

**\*\*\* NOTE \*\*\***

This document provides general procedures and an example of how to add a new variable to the UPP code. Please keep in mind it may not be an exhaustive step-by-step depending on your particular situation. While we can provide general assistance for adding a new variable, users should be aware that this requires good knowledge of Fortran and thorough understanding of the code.

We encourage users to contact us at [upp-help@ucar.edu](mailto:upp-help@ucar.edu) to make us aware of modifications you are making. In some cases, if we determine the changes you are making may be relevant for operational and/or community purposes, we will be interested in incorporating your changes into the code base for support and future release. We would then work with you to make this possible.

### General Instructions

- 1) Allocate the field: ALLOCATE.f  
*This file is the instantiation or allocation of the variable. Note that the variables are defined based on the parallel processing capability of UPP - use an example from the file.*
- 2) Deallocate the field: DEALLOCATE.f  
*All good programmers give back their resources when they are done. Please update this routine to return your resource to the system.*
- 3) Declare the new variable: VRBLS2D\_mod.f, VRBLS3D\_mod.f or VRBLS4D\_mod.f  
*The variable is declared in one of these module defining files depending on its dimension.*
- 4) Define field for grib1: RQSTFLD.f  
*This file contains a list of all possible fields to be output by UPP, corresponding key-word character string user places in wrf\_cntrl.parm file, UPP ID for internal code, grib IDs.*
- 5) Read model output: INITPOST.F  
*This file is used for reading the model output files.*
- 6) Determine routine for filling variable: e.g SURFCE.f, MDLFLD.f, MDL2P.f, etc.  
*This is the place that you will fill the array with the data and call gribit to output the field.*
- 7) Define table/grib2 parameters/ for grib2 output: params\_grib2\_tbl\_new  
*This table contains the necessary parameter information for grib2 fields.*

- 8) Define the field for grib2 output: post\_avlbfls.xml  
*This file is used for defining all available grib2 fields.*
- 9) Define control file entry for output: postcntrl.xml & postxconfig-NT.txt  
*These files are used for controlling which fields are output by UPP for grib2.*
- 10) Define output control file: wrf\_cntrl.parm  
*This file is used for controlling which fields are output by UPP for grib1.*

### **Example Procedure: Steps for adding the new variable 'ACLHF'**

- This example illustrates a new variable from the WRF output that will be read into UPP and directly output into the grib output files (i.e. no additional computations/calculations are needed for the field).
- Note that while grib1 procedures are provided, we have moved to grib2 almost exclusively. As such support for grib1 additions is limited.
- Additions to each of the routines are highlighted in **green**.
- Locations of **routines** are in /UPPV3.2/src/unipost unless specified otherwise.
- A sample wrfout file for the following procedures is available for download from:
  - <https://dtcenter.org/upp/users/downloads/data/AddNewVariableData.tar.gz>
  - This data is the 6-hr forecast of a WRF initialization of 2009-12-17\_12:00:00

New variable to add: **ACLHF**

```
float ACLHF(Time, south_north, west_east) ;
  ACLHF:FieldType = 104 ;
  ACLHF:MemoryOrder = "XY " ;
  ACLHF:description = "ACCUMULATED UPWARD LATENT HEAT FLUX AT THE SURFACE" ;
  ACLHF:units = "J m-2" ;
  ACLHF:stagger = "" ;
  ACLHF:coordinates = "XLONG XLAT" ;
```

#### **1. Allocate the new variable in *ALLOCATE\_ALL.f***

This file is the instantiation or allocation of the variable. Note that the variables are defined based on the parallel processing capability of UPP - use an example from the file.

#### ***User procedure:***

- Add in VRBLS2D section as:  
**allocate(aclhf(im,jsta\_2l:jend\_2u))**

#### **2. De-allocate the variable to give the resources back in *DEALLOCATE.f***

All good programmers give back their resources when they are done. Please update this routine to return your resource to the system.

**User procedure:**

- Add as:  
**deallocate(ac1hf)**

3. Declare the new variable in the appropriate file depending on its dimensions;  
**VRBLS2D\_mod.f, VRBLS3D\_mod.f** or **VRBLS4D\_mod.f**

**User procedure:**

- **ACLHF** is a 2-dimensional field, so declare it in **VRBLS2D\_mod.f**
- Add to the end of the first section of allocations as:  
**ACLHF(:,:)**

4. List the new variable in **RQSTFLD.f** which includes a list of all possible fields to be output by UPP, as well as the corresponding key-word character string the user places in wrf\_cntrl.parm file, the UPP ID for internal code, and grib IDs. Be sure to pick a unique identifier that is not already used for the new variable. The unique identifier or index are typically assigned in groups - hopefully a community area will be added in the future - or a defined method to avoid overwriting others values. Right now using 900's for community contributions.

**Example entry:**

! HWRF addition for v\_flux as pass through variable:

```
DATA IFILV(901),AVBL(901),IQ(901),IS(901),AVBLGRB2(901) &  
& /1,'MODEL SFC V WIND STR',125,001, &  
& 'V_FLX ON surface'/
```

Where:

- **IFILV** Identifies field as MASS/VELOCITY point
- **AVBL** is the model output character string variable name (grib1)
- **AVBLGRB2** is the model output character string variable name (grib2)
- **IQ** is the GRIB PDS OCTET 9 (table 2) - Indicator of parameter and units
- **IS** is the GRIB PDS OCTET 10 (table 3&3a) - Indicator of type of level or layer
- A **UNIQUE** array location UPP uses to store this variable in parallel arrays (e.g. **901**)

**User procedure:**

- A latent heat flux variable (LHTFL) was found in the GRIB1 parameter tables, so add a new unused parameter number (237) using Table 130 to define the new field.  
<http://www.nco.ncep.noaa.gov/pmb/docs/on388/table2.html>
- Used level type surface, which is 001  
<http://www.nco.ncep.noaa.gov/pmb/docs/on388/table3.html>
- Add as:  
**DATA IFILV(950),AVBL(950),IQ(950),IS(950),AVBLGRB2(950) &**  
**& /1,'ACC SFC LAT HEAT FX ',237,001, &**  
**& 'ACC LHTFL ON surface ' !Table 130**

5. Read the model output field from the wrfout file by adding the new variable into **INITPOST.F**. This file is used for reading the WRF-ARW model output files in netcdf format.

**User procedure:**

- Add using the 2D variable **SNDEPAC** (snowfall accumulation), which is also a 2D surface based accumulation field, as a template by following it through the routine.
- Add to top section of the routine in 'use vrbls2d' to initiate the new variable as:

**aclhf**

- Read in the new variable as:

**VarName='ACLHF'**

```
call getVariable(fileName,DateStr,DataHandle,VarName,DUMMY,      &  
IM,1,JM,1,IM,JS,JE,1)
```

```
do j = jsta_2l, jend_2u
```

```
do i = 1, im
```

```
ACLHF ( i, j ) = dummy ( i, j )
```

```
end do
```

```
end do
```

```
! print*, 'ACLHF at ',ii,jj, ' = ',ACLHF(ii,jj)
```

6. Determine the correct routine to add the new variable to (e.g. **SURFCE.f**, **MDLFLD.f**, **MDL2P.f**, etc). You will need to determine the correct routine to add your field into; this is the place that you will fill the array with the data and call grubit to output the field. The correct routine will depend on what your field is. For example, if you have a new diagnostic called foo, and you want it interpolated to pressure levels, you would need to add it to MDL2P.f. If foo was only a surface variable, you would add it to SURFCE.f. If you wanted foo on native model levels, you would add it to MDLFLD.f. If you're not sure which routine to add the new variable to, choose a similar variable as a template.

*Note:* This is also where you would add any calculations needed for your new variable, should it be required.

**User procedure:**

- Treat **ACLHF** like a surface field (**SURFCE.f**)
- Using the variable **SNDEPAC** (accumulated depth of snowfall) as a template which is also an accumulated field that is just being read through and output, similar to what we want.
- Add in top section in 'use vrbls2d, only' to initiate the new variable as:

**aclhf**

- Add in main section using the template variable as a guide.
  - Note that ID(02), which is the ID for table version number, is added and set to 130. This is the table that we are adding the new variable to.

- The block of code in **blue** is for metadata for the accumulation field being added in this example and the user does not need to edit it and it is not always needed. For example, for an instantaneous field, you would not need that block.

```

! ACCUM UPWARD LATENT HEAT FLUX AT SURFACE
IF (IGET(950).GT.0) THEN
  ID(1:25) = 0
  ID(02) = 130
  ITPREC   = NINT(TPREC)
!mp
  IF(ITPREC .NE. 0) THEN
    IFINCR = MOD(IFHR,ITPREC)
    IF(IFMIN .GE. 1)IFINCR = MOD(IFHR*60+IFMIN,ITPREC*60)
  ELSE
    IFINCR = 0
  ENDIF
!mp
  ID(18)   = 0
  ID(19)   = IFHR
  IF(IFMIN .GE. 1)ID(19)=IFHR*60+IFMIN
  ID(20)   = 4
  IF (IFINCR.EQ.0) THEN
    ID(18) = IFHR-ITPREC
  ELSE
    ID(18) = IFHR-IFINCR
    IF(IFMIN .GE. 1)ID(18)=IFHR*60+IFMIN-IFINCR
  ENDIF
  IF (ID(18).LT.0) ID(18) = 0
  if(grib=='grib1') then
    DO J=JSTA,JEND
      DO I=1,IM
        GRID1(I,J) = ACLHF(I,J)
      ENDDO
    ENDDO
    CALL GRIBIT(IGET(950),LVLS(1,IGET(950)), GRID1,IM,JM)
  elseif(grib=='grib2') then
    cfld=cfld+1
    fld_info(cfld)%ifld=IAVBLFLD(IGET(950))
    fld_info(cfld)%ntrange=1
    fld_info(cfld)%tinvstat=IFHR-ID(18)
!$omp parallel do private(i,j,jj)
  do j=1,jend-jsta+1
    jj = jsta+j-1
    do i=1,im
      datapd(i,j,cfld) = ACLHF(i,jj)

```

```
    enddo
  enddo
endif
ENDIF
```

7. For grib2 output, add the new variable to /UPPV3.2/src/lib/g2tmpl/*params\_grib2\_tbl\_new*. For all current UPP output fields, this table lists, in order, the:

- Discipline ([http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2\\_table0-0.shtml](http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2_table0-0.shtml))
- Category ([http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2\\_table4-1.shtml](http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2_table4-1.shtml))
- Parameter Number ([http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2\\_table4-2.shtml](http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2_table4-2.shtml))
- Table information (0 for parameters from the WMO table; 1 for parameters from the local NCEP table)
- Abbreviated Variable Name (from the parameters table)

**User procedure:**

- Since there is already a latent heat flux (LHTFL) parameter in this table, create a new Latent Heat Flux parameter so as to not overwrite the current one, just in case you want both to be output
- Latent heat flux is a meteorological field (discipline=0)
- Latent heat flux is a temperature product (category=0)
- Pick an unused parameter number from the table defined by discipline=0 and category=0 (Table 4.2-0-0: [http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2\\_table4-2-0-0.shtml](http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2_table4-2-0-0.shtml)). In this case, the unused parameter number 205 was chosen.
- Add using the NCEP local table (table=1)
- Choose an abbreviated parameter name to describe your field (e.g. ACLHF)
- Add as:

```
0 0 205 1 ACLHF
```

8. Add the new variable to the /UPPV3.2/parm/*post\_avblflds.xml*, which lists all fields available for output in GRIB2 format.

- Post\_avblfldidx: the unique array number given in the RQSTFLD.f routine.
- Shortname: name describing the variable and level type
- Pname: the abbreviation for your variable
- Table info: table used if not standard WMO
- Fixed\_sfc1\_type: level type
- Scale: precision of data written out to grib2 file

**User procedure:**

- Add as:

```
<param>
  <post_avblfldidx>950</post_avblfldidx>
  <shortname>ACC_LATENT_HEAT_FLUX_ON_SURFACE</shortname>
  <pname>ACLHF</pname>
```

```

    <table_info>NCEP</table_info>
    <fixed_sfc1_type>surface</fixed_sfc1_type>
    <scale>4.0</scale>
</param>

```

9. Add the new variable to the /UPPV3.2/parm/*postcntrl.xml* file, which lists all fields and levels you wish to output for GRIB2. Remake the /UPPV3.2/parm/*postxconfig-NT.txt* file, which contains the information from the xml that UPP reads.

- See the User's guide on steps for creating the text control file

**User procedure:**

- Add as:
 

```

<param>
  <shortname>ACC_LATENT_HEAT_FLUX_ON_SURFACE</shortname>
  <pname>ACLHF</pname>
  <scale>4.0</scale>
</param>

```

10. Add the new variable to the /UPPV3.2/parm/*wrf\_cntrl.parm* file, which lists all fields and levels you wish to output for GRIB1.

**User procedure:**

- Add as:
 

```

(ACC SFC LAT HEAT FX ) SCAL=( 4.0)
L=(10000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000)

```

 \*\* Note the first entry is turned from 0 to 1. This turns on the first level.

11. Run clean on the code and recompile the code to include the changes before running your UPP run script.

**User procedure:**

```

>> ./clean -a
>> ./configure
>> ./compile >& compile.log &

```

12. Assuming the modified code compiled successfully and you were able to produce grib output, you can check the grib file for your new variable.

**GRIB2** output of the new variable from this example procedure (using the wgrib2 utility if available on your system).

- The new variable will not be defined by the variable name. Instead it will be defined using the grib2 parameter information you entered into *params\_grib2\_tbl\_new* from step 7 of this procedure.

456:43204412:vt=2009121718:surface:6 hour fcst:var **discipline=0** center=7 **local\_table=1**  
**parmcats=0 parm=205:**

ndata=121002:undef=0:mean=1.97108e+06:min=-1.12e+06:max=2.406e+07  
grid\_template=30:winds(grid):  
Lambert Conformal: (402 x 301) input WE:SN output WE:SN res 8  
Lat1 14.807213 Lon1 231.818604 LoV 258.040009  
LatD 38.270000 Latin1 38.270000 Latin2 38.270000  
LatSP 0.000000 LonSP 0.000000  
North Pole (402 x 301) Dx 15000.000000 m Dy 15000.000000 m mode 8

**GRIB1** output of the new variable from this example procedure (using the wgrib utility if available on your system).

- The new variable will not be defined by the variable name. Instead it is defined by **kpds5=237** (grib1 ID) and **kpds6=1** (level type) you gave it in **RQSTFLD.f** in step 4.
- For this particular variable, the accumulation period is shown, due to the addition of the metadata block in the SURFCE.f routine.

rec 319:59903982:date 2009121712 **var237 kpds5=237 kpds6=1** kpds7=0 levels=(0,0)  
grid=255 sfc **0-24hr acc:**

**var237=undefined**

timerange 0 P1 6 P2 0 TimeU 1 nx 402 ny 301 GDS grid 3 num\_in\_ave 0 missing 0  
center 7 subcenter 0 process 125 **Table 130** scan: WE:SN winds(grid)  
Lambert Conf: Lat1 14.807000 Lon1 231.819000 Lov 258.040000  
Latin1 38.270000 Latin2 38.270000 LatSP 0.000000 LonSP 0.000000  
North Pole (402 x 301) Dx 15.000000 Dy 15.000000 scan 64 mode 136  
min/max data -1.1217e+06 2.40583e+07 num bits 12 BDS\_Ref -112.17 DecScale -4  
BinScale 0