517 6405928.812)). Utiliser préférentiellement le [SRID 2154](#) (à défaut le [SRID 4326](#)).
- `precision` [INT(4)] : estimation en mètres d'une zone tampon autour de l'objet géographique. Cette précision peut inclure la précision du moyen technique d'acquisition des coordonnées (GPS,...) et/ou du protocole naturaliste.
- **date\_min** [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure du jour, dans le système local de l'observation dans le calendrier grégorien. En cas d'imprécision, cet attribut représente la date la plus ancienne de la période d'imprécision. La date doit être écrite suivant la norme ISO8601. L'heure est dans le fuseau horaire de la zone d'observation. Ce champ est obligatoire et ne peut pas contenir de valeur nulle.

- **date\_max** [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure du jour, dans le système local de l'observation dans le système grégorien. Lorsqu'une observation est faite sur un jour, les dates et heures de début et de fin sont les mêmes (cas le plus courant). La date doit être écrite suivant la norme ISO8601. L'heure est dans le fuseau horaire de la zone d'observation. Ce champ est obligatoire et ne peut pas contenir de valeur nulle. Ce champ doit contenir une date supérieur ou égale à celle du champ "date\_min".
- **validator** [VARCHAR(1000)] : personne ayant procédé à la validation (et organisme). Voir [le détail du format à plat des infos sur une personne](#).
- **validation\_comment** [TEXT] : commentaire sur la validation.
- **validation\_date** [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure de validation de l'observation.
- **observers** [VARCHAR(1000)] : Nom, prénom, et organisme de la ou des personnes ayant réalisé l'observation. Voir [le détail du format à plat des infos sur une personne](#).
- **determiner** [VARCHAR(1000)] : Prénom, nom et organisme de la ou les personnes ayant réalisé la détermination taxonomique de l'observation. Voir [le détail du format à plat des infos sur une personne](#).
- **determination\_date** [DATE(YYYY-MM-DD HH:MM:SS)] : date/heure de la dernière détermination du taxon de l'observation dans le calendrier grégorien.
- **code\_digitiser** [VARCHAR(50)] : code alphanumérique permettant d'identifier l'utilisateur à l'origine de l'observation (=observer). Sert de lien avec la ressource USER et son champ "identifier".
- **code\_nomenclature\_determination\_method** [VARCHAR(25)] (=id\_nomenclature\_determination\_method) : code alphanumérique de la valeur du type de nomenclature METH\_DETERMINE (champ libre dans OccTax) ([Voir les codes disponibles](#)).
- **comment\_context** [TEXT] : description libre du contexte de l'observation, aussi succincte et précise que possible. Informations sur le lieu (=où), le moment (=quand), la méthode (=comment) et la personne ayant réalisé l'observation (=qui).
- **comment\_description** [TEXT] : description libre de l'observation, aussi succincte et précise que possible. Informations sur le(s) individu(s) observé(s) (=quoi).
- **additional\_data** [JSON] (=champs\_addi) : permet d'associer des champs complémentaires à la synthèse si toutes les entrées ne partagent pas les mêmes champs. Les valeurs de ce champ doivent être [au format JSON](#) et [être valide](#).
- **meta\_create\_date** [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure de création de l'enregistrement de l'observation.
- **meta\_update\_date** [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure de mise à jour de l'enregistrement de l'observation.
- **meta\_last\_action** [CHAR(1)] (=last\_action) : permet d'identifier les lignes ajoutées depuis le dernier import ("I"), modifiées ("U") ou supprimées ("D").

## Format SOURCE d'import

- But : Permet de décrire la source des données.
- Table GeoNature : "gn\_synthese.t\_sources".

## Description du format SOURCE

Pour chaque ligne : nom\_du\_champ [format du champ] (=nom\_champ\_table\_geonature) : description du champ.. Les champs **en gras** sont obligatoires.

- **name** [VARCHAR(255)] (=name\_source) : correspond au champ "name\_source". Sert de lien

- avec la ressource SYNTHESE et son champ "code\_source".
- desc [TEXT] (=desc\_source) : description de la source des données. Correspond au champ "desc\_source"
  - entity\_source\_pk\_field [VARCHAR(255)] : nom du champ dans les données sources servant de clé primaire et dont la valeur est présente dans le champ "entity\_source\_pk\_value" de la table SYNTHESE d'import.
  - url [VARCHAR(255)] (=url\_source) : adresse web décrivant la source des données ou permettant d'accéder aux données sources. Correspond au champ "url\_source".
  - additional\_data [JSON] (=champs\_addi) : permet d'associer des champs complémentaires à la source si toutes les entrées ne partagent pas les mêmes champs. Les valeurs de ce champ doivent être [au format JSON](#) et être valide.
  - meta\_create\_date [DATE YYYY-MM-DD HH:MM:SS] : date et heure de création de l'enregistrement de la source.
  - meta\_update\_date [DATE YYYY-MM-DD HH:MM:SS] : date et heure de mise à jour de l'enregistrement de la source.
  - meta\_last\_action [CHAR(1)] : permet d'identifier les lignes ajoutées depuis le dernier import ("I"), modifiées ("U") ou supprimées ("D").

## Format DATASET d'import

- But : permet de fournir les informations sur les jeux de données.
- Table GeoNature : "gn\_meta.t\_datasets".
- Standard : [Métadonnées v1.3.10](#)
- Correspondance GeoNature/Standard : [gn\\_meta.t\\_datasets](#) et [Métadonnées](#)

### NOTES :

- pour éviter trop de complexité, nous ne tenons pas compte des "protocoles" liées au jeu de données, ni des acteurs de type "personne". Si cela s'avérait nécessaire, il est possible de les ajouter manuellement une fois l'import effectué.
- Dans la table GeoNature correspondante à ce fichier, les nomenclatures ne peuvent pas être nulle. Il faut donc indiquer la valeur par défaut ou ne pas inclure la colonne dans le fichier csv.

## Description du format DATASET

Pour chaque ligne : nom\_du\_champ [format du champ] (=nom\_champ\_table\_geonature) : description du champ.. Les champs **en gras** sont obligatoires.

- unique\_id\_sinp [UUID] (=unique\_dataset\_id) : UUID du jeu de données si pré-existant.
- **code\_acquisition\_framework** [VARCHAR(255)] (=id\_acquisition\_framework) : code/nom du cadre d'acquisition auquel le jeu de données appartient. Sert de le lien avec la ressource ACQUISITION\_FRAMEWORK et son champ "name".
- **name** [VARCHAR(255)] (=dataset\_name) : nom du jeu de donnée.
- **shortname** [VARCHAR(255)] (=dataset\_shortname) : libellé court du jeu de données (30 caractères). Permet de faire le lien avec la ressource SYNTHESE et son champ "code\_dataset".
- **desc** [TEXT] (=dataset\_desc) : description du jeu de données.
- code\_nomenclature\_data\_type [VARCHAR(25)] (=id\_nomenclature\_data\_type) : code de la valeur du type de nomenclature DATA\_TYP ([TypeDonneesValue](#)).
- keywords [TEXT] : mot(s)-clé(s) représentatifs du jeu de données.

- **marine\_domain** [BOOL] : indique si le jeu de données concerne le domaine marin. Ne peut pas être NULL.
- **terrestrial\_domain** [BOOL] : indique si le jeu de données concerne le domaine terrestre. Ne peut pas être NULL.
- **code\_nomenclature\_dataset\_objectif** [VARCHAR(25)] (=id\_nomenclature\_dataset\_objectif) : code de la valeur du type de nomenclature *JDD\_OBJECTIFS* ([ObjectifJeuDonneesValue](#)).
- **bbox\_west** [FLOAT] : extrême ouest de l'emprise géographique du jeu de données (rectangle permettant d'englober le jeu de données et défini par 4 extrêmes : nord, sud, est, et ouest). La valeur doit correspondre à la longitude ouest dans le SRID 4326.
- **bbox\_est** [FLOAT] : extrême est de l'emprise géographique du jeu de données (rectangle permettant d'englober le jeu de données et défini par 4 extrêmes : nord, sud, est, et ouest). La valeur doit correspondre à la longitude est dans le SRID 4326.
- **bbox\_south** [FLOAT] : extrême sud de l'emprise géographique du jeu de données (rectangle permettant d'englober le jeu de données et défini par 4 extrêmes : nord, sud, est, et ouest). La valeur doit correspondre à la latitude sud dans le SRID 4326.
- **bbox\_north** [FLOAT] : extrême nord de l'emprise géographique du jeu de données (rectangle permettant d'englober le jeu de données et défini par 4 extrêmes : nord, sud, est, et ouest). La valeur doit correspondre à la latitude nord dans le SRID 4326.
- **code\_nomenclature\_collecting\_method** [VARCHAR(25)] (=id\_nomenclature\_collecting\_method) : code de la valeur du type de nomenclature *METHO\_RECUEIL* ([MethodeRecueilValue](#)).
- **code\_nomenclature\_data\_origin** [VARCHAR(25)] (=id\_nomenclature\_data\_origin) : code de la valeur du type de nomenclature *DS\_PUBLIQUE* ([DSPubliqueValue](#) [[Standard OccTax](#)]).
- **code\_nomenclature\_source\_status** [VARCHAR(25)] (=id\_nomenclature\_source\_status) : code de la valeur du type de nomenclature *STATUT\_SOURCE* ([StatutSourceValue](#)).
- **cor\_territory** [TEXT(ARRAY-ARRAY)] : champ de type tableau de tableau de textes. Les tableaux de textes du plus bas niveau contiendront comme première valeur le code de nomenclature correspondant au champ "id\_nomenclature\_territory" (= *TERRITOIRE* [TerritoireValue](#)) de la table "gn\_meta.cor\_dataset\_territory" et comme seconde valeur une description du territoire (champ "territory\_desc) ou une chaîne vide. Exemple :

```
"{"REU", ""}, {"MYT", ""}]"
```

- **cor\_actors\_organism** [TEXT(ARRAY-ARRAY)] : champ de type tableau de tableau de textes. Les tableaux de textes du plus bas niveau contiendront comme première valeur le nom de l'organisme et comme seconde valeur le code de nomenclature correspondant au champ "id\_nomenclature\_actor\_role" (= "ROLE\_ACTEUR" / [RoleActeurValue](#)) de la table "gn\_meta.cor\_dataset\_actor". Exemple :

```
"{"CBN-Alpin", "1"}, {"PNE", "5}]"
```

- **cor\_actors\_user** [TEXT(ARRAY-ARRAY)] : champ de type tableau de tableau de textes. Les tableaux de textes du plus bas niveau contiendront comme première valeur l'identifiant de l'utilisateur/personne et comme seconde valeur le code de nomenclature correspondant au champ "id\_nomenclature\_actor\_role" (= "ROLE\_ACTEUR" / [RoleActeurValue](#)) de la table "gn\_meta.cor\_dataset\_actor". Exemple :

```
"{"jpmilcent", "1"}, {"pmartin", "5}]"
```

- **additional\_data** [JSON] : permet d'associer des champs complémentaires au jeu de données si toutes les entrées ne partagent pas les mêmes champs. Les valeurs de ce champ doivent être [au format JSON](#) et être [valide](#).
- **meta\_create\_date** [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure de création de

l'enregistrement du jeu de données.

- `meta_update_date` [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure de mise à jour de l'enregistrement du jeu de données.
- **meta\_last\_action** [CHAR(1)] : permet d'identifier les lignes ajoutées depuis le dernier import ("I"), modifiées ("U") ou supprimées ("D").

## Format de la table ACQUISITION\_FRAMEWORK d'import

- But : permet de fournir les informations sur les cadres d'acquisition des jeux de données.
- Tables GeoNature : `gn_meta.t_acquisition_framework` et aux tables liées `cor_acquisition_framework_actor`, `cor_acquisition_framework_objectif`, `cor_acquisition_framework_publication`, `cor_acquisition_framework_voletsinp` et `inp_datatype_publications`.
- Standard : [Métadonnées v1.3.10](#)
- Correspondance GeoNature/Standard : [gn\\_meta.t\\_acquisition\\_framework](#) et [Métadonnées](#)

## Description du format ACQUISITION\_FRAMEWORK

Pour chaque ligne : `nom_du_champ` [format du champ] (=nom\_champ\_table\_geonature) : description du champ.. Les champs **en gras** sont obligatoires.

- `unique_id` [UUID] (=unique\_acquisition\_framework\_id) : UUID du cadre d'acquisition si pré-existant.
- **name** [VARCHAR(255)] (=acquisition\_framework\_name) : nom complet du cadre d'acquisition. Sert de le lien avec la ressource DATASET et son champ `code_acquisition_framework` mais aussi avec le champ `parent_code`.
- **desc** [TEXT] (=acquisition\_framework\_desc) : description du cadre d'acquisition.
- `code_nomenclature_territorial_level` [VARCHAR(25)] (=id\_nomenclature\_territorial\_level) : code alphanumérique de la valeur du type de nomenclature `NIVEAU_TERRITORIAL` ([NiveauTerritorialValue](#)).
- `territory_desc` [TEXT] : description du territoire concerné.
- `keywords` [TEXT] : mot(s)-clé(s) représentatifs du cadre d'acquisition, séparés par des virgules.
- `code_nomenclature_financing_type` [VARCHAR(25)] (=id\_nomenclature\_financing\_type) : code du type de financement pour le cadre d'acquisition, tel que défini dans la nomenclature `TYPE_FINANCEMENT` ([TypeFinancementValue](#)).
- `target_description` [TEXT] : description de la cible taxonomique ou géologique pour le cadre d'acquisition (Ex. : pteridophyta).
- `ecologic_or_geologic_target` [TEXT] : champ composé de `cd_nom` de TaxRef, séparés par des points virgules, s'il s'agit de taxons, ou de `cd_hab` de HabRef, séparés par des points virgules, s'il s'agit d'habitats.
- `parent_code` [VARCHAR(255)](=acquisition\_framework\_parent\_id) : si ce cadre d'acquisition est contenu dans un autre, le code/nom du cadre acquisition parent sera indiqué. Sert de le lien avec la ressource ACQUISITION\_FRAMEWORK parente et son champ "name".
- `is_parent` [BOOL] : indique si ce dispositif est un métacadre, et donc s'il contient d'autres cadres d'acquisition.
- **start\_date** [DATE(YYYY-MM-DD)] (=acquisition\_framework\_start\_date) : date de lancement du cadre d'acquisition. **Ne peut pas être NULL.**
- `end_date` [DATE(YYYY-MM-DD)] (=acquisition\_framework\_end\_date) : date de clôture du cadre d'acquisition. Si elle n'est pas remplie, on considère que le cadre est toujours en activité.

- `cor_objectifs` [TEXT(ARRAY)] : champ de type tableau de textes. Le tableau contiendra une succession de codes de nomenclature correspondant au champ "`id_nomenclature_objectif`" (= "CA\_OBJECTIFS" / [ObjectifsValue](#)) de la table "`gn_meta.cor_acquisition_framework_objectif`". Exemple :

```
'{'4','5','6'}'
```

- `cor_voletsinp` [TEXT(ARRAY)] : champ de type tableau de textes. Le tableau contiendra une succession de codes de nomenclature correspondant au champ "`id_nomenclature_voletsinp`" (= "VOLET\_SINP" / [VoletSINPValue](#)) de la table "`gn_meta.cor_acquisition_framework_voletsinp`". Exemple :

```
'{'1'}'
```

- `cor_actors_organism` [TEXT(ARRAY-ARRAY)] : champ de type tableau de tableau de textes. Les tableaux de textes du plus bas niveau contiendront comme première valeur le nom de l'organisme et comme seconde valeur le code de nomenclature correspondant au champ "`id_nomenclature_actor_role`" (= "ROLE\_ACTEUR" / [RoleActeurValue](#)) de la table "`gn_meta.cor_acquisition_framework_actor`". Exemple :

```
'{{'CBN-Alpin','1'},{'PNE','5'}}'
```

- `cor_actors_user` [TEXT(ARRAY-ARRAY)] : champ de type tableau de tableau de textes. Les tableaux de textes du plus bas niveau contiendront comme première valeur l'identifiant de l'utilisateur/personne et comme seconde valeur le code de nomenclature correspondant au champ "`id_nomenclature_actor_role`" (= "ROLE\_ACTEUR" / [RoleActeurValue](#)) de la table "`gn_meta.cor_acquisition_framework_actor`". Exemple :

```
'{{'jpmilcent','1'},{'pmartin','5'}}'
```

- `cor_publications` [JSON] : champ contenant un tableau d'objets [au format JSON valide](#). Ce champ correspond à la table `cor_acquisition_framework_publication` de GeoNature. Chaque objet du JSON représentera une publication avec les attributs suivant :
  - **reference** : attribut obligatoire contenant la référence complète de la publication au [format ISO 690](#).
  - `url` : permalien permettant de la publication si disponible sur le web.
  - `uuid` : UUID de cette publication.
- `additional_data` [JSON] : permet d'associer des champs complémentaires au cadre d'acquisition si toutes les entrées ne partagent pas les même champs. Les valeurs de ce champ doivent être [au format JSON](#) et être valide.
- `meta_create_date` [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure de création de l'enregistrement du jeu de données.
- `meta_update_date` [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure de mise à jour de l'enregistrement du jeu de données.
- `meta_last_action` [CHAR(1)] : permet d'identifier les lignes ajoutées depuis le dernier import ("`I`"), modifiées ("`U`") ou supprimées ("`D`")

## Format ORGANISM d'import

- **But** : permet de fournir les informations sur les organismes liées aux jeux de données et cadres d'acquisition.

- Table GeoNature : "*utilisateurs.bib\_organismes*".

## Description du format ORGANISM

Pour chaque ligne : `nom_du_champ [format du champ] (=nom_champ_table_geonature)` : description du champ.. Les champs **en gras** sont obligatoires.

- `unique_id [UUID] (=uuid_organisme)` : UUID de l'organisme si pré-existant dans les données sources. Correspond au champ "uuid\_organisme".
- **name** [VARCHAR(100)] (=nom\_organisme) : correspond au champ "nom\_organisme". Sert de lien avec les tables DATASET (*cor\_actors\_organism*) et ACQUISITION\_FRAMEWORK (*cor\_actors\_organism*).
- `address [VARCHAR(128)] (=adresse_organisme)` : correspond au champ "adresse\_organisme".
- `postal_code [VARCHAR(5)] (=cp_organisme)` : correspond au champ "cp\_organisme".
- `city [VARCHAR(100)] (=ville_organisme)` : correspond au champ "ville\_organisme".
- `phone [VARCHAR(14)] (=tel_organisme)` : correspond au champ "tel\_organisme".
- `fax [VARCHAR(14)] (=fax_organisme)` : correspond au champ "fax\_organisme".
- `email [VARCHAR(100)] (=email_organisme)` : correspond au champ "email\_organisme".
- `url [VARCHAR(255)] (=url_organisme)` : correspond au champ "url\_organisme".
- `logo_url [VARCHAR(255)] (=url_logo)` : correspond au champ "url\_logo".
- `additional_data [JSON]` : permet d'associer des champs complémentaires à l'organisme si toutes les entrées ne partagent pas les même champs. Les valeurs de ce champ doivent être [au format JSON](#) et être [valide](#).
- **meta\_create\_date** [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure de création de l'enregistrement de l'organisme.
- `meta_update_date [DATE(YYYY-MM-DD HH:MM:SS)]` : date et heure de mise à jour de l'enregistrement de l'organisme.
- **meta\_last\_action** [CHAR(1)] : permet d'identifier les lignes ajoutées depuis le dernier import ("I"), modifiées ("U") ou supprimées ("D").

## Format USER d'import

- But : permet de fournir les informations sur les utilisateurs, c'est à dire les personnes désireuses d'accéder aux informations demandant un compte. Normalement, ce fichier ne devrait être utilisé que pour l'import initial mais des champs "meta\_\*" sont fournis s'il s'avérait nécessaire de l'utiliser aussi en mise à jour.
- Table GeoNature : "*utilisateurs.t\_roles*"

**NOTES** : nous ne gérons pas dans cette version la notion de groupe de GeoNature.

## Description du format USER

Pour chaque ligne : `nom_du_champ [format du champ] (=nom_champ_table_geonature)` : description du champ.. Les champs **en gras** sont obligatoires.

- `unique_id [UUID] (=uuid_role)` : UUID associé à l'utilisateur.
- **identifiant** [VARCHAR(100)] (=identifiant) : identifiant/login de l'utilisateur. Peut éventuellement correspondre à la première partie de l'email (avant le @). Sert de lien avec la ressource

SYNTHESE (*code\_digitiser*).

- **firstname** [VARCHAR(50)] (=prenom\_role) : prénom.
- **name** [VARCHAR(50)] (=nom\_role) : nom de famille.
- **email** [VARCHAR(250)] : courriel.
- **code\_organism** [VARCHAR(100)] (=id\_organisme) : code alphanumérique permettant d'identifier l'organisme de l'utilisateur (voir ORGANISM). Sert de le lien avec la ressource ORGANISM et son champ "name".
- **comment** [TEXT] (=remarques) : contient des éventuelles informations internes (date d'importation, par exemple).
- **enable** [BOOL] (=active) : indique si l'utilisateur est actif (=true) et donc peut se connecter ou pas (=false).
- **additional\_data** [JSON] (=champs\_addi) : permet d'associer des champs complémentaires à l'utilisateur si toutes les entrées ne partagent pas les même champs. Les valeurs de ce champ doivent être [au format JSON](#) et être valide.
- **meta\_create\_date** [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure de création de l'enregistrement de l'utilisateur.
- **meta\_update\_date** [DATE(YYYY-MM-DD HH:MM:SS)] : date et heure de mise à jour de l'enregistrement de l'utilisateur.
- **meta\_last\_action** [CHAR(1)] : permet d'identifier les lignes ajoutées depuis le dernier import ("I"), modifiées ("U") ou supprimées ("D").

## Format META\_ADDITIONAL\_DATA d'import

- But : permet de fournir la description des champs additionnels présent dans les fichiers de données à importer.
- Standard : [OccTax v2](#) extension *AttributAdditionnel*.

## Description du format META\_ADDITIONAL\_DATA

Pour chaque ligne : nom\_du\_champ [format du champ] (=nom\_attribut\_standard) : description du champ.. Les champs **en gras** sont obligatoires.

- **resource** [VARCHAR(50)] : nom de la ressource (=fichier d'import) pour laquelle le champ additionnel est utilisé. Valeurs : *synthese, source, dataset, acquisition\_framework, organism, user*.
- **name** [VARCHAR(50)] (=nomAttribut) : nom du champ additionnel.
- **desc** [TEXT] (=definitionAttribut) : définition/description du contenu du champ additionnel.
- **keyword** [VARCHAR(50)] (=thematiqueAttribut) : un mot-clé indiquant la catégorie/thématique du champ.
- **type** [VARCHAR(25)] (=typeAttribut) : indique si l'attribut additionnel est de type quantitatif (=QTA), qualitatif (=QUAL) ou inconnu (=NSP).
- **unity** [VARCHAR(50)] (=uniteAttribut) : unité du système international du champ additionnel (Ex. : °C, m, kg). **Obligatoire** si *type = QUAL*.

From:

<http://wiki-sinp.cbn-alpin.fr/> - **CBNA SINP**

Permanent link:

<http://wiki-sinp.cbn-alpin.fr/database/import-formats?rev=1614176912>

Last update: **2021/02/24 14:28**

