

# Description de la tâche FCQODB et du fichier BDM\_CQ

## Table des matières

I. Tâche FcqOdb.....	2
1-Namelist.....	2
a)Modele.....	2
b)LSolVerif.....	2
c)LTempVerif.....	2
d)LPilotVerif.....	2
e)LbiasCorr.....	3
II.Format du fichier rejet.....	3
1-L'entête général.....	3
2-Entête d'une section.....	3
3-Ligne de données pour les nouvelles sections.....	3
a)SOLVERIF.....	4
b)TEMPVERIF.....	4
c)PILOTVERIF.....	4
4-Constitution des flags.....	5
a)constitution de flag1 – code qualité.....	5
b)Constitution de flag2 – code utilisation.....	5
5-Codification du type de niveau TEMPVERIF (niv).....	5
6-Correspondance varno et paramètres.....	6
7-Correspondance codmess et type de message.....	6

Historique :

V 1.0 : mise en place sur le CY41t1\_op1

# Description de la tâche FCQODB et du fichier BDM\_CQ

## I. Tâche FcqOdb

La fonction de cette tâche est de générer un fichier ASCII formaté qui est renvoyé vers la BDMO pour utilisation ultérieure (génération de scores). Elle utilise en entrée la base ECMA issue du matchup.

### 1- Namelist

Une namelist est nécessaire pour spécifier le type de sections à générer dans le fichier *BDM\_CQ*. Ci-dessous un exemple de namelist.

```
&NADIRS  
  Modele="ARPEGE",  
  LSolVerif=.TRUE.,  
  LTempVerif=.TRUE.,  
  LPilotVerif=.TRUE.,  
  LBiasCorr = .TRUE.  
/
```

#### a) *Modele*

De type chaîne de caractères (longueur 9), elle indique le modèle à partir duquel on génère le fichier. Les valeurs possible (au 07/2015) sont :

ARPEGE  
AEARP  
AROME  
AROMEPI  
VARPACK  
CALEDONIE  
REUNION  
POLYNESIE  
ANTI-GUYA

#### b) *LSolVerif*

De type logique, cette variable autorise la génération d'une section pour la table SOLVERIF si initialisée à *.TRUE.*. Si elle prend la valeur *.FALSE.*, les sections pour les tables SOLOMM, RADOMEH, SHIP, SYNOP, SYNOR seront générées (compatibilité ascendante des sections avec les précédentes versions).

#### c) *LTempVerif*

De type logique, cette variable autorise la génération d'une section pour la table TEMPVERIF si initialisée à *.TRUE.*. Dans le cas contraire, les anciennes sections TEMP, TEMPSHIP, TEMPDROP et TEMPMOBIL seront générées.

#### d) *LPilotVerif*

Comme pour *LTempVerif*, cette variable autorise la génération d'une section pour la table

## Description de la tâche FCQODB et du fichier BDM\_CQ

PILOTVERIF si elle initialisée à .TRUE.. Dans le cas contraire, les anciennes sections PILOT, PILOTSHIP et PILOTMOBIL seront générées.

### e) *LbiasCorr*

Cette variable n'a d'influence que lorsque une ou plusieurs des 3 variables précédentes prennent la valeur .TRUE.. Dans le cas où la *LBiasCorr* prend la valeur .TRUE., les valeurs des paramètres météorologiques présents dans les sections concernées sont celles qui sont débiaisées. Si elle prend la valeur .FALSE., les valeurs des paramètres météorologiques sans aucune correction sont données<sup>1</sup> (fonctionnement historique).

## II. Format du fichier *BDM\_CQ*

Le fichier *BDM\_CQ* comporte une ligne d'entête général, et un nombre variable de sections. Chaque section, quand à elle, possède un entête suivi d'un certain nombre de lignes de données.

### 1- L'entête général

```
'MODELE' 'CUTOFF' 'DATE_RESEAU' 'DATE_RUN'
```

format fortran :

```
"('',a9,'',i8,i6.6,'',a8,a6,'')"
```

*MODELE* est une chaîne de longueur 9. Sa valeur par défaut est chaîne vide. Prend la valeur de la variable *Modele* initialisée par *namelist*.

*CUTOFF* est une chaîne de longueur 6. Sa valeur indique de quel type de réseau il s'agit. Elle est fournie par *DSI/OP/IGA* pour les chaînes opérationnelles et fixée à la valeur *TEST* dans les autres cas.

*DATE\_RESEAU* est une chaîne de longueur 14. Sa valeur, qui correspond à la date pivot, est de la forme *YYYYDDMMhhmmss*.

*DATE\_RUN* est une chaîne de longueur 14. Sa valeur, qui correspond à la date d'exécution de l'analyse, est de la forme *YYYYDDMMhhmmss*.

### 2- Entête d'une section

```
'FLAG' 'TABLE' 'nlines ndata'
```

format fortran :

```
"('FLAG',a20,'',i7,1x,i2)"
```

*FLAG* est une constante suivie d'un blanc (longueur 5 caractères).

*TABLE* est une chaîne de longueur 20. Sa valeur donne le nom de la table dans la quelle doivent être stockées les informations.

*nlines* donne le nombre de lignes de données qui doivent être lues.

*ndata* indique le nombre de blocs à lire sur une ligne de données.

<sup>1</sup> la valeur de l'humidité spécifique est recalculée dans l'analyse des modèles en tenant compte de la correction de biais de la température. Si cette valeur est inférieure ou égale à 1.E10-09, elle est forcée à *RMDI* (valeur manquante). **Donc, quelque soit la valeur prise par *LbiasCorr*, la valeur récupérée pour l'humidité spécifique est toujours celle calculée par l'analyse (donc débiaisée).**

# Description de la tâche FCQODB et du fichier BDM\_CQ

## 3- Ligne de données pour les nouvelles sections

une ligne de données se décompose en 2 parties : une partie 'métadonnées' (au sens large) et une partie 'data'. A noter que la partie 'data' est composée de plusieurs quadruplés dont **l'ordre n'est pas garanti dans le futur**, d'où la présence du *varno* (en début de chaque quadruplé), qui permet d'identifier avec certitude le type de paramètre à traiter.

### a) SOLVERIF

```
Lat lon 'dateheure' 'statid' codemess alti pmer ndata(varno valeur flag1 flag2)
```

format fortran :

```
"( 2(f11.5,1x),1x,'''',i8.8,i6.6,''' ''',a8,' '' ',I3.3,1x,f7.1,1x,I6.6,1x,ndata(1x,i3,1x,ES17.10,2(1x,i2)),1x)"
```

Si l'altitude est manquante, *alti* (m) prend la valeur 99999.9.

Si la pression réduite au niveau de la mer est absente, *pmer* (Pa) prend la valeur 999999.

Si un paramètre météorologique est absent, *valeur* est mise à -2.1474836470E+09, *varno* prend la valeur 0, *flag1* et *flag2* prennent la valeur 99 (cf. chapitres 4 et 6).

**NB : une partie du quadruplet concernant les précipitations (varno 80) est particulière : *flag1* représente en fait la durée de cumul en heures.**

### b) TEMPVERIF

```
Lat lon 'dateheure' 'statid' codemess alti press niv ndata(varno valeur flag1 flag2)
```

format fortran :

```
"( 2(f11.5,1x),1x,'''',i8.8,i6.6,''' ''',a8,' '' ',I3.3,1x,f7.1,2x,i8,1x,i2,ndata(1x,i3,1x,ES17.10,2(1x,i2)))"
```

Si l'altitude est manquante, *alti* (m) prend la valeur 99999.9.

Si un paramètre météorologique est absent, *valeur* est mise à -2.1474836470E+09, *varno* prend la valeur 0, *flag1* et *flag2* prennent la valeur 99 (cf. chapitres 4 et 6).

*press* (pa) est le niveau pression auquel sont faites les mesures.

Le paramètre *niv* caractérise le type de niveau (cf. chapitre 5)

### c) PILOTVERIF

```
Lat lon 'dateheure' 'statid' codemess alti nivp nivh niv ndata(varno valeur flag1 flag2)
```

format fortran :

```
"( 2(f11.5,1x),1x,f7.1,1x,'''',i8.8,i6.6,''' ''',a8,' '' ',I3.3,1x,f7.1,1x,1x,i8,1x,i8,1x,i2,ndata(1x,i3,1x,ES17.10,2(1x,i2)))"
```

Si l'altitude de la station est manquante, *alti* (m) prend la valeur 99999.9.

Si *nivp* (niveau pression Pa) est renseigné, *nivh* (niveau m<sup>2</sup>/s<sup>2</sup>) est nul et vice et versa.

Le paramètre *niv* caractérise le type de niveau (cf. chapitre 5)

Si un paramètre météorologique est absent, *valeur* est mise à -2.1474836470E+09, *varno* prend la valeur 0, *flag1* et *flag2* prennent la valeur 99 (cf. chapitres 4 et 6).

# Description de la tâche FCQODB et du fichier BDM\_CQ

## 4- Constitution des flags

### a) constitution de flag1 – code qualité

$$\text{Flag1} = \text{I1} * 10 + \text{I2}$$

avec :

I1    1    absent de la liste noire  
      2    sur liste noire

I2    0    pas d'information  
      1    correct  
      2    incorrect

99    paramètre manquant

### b) Constitution de flag2 – code utilisation

$$\text{Flag2} = \text{I3}$$

avec :

I3    0    pas d'information  
      1    utilisé  
      2    non utilisé

99    paramètre manquant

## 5- Codification du type de niveau TEMPVERIF & PILOTVERIF (*niv*)

La ou les caractéristiques d'un niveau sont codés sur 6 bits et prennent les valeurs suivantes :

Niveau standard	Niveau significatif de température	Niveau significatif de vent	Niveau de vent maximum	Tropopause	Surface
32	16	8	4	2	1

Ainsi, pour un niveau standard et significatif de vent, *niv* prendra la valeur 40 et s'écrira 101000 en binaire.

## Description de la tâche FCQODB et du fichier BDM\_CQ

### 6- Correspondance varno et paramètres

<b>Varno</b>	<b>Paramètre</b>	<b>Unité</b>	<b>SOLVERIF</b>	<b>TEMPVERIF</b>	<b>PILOTVERIF</b>
0	ABSENT		X	X	X
1	Géopotentiel	m <sup>2</sup> /s <sup>2</sup>	X	X	
2	T	°K		X	
3	U			X	X
4	V			X	X
7	Q	Kg/Kg		X	
29	Hu	%/100		X	
39	T2m	°K	X		
41	U10m		X		
42	V10m		X		
58	Hu2m	%/100	X		
80	RR	Kg/m <sup>2</sup> /s	X		
91	Neb	%	X		

### 7- Correspondance codmess et type de message

<b>codmess</b>	<b>message</b>
11	Synop
14	Synop automatique
15	Synor
16	Radome
21	Ship
24	Ship automatique
32	Pilot
33	Pilotship
35	Temp
36	Tempship
37	Tempmobil
132	Pilotmobil
135	Tempdrop