

USER'S GUIDE TO ADD NEW SURFACE VARIABLES OR NEW SURFACE ATTRIBUTES IN ARPEGE/IFS, ALADIN, AROME: CYCLE 46T1R1.

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

February 12, 2019

Contents

1	Introduction.	2
2	Add a new surface variable: main features.	2
2.1	No new group creation.	2
2.2	New group creation with specification of individual variables.	3
2.3	New group creation without specification of individual variables.	5
3	Add a new surface variable or group: additional features when read/write this surface variable on ARPEGE files.	6
4	Add a new surface variable: additional features when post-processing it.	6
5	Add a new surface variable: additional features when assimilating it.	7
6	Add a new surface variable: additional features when applying nudging on it.	7
7	Add a new surface attribute.	8
8	Add a new surface field or group in climatological files (configuration 923/E923, METEO-FRANCE purpose only).	8
9	References.	8

1 Introduction.

The new surface structure has been introduced by ECMWF (Mats Hamrud) in CY31R2, and now the surface buffers are split into a “SP” one (prognostic surface variables) and a “SD” one (diagnostic surface variables). The word “surface” can be sometimes misleading because we find not only surface or internal soil quantities, but also upper-air quantities at a specified level (temperature at 10 meters, height of the top of the PBL, total cloudiness). “SP” buffers are split into several groups, “SD” buffers also. The total number of fields reaches a total close to 200, and new fields are regularly added.

Adding new surface fields is not so easy that one could believe (even if the number of routines to be modified is rather small) and the purpose of the present paper is to provide a user’s guide for people wanting to introduce new surface variables or new surface attributes. This is not a comprehensive documentation of all the surface features. Some other internal notes describe the new surface data flow.

Three types of “SD” surface variables can be found (independently of the different groups):

- The constants (for example the land-sea mask), which do not evolve and which are always input data in the physics.
- The diagnostics (for example the temperature at 10 meters), which are always output data in the physics: they are not conserved from one timestep to the following one.
- The pseudo-prognostic fields (like the Charnock constant), which are input of the physics, modified by the physics and the new value is conserved for the input of physics at the following timestep.

Additionally to that, some of them are read on a file; some of them can be post-processed; some of them can be assimilated.

Note that there are some other groups of “surface” fields which have not been put in the “SP” and “SD” buffers: for example some AROME surface fields which are still stored in GPARBUF. Such quantities are out of the scope of this paper.

We first start to list the common features to be added when introducing surface fields (read or not on a file, post-processed or not, assimilated or not). Some details will be then given to describe:

- how to read/write new surface fields on files.
- how to post-process new surface fields.
- how to assimilate new surface fields.
- how to apply nudging on new surface fields.
- how to add new attributes.
- how to add a new group in SP or SD.

2 Add a new surface variable: main features.

2.1 No new group creation.

For the time being, the new variable is assumed to enter an existing group. To simplify, we assume that the new variable will have a generic code name NEWVAR.

* **List of routines to update:** For the time being we assume that we do not add any new attribute. There is a minimal list of routines which should be updated:

- module/surface_fields_mix.F90
- module/field_definitions.F90
- module/field_gfl_wrapper.F90
- setup/su_surf_fds.F90
- in some cases, diab/cpg.F90
- in some cases, diab/cpwts.F90
- in some cases, module/traj_surface_mod.F90
- in some cases, module/gom_mod.F90
- in some cases, phys.dmn/mf_phys.F90

* **The first questions to answer:**

- The first question to answer is to know if this is a prognostic variable (which must enter SP), or a diagnostic one (which must enter SD). If it must enter SD, is it a constant, a diagnostic or a pseudo-prognostic field?
- The second question is to determine in what group it must enter. For example, if this is a climatological constant or a geometric constant, it must enter SD, in the group VARSF.
- Is it a 3D field (several layers) or a 2D field (one layer)?

* **module/surface_fields_mix.F90:** Now we assume that we have answered the previous questions. In SURFACE_FIELDS_MIX, we go in the topics “Group specific type definitions” and:

- We add the attribute Y[NEWVAR] in the adequate type definition TYPE_SFL_[group]. The attribute is itself a derivated-type variable (of type TYPE_SURF_MTL_2D or TYPE_SURF_MTL_3D according to the fact that this is a 2D variable or a 3D variable). A comment should always be provided at this place to know what is the variable NEWVAR.

* **module/field_definitions.F90 and module/field_gfl_wrapper.F90:** Reference to the new surface variable must be added in these two modules too.

* **setup/su_surf_fds.F90:** Go in the right group. For each group, the code is divided into 3 parts:

- A part which does some allocations and memory equivalences, and calls INI_SFLP3 or INI_SFLP2.
- A part which set-ups attributes for useful (allocated) fields (separate LECMWF and .NOT.LECMWF codes). This part calls SETUP_SFLP3 or SETUP_SFLP2. This is the only part where some complicated tests are allowed.
- A part calling FINALISE_SFLP3 or FINALISE_SFLP2. In particular these routines set-up attributes YSD_[group]-Y[variable]_LSET or YSP_[group]-Y[variable]_LSET saying if surface fields are active or not.

Individual fields must appear in the same order in these 3 parts.

When a new field is introduced:

- In part 1, a line must be added:

```
YS[P or D]_[group]%Y[NEWVAR] => YS[P or D]_[group]%Y[group] (JPMAXSFLDS)
```

- In part 2, two statements must be added (example given with SETUP_SFLP2, but that must be SETUP_SFLP3 for some groups):

```
IF (test) THEN
  YS[P or D]_[group]%Y[NEWVAR] => YS[P or D]_[group]%Y[group] (YS[P or D]_[group]D%IPTR)
  CALL SETUP_SFLP2(YS[P or D]_[group]D,YS[P or D]_[group]%Y[NEWVAR],optional arguments)
ENDIF
```

The optional arguments to be given vary according to the field; for fields read on files the GRIB name or the ARPEGE file identifier appears; if KREQIN=1 that means that the field is read on a file.

* **phys_dmn/mf_phys.F90:** Some specific cautions should be taken for pseudo-historic “SD” fields, for purely diagnostic calls of MF_PHYS (this is the case for LDCONF, sort of timestep zero in the XFU diagnostics). For such calls of MF_PHYS the initial content of surface variables should be saved in a local array, and restored at the end of MF_PHYS (I do not know if this question is relevant also for EC_PHYS).

2.2 New group creation with specification of individual variables.

The new variable is now assumed to enter a new group. To simplify, we assume that the new variable will have a generic code name NEWVAR, that the new group will have a generic code name NG (long version of the name: NGRUP), and that there will be one dimension INGRUP to increment.

* **List of routines to update:** For the time being we assume that we do not add any new attribute. There is a minimal list of routines which should be updated:

- fullpos/fpchresol.F90
- module/surface_fields_mix.F90
- module/surface_fields_oper_traj.F90
- module/field_definitions.F90

- module/field_gfl_wrapper.F90
- setup/su_surf_fds.F90
- setup/falsify_surf_fields.F90
- in some cases, phys.dmn/mf_phys.F90

* **The first questions to answer:**

- The first question to answer is to know if this is a prognostic variable (which must enter SP), or a diagnostic one (which must enter SD). If it must enter SD, is it a constant, a diagnostic or a pseudo-prognostic field?
- Is it a 3D field (several layers) or a 2D field (one layer)?

* **module/surface_fields_mix.F90:** Now we assume that we have answered the previous questions.

- The new type TYPE_SFL_[NGRUP] must be defined (paragraph “Group specific type definitions”), for example (the new variable is assumed here to be a 2D one entering ‘SD’):

```
! * Group NG=NGRUP: ??? (provide comments)
TYPE TYPE_SFL_[NGRUP]
TYPE(TYPE_SURF_MTL_2D), POINTER :: Y[NEWVAR] ! ??? (provide comments)
TYPE(TYPE_SURF_MTL_2D), POINTER :: Y[NG](:)
END TYPE TYPE_SFL_[NGRUP]
```

- In the part “Data structures” (“Prognostic (multi time level) fields” if “SP”, or “one time level fields” if “SD”), add:

```
! NGRUP
REAL(KIND=JPRB), ALLOCATABLE :: SD_[NG] (:, :, :)
TYPE(TYPE_SURF_GEN) :: YSD_[NG]D
TYPE(TYPE_SFL_[NGRUP]) :: YSD_[NG]
```

- Definition of type TYPE_SURF must be updated.
- If the new group enters SP, routines GPPOPER and GPOPER should be updated; if the new group enters SD, routine GPOPER should be updated. GPOPER_2 must be called for 2D fields, GPOPER_3 for 3D fields.
- The new group must be added in most of encapsulated routines, the name of which ends by _SURF.

* **module/field_definitions.F90 and module/field_gfl_wrapper.F90:** Reference to the new group and to the individual variables entering this group must be added in these two modules too.

* **module/surface_field_oper_traj.F90, fullpos/fpchresol.F90 and setup/falsify_surf_fields.F90:** Reference to the new group must be added in these routines too.

* **setup/su_surf_fds.F90:** In SU_SURF_FLDS, we add the corresponding code for a new group in section 1 if this is a prognostic variable, in section 2 if this is a diagnostic variable. That provides a piece of code which is divided into 3 parts and looks like (the new variable is assumed here to be a 2D one entering ‘SD’):

```
! * Group NG=NGRUP

! - call to INI_SFLP2, allocation:
IF (ASSOCIATED(YSD_[NG]%Y[NG])) DEALLOCATE(YSD_[NG]%Y[NG])
ALLOCATE(YSD_[NG]%Y[NG] (JPMAXSFLDS))
CALL INI_SFLP2(YSD_[NG]D, YSD_[NG]%Y[NG], .FALSE., 'NGRUP - SD_[NG] ')
YSD_[NG]%Y[NEWVAR] => YSD_[NG]%Y[NG] (JPMAXSFLDS)

! - set-up attributes for active fields:
IF (LECMWF) THEN
  IF (test) THEN
    YSD_[NG]%Y[NEWVAR] => YSD_[NG]%Y[NG] (YSD_[NG]D/IPTR)
    CALL SETUP_SFLP2(YSD_[NG]D, YSD_[NG]%Y[NEWVAR], ... optional arguments ...)
  ENDIF
ELSE
  IF (test) THEN
```

```

    YSD_[NG]%Y[NEWVAR] => YSD_[NG]%Y[NG] (YSD_[NG]D%IPTR)
    CALL SETUP_SFLP2(YSD_[NG]D,YSD_[NG]%Y[NEWVAR],... optional arguments ...)
ENDIF
ENDIF

```

! - set-up information that couldn't be set until the total number of [NGRUP] fields was known.
CALL FINALISE_SFLP2(YDSURF,YSD_[NG]D,YSD_[NG]%Y[NG],... optional arguments ...)

Remarks:

- Call to INLSFLP2: argument after INGRUP is .FALSE. for diagnostic fields, .TRUE. for prognostic fields.
- About the optional arguments of SETUP_SFLP2, one can specify for example KGRIB (GRIB code), CDNAME (field name for the ARPEGE files), KTRAJ, KREQIN (KREQIN=1 means that the field must be read on a file).
- For a 3D variable, SETUP_SFLP3 must be called instead of SETUP_SFLP2.
- Sometimes no field is allocated for LECMWF (resp. for .NOT.LECMWF): in this case, in paragraph “set-up attributes for active fields”, we may find empty code under LECMWF (resp. under .NOT.LECMWF).

* **phys_dmn/mf_phys.F90:** If the new variable is a pseudo-historic “SD” field, the same thing as for a new variable in a already existing class must be done.

* **other routines:** Other routines may have to be modified, in particular those requiring dummy arguments YDSURF, PSP_[NG] or PSD_[NG].

2.3 New group creation without specification of individual variables.

We now consider creation of a new group, which may contain several variables and where it is not necessary to give an individual name to each variable. This is the case for example for group SFO=SFORC.

We assume that the new group will have a generic code name NG (long version of the name: NGRUP), and that there will be one dimension INGRUP to increment.

* **List of routines to update:** For the time being we assume that we do not add any new attribute. There is a minimal list of routines which should be updated:

- module/surface_fields_mix.F90
- setup/su_surf_fds.F90
- fullpos/fpchresol.F90
- module/surface_fields_oper_traj.F90
- setup/falsify_surf_fields.F90
- in some cases, phys_dmn/mf_phys.F90

* **The first questions to answer:**

- The first question to answer is to know if this group contains prognostic variables (which must enter SP), or diagnostic variables (which must enter SD). If it must enter SD, does this group contain constants, diagnostic fields or pseudo-prognostic fields?
- Does this group contain 3D fields (several layers) or 2D fields (one layer)?

* **module/surface_fields_mix.F90:** Now we assume that we have answered the previous questions.

- The new type TYPE_SFL_[NGRUP] must be defined (paragraph “Group specific type definitions”), for example (the new variable is assumed here to be a 2D one entering ‘SD’):

```

! * Group NG=NGRUP: ??? (provide comments)
TYPE TYPE_SFL_[NGRUP]
TYPE(TYPE_SURF_MTL_2D),POINTER :: Y[NG](:)
END TYPE TYPE_SFL_[NGRUP]

```

- In the part “Data structures” (“Prognostic (multi time level) fields” if “SP”, or “one time level fields” if “SD”), add:

```

! NGRUP
REAL(KIND=JPRB),ALLOCATABLE :: SD_[NG] (:,:,)
TYPE(TYPE_SURF_GEN)      :: YSD_[NG]D
TYPE(TYPE_SFL_[NGRUP])  :: YSD_[NG]

```

- If the new group enters SP, routines GPPOPER and GPOPER should be updated; if the new group enters SD, routine GPOPER should be updated. GPOPER_2 must be called for 2D fields, GPOPER_3 for 3D fields.
- The new group must be added in most of encapsulated routines, the name of which ends by _SURF.

* **setup/su_surf flds.F90:** In SU_SURF_FLDS, we add the corresponding code for a new group in section 1 if this is a prognostic variable, in section 2 if this is a diagnostic variable. That provides a piece of code which is divided into 3 parts and looks like (the new variable is assumed here to be a 2D one entering 'SD'):

```

! * Group NG=NGRUP

! - call to INI_SFLP2, allocation:
IF(ASSOCIATED(YSD_[NG]%Y[NG])) DEALLOCATE(YSD_[NG]%Y[NG])
ALLOCATE(YSD_[NG]%Y[NG] (JPMAXSFLDS))
CALL INI_SFLP2(YSD_[NG]D, YSD_[NG]%Y[NG], INGRUP, .FALSE., 'NGRUP - SD_[NG]  ')

! - set-up attributes for active fields:
IF(LECMWF) THEN
  DO JFLD=1, YSD_[NG]D%NUMFLDS
    CALL SETUP_SFLP2(YSD_[NG]D, YSD_[NG]%Y[NG] (JFLD), ... optional arguments ...)
  ENDDO
ELSE
  DO JFLD=1, YSD_[NG]D%NUMFLDS
    CALL SETUP_SFLP2(YSD_[NG]D, YSD_[NG]%Y[NG] (JFLD), ... optional arguments ...)
  ENDDO
ENDIF

! - set-up information that couldn't be set until the total number of [NGRUP] fields was known.
CALL FINALISE_SFLP2(YDSURF, YSD_[NG]D, YSD_[NG]%Y[NG], ... optional arguments ...)

```

Remarks: cf. remarks given in part 2.3.

* **phys_dmn/mf_phys.F90:** If the new group contains pseudo-historic "SD" field, the same thing as for a new variable in a already existing class must be done.

* **module/surface_field_oper_traj.F90, fullpos/fpchresol.F90 and setup/falsify_surf_fields.F90:** Reference to the new group must be added in these routines too.

* **other routines:** Other routines may have to be modified, in particular those requiring dummy arguments PSP_[NG] or PSD_[NG].

3 Add a new surface variable or group: additional features when read/write this surface variable on ARPEGE files.

This topic will not be described in detail.

It is recommended to check the occurrences of use of YDSURF in the sugrid.. or iogrid.. named routines and their callees.

4 Add a new surface variable: additional features when post-processing it.

Modifications will touch at least the following pieces of code.

* **module/yomafn.F90 and namelist/namafn.nam.h:** Variable GFP_[NEWVAR] must be added; it is desirable that the name into [NEWVAR] will be identical to the one used in the surface variables.

* **module/parfpos.F90:** If necessary, value of JPOSSGP must be increased.

* **other routines:** Adding a new variable GFP_[NEWVAR] may generate modifications in routines using GFP..., in particular:

- setup/suafn1.F90, setup/suafn2.F90 and setup/suafn3.F90.
- setup/sucfu.F90 and setup/suxfu.F90.
- some arpifs/fullpos and aladin/fullpos routines: check those using GFP..., requiring YDSURF, PSP_[NG] or PSD_[NG] as dummy arguments, or using module **SURFACE_FIELDS_MIX**.
- more generally all routines involved in FULL-POS requiring YDSURF, PSP_[NG] or PSD_[NG] as dummy arguments, or using module **SURFACE_FIELDS_MIX**.

For ordering of the GFP..., if possible, keep the same order in YOMAFN, NAMAFN, SUAFN1, SUAFN2, SUAFN3.

More generally, look at all routines using YOMAFN to see if some physical surface post-processable variables are used, and if the new post-processable variable requires code updating.

5 Add a new surface variable: additional features when assimilating it.

There are several parts of the code to be checked (updates may be required), at least:

- the GOMS dataflow.
- the observation interpolator.
- canari/caprsurf.F90.

* **List of routines to update in the GOMS dataflow:** There is a minimal list of routines which should be updated:

- module/gom_mod.F90
- obs_preproc/sugoms.F90
- a subset of the routines using the following variables of module/gom_mod.F90: GID%[NG]_[newvar] (pointer for array GOM) must be checked, and in particular those where individual surface variables are mentioned (mostly in directories op_obs and pp_obs).
- op_obs/cobs.F90

Only a small subset of surface fields are present in the GOMS dataflow.

For example:

- in module/gom_mod.F90 update definition of types TYPE_GOM_ID, and TYPE_GVAR, update calculation of variable GID.
- in obs_preproc/sugoms.F90, set-up GOM%LINTERP(:,GID%[NG]_[newvar]) and GOM%INTERP_TYPE(:,GID%[NG]_[newvar]).
- in op_obs/cobs.F90, call GOM_MODEL_SET for new surface variable, using pointer GID%[NG]_[newvar].

* **Updating TL and AD codes:** TL and AD codes must be updated, for example op_obs/cobsad.F90.

6 Add a new surface variable: additional features when applying nudging on it.

Nudging is used for climatic simulations.

This topic is not described in detail for the time being (it has to be written by people having a good knowledge into nudging). Some of the variables XVU.. of YOMNUD contain reference values for the nudging of surface prognostic variables (surface moisture for example). When adding a new variable which can be nudged the question of updating the following routines must be answered:

- climate/updsst.F90 (and its call in adiab/cpg_gp.F90).
- dia/cpnudg.F90 (and its call in phys_dmn/mf_phys.F90).

A new XVU.. must be added in YOMNUD and all routines using the XVU.. must be updated: expect modifications in the following routines:

- dia/cpnudg.F90
- climate/updnud.F90
- climate/updsst.F90
- module/yomnud.F90

7 Add a new surface attribute.

We assume that we want to add a new attribute to the following data structures: TYPE_SURF_MTL_2D (resp. TYPE_SURF_MTL_3D): the generic name of this attribute is NEWATTR.

* **List of routines to update:** There is a minimal list of routines which should be updated:

- module/surface_fields_mix.F90
- setup/su_surf_fds.F90

* **module/surface_fields_mix.F90 and setup/su_surf_fds.F90:**

- New attribute NEWATTR must be added in the definition of TYPE_SURF_MTL_2D (resp. TYPE_SURF_MTL_3D).
- Update INLSFLP2 and SETUP_SFLP2 for 2D variables (resp. INLSFLP3 and SETUP_SFLP3 for 3D variables).
- In some cases this new attribute may become a new optional dummy argument of SETUP_SFLP2 (2D) or SETUP_SFLP3 (3D). Update the calls to SETUP_SFLP2 and SETUP_SFLP3 when required in SU_SURF_FLDS.

8 Add a new surface field or group in climatological files (configuration 923/E923, METEO-FRANCE purpose only).

To simplify, we assume that the new variable will have a generic code name NEWVAR. If a new group must be created for that, the new group will have a generic code name NG (long version of the name: NGRUP).

* **List of routines to update if the new variable enters an existing group:** Once known the group where the new variable must enter, the proper INCLI.. routine (INCLI2 to INCLI10) must be updated. Reading of the new surface field on a database, interpolating it, and writing it on an ARPEGE file, must be added in the proper INCLI..

* **List of routines to update if the new variable enters a new group:** It must be questioned if a new INCLI.. routine must be created or if the code relative to the new variable can enter an existing INCLI.. . In the first case, a new INCLI.. routine must be written, with the same structure as in INCLI2 to INCLI10 (reading of the new surface field on a database, interpolating it, and writing it on an ARPEGE file). In the last case, see the previous paragraph.

* **Additional remarks:** New INCLI.. routines must be called by INCLI0. For each new variable it must be questioned if there is only an annual climatological value or if there are different monthly climatological values.

9 References.

- (IDGOMU) Geer, A., 2013:
GOM user guide: interpolation to observation space. Internal note, 3pp.