Déclaration des flotteurs pour le décodeur Matlab

Cette note décrit la procédure de déclaration d’un flotteur NKE pour le décodeur Matlab.

La déclaration d’un flotteur doit être suivie d’un décodage des données reçues et de la vérification des données décodées.

[Déclaration des flotteurs pour le décodeur Matlab 1](#_Toc422910718)

[1. Déclaration des flotteurs Argos et Iridium 2](#_Toc422910719)

[1.1. Création de la liste des flotteurs à déclarer 2](#_Toc422910720)

[1.2. Création d’un export de la base de données pour les flotteurs à déclarer 2](#_Toc422910721)

[1.3. Mise à jour du fichier \_provor\_floats\_information\_co.xls 2](#_Toc422910722)

[1.4. Enrichissement de l’export de la base de données 2](#_Toc422910723)

[1.4.1. Enrichissement 2](#_Toc422910724)

[1.4.2. Contrôle des informations obligatoires 3](#_Toc422910725)

[1.4.3. Reporter les enrichissements dans la base de données 3](#_Toc422910726)

[1.5. Génération des fichiers JSON de méta-données 3](#_Toc422910727)

[2. Déclaration des flotteurs Remocean 4](#_Toc422910728)

[2.1. Création de la liste des flotteurs à déclarer 4](#_Toc422910729)

[2.2. Collecte des renseignements sur les flotteurs à déclarer 4](#_Toc422910730)

[2.3. Création d’un export de la base de données pour les flotteurs à déclarer 5](#_Toc422910731)

[2.4. Mise à jour du fichier \_provor\_floats\_information\_co.xls 5](#_Toc422910732)

[2.5. Enrichissement de l’export de la base de données 5](#_Toc422910733)

[2.5.1. Enrichissement 5](#_Toc422910734)

[2.5.2. Contrôle des informations obligatoires 5](#_Toc422910735)

[2.5.3. Reporter les enrichissements dans la base de données 6](#_Toc422910736)

[2.6. Première génération des fichiers JSON de méta-données 6](#_Toc422910737)

[2.7. Récupération des coefficients de calibration transmis par le flotteur 6](#_Toc422910738)

[2.8. Seconde génération des fichiers JSON de méta-données 6](#_Toc422910739)

[3. Préparation, décodage et contrôle des données 7](#_Toc422910740)

[3.1. Préparation des données 7](#_Toc422910741)

[3.1.1. Pour les flotteurs Argos, préparation des données Argos 7](#_Toc422910742)

[3.1.2. Pour les flotteurs Iridium, duplication des données Iridium 8](#_Toc422910743)

[3.1.1. Pour les flotteurs Rémocéan, duplication des données Iridium 8](#_Toc422910744)

[3.2. Décodage des données 8](#_Toc422910745)

[3.3. Contrôle ‘visuel’ des données décodées 8](#_Toc422910746)

[3.4. Contrôle final des données décodées 9](#_Toc422910747)

[4. Finalisation de la déclaration des flotteurs 10](#_Toc422910748)

[4.1. Génération des fichiers JSON d’information 10](#_Toc422910749)

[4.2. Copie des fichiers vers l’espace d’exploitation 10](#_Toc422910750)

[4.3. Finalisation de la déclaration 10](#_Toc422910751)

[4.3.1. Flotteurs Argos 10](#_Toc422910752)

[4.3.2. Flotteurs Iridium 10](#_Toc422910753)

[4.3.3. Flotteurs Remocean 10](#_Toc422910754)

[5. Contrôles de routine 11](#_Toc422910755)

[5.1. Contrôle des fichiers Argos cycle 11](#_Toc422910756)

[5.2. Comparaison des informations de la base avec celles des fichiers JSON de méta-données 11](#_Toc422910757)

# Déclaration des flotteurs Argos et Iridium

## Création de la liste des flotteurs à déclarer

Les flotteurs managés par le décodeur Matlab sont listés dans le fichier ***\_provor\_floats\_information\_co.txt*** (.xls).

Les nouveaux flotteurs déployés sont initialement déclarés dans la base de données (qui est et demeure la référence) par la saisie des informations de leur fiche de mise à l’eau. Périodiquement, il est donc nécessaire de déclarer ces nouveaux flotteurs dans le décodeur Matlab.

Créer, via une requête SQL la liste des flotteurs déclarés en base mais non présents dans le fichier ***\_provor\_floats\_information\_co.txt***, i.e. flotteurs NKE dont la version est listée en colonne « N » mais dont le numéro WMO n'est pas en colonne « A ».

Classer le résultat de cette requête par version de flotteur puis par numéro WMO.

On crée ainsi une liste ***new\_argos\_yyyymmdd.txt*** ou ***new\_iridium\_yyyymmdd.txt***.

## Création d’un export de la base de données pour les flotteurs à déclarer

Générer un export CSV de la base pour les flotteurs de la liste ***new\_argos/iridium\_yyyymmdd.txt***.

Dans cet export, remplacer chaque retour à la ligne (ALT + 010) par un espace (et faire la mise à jour en base pour ne plus être gêné par cela à l'avenir).

Copier le contenu du fichier .csv obtenu dans un fichier .txt (copie des colonnes « A » à « E »).

On crée ainsi un fichier ***new\_argos/iridium\_meta\_yyyymmdd.txt***.

## Mise à jour du fichier \_provor\_floats\_information\_co.xls

Utiliser l'outil **get\_meta\_data\_from\_data\_base** pour générer, à partir de l'export de la base, un fichier CSV contenant les informations nécessaires au fichier ***\_provor\_floats\_information\_co.xls***.

Mettre à jour le fichier ***\_provor\_floats\_information\_co.xls*** puis le fichier ***\_provor\_floats\_information\_co.txt***.

## Enrichissement de l’export de la base de données

Le contenu de la base doit être enrichi avec des informations que l'on peut déduire de la version Coriolis du flotteur et avec des valeurs par défaut.

### Enrichissement

Ces ajouts sont réalisés par l'outil **generate\_csv\_meta** qui produit un fichier CSV enrichi pour certains TECH\_PARAMETER\_ID. Les ajouts effectués sont listés dans le log de l’outil par un message INFO (il ne doit pas y avoir de message ERROR lors de l’utilisation de cet outil).

L’outil effectue également le changement du DIM\_LEVEL associé aux capteurs et aux paramètres. Ils sont alors sur 3 digits, le premier servant à identifier le capteur d'appartenance (0: CTD, 1: OPTODE, 2: OCR, 3: ECO3, 4: FLBB, 5: CROVER, 6: SUNA). Cela rend plus aisé l'accès aux différents capteurs et paramètres associés (notamment pour leur mise à jour).

Les données du fichier CSV produit doivent remplacer les informations de l'export de la base (***new\_argos/iridium\_meta\_yyyymmdd.txt***) pour les TECH\_PARAMETER\_ID concernés.

On peut relancer **generate\_csv\_meta** avec l'export de la base mis à jour jusqu'à ne plus avoir d'ajout effectué par l'outil (plus de message INFO dans le log).

### Contrôle des informations obligatoires

L'export de base obtenu doit ensuite être vérifié pour contrôler que les informations obligatoires (pour le checker) y sont. L'outil **generate\_csv\_meta\_mandatory** effectue cette vérification.

La colonne  « D » du fichier CSV produit:

* Peut comporter la mention « MANDATORY (can be replaced by 'n/a') » : cela signifie que l'information est obligatoire mais que l'on peut mettre 'n/a' si elle est absente. Cela sera fait par l'outil de génération des fichiers JSON de méta-données,
* Ne doit pas comporter la mention « MANDATORY » : dans le cas contraire, ces champs doivent **impérativement** être renseignés si l'on veut que le fichier meta.nc passe le checker.

### Reporter les enrichissements dans la base de données

Une fois l'export de base enrichi, mettre à jour la base avec son contenu (pour archivage). Avant cette opération, penser à exclure les champs vides.

## Génération des fichiers JSON de méta-données

Générer les fichiers JSON de méta-données avec les outils **generate\_json\_float\_meta\_argos** ou **generate\_json\_float\_meta\_ir\_sbd** et déplacer les fichiers produits dans les répertoires ad hoc du décodeur.

# Déclaration des flotteurs Remocean

## Création de la liste des flotteurs à déclarer

Les flotteurs managés par le décodeur Matlab sont listés dans le fichier ***\_provor\_floats\_information\_co.txt*** (.xls).

Les nouveaux flotteurs déployés sont initialement déclarés dans la base de données (qui est et demeure la référence) par la saisie des informations de leur fiche de mise à l’eau. Périodiquement, il est donc nécessaire de déclarer ces nouveaux flotteurs dans le décodeur Matlab.

Créer, via une requête SQL la liste des flotteurs déclarés en base mais non présents dans le fichier ***\_provor\_floats\_information\_co.txt***, i.e. flotteurs NKE dont la version est listée en colonne « N » mais dont le numéro WMO n'est pas en colonne « A ».

Classer le résultat de cette requête par version de flotteur puis par numéro WMO.

On crée ainsi une liste ***new\_rem\_yyyymmdd.txt***

## Collecte des renseignements sur les flotteurs à déclarer

Les informations relatives aux configurations des flotteurs sont disponibles sur le site <http://www.oao.obs-vlfr.fr/BD_FLOAT/moteurRechercheFloat2.php> (poteau/poteau).

Pour chaque flotteur à déclarer, il est nécessaire de récupérer :

1. La feuille de présentation du flotteur (pour connaitre les capteurs portés par le flotteur et le type d’utilisation de l’Eco3 s’il est présent). Imprimer la page générale de présentation du flotteur dans un fichier *WMO*\_1.pdf.
2. Feuille de mise à l’eau du flotteur (.xlsm),
3. Configuration du flotteur à la mise à l’eau (sauvegardée dans un fichier *WMO*\_2.txt),
4. Lorsqu’une Optode est installée sur le flotteur, le certificat de calibration de l’Optode.

Renseigner le fichier ***Liste\_Remocean\_20150216.xlsx*** qui résume les configurations des différents Rémocéans gérés par le décodeur Matlab.

L’information « Firm. Version » est extraite des configurations au lâcher lors de la génération des fichiers JSON de méta-données.

L’attribution de la version Coriolis dépend des capteurs installés sur le flotteur :

* 5.9: Eco3 (BETA\_BACKSCATTERING700 & FLUORESCENCE\_CDOM) sans Optode,
* 5.91: Eco3 (BETA\_BACKSCATTERING700 & FLUORESCENCE\_CDOM) et Optode calibrée avec “Aanderaa standard calibration”,
* 5.92: Eco3 (BETA\_BACKSCATTERING700 & FLUORESCENCE\_CDOM) et Optode calibrée avec “Stern-Volmer”,
* 6.0: Eco3 (BETA\_BACKSCATTERING700 & BETA\_BACKSCATTERING532) sans Optode (aucun flotteur à ce jour),
* 6.01: Eco3 (BETA\_BACKSCATTERING700 & BETA\_BACKSCATTERING532) et Optode calibrée avec “Stern-Volmer”.

## Création d’un export de la base de données pour les flotteurs à déclarer

Générer un export CSV de la base pour les flotteurs de la liste ***new\_rem\_yyyymmdd.txt***.

Dans cet export, remplacer chaque retour à la ligne (ALT + 010) par un espace (et faire la mise à jour en base pour ne plus être gêné par cela à l'avenir).

Copier le contenu du fichier .csv obtenu dans un fichier .txt (copie des colonnes « A » à « E »).

On crée ainsi un fichier ***new\_rem\_meta\_yyyymmdd.txt***.

## Mise à jour du fichier \_provor\_floats\_information\_co.xls

Utiliser l'outil **get\_meta\_data\_from\_data\_base** pour générer, à partir de l'export de la base, un fichier CSV contenant les informations présentes dans ***\_provor\_floats\_information\_co.xls***.

Mettre à jour le fichier ***\_provor\_floats\_information\_co.xls*** puis le fichier ***\_provor\_floats\_information\_co.txt***.

## Enrichissement de l’export de la base de données

Le contenu de la base doit être enrichi avec des informations que l'on peut déduire de la version du flotteur et avec des valeurs par défaut.

### Enrichissement

Ces ajouts sont réalisés par l'outil **generate\_csv\_meta\_remocean** qui produit un fichier CSV enrichi pour certains TECH\_PARAMETER\_ID. Les ajouts effectués sont listés dans le log de l’outil par un message INFO (il ne doit pas y avoir de message ERROR lors de l’utilisation de cet outil).

L’outil effectue également le changement du DIM\_LEVEL associé aux capteurs et aux paramètres: sur 3 digits, le premier servant à identifier le capteur d'appartenance (0: CTD, 1: OPTODE, 2: OCR, 3: ECO3, 4: FLBB, 5: CROVER, 6: SUNA). Cela rend plus aisé l'accès aux différents capteurs et paramètres associés (notamment pour leur mise à jour).

Les données du fichier CSV produit doivent remplacer les informations de l'export de la base (***new\_rem\_meta\_yyyymmdd.txt***) pour les TECH\_PARAMETER\_ID concernés.

On peut relancer **generate\_csv\_meta\_remocean** avec l'export de la base mis à jour jusqu'à ne plus avoir d'ajout effectué par l'outil (plus de message INFO dans le log).

### Contrôle des informations obligatoires

L'export de base obtenu doit ensuite être vérifié pour contrôler que les informations obligatoires (pour le checker) y sont. L'outil **generate\_csv\_meta\_mandatory** effectue cette vérification.

La colonne  « D » du fichier CSV produit:

* Peut comporter la mention « MANDATORY (can be replaced by 'n/a') » : cela signifie que l'information est obligatoire mais que l'on peut mettre 'n/a' si elle est absente. Cela sera fait par l'outil de génération des fichiers JSON de méta-données,
* Ne doit pas comporter la mention « MANDATORY » : dans le cas contraire, ces champs doivent **impérativement** être renseignés si l'on veut que le fichier meta.nc passe le checker.

### Reporter les enrichissements dans la base de données

Une fois l'export de base enrichi, mettre à jour la base avec son contenu (pour archivage). Avant cette opération, penser à exclure les champs vides.

## Première génération des fichiers JSON de méta-données

Générer les fichiers JSON **temporaires** de méta-données avec les outils **generate\_json\_float\_meta\_remocean** ou **generate\_json\_float\_meta\_remocean\_flbb** et déplacer les fichiers produits dans les répertoires ad hoc du décodeur.

Il est au préalable nécessaire de renseigner la liste des capteurs montés sur chaque flotteur (cf. lignes ~250-300 de ces outils).

Conserver la version de firmware listée dans le log de l’outil pour la reporter dans le fichier ***\_provor\_floats\_information\_co.xls*** (et ***Liste\_Remocean\_20150216.xlsx***).

## Récupération des coefficients de calibration transmis par le flotteur

Une duplication des données Iridium (cf 3.1.1) suivi d’un décodage avec decode\_provor\_2\_csv permet de récupérer, dans le fichier CSV produit :

* Les coefficients de calibration de l’Eco3 et de l’OCR (flotteurs de Villefranche) et du FLBB (flotteurs Indiens). Renseigner le fichier ***calib\_coef.xlsx*** (et dupliquer son contenu dans le fichier ***calib\_coef.txt***),
* L’information ‘show mode’ pour renseigner le fichier ***show\_mode.xlsx*** (et dupliquer son contenu dans le fichier ***show\_mode.txt***),
* Les informations 'pixel begin/end' du SUNA pour renseigner le fichier ***output\_pixel.xlsx*** (et dupliquer son contenu dans le fichier ***output\_pixel.txt***).

## Seconde génération des fichiers JSON de méta-données

Générer les fichiers JSON **définitifs** de méta-données avec les outils **generate\_json\_float\_meta\_remocean** ou **generate\_json\_float\_meta\_remocean\_flbb** et déplacer les fichiers produits dans les répertoires ad hoc du décodeur.

# Préparation, décodage et contrôle des données

Le décodage des données doit être précédé d’une étape de préparation des données dont le contenu dépend du type de transmission du flotteur.

## Préparation des données

### Pour les flotteurs Argos, préparation des données Argos

La déclaration des flotteurs peut être faite avant leur mise à l'eau, cependant, la faire après la transmission de quelques cycles permet de les déclarer et en même temps de vérifier qu'ils se décodent correctement (qu'ils sont bien « calés » temporellement et que leur méta-données sont correctement renseignées).

Lorsqu’un flotteur est déclaré après avoir déjà émis, il est nécessaire de préparer les données Argos reçues. Cette étape nécessite d'avoir déclaré le flotteur pour le décodeur (dans le fichier ***\_provor\_floats\_information\_co.txt***) et d'avoir créé son fichier JSON de méta-données.

La préparation des données Argos consiste à créer des fichiers cycles identifiés (numéro WMO et numéro de cycle renseignés dans le nom du fichier) avec les données reçues.

#### Contrôle du contenu du spool Argos

Avant d'effectuer la préparation des données Argos, il est nécessaire de vérifier qu'aucun fichier Argos de ce flotteur, présent dans le spool ne risque d'être archivé pendant la préparation des données (sinon les données de ce fichier seront perdues puisque nous travaillons sur une copie des fichiers déjà archivés).

L'outil **check\_argos\_spool\_directory** permet de contrôler le contenu du spool.

L’archivage est la première étape du décodage qui est effectué toutes les heures (à HH:15:00). Lorsque l’on a vérifié (*comment ?*) qu’aucun archivage n’est en cours, il est nécessaire de ‘bloquer’ les identifiants Argos des données en cours de préparation en renseignant le fichier *XXX*. Ainsi, les données associées à ces identifiants de ne seront plus pris en compte lors de l’archivage ou du décodage.

#### Préparation des données Argos

La préparation des données Argos s'effectue alors en trois étapes.

##### Copie des données

L'outil **copy\_argos\_files\_in\_archive\_cycle** effectue une copie des données Argos archivées vers un répertoire de travail.

##### Préparation des données

L'outil **process\_argos\_data** effectue le traitement des données Argos du répertoire de travail.

Cet outil enchaine les 4 outils:

* STEP1: **split\_argos\_cycles**: Eclatement des données par balise Argos et par passage satellite,
* STEP2: **delete\_double\_argos\_split**: Suppression des passages satellite identiques,
* STEP3: **create\_argos\_cycle\_files**: Création des fichiers cycles (i.e. concaténation temporelle des passages satellite avec un changement de fichier dès que le flotteur n'a pas émis pendant au moins 18 heures),
* STEP4 (FINAL): **move\_and\_rename\_argos\_files**: Identification des fichiers cycles (i.e. détermination du numéro de cycle, algorithme prenant en compte le contenu du fichier JSON de méta-données).

##### Contrôle du résultat

L'outil **check\_argos\_cycle\_files** permet de contrôler le découpage réalisé.

#### Mise en exploitation des fichiers Argos préparés

Les fichiers produits doivent ensuite remplacer les fichiers existants dans l'espace d'archivage des données Argos (on peut utiliser l'outil **select\_argos\_files** pour regrouper les répertoires Argos d'un ensemble de flotteurs).

### Pour les flotteurs Iridium, duplication des données Iridium

Les données des flotteurs Iridium ne nécessitent pas de préparation.

Pour en effectuer le décodage (décodage de contrôle par l'opérateur) il est toutefois nécessaire d'effectuer une copie des données reçues, de l'espace d'archivage (DIR\_INPUT\_RSYNC\_DATA) vers l'espace d'exploitation (IRIDIUM\_DATA\_DIRECTORY).

Cette opération s'effectue avec l’outil **copy\_iridium\_mail\_files**.

Notons que la configuration du flotteur au lâcher est transmise lors du prélude. Si le SBD en question a été collecté après la date de mise à l'eau du flotteur, il ne sera pas décodé. Il est alors possible de le récupérer en changeant temporairement cette date, en décodant ces données et en insérant en base les informations décodées.

### Pour les flotteurs Rémocéan, duplication des données Iridium

Les données des flotteurs Rémocéans ne nécessitent pas de préparation.

Pour en effectuer le décodage (décodage de contrôle par l'opérateur) il est toutefois nécessaire d'effectuer une copie des données reçues, de l'espace d'archivage (DIR\_INPUT\_RSYNC\_DATA) vers l'espace d'exploitation (IRIDIUM\_DATA\_DIRECTORY).

Cette opération s'effectue avec l’outil **copy\_remocean\_sbd\_files** (Remocean de Villefranche) ou **copy\_iridium\_mail\_files** (FLBB indiens).

## Décodage des données

Les données peuvent ensuite être décodées (avec **decode\_provor\_2\_csv** et **decode\_provor\_2\_nc**).

Lors de la génération des fichiers NetCDF, il faut penser à positionner le flag APPLY\_RTQC à 1 pour l’étape finale de contrôle présentée au §3.4.

## Contrôle ‘visuel’ des données décodées

Un parcours rapide du fichier CSV issu du décodage permet de vérifier que le décodage semble correct (regarder par exemple l'offset de l'horloge et quelques données mesurées).

Un contrôle visuel des données permet de vérifier globalement qu'il n'y a pas de problème, de re-caler le flotteur si nécessaire et de vérifier ses méta-données. En cas d'anomalie, on reviendra au détail des données du fichier CSV.

L'outil **nc\_trace\_times** permet de vérifier, la durée du cycle, la périodicité des mesures en dérive, les immersions de parking et de profil et plus généralement la qualité de la transmission et la cohérence des informations temporelles.

L'outil **nc\_trace\_disp** permet de vérifier la position de lâcher et les déplacements du flotteur.

L'outil **nc\_trace\_param** permet de vérifier les mesures des profils.

La correction de certaines méta-données peut nécessiter une mise à jour du fichier ***\_provor\_floats\_information\_co.txt***) ou des fichiers JSON de méta-données.

## Contrôle final des données décodées

Le dernier contrôle à appliquer aux fichiers NetCDF produits concerne le passage des contrôles du checker du GDAC Argo.

Ceci est effectué avec l’outil **nc\_check\_file\_format** en s’assurant au préalable que la version utilisée du checker est à jour.

# Finalisation de la déclaration des flotteurs

## Génération des fichiers JSON d’information

Les versions **decode\_provor\_2\_nc\_rt** et **decode\_provor\_2\_nc\_dm** du décodeur sont utilisées en exploitation pour assurer le décodage ‘temps réel’ des données arrivant à Coriolis.

Ces décodeurs n’utilisent pas le fichier ***\_provor\_floats\_information\_co.txt*** mais des fichiers JSON d’information spécifiques à chaque flotteur. L’outil **generate\_json\_float\_info** permet de générer, à partir d’un fichier ***\_provor\_floats\_info.txt* (**obtenu en copiant les colonnes « A » à « N » du fichier ***\_provor\_floats\_information\_co.xls***), les fichiers JSON d’information (un fichier JSON est généré pour chaque ligne du fichier d’entrée).

## Copie des fichiers vers l’espace d’exploitation

Rappelons :

* Que les nouveaux fichiers JSON de méta-données ont été copiés dans les répertoires ad hoc du décodeur (cf. 1.5 et 2.8),
* Que les données Argos préparées ont été copiées dans l'espace d'archivage des données Argos.

Il reste alors à copier les nouveaux fichiers JSON d’information dans le répertoire ad hoc du décodeur.

## Finalisation de la déclaration

La déclaration des nouveaux flotteurs s’achève alors par les actions suivantes.

### Flotteurs Argos

Supprimer les identifiants Argos concernés dans le fichier *XXX* (cf. 3.1.1.1).

### Flotteurs Iridium

*XXX*

### Flotteurs Remocean

*XXX*

# Contrôles de routine

## Contrôle des fichiers Argos cycle

Le numéro de cycle de chaque fichier Argos est déterminé de manière automatique lors de l'archivage du fichier en question.

Ces numéros doivent être vérifiés en particulier lorsque le flotteur dysfonctionne (mauvaise transmission, saut d'horloge, remontée d'urgence, etc…) ou émet en mode EOL.

Ce contrôle permet d'éviter que des données soient diffusées avec des numéros de cycles incorrects (en particulier pour les opérateurs chargés du temps différé).

L'outil **check\_argos\_hidden\_cycle\_files** permet de vérifier qu'aucun fichier Argos cycle n'est masqué au décodeur (pas de numéro WMO de flotteur dans le nom du fichier). Cette anomalie ne peut intervenir que suite à une erreur de l’opérateur.

L'outil **check\_argos\_cycle\_files** permet de contrôler les fichiers Argos cycle.

## Comparaison des informations de la base avec celles des fichiers JSON de méta-données

Toute modification en base d'une information utilisée dans le fichier meta.nc ou d’un coefficient de calibration doit être également reportée dans le fichier JSON de méta-données.

Une génération de tous les fichiers JSON de méta-données à partir d'un export de la base puis leur comparaison avec les fichiers en exploitation permet de s'assurer que les fichiers meta.nc seront conformes au contenu de la base.