IO IN THE CYCLE 45 OF ARPEGE/IFS: CALL-TREE ASPECTS.

YESSAD K. (METEO-FRANCE/CNRM/GMAP/ALGO)

June 28, 2017

Abstract:

This documentation gives an overview of the IO (file reading and writing) done in the most usual configurations. This is not a technical documentation about IO but rather a documentation describing entries to IO routines. It is not a comprehensive documentation for all configurations but sticks to the main configurations used for example in the MF operational suite. When locating file reading and writing in the code one has realized that this is not so easy to find where these IO are done in the code. The aim of this documentation is more precisely to give, for a subset of configurations, the list of files read and written, where they are read or written in the code, the local name of the file and the logical unit involved.

Résumé:

Cette documentation donne une vision des entrées-sorties (lecture et écriture sur fichiers) faites dans les configurations les plus utilisées. Ce n'est pas une documentation technique sur les IO mais plutôt une documentation donnant les points d'entrée aux routines lisant ou écrivant des fichiers. Ce n'est pas une documentation détaillée pour toutes les configurations; elle s'en tient aux principales configurations utilisées par exemple dans l'environnement opérationnel à METEO-FRANCE. Rechercher l'emplacement des lectures-écritures sur fichiers dans le code n'est pas vraiment aisé. Le but de cette documentation est plus précisément de donner, pour un certain jeu de configurations, la liste des fichiers lus et écrits, quelle partie de code les lit ou les écrit, le nom local des fichiers et le numéro de leur unité logique.

Contents

1 Introduction and purpose.

For a subset of usual configurations, one describes the list of files read and written, where they are read or written in the code, the local name of the file and the logical unit involved. This task is not done for a comprehensive list of configurations but rather for a subset of usual configurations which are listed below:

- ARPEGE configuration 1 with real data, no initialisation, with historical files writing (containing CFU and XFU fields).
- ALADIN or AROME configuration 1 with real data, no initialisation, with historical files writing (containing CFU and XFU fields).
- ARPEGE configuration 1 with real data, no initialisation, reading and writing restart files.
- ARPEGE configuration 1 with real data doing DFI initialisation (**LDFI**=T).
- Off-line FULL-POS configuration with departure and arrival climatologies.
- ARPEGE configuration 1 with real data, no initialisation, with in-line FULL-POS configuration with departure and arrival climatologies.
- ARPEGE configuration 1 with real data, no initialisation, and screening.
- ARPEGE configuration 1 with real data, no initialisation, and trajectory update.
- **LTRAJHR**=F incremental 4D-VAR minimisation (configuration 131).
- **LTRAJHR**=T incremental 4D-VAR minimisation (configuration 131).
- ARPEGE configuration 801 with real data used for sensitivity studies.
- ARPEGE configuration 601 with real data like the MF-PEARP one.
- ARPEGE configuration 701 with real data for CANARI surface assimilation (cf. MF operational suite).
- ARPEGE configuration 901 to convert ECMWF MARS files into MF ARPEGE files.
- Configuration to provide dilatation/contraction matrices.
- ARPEGE configuration 923 to provide climatological files.
- ARPEGE configuration 931 to provide SST files.
- ARPEGE configuration 932 to provide ice content files.

The following information will be given:

- List of files read and written.
- Local name of these files: they are often in some global variables of **YOMOPH0**, **YOMIOS** or **YOMFPOP** (FULL-POS), and respectively computed in **SUOPH0**, **SUIOS** and **SUFPOPH**. Some of them can be modified via the namelists **NAMOPH** and **NAMIOS**. About SURFEX files local names of some files are in **MODD_IO_SURF_ARO** (project "mse") and can be modified via namelist **NAMPHMSE**.
- "**KFILE**" of files (files read via **SUINIF** or **SUSPEC**, **SUGRIDF** and **SUGRIDU**, dummy argument of **SUINIF**).
- Type of files (FA, LFI, GRIB, ODB, BUFR, ASCII, IEEE binary, GSA).
- Logical unit: logical units are often in some global variables of **YOMLUN** (computed in **SULUN**).
- A piece of organigramme (at least for the FA, LFI and GRIB files) to easily locate in the ARP/ALD code where the files are read/written.

Description of deep call tree of IO routines is not done there; the aim of the current documentation is just to give the entry of the call tree.

For additional information about file formats, the reader can report on documentations (IDBUFR) for BUFR format, (IDFA) for ARPEGE-FA format, (IDGRIB), (IDGRIC) and (IDGRIG) for GRIB format, (IDGSA) for GSA format, (IDLFI) for LFI format, (IDODBO), (IDODBT), (IDODBU), (IDODBZ) and (TDECODB) for ODB format. For technical aspects and deeper call tree one can refer to documentation (IDTECIO).

2 Routines doing file reading/writing.

2.1 Preliminary remarks.

Excepted for configurations using BUFR, GRIB, ASCII, IEEE binary or GSA data, most files can be read/written in FA (ARPEGE) files for MF applications and in GRIB format for ECMWF applications. For a limited subset of files only the GRIB format is used (this is the case of the increment of the minimization) and both MF and ECMWF configurations can use them. For a limited subset of files only the FA (or LFI) format is used (in this case ECMWF cannot use them). METEO-FRANCE uses to read/write ARPEGE files due to historical operational constraints but in a theoretical point of view the use of GRIB files would be possible (at least for the points where the GRIB routines are the mirror of the FA routines, what is not always the case). ECMWF uses to read/write GRIB files.

There is a limited list of keys saying if FA or GRIB files must be read/written:

- When reading, the key **NGRIBFILE** is generally used (1 for GRIB files, 0 for ARPEGE files).

- When writing, the key **LARPEGEF** is generally used.

- There are specific keys **LARPEGEF_TRAJHR** to read trajectory files and **LARPEGEF_TRAJBG** to read background files in the **LTRAJHR**=T incremental 4D-VAR minimisation.

- There remains pieces of code where the choice between GRIB of FA files is done according to the value of the key **LECMWF** (this is not a clean feature).

- **NGRIBFILE** is in **YOMARG** and **NAMARG**. Other keys are in **YOMCT0** and **NAMCT0** and default values are computed in **SUCT0**.

As already said, the GRIB routines are not always the mirror of the ARPEGE files routines, and this is a point that makes a difficulty to import work done at ECMWF towards METEO-FRANCE (or vice-versa). The other problem generated by this situation is that a lot of work of code rewriting would be necessary if METEO-FRANCE decided to abandon FA files and to use only GRIB files. For example, METEO-FRANCE has historical files, but ECMWF rather uses model-level or pressure-level post-processing files (the distinction between post-processing files and historical files is clear at METEO-FRANCE but not at ECMWF). Another point of difference is the writing of post-processing files. METEO-FRANCE rather does a distinction between spectral fields (**WRSFP**) and grid-point fields (**WRHFP**) but no distinction between the different vertical coordinates of post-processing (model level, pressure, geopotential height, potential temperature, potential vorticity). On the contrary ECMWF does a distinction between the different vertical coordinates of post-processing but does no distinction between spectral data writing and grid-point writing: **WRMLFP** for model level, **WRPLFP** for pressure, potential temperature, potential vorticity level.

GRIB files require specific data (with a specific setup) which are not required for ARPEGE files, and this setup is generally written for ECMWF needs (for example the list of surface fields is not the same at ECMWF and METEO-FRANCE due to the use of different physical packages and this would be a new source of difficulties of adaptations if METEO-FRANCE decided to use GRIB files).

2.2 File reading.

The most frequent readings read files containing upper air spectral data (velocity potential, stream function, temperature, logarithm of surface pressure, spectral humidity), upper air grid-point data (grid-point GFL like liquid water, ice, cloud fraction, etc), surface spectral data (logarithm of surface pressure), surface spectral constants (surface orography), surface grid-point data. One generally finds the call of the following routines:

- A call to **SUSPEC** to read upper air "GMV" data, upper air spectral "GFL" data, and logarithm of surface pressure (which calls **SUSPECA** or **SUSPECA_GP** for FA files and **SUSPECG** for GRIB files). **SUSPEC** also (optionally) reads the spectral surface orography.

- A call to **SUGRIDF** to read surface grid-point data (which generally calls **SUGRIDA** for FA files and **SUGRIDG** for GRIB files).

- A call to **SUGRIDU** to read upper-air grid-point "GFL" data (which calls **SUGRIDUA** for FA files and **SUGRIDUG** for GRIB files).

These routines are sometimes called separately (for example in **SU1YOM** or **SUECGES**) and sometimes called via an interface routine **SUINIF** (this is the case in **CSTA** and **SUVAZX**).

There are several entries calling either **SUINIF** or the sequence (or a subset of the sequence) **SUSPEC**+**SUGRIDF**+**SUGRIDU**: in the following parts of the documentation we will mention the following entries: **SUECGES** (called by **SU0YOMB**), **SUVAZX** (called by **SU1YOM**), **SU1YOM**, **CSTA** (called by **CNT3**). Such a spread list of entries does not make easy the understanding of the part of the code reading files.

In LAM models (for example ALADIN, AROME) one must also read lateral boundary conditions files and the entry **SUINIF** (called by **CNT3**+**CSTA**) is replaced by the entry **SUEINIF** (called by **CNT3**+**ELSAC**).

Some other fields or files can be read for particular applications: CFU and XFU diagnostics (via the routines **SUGRCFU** and **SUGRXFU** coded only for FA files, not for GRIB files), departure and arrival climatologies for FULL-POS, preconditioning vectors for configuration 131, increment produced by the previous minimization (configuration 131), etc. The entries reading such data are generally spread in the code so it is not obvious to find them and some of them will be described later. Configuration 131, configuration 701, screening, trajectory update read specific files (BUFR, GRIB, ASCII, IEEE binary or GSA), the reading of them being often done in specific projects (other than ARP/ALD) and not described in detail in the current documentation.

When the **NSUPERSEDE** is set to 1, the frame of some files (initial conditions for example) is read in **SUARG** (called by **SU0YOMA**) to get some information (date and geometry). The call to **SUARG** will not be mentioned in the following descriptions.

When RRTM radiation scheme is used, additional files containing constants must be read (local names: C11CLIM, C12CLIM, C22CLIM, CCL4CLIM, ECOZC, GCH4CLIM, GCO2CLIM, GOZOCLIM, MCH4CLIM, MCO2CLIM, MOZOCLIM, N2OCLIM, NO2CLIM, RADRRTM). These ASCII files are read in different radiation set-up routines.

SRTM radiation scheme also requires some additional files containing constants.

2.3   File writing.

The main entry to write files is **IOPACK** (called by **STEPO**) and **IOPACK** calls a lot of **WR.....** routines to write data according to the application desired.

For example, to write historical files one calls **WRMLPPA** for FA files or **WRMLPPG** for GRIB files. In the call-tree under **WRMLPPA** we find:

- **WRSPECA** to write upper air spectral data and logarithm of surface pressure.
- **WRGRIDA**-named routines to write surface grid-point data.
- **WRGRIDUA**-named routines to write upper air grid-point data (grid-point GFL).
- **WRXFU** and **WRFU**-named routines to write XFU and CFU.
- **WRGP2FA** to write all grid-point fields, if **LUSEWRGRIDALL**=.T..

This code is called under **IOPACK**.

To write post-processing files one calls **WRHFP**+**WRSFP** for FA files and **WRMLFP**+**WRPLFP** for GRIB files. The main entry to write post-processing files is **IOFPOS** (called by **STEPO_FPOS**).

If **LUSEWRGRIDALL**=.F., call-tree under **WRMLPPA** is:

```
WRMLPPA ->
* WRSPECA
* WRGRIDA
* WRGRIDUA
* WRXFU (write XFU)
* WRFU (write CFU)
```

If **LUSEWRGRIDALL**=.T., call-tree under **WRMLPPA** is:

```
WRMLPPA ->
* WRSPECA
* WRGRIDALL
```

and a lot of operations are done under **WRGRIDALL**.

Routines called by **WRMLPPA** will not be repeated in the remaining part of this documentation.

Some other fields or files can be written for particular applications: preconditioning vectors for configuration 131, increment produced by the minimization (configuration 131), restart files, etc. The entries reading such data are generally spread in the code so it is not obvious to find them and some of them will be described later. Configuration 131, configuration 701, screening, trajectory update write specific files (BUFR, GRIB, ASCII, IEEE binary or GSA), the writing of them being often done in specific projects (other than ARP/ALD) and not described in detail in the current documentation.

3 ARPEGE configuration 1 with real data, no initialisation, with historical files writing (containing CFU and XFU fields).

3.1 Files read.

∗ **Initial conditions.**

- Local name: **CFNISH**=ICMSH[CODE]INIT.
- **KFILE**: 0.
- Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
- Logical unit: **NINISH**=81.
- Entry reading them (piece of organigramme):

```
CNT3 -> CSTA -> SUINIF -> SUSPEC+SUGRIDF+SUGRIDU
                          SUGRCFU (read CFU)
                          SUGRXFU (read XFU)
```

3.2 Files written.

∗ **Historical files.**

- Local name: **CFNSH**=ICMSH[CODE]+[eeee]. [eeee] is the time range (in hours or timesteps according to the value of variable **LINC**) on 4 digits.
- Type: FA if **LARPEGEF**=.T., GRIB if **LARPEGEF**=.F. .
- Logical unit: **NTRJSH**=71.
- Entry writing them (piece of organigramme):

```
FA files:    CNT4 -> STEPO -> IOPACK -> WRMLPP -> WRMLPPA
GRIB files:  CNT4 -> STEPO -> IOPACK -> WRMLPP -> WRMLPPG
```

∗ **DDH files.**

- Local name: **CFNDDH**=DHFTT[CODE]IMIN+[eeee].
- Type: LFA.
- Logical unit: **NPODDH**=53.
- Entry writing them (piece of organigramme):

```
CNT4 -> STEPO -> POSDDH -> PPFIDH
```

- Restriction of use: MF applications only.

3.3 Remarks.

- "CODE" is the four letter code of the experiment (example ARPE).
- ECMWF rather writes model data via FULL-POS.

```
CNT4 -> STEPO -> IOPACK -> WRMLPPLG
```

4 ALADIN or AROME configuration 1 with real data, no initialisation, with historical files writing (containing CFU and XFU fields).

4.1 Files read.

∗ **Initial conditions.**

- Local name: **CFNISH**=ICMSH[CODE]INIT.
- **KFILE**: 0.
- Type: FA (GRIB not coded).
- Logical unit: **NINISH**=81.
- Entry reading them (piece of organigramme):

```
CNT3 -> ELSAC -> SUEINIF -> SUSPEC+SUGRIDF+SUGRIDU
                            SUGRCFU (read CFU)
                            SUGRXFU (read XFU)
```

∗ **Additional files for SURFEX.**

- Local names: **CFILE**=ICMSH[CODE]INIT.sfx (prognostic SURFEX fields), **CFILEPGD**=Const.Clim.sfx (constant SURFEX fields), ecoclimapI_covers_param.bin, ecoclimapII_eu_covers_param.bin and ecoclimapII_af_covers_param.bin (town/land/sea masks).
- Type: FA (GRIB not coded) for files appended by "sfx", binary files for the other ones.
- Logical unit: computed in routine FMLOOK, not explicitly given (gives 33 for ICMSH[CODE]INIT.sfx).
- Entry reading them (piece of organigramme):

```
SUOYOMB -> SUPHY -> SUPHMF -> SUPHMSE -> SUPHMSE_SURFACE -> AROINI_SURF -> SUGRIDSFX
                                                                           AROINI_SURFB
```

- Restriction of use: MF applications only.

∗ **Lateral boundary conditions.**

- Local name: **CFNISH**=ELSCF[CODE]ALBC[nnn]. [nnn]= 3 digit numbering (000, 001, etc.).
- **KFILE**: 9.
- Type: FA (GRIB not coded).
- Logical unit: **NINISH**=81.
- Entry reading them (piece of organigramme):

```
CNT3 -> ELSAC -> SUEINIF -> SUSPEC+SUGRIDF+SUGRIDU
```

4.2 Files written.

∗ **Historical files.**

- Local name: **CFNSH**=ICMSH[CODE]+[eeee]. [eeee] is the time range (in hours or timesteps according to the value of variable **LINC**) on 4 digits.
- Type: FA (GRIB not available).
- Logical unit: **NTRJSH**=71.
- Entry writing them (piece of organigramme):

```
CNT4 -> STEPO -> IOPACK -> WRMLPPA
```

* **Additional files for SURFEX.**
  - Local name: AROMOUT_.[eeee].fa and AROMOUT_.[eeee].des (surface prognostic and diagnostic fields).
  - Type: FA (GRIB not coded).
  - Logical unit: computed in routine FMLOOK, not explicitly given.
  - Entry writing them (piece of organigramme):

        CNT4 -> STEPO -> IOPACK -> ARO_SURF_DIAGH -> WRMLPPA -> WRSFX

  - Restriction of use: MF applications only.

* **DDH files.**
  - Local name: **CFNDDH**=DHFTT[CODE]IMIN+[eeee].
  - Type: LFA.
  - Logical unit: **NPODDH**=53.
  - Entry writing them (piece of organigramme):

        CNT4 -> STEPO -> POSDDH -> PPFIDH

  - Restriction of use: MF applications only.

5 ARPEGE configuration 1 with real data, no initialisation, reading and writing restart files.

5.1 Files read.

∗ Restart files read.

- Local name:
    - **CFRCF**=RFILE[CODE] (control file).
    - **CIOSPRF**=RSPEC[CODE] (spectral data)
- Type: binary.
- Logical unit:
    - **NULRCF**=94.
- Entry reading them (piece of organigramme):

        CNT3 -> CNT3_GLO -> RERESF

5.2 Files written.

∗ Restart files written.

- Local name:
    - **CFRCF**=RFILE[CODE] (control file).
    - **CIOSPRF**=RSPEC[CODE] (spectral data)
- Type: binary.
- Logical unit:
    - **NULRCF**=94.
- Entry writing them (piece of organigramme):

        CNT4 -> STEPO -> IOPACK -> WRRESF

5.3 Remarks.

- Restart is not coded in LAM models.
- Same type of files for MF and ECMWF applications.

6 ARPEGE configuration 1 with real data doing DFI initialisation (LDFI=T).

6.1 Files read.

∗ **Initial conditions.**

- Local name: **CFNISH**=ICMSH[CODE]INIT.
- **KFILE**: 0.
- Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
- Logical unit: **NINISH**=81.
- Entry reading them (piece of organigramme):

```
CNT3 -> CSTA -> SUINIF -> SUSPEC+SUGRIDF+SUGRIDU
                          SUGRCFU (read CFU)
                          SUGRXFU (read XFU)
```

∗ **Bias file when reading it.**

- Local name: **CFNBS**=ICMSH[CODE]BIAS.
- **KFILE**: 3.
- Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
- Logical unit: **NBIAS**=90.
- Entry reading them (piece of organigramme):

```
ARP: CNT3 -> DFI -> DFI2 -> SUSPEC
ALD: CNT3 -> ELSAC -> DFI -> DFI2 -> SUSPEC
```

6.2 Files written.

∗ **Bias file when writing it.**

- Local name: **CFNSH**=ICMSH[CODE]+0000.
- Type: FA if **LARPEGEF**=.T., GRIB if **LARPEGEF**=.F. .
- Logical unit: **NTRJSH**=71.
- Entry writing them (piece of organigramme):

```
ARP: DFI3 -> STEPO -> IOPACK -> WRMLPP -> WRMLPPA
ALD: DFI3 -> STEPO -> IOPACK -> WRMLPPA
```

7 Off-line FULL-POS configuration with departure and arrival climatologies.

Example taken is a FULL-POS 5 domains, **CFPFMT**='LALON', **NFPOS**=1, but the following description remains valid for most of the off-line FULL-POS configurations.

7.1 No use of the IO server "FP_SERV".

7.1.1 Files read.

∗ **Initial conditions.**
- Local name: **CFNISH**=ICMSH[CODE]INIT.
- **KFILE**: 0.
- Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
- Logical unit: **NINISH**=81.
- Entry reading them (piece of organigramme):

```
CNT3 -> CSTA -> SUINIF -> SUSPEC+SUGRIDF+SUGRIDU
                          SUGRCFU (read CFU)
                          SUGRXFU (read XFU)
```

∗ **Input geometry climatology.**
- Local name: Const.clim (hard coded name).
- **KFILE**: 10.
- Type: FA (GRIB not available).
- Logical unit: **NULCL1**=10.
- Entry reading them (piece of organigramme):

```
CNT3 -> CSTA -> SUINIF -> SUGRIDF -> SUGRCLIA
```

∗ **Output geometry climatology.**
- Local name: Const.clim.[DDDDDD] (hard coded name). [DDDDDD] is the name of the domain (for example GLOB15).
- Type: FA if **LARPEGEF**=.T., GRIB if **LARPEGEF**=.F. .
- Logical unit: **NULFP01**=54, and optionally **NULFP02**=50, **NULFP03**=43, **NULFP04**=30, **NULFP05**=19, ... (one per domain if there are several domains) when required.
- Entry reading them (piece of organigramme):

```
SUOYOMB -> SUBFPOS -> SUFPRFPBUF_CLIM -> RDCLIMO (FA) or RDECCLIMO (GRIB)
```

∗ **Filtering matrices.**
- Local name: matrix.fil.[DDDDDD] (hard coded name).
- Type: LFI.
- Logical unit: **NULFP01**=54, and optionally **NULFP02**=50, **NULFP03**=43, **NULFP04**=30, **NULFP05**=19, ... (one per domain if there are several domains) when required.
- Entry reading them (piece of organigramme):

```
SUOYOMB -> SUBFPOS -> SUFPF -> RDFPFILTER
```

- Restrictions of use: METEO-FRANCE applications only when the departure geometry is a stretched ARPEGE one.

7.1.2   Files written.

∗ **Post-processing files.**

- Local name: **CFPFN**(domain)=PF[CODE][DDDDDD]+0000.
- Type: FA if **LARPEGEF**=.T., GRIB if **LARPEGEF**=.F. .
- Logical unit: **NULFP01**=54, and optionally **NULFP02**=50, **NULFP03**=43, **NULFP04**=30, **NULFP05**=19, ... (one per domain if there are several domains) when required.
- Entry writing them (piece of organigramme):

```
FA  : CNT4 -> DYNFPOS -> IOFPOS -> WRHFP (gp fields) and WRSFP (spectral fields)
      CNT4 -> GRIDFPOS -> WRHFP (surface gp fields, CFU, XFU)
GRIB: CNT4 -> DYNFPOS -> IOFPOS -> WRPLFP, WRMLFP
      CNT4 -> GRIDFPOS -> WRMLFP
```

7.1.3   Configuration 903.

Configuration 903 is already coded and will replace configuration 1 for off-line FULL-POS.

- replace "**CNT3** − > **CSTA** − > **SUINIF**" by "**CPREP3** − > **SUFPINIF**".
- **SUFPRFPBUF_CLIM**, **DYNFPOS**, **GRIDFPOS** are called by **CPREP3**.

7.2   Use of the IO server "FP_SERV".

The call-tree may be different (not described).

8 ARPEGE configuration 1 with real data, no initialisation, with in-line FULL-POS configuration with departure and arrival climatologies.

8.1 Files read.

Same ones as for off-line FULL-POS configurations.

8.2 Files written.

Same ones as for off-line FULL-POS configurations but the local name of the post-processing files is now **CFPFN**(domain)=PF[CODE][DDDDDD]+[eeee] ([eeee] is the time range, in hours or timesteps according to the value of variable **LINC**, on 4 digits).

9 ARPEGE configuration 1 with real data, no initialisation, and screening.

The example given matches the screening of the operational MF suite excepted that we assume that there is no DFI initialisation. In the operational MF suite, screening runs with a DFI initialisation and there is an additional bias output file.

9.1 Files read.

∗ **Initial conditions.** See section 3

∗ **Additional files read.** This configuration reads:

- ECMA: local name = ECMA.[part], type = ODB.

and a significant amount of other files (with different formats) not detailed in this documentation. Entry reading them is not described.

9.2 Files written.

∗ **Historical files.** See section 3

∗ **Additional files written.**

- Local name and type:

  - ECMA_screened: local name = ECMA._screened[part], type = ODB.
  - Varbc out: local name = VARBC.cycle, type = ASCII.

- Entry writing them (piece of organigramme): specific entries not in ARP/ALD, not described.

10 ARPEGE configuration 1 with real data, no initialisation, and trajectory update.

The example given matches the last trajectory update of the operational MF suite excepted that we assume that there is no DFI initialisation. In the operational MF suite, the last trajectory update runs with a DFI initialisation and there is an additional bias input file.

10.1 Files read.

∗ **Initial conditions.**   See section 3

∗ **Low resolution background (case NUPTRA>0).**
- Local name: **CFNFGI**=ICMSH[CODE]FGIN.
- **KFILE**: 6.
- Type: FA (**LECMWF**=.F.).
- Logical unit: **NULASE**=16.
- Entry reading them (piece of organigramme):

        CNT3 -> UPSPEC -> RDFPINC -> SUINIF -> SUSPEC+SUGRIDF+SUGRIDU

- Restrictions of use: METEO-FRANCE applications only.

∗ **Incremented low resolution analysis (case NUPTRA>0).**
- Local name: **CFNANI**=ICMSH[CODE]ANIN.
- **KFILE**: 7.
- Type: FA (**NGRIBFILE**=0, **LECMWF**=.F.).
- Logical unit: **NULASS**=17.
- Entry reading them (piece of organigramme):

        CNT3 -> UPSPEC -> RDFPINC -> SUINIF -> SUSPEC+SUGRIDF+SUGRIDU

- Restrictions of use: METEO-FRANCE applications only.

∗ **Low resolution increment (case NUPTRA>0).**
- Local name: **CFNANI**=ICMSH[CODE]ANIN.
- **KFILE**: 7.
- Type: GRIB (**NGRIBFILE**=1, **LECMWF**=.T.).
- Logical unit: **NULASS**=17.
- Entry reading them (piece of organigramme):

        CNT3 -> UPSPEC -> RDFPINC -> SUINIF -> at least SUSPEC

- Restrictions of use: ECMWF applications only.

∗ **Additional files read.**   This configuration reads:
- CCMA: local name = CCMA, type = ODB.

and a significant amount of other files (with different formats) not detailed in this documentation. Entry reading them is not described.

10.2 Files written.

∗ Output trajectory.

- Local name: **CFNSH**=ICMSH[CODE]+0000.
- Type: FA if **LARPEGEF**=.T., GRIB if **LARPEGEF**=.F. .
- Logical unit: **NTRJSH**=71.
- Entry writing them (piece of organigramme):

```
FA files:    CNT4 -> STEPO -> IOPACK -> WRMLPP -> WRMLPPA
GRIB files:  CNT4 -> STEPO -> IOPACK -> WRMLPP -> WRMLPPG
```

∗ Additional files written.

- Local name and type:
    - CCMA_traj: local name = CCMA, type = ODB.
- Entry writing them (piece of organigramme): specific entries not in ARP/ALD, not described.

11 LTRAJHR=F incremental 4D-VAR minimisation (configuration 131).

The example given matches the minimizations done in the operational MF suite but we consider rather an intermediate minimization in a system with at least 3 minimizations: in the current example, there are preconditioning vectors read and preconditioning vectors written. This example assumes that the minimizer is **CONGRAD**.

11.1 Files read.

∗ **Low resolution (non incremented) background.**
- Local name: **CFNRF**=ICMRF[CODE]0000.
- **KFILE**: 2.
- Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
- Logical unit: **NFGISH**=83.
- Entry reading them (piece of organigramme):

      SU0YOMB -> SUECGES -> SUSPEC

∗ **Low resolution incremented guess (initial conditions for the trajectory).**
- Local name: **CFNGSH**=ICMSH[CODE]IMIN.
- **KFILE**: 1.
- Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
- Logical unit: **NINMSH**=79.
- Entry reading them (piece of organigramme):

      SU0YOMB -> SUECGES -> SUGRIDU+SUGRIDF
      CVA1 -> SU1YOM -> SUINIF -> SUSPEC+SUGRIDU+SUGRIDF

- Remark: grid-point fields are read twice, first under **SUECGES**, then under **SUVAZX**.

∗ **Increment coming from the previous minimization.**
- Local name: traj00+0000000 (hard coded name).
- Type: GRIB.
- Entry reading them (piece of organigramme):

      CVA1 -> SU1YOM -> SUVAZX -> GETMINI

∗ **Preconditioning vectors coming from the previous minimization.**
- Local name: MEMINI (hard coded name).
- Type: GRIB.
- Entry reading them (piece of organigramme):

      CNT0 -> CVA1 -> PREPPCM -> GETMINI

∗ **Additional files read.** This configuration reads:
- CCMA: local name = CCMA, type = ODB.

and a significant amount of other files (with different formats) not detailed in this documentation. Entry reading them is not described.

11.2 Files written.

∗ **Minim_analyse (incremented guess with the increment of the current minimization).**

- Local name: MX[CODE]999+0000 (local variable computed from **CFNAN** in **INIFAOUTINFO** (FA) or **WROUTSPGB** (GRIB)).
- Type: FA if **LARPEGEF**=.T., GRIB if **LARPEGEF**=.F. .
- Logical unit: **NTRJSH**=71.
- Entry writing them (piece of organigramme):

      STEPOTL -> IOPACK -> WRMLPPA

∗ **Minim_stp (guess without the increment of the current minimization, but with the increment of the previous minimizations?).**

- Local name: **CFNAN**=MX[CODE]000+0000.
- Type: FA if **LARPEGEF**=.T., GRIB if **LARPEGEF**=.F. .
- Logical unit: **NTRJSH**=71.
- Entry writing them (piece of organigramme):

      STEPOTL -> IOPACK -> WRMLPPA

- Remark: file identical to the input file ICMSH[CODE]IMIN or not?

∗ **Increment produced by the current minimization.**

- Local name: traj01+0000000 (hard coded name).
- Type: GRIB.
- Entry writing them (piece of organigramme):

      CNT0 -> CVA1 -> CVA2 -> SAVMINI or SAVMINI2

∗ **Preconditioning vectors produced by the current minimization.**

- Local name: MEMINI (hard coded name).
- Type: GRIB.
- Entry writing them (piece of organigramme):

      CNT0 -> CVA1 -> CVA2 -> XFORMEV -> SAVMINI or SAVMINI2

∗ **Additional files written.**

- Local name and type:
  - CCMA_min: local name = CCMA, type = ODB.
  - Varbc out: local name = VARBC.cycle, type = ASCII.
- Entry writing them (piece of organigramme): specific entries not in ARP/ALD, not described.

11.3 Remarks.

File **CFNISH**=ICMSH[CODE]INIT is not read in this configuration (excepted maybe in **SUARG** when **NSUPERSEDE** is set to 1).

12 LTRAJHR=T incremental 4D-VAR minimisation (configuration 131).

The only difference with the example shown in chapter 11 is the switch on of the option **LTRAJHR**=.T. : additional trajectory files must be read.

12.1 Files read.

∗ **Low resolution (non incremented) background.**

- Local name: **CFNRF**=ICMRF[CODE]0000 (entry via **SUSPEC**) and **CFNTRAJBG**=ICMRF[CODE]0000 (entry via **READ_TRAJECTORY**).
- **KFILE**: 2.
- Type: FA if **NGRIBFILE**=0 and **LARPEGEF_TRAJBG**=.T., GRIB if **NGRIBFILE**=1 and **LARPEGEF_TRAJBG**=.F. .
- Logical unit: **NFGISH**=83.
- Entry reading them (piece of organigramme):

```
SUOYOMB -> SUECGES -> SUSPEC
```

and

```
SUOYOMB -> SUECGES -> READ_TRAJECTORY ->
* READ_SURFGRID_TRAJ (GRIB) or READ_SURFGRID_TRAJ_FROMFA (FA)
* READ_TRAJ_SPEC ->
  - READ_SPEC (GRIB) or READ_SPEC_FROMFA (FA)
* READ_TRAJ_GRID ->
  - READ_GRID_TRAJ (GRIB) or READ_GRID_TRAJ_FROMFA (FA)
  - READ_SPEC (GRIB) or READ_SPEC_FROMFA (FA)
```

∗ **Low resolution incremented guess (initial conditions for the trajectory).**

- Local name: **CFNGSH**=ICMSH[CODE]IMIN.
- **KFILE**: 1.
- Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
- Logical unit: **NINMSH**=79.
- Entry reading them (piece of organigramme):

```
SUOYOMB -> SUECGES -> SUGRIDF
SU1YOM -> SUINIF -> SUSPEC+SUGRIDU+SUGRIDF
```

- Remark: surface grid-point fields are read twice, first under **SUECGES**, then under **SUVAZX**.

and additionally:

- Local name: **CFNTRAJHR**=ICMSH[CODE]IMIN or TRAJ[CODE]+0000 (both files must be identical for METEO-FRANCE purpose).
- **KFILE**: 1.
- Type: FA if **LARPEGEF_TRAJHR**=.T., GRIB if **LARPEGEF_TRAJHR**=.F. .
- Logical unit: **NULTRAJHR**=**NINMSH**=79.
- Entry reading them (piece of organigramme):

```
SUOYOMB -> SUECGES -> READ_TRAJECTORY ->
* READ_SURFGRID_TRAJ (GRIB) or READ_SURFGRID_TRAJ_FROMFA (FA)
* READ_TRAJ_SPEC ->
  - READ_SPEC (GRIB) or READ_SPEC_FROMFA (FA)
* READ_TRAJ_GRID ->
  - READ_GRID_TRAJ (GRIB) or READ_GRID_TRAJ_FROMFA (FA)
  - READ_SPEC (GRIB) or READ_SPEC_FROMFA (FA)
```

- Remark: for METEO-FRANCE purpose TRAJ[CODE]+0000 should not be read: the code is forced to read ICMSH[CODE]IMIN. The code actually enters READ_SPEC_FROMFA which does nothing in this case.

∗ **High resolution trajectory transformed into low resolution.**

- Local name: **CFNTRAJHR**=TRAJ[CODE]+[nnnn] ([nnnn] is a numbering index on 4 digits).
- Type: FA if **LARPEGEF_TRAJHR**=.T., GRIB if **LARPEGEF_TRAJHR**=.F. .
- Logical unit: **NULTRAJHR**=72.
- Entry reading them (piece of organigramme):

```
CNT4 -> READ_TRAJECTORY ->
* READ_SURFGRID_TRAJ (GRIB) or READ_SURFGRID_TRAJ_FROMFA (FA)
* READ_TRAJ_SPEC ->
  - READ_SPEC (GRIB) or READ_SPEC_FROMFA (FA)
* READ_TRAJ_GRID ->
  - READ_GRID_TRAJ (GRIB) or READ_GRID_TRAJ_FROMFA (FA)
  - READ_SPEC (GRIB) or READ_SPEC_FROMFA (FA)
```

∗ **Additional GRIB files to get headers (useful only when GRIB files are read for trajectory and background).**

- Local names: "main_gg" (trajectory, case **LTRAJHR_ALTI**=T), "lowres_gg" (trajectory, case **LTRAJHR_SURF**=T), "backgr_gg[nn]" with [nn]=00,01,..=numbering of the update (background, case **LTRAJHR_ALTI**=T).
- Type: GRIB only (case **LARPEGEF_TRAJHR**=F, **LARPEGEF_TRAJBG**=F).
- Logical unit: none.
- Entry reading them (piece of organigramme):

```
SUOYOMB -> SUALLT -> ALLOCATE_TRAJECTORY -> INIT_TRAJECTORY
```

∗ **Increment coming from the previous minimization.** cf. case **LTRAJHR**=F.

∗ **Preconditioning vectors coming from the previous minimization.** cf. case **LTRAJHR**=F.

∗ **Additional files read.** cf. case **LTRAJHR**=F.

12.2 Files written.

cf. case **LTRAJHR**=F.

12.3 Remarks.

File **CFNISH**=ICMSH[CODE]INIT is not read in this configuration (excepted maybe in **SUARG** when **NSUPERSEDE** is set to 1).

13 ARPEGE configuration 801 with real data used for sensitivity studies.

13.1 Files read.

* **Reference file.**
  - Local name: **CFNRF**=ICMRF[CODE]0000.
  - **KFILE**: 2.
  - Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
  - Logical unit: **NFGISH**=83.
  - Entry reading them (piece of organigramme):

        STEPO -> IOPACK -> RD801 -> SUSPEC

* **Minimization file.**
  - Local name: **CFNGSH**=ICMSH[CODE]IMIN.
  - **KFILE**: 1.
  - Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
  - Logical unit: **NINMSH**=79.
  - Entry reading them (piece of organigramme):

        SU1YOM -> SUVAZX -> SUINIF -> SUSPEC+SUGRIDU+SUGRIDF

* **Initial conditions.**
  - Local name: **CFNISH**=ICMSH[CODE]INIT.
  - **KFILE**: 0.
  - Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
  - Logical unit: **NINISH**=81.
  - Entry reading them (piece of organigramme):

        SU1YOM -> SUSPEC+SUGRIDF+SUGRIDU

13.2 Files written.

* **Gradient at initial timestep.**
  - Local name: **CFNGR**=MG[CODE]000+0000.
  - Type: FA if **LARPEGEF**=.T., GRIB if **LARPEGEF**=.F. .
  - Logical unit: **NTRJSH**=71.
  - Entry writing them (piece of organigramme):

        ARP: IOPACK -> WRMLPP -> WRMLPPA
        ALD: IOPACK -> WRMLPPA

* **Gradient at final timestep.**
  - Local name: **CFNGR**=MG[CODE]000+[eeee]. [eeee] is the time range (in hours or timesteps according to the value of variable **LINC**) on 4 digits.
  - Type: FA if **LARPEGEF**=.T., GRIB if **LARPEGEF**=.F. .
  - Logical unit: **NTRJSH**=71.
  - Entry writing them (piece of organigramme):

        ARP: IOPACK -> WRMLPP -> WRMLPPA
        ALD: IOPACK -> WRMLPPA

14 ARPEGE configuration 601 with real data like the MF-PEARP one.

14.1 Files read.

∗ **Minimization file.**
- Local name: **CFNGSH**=ICMSH[CODE]IMIN.
- **KFILE**: 1.
- Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
- Logical unit: **NINMSH**=79.
- Entry reading them (piece of organigramme):

```
SU1YOM -> SUVAZX -> SUINIF -> SUSPEC+SUGRIDU+SUGRIDF
```

∗ **Initial conditions.**
- Local name: **CFNISH**=ICMSH[CODE]INIT.
- **KFILE**: 0.
- Type: FA if **NGRIBFILE**=0, GRIB if **NGRIBFILE**=1 .
- Logical unit: **NINISH**=81.
- Entry reading them (piece of organigramme):

```
SU1YOM -> SUSPEC+SUGRIDF+SUGRIDU
```

14.2 Files written.

∗ **Singular vectors at the initial and final instants, for METEO-FRANCE applications.**
- Local name: SV[CODE][nnn]+0000 (initial instant), SV[CODE][nnn]+[eeee] (final instant). [eeee] is the time range (in hours or timesteps according to the value of variable **LINC**) on 4 digits.
- Type: FA.
- Logical unit: **NTRJSH**=71.
- Entry writing them (piece of organigramme):

```
ARP: CUN1 -> CUN2 -> NALAN1 -> WRTSV -> STEPO -> IOPACK -> WRMLPPA
ALD: CUN1 -> CUN2 -> NALAN1 -> EWRTSV -> STEPO -> IOPACK -> WRMLPPA
```

∗ **Singular vectors, for ECMWF applications.**
- Local name: svifs (hard coded name).
- Type: GRIB.
- Entry writing them (piece of organigramme):

```
CUN1 -> CUN2 -> NALAN1 -> WRTLLCZ
```

∗ **File "lczwrk" for each processor.**
- Local name: lczwrk[ppp] (hard coded name); [ppp] is the numbering of the processor on 3 digits.
- Type: ?.
- Logical unit: **NULUSR1**=95.
- Entry writing them (piece of organigramme):

```
CUN1 -> CUN2 -> NALAN1
```

15 ARPEGE configuration 701 with real data for CANARI surface assimilation (cf. MF operational suite, LAEINC=.T.).

15.1 Files read.

∗ **Initial conditions (Guess6).**
- Local name: **CFNISH**=ICMSH[CODE]INIT.
- **KFILE**: 0.
- Type: FA.
- Logical unit: **NINISH**=81.
- Entry reading them (piece of organigramme):

      CNT0 -> CAN1 -> SUINIF -> SUSPEC+SUGRIDF+SUGRIDU

∗ **Nearest month climatology file (Clim_1).**
- Local name: ICMSH[CODE]CLI2 (hard coded name).
- **KFILE**: 0.
- Type: FA.
- Logical unit: 18 (hard coded).
- Entry reading them (piece of organigramme):

      CNT0 -> CAN1 -> SUINIF -> SUGRIDF -> SUGRCLIA

∗ **Current month climatology file (Clim_2).**
- Local name: ICMSH[CODE]CLIM (hard coded name).
- **KFILE**: 0.
- Type: FA.
- Logical unit: 17 (hard coded).
- Entry reading them (piece of organigramme):

      CNT0 -> CAN1 -> SUINIF -> SUGRIDF -> SUGRCLIA

∗ **Low resolution (incremented) analysis (Traj).**
- Local name: **CFNANI**=ICMSH[CODE]ANIN.
- **KFILE**: 7.
- Type: FA.
- Logical unit: **NULASS**=17.
- Entry reading them (piece of organigramme):

      CNT0 -> CAN1 -> CANARI -> RDFPINC -> SUINIF -> SUSPEC+SUGRIDF+SUGRIDU

∗ **Low resolution (non incremented) background (Guess3+3).**
- Local name: **CFNFGI**=ICMSH[CODE]FGIN.
- **KFILE**: 6.
- Type: FA.
- Logical unit: **NULASE**=16.
- Entry reading them (piece of organigramme):

      CNT0 -> CAN1 -> CANARI -> RDFPINC -> SUINIF -> SUSPEC+SUGRIDF+SUGRIDU

* **File containing regression coefficients for soil moisture (Isba).**
  - Local name: fort.61 (hard coded name).
  - Type: ASCII.
  - Logical unit: 61 (hard coded).
  - Entry reading them (piece of organigramme):

        CNT0 -> CAN1 -> CANARI -> CAVEGI

* **SST file.**
  - Local name: ICMSH[CODE]SST (hard coded name).
  - Type: FA.
  - Logical unit: **NULCL2**=11.
  - Entry reading them (piece of organigramme):

        CNT0 -> CAN1 -> CANARI -> CACLSST

* **sea ice file (SEAICE).**
  - Local name: ICMSH[CODE]SEAICE (hard coded name).
  - Type: FA.
  - Logical unit: **NULCL1**=12.
  - Entry reading them (piece of organigramme):

        CNT0 -> CAN1 -> CANARI -> CACLSST

* **Additional files read.**   This configuration reads:
  - CCMA: local name = CCMA_CAN, type = ODB.

and some other files (with different formats) not detailed in this documentation. Entry reading them is not described.

15.2   Files written.

* **Output (analysed) file.**
  - Local name: **CFNSH**=ICMSH[CODE]+0000.
  - Type: FA.
  - Logical unit: **NTRJSH**=71.
  - Entry writing them (piece of organigramme):

        CNT0 -> CAN1 -> CANARI -> STEPO -> IOPACK -> WRMLPPA

* **Additional files written.**
  - Local name and type:
    - CCMA_CANS: local name = CCMA_CAN, type = ODB.
  - Entry writing them (piece of organigramme): specific entries not in ARP/ALD, not described.

15.3   Remarks.

The configuration described above is not designed to be run at ECMWF (no GRIB version for some of the files).

16 ARPEGE configuration 901 to convert ECMWF MARS files into MF ARPEGE files.

16.1 Files read.

∗ **Initial GRIB/MARS files.** One or several GRIB files: the example given (cf. conf 901 under OLIVE) reads five GRIB files.

- Local names:

    - Ecmwf_gausss: ICMUAb001INIT+[eeee].
    - Ecmwf_gaussq: ICMGGa001INIT+[eeee].
    - Ecmwf_spectrd: ICMUAa001INIT+[eeee].
    - Ecmwf_spectrt: ICMSHa001INIT+[eeee].

    These default names can be modified via namelist **NAMMARS**.

- Type: GRIB.
- Logical unit: **NINISH**=81.
- Entry reading them (piece of organigramme):

        CPREP1

∗ **Climatology for ARPEGE files.**
- Local name: Const.clim (hard coded name).
- **KFILE**: 10.
- Type: FA (GRIB not available).
- Logical unit: **NULCL1**=10.
- Entry reading them (piece of organigramme):

        CPREP1

16.2 Files written.

∗ **Output ARPEGE file.**
- Local name: CN90x[CODE]INIT (hard coded name).
- Type: FA.
- Logical unit: **NPOSSH**=51.
- Entry writing them (piece of organigramme):

        CPREP1

17   Configuration to provide dilatation/contraction matrices.

17.1   Files read.

None.

17.2   Files written.

∗ Dilatation and contraction matrices.

- Local name: MATDILA for dilatation and MATCONT for contraction (hard coded names).
- Type: LFI.
- Logical unit: **IULDILA**=46 for dilatation and **IULCONT**=47 for contraction.
- Entry writing them (piece of organigramme, in project "utilities"):

        MASTER911 -> SUDIL -> DILATB

18 ARPEGE configuration 923 to provide climatological files (MF only).

18.1 Part 1.

18.1.1 Files read (common for all months).

∗ **Various surface data.**

- Local names: Water_percentage, Oro_mean, etc... (hard coded names).
- Type: IEEE binary.
- Logical unit: 31 to 39 (hard coded).
- Entry reading them (piece of organigramme): **INCLI1**.

∗ **Input orography when required.**

- Local name: Const.Clim.Surfa (hard coded name).
- Type: FA.
- Logical unit: 41 (hard coded).
- Entry reading them (piece of organigramme): **INCLI1**.

18.1.2 Files written (common for all months).

∗ **Output climatology (surface orography field).**

- Local name: Const.Clim (hard coded name).
- Type: FA.
- Logical unit: 41 (hard coded).
- Entry writing them (piece of organigramme): **INCLI1**.

18.2 Part 2.

18.2.1 Files read (common for all months).

∗ **Various surface data.**

- Local names: itp_GL, alb_GL, emi_GL, dps_GL, arg_GL, sab_GL, vgx_GL, dpr_GL (hard coded names).
- Type: IEEE binary or ASCII according to **LIEEE**.
- Logical unit: 11 to 18 (hard coded).
- Entry reading them (piece of organigramme): **INCLI2**.

∗ **Input climatology (output of part 1).**

- Local name: Const.Clim.Surfa (hard coded name).
- Type: FA.
- Logical unit: 3 (hard coded).
- Entry reading them (piece of organigramme): **INCLI2**.

18.2.2 Files written (common for all months).

∗ **Output climatology.**

- Local name: Const.Clim (hard coded name).
- Type: FA.
- Logical unit: 3 (hard coded).
- Entry writing them (piece of organigramme): **INCLI2**.

18.3 Part 3.

18.3.1 Files read (common for all months).

∗ **Various surface data.**

- Local names: N108_GL (hard coded name).
- Type: IEEE binary or ASCII according to **LIEEE**.
- Logical unit: 10 (hard coded).
- Entry reading them (piece of organigramme): **INCLI3**.

∗ **Input climatology (output of part 2).**

- Local name: Const.Clim.Surfa (hard coded name).
- Type: FA.
- Logical unit: 3 (hard coded).
- Entry reading them (piece of organigramme): **INCLI3**.

18.3.2 Files written (one per month).

∗ **Output climatology.**

- Local name: Const.Clim (hard coded name).
- Type: FA.
- Logical unit: 3 (hard coded).
- Entry writing them (piece of organigramme): **INCLI3**.

18.4 Parts 4 to 9.

18.4.1 Files read (one per month).

∗ **Various surface data.**

- Local names: hard coded names in the INCLI[part] routine.
- Type: IEEE binary or ASCII according to **LIEEE**.
- Logical unit: generally between 10 and 25 (hard coded).
- Entry reading them (piece of organigramme): **INCLI**[part].

∗ **Input climatology (output of part [part]-1).**

- Local name: Const.Clim.Surfa (hard coded name).
- Type: FA.
- Logical unit: 3 (hard coded).
- Entry reading them (piece of organigramme): **INCLI**[part].

18.4.2 Files written (one per month).

∗ **Output climatology.**

- Local name: Const.Clim (hard coded name).
- Type: FA.
- Logical unit: 3 (hard coded).
- Entry writing them (piece of organigramme): **INCLI**[part].

Output of part 9 provides the final climatologies.

19 ARPEGE configuration 931 to provide SST files.

19.1 Files read.

∗ **SST NESDIS file (SSTnesdis).**
- Local name: sst.nesdis.bdap (hard coded name).
- Type: IEEE binary.
- Logical unit: 3 (hard coded).
- Entry reading them (piece of organigramme): **CNT0 − > CSSTBLD − > INCLITC**

∗ **Climatology (Global_Clim.**
- Local name: ICMSH[CODE]INIT (hard coded name).
- Type: FA (GRIB not available).
- Logical unit: **NINISH**=81.
- Entry reading them (piece of organigramme): **CNT0 − > CSSTBLD − > INCLITC**

19.2 Files written.

∗ **ARPEGE SST file.**
- Local name: ICMSH[CODE]SST (hard coded name).
- Type: FA.
- Logical unit: 15 (hard coded).
- Entry writing them (piece of organigramme): **CNT0 − > CSSTBLD − > INCLITC**

19.3 Remarks.

Configuration for MF applications only.

20 ARPEGE configuration 932 to provide ice content files.

20.1 Files read.

∗ **Sea ice content file (Ice_content).**
- Local name: ice_content (hard coded name).
- Type: ASCII.
- Logical unit: 16 (hard coded).
- Entry reading them (piece of organigramme): **CNT0 − > CSEAICE**

∗ **Climatology (Global_Clim.**
- Local name: ICMSH[CODE]INIT (hard coded name).
- Type: FA (GRIB not available).
- Logical unit: **NULCL1**=10.
- Entry reading them (piece of organigramme): **CNT0 − > CSEAICE**

20.2 Files written.

∗ **ARPEGE sea ice file.**
- Local name: ICMSH[CODE]SEAICEF (hard coded name).
- Type: FA.
- Logical unit: **NULCL2**=11.
- Entry writing them (piece of organigramme): **CNT0 − > CSEAICE**

20.3 Remarks.

Configuration for MF applications only.

21 References and internal notes.

21.1 Documentations about file formats.

- (IDBUFR) Thorpe, W., 1998: A guide to the WMO code form FM 94 BUFR. Internal note, 89pp, available on the internet server "http://rda.ucar.edu/docs/formats/bufr/" (file "bufr.pdf").
  (Remark: there are other "pdf" files dated from 2002 giving additional documentation).

- (IDFA) Clochard, J., R. El Khatib and D. Paradis, 2002: FA subroutines of the ARPEGE/ALADIN files package (Interfaces de manipulation des fichiers ARPEGE/ALADIN) Internal note, 34 pp; is an updated version of (NTA17) valid for cycle 24, available in English and French on the internet server "http://www.umr-cnrm.fr/gmapdoc/" (topic "files and data").

- (IDGRIB) (updated version of (IDGRIG) in 2000, in HTML form, see below).

- (IDGRIC) GRIB Format, Edition 1. Internal note, available on the internet server "http://weather.unisys.com/wxp/Appendices/Formats/GRIB.html".

- (IDGRIG) Dey, C.H., 1998: The WMO format for the storage of weather product information and the exchange of weather product messages in gridded binary form as used by NCEP central operations. Internal note, 106pp, available on the internet server "http://rda.ucar.edu/docs/formats/" (topic "WMO GRIB Edition 1, Adobe PDF").

- (IDGSA) Bouttier, F., 1996: Specifications of the GSA file format (version 1.1). Internal note, available on the internet server "http://intra.cnrm.meteo.fr/assim/doc/gsaspec.html".

- (IDLFI) Bouttier, F., and J. Clochard, 2002: The LFI file access library (valid for cycle 24T1). (internal note, 8 pp; is an updated version of (NTA12)). Available on the internet server "http://www.umr-cnrm.fr/gmapdoc/" (topic "files and data").

- (IDODBO) Kertész, S., 2001: A short overview of ODB. Internal note, 14 pp, available on the internet server "http://www.umr-cnrm.fr/gmapdoc/" (topic "ODB").

- (IDODBT) Puech, D., 2009: Documentation technique sur ODB. compilation of internal notes (generally in French) available on the internet server "http://www.umr-cnrm.fr/gmapdoc/" (topic "ODB").

- (IDODBU) Kertész, S., 2002: Using ODB in ALADIN. Internal note, 21pp, available on the internet server "http://www.umr-cnrm.fr/gmapdoc/" (topic "ODB").

- (IDODBZ) Fouilloux, A., 2013: ODB documentation (several .tar files, restricted access on ECMWF intranet).

- (TDECODB) Saarinen, S., 2004: IFS technical documentation (CY28R1). Part I: ODB.

21.2 Some internal notes and other ARPEGE notes.

- (TDECDAS) 2016: IFS technical documentation (CY41R2). Part II: data assimilation.

- (TDECTEC) 2016: IFS technical documentation (CY41R2). Part VI: technical and computational procedures. Available at "https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation".

- (IDBAS) Yessad, K., 2017: Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 45 of ARPEGE/IFS (internal note).

- (IDLAM) Zagar, M., and C. Fischer, 2007: The ARPEGE/ALADIN Tech'Book: Implications of LAM aspects on the global model code for CY33/AL33. Internal note, 31pp. Available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/".

- (IDTECIO) Marguinaud, Ph., 2013: ARPEGE/ALADIN/AROME IO in 39t1. Internal note, 18pp. Available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/".