General information

Get the examples as postcript file:examples.ps

Making maps with the old MetgraF usually required the editing of a rather long namelist file, often given the name 'dat.dat'. Editing of dat.dat was done either interactively using pull-down menus, or directly in an editor. Presently, in MetgraF_2, maps are produced by short batch scripts that accesses libraries of map- and field- layout characteristics. The resulting maps are in postscript format and can be viewed on the workstation or sent to the printer. After conversion to the gif-format, the maps can easily be distributed to other media and users such as the intranet or the internet. An interactive version of MetgraF_2 will be available later.

There main difference when making maps with MetgraF_2 is in the structure of the scripts and the contents of the namelists. All information that is unique to a certain map, i.e. the 'experiment' name, the location of the files, the date and time and the forecast length, are set in one new namelist, called 'namfil'. The map-layout is, as before, set in 'namgeo', while the field selection and drawing instructions are given in 'namsel', also as it used to. A new namelist, called 'namlog', is used to define the destination of error- and diagnostic messages, necessary when MetgraF_2 is used in the RiPP environment.

The namelist variables in 'namsel' are the same as in the older MetgraF, and we refer to existing MetgraF documentation for a detailed description. There is however one important change in that a separate 'namsel' is used for each field. Hence all variables that used to be arrays, are now single valued, such as ipar1 and ilev1. For multi field maps one just adds another 'namsel' after the previous, up to six of them are allowed by the 'parameter'-statement in the code.

A namelist called 'namval' is used to define symbols and texts that are to be written on the map, an example is 'dots' and names for cities and towns.

Observations can, at present, only be plotted from observation files in the internal SMHI format. A facility to plot them from BUFR-files will be added as soon as time allows us to implement it. The obs-plotting requires many options and selections, and there are many, many parameters in the obs namelist 'namobs'. Most of them are however preset by default. Details of how to change the defaults can be obtained from Bo Lindgren at SMHI.

A novel feature is that the final namelist input to a plot job is assembled from several small files which can be kept on-line in MetgraF_2 directories. This allows for easily repeated plots with frozen characteristics from different experiments and times, but it also allows for easy addition of new layouts and contents. The emphasis has been on flexibility and ease-of-use at the same time.

The order of the namelist components is basically free. Each component is processed as it appears in the final input to MetgraF_2. This allows for great flexibility in the design of maps and clusters of maps.

Map example 1

A sample MetgraF_2 map job for maps from SMHI'S operational HIRLAM-44, with comments in *Italic*, follows. Get the example: map1

#!/usr/bin/ksh
#
ulimit -c 0

```
#set -kx
#
RED_FILE_PATH=/proj/iflocal/src/metgr/coast/ ; export RED_FILE_PATH
COAST_FILE_PATH=/proj/hirfou/ulf/MetgraF_2/coast/ ; export COAST_FILE_PATH
```

This sets the path to the coast lines. If a coastline is not available for the given projection, MetgraF will create one at COAST_FILE_PATH.

#
rm namelists *.key

This takes away some crap. The .key files are table of contents files for the grib files. Here the date, levels, parameter etc. are listed. It's a good habit to remove them since old .key files may cause troubles.

#
MG2=~iflocal/MetgraF_2

Sets the path to the MetgraF library

```
#
GE0=411S
YEAR1='date +"%Y"'
M01='date +"%m"'
DA1='date +"%d"'
H01=00
INTV=06
```

GEO tells which namgeo to use. The rest sets todays date and time.

```
for LN1 in 12
do
```

This is a Unix script loop over LN1 which is the forecast length.

#
for map_number in 5
do

Tells us that map_number 5 shall be used.

```
#
if [ $map_number = '1' ]
then
    SEL1=Wind_300
    SEL2=GeoPot_300
elif [ $map_number = '2' ]
then
    SEL1=Temp_500
    SEL2=GeoPot_500
elif [ $map_number = '3' ]
```

```
then
   SEL1=EqTheta_850
   SEL2=GeoPot_850
elif [ $map_number = '5' ]
then
   SEL1=10mWind
   SEL2=Pms1
```

Selects namelist for plotting 10m wind and Mean sea level pressure.

```
elif [ $map_number = '6' ]
then
  SEL1=2mTemp
  SEL2=Pmsl
elif [ $map_number = '7' ]
then
  SEL1=Tot.Cloud
  SEL2=Pmsl
elif [ $map_number = '8' ]
then
  INTV=03
  SEL1=isallobars
  SEL2=Pmsl
  TOPTEXT='Marktryck, isallobarer och markobsar'
elif [ $map_number = '9' ]
then
  INTV=06
  SEL1=6hRain
  SEL2=Pmsl
elif [ $map_number = '10' ]
then
  INTV=06
  SEL1=6hRainDig
  SEL2=Pmsl
elif [ $map_number = '11' ]
then
  INTV=06
  SEL1=2mTempDig
  SEL2=Pmsl
fi
#
((LN2 = ${LN1} - {INTV}))
if [ $LN2 -le 9 ]
then
LN2=0$LN2
fi
#
GRIBFIL1=fc${YEAR1}${MO1}${DA1}${HO1}${LN1}
GRIBFIL2=fc${YEAR1}${MO1}${DA1}${HO1}${LN2}
```

Builds the filenames. Here these names refer to model level files. I you want pressure level files add pp at the end of the filenames.

```
#
cat > logfile << *EOLOGFILE
    &namlog
        log_print =.t.,
        log_unit =6,
        log_file ='metg.log',
    &end
*EOLOGFILE</pre>
```

Namelist for log files. In this case extra information are given on output 6 (the screen). If log_unit is set to 7 the output will be written to metg.log

```
#
cat > datefile << *EOINFILE</pre>
&namfil
                 ='Svenska',
      lingua
      lfield
                 =.true.,
      gribdir_1 ='/local_disk/hirlam.2.7.15/hl_arc/ulf',
      gribfile_1 ='$GRIBFIL1',
                 = 19{YEAR1},
      iyr1
                 = {MO1},
      imo1
      ida1
                 = \{DA1\},
                 = ${H01},
      iho1
      iln1
                 =  {LN1},
      gribfile_2 ='$GRIBFIL2',
                 = {LN2},
      iln2
                 ='SMHI HIRLAM 44. Daily Operations.',
      filtext
      top_text(1)='SMHI HIRLAM_44 Nordenkartan',
      top_text(2) = '$TOPTEXT',
      bot_text(1)='${YEAR1}${MO1}${DA1} ${HO1}UTC+${LN1}h',
      bot_text(2)='$BOTTEXT',
      top_hgt
                 =0.3,
      bot_hgt
                 =0.2744,
      top_font
                 =10,
      bot_font
                 =10,
 &end
*EOINFILE
```

This gives the information of where to find the file and which files to use. If you leave out date and time it will be read from the files. Post processed files has the extension pp on their file names.

```
#
#
cat logfile > namelists
cat ${MG2}/Maps/namelists/namgeo/${GE0} >> namelists
cat datefile >> namelists
cat ${MG2}/Maps/namelists/namsel/${SEL1} >> namelists
cat ${MG2}/Maps/namelists/namsel/${SEL2} >> namelists
```

cat \${MG2}/Maps/namelists/namval/SvOrter >> namelists
#

Writes your choices to the file namelists. The path /MG2/Maps/namelists/namval/ tells you where the predined namelists are stored.

```
PIXFILE=${SEL1}_${SEL2}+${LN1}h
```

Builds the plot file name from what you have chosen.

#
#cat namelists
\$MG2/bin/metgraf_b namelists \$PIXFILE.ps L a4
#

Executes MetgraF with the namelist input, the output filename, L for landscape orientation (P portrait) and a4 for the paper size.

done done exit

Map example 2

This is a more straightforward way to do it. Get the example: map2

```
#!/usr/bin/ksh
#
RED_FILE_PATH=/local_disk/MetgraF_2/coast/ ; export RED_FILE_PATH
COAST_FILE_PATH=/local_disk/MetgraF_2/coast/ ; export COAST_FILE_PATH
#
```

This sets the path to the coast lines. If a coastline is not available for the given projection, MetgraF will create one.

```
#
rm namelists *.key
#
```

This takes away some crap. The .key files are table of contents files for the grib files. Here the date, levels, parameter etc. are listed.

```
YEAR='date +"%Y"'
MONTH='date +"%m"'
DAY='date +"%d"'
HOUR=00
LENGTH=024
```

Sets the date and time to todays 00Z+24h forecast.

cat > namelists << *EOINFILE</pre>

This puts the following (until EOINFILE) to a file named namelists

```
&namlog
    log_print =.true.,
    log_unit = 6,
    log_file ='metg.log',
&end
```

Namelist for log files. In this case extra information are given on output 6 (the screen). If log_unit is set to 7 the output will be written to metg.log

```
&namgeo
    ltotal = .t.,
    scale = 35.,
    maprep = 10,
&end
```

This is the map namelist. Itotal means that the whole model area is plotted. maprep gives the projection, in this case rotated polar stereo graphic, which is equal to the Hirlam grid.

```
&namfil
    gribdir_1 ='/local_disk/hirlam.2.7.15/hl_arc/sno',
    gribfile_1 ='fc8701100006pp',
    kand
```

&end

This gives the information of where to find the file and which file. You may also give date, time and forecast length here to be sure you get the right date. Year shall be given with YYYY (1998). Post processed files has the extension pp on their file names.

&namsel

```
= 01,
     ipar1
     ilev1
              = 00,
     ityp1
              = 103,
     conint
              = 5.,
              = 0.01,
     sca
     iconcol = 002,
&end
&namsel
     ipar1
              = 33,
     ilev1
              = 10,
              = 105,
     ityp1
&end
```

Namelists for parameter selection. Here Pressure and wind are given.

```
*EOINFILE
#
/local_disk/MetgraF_2/bin/metgraf_b namelists plot.ps L a4
```

Executes MetgraF with the namelist input, the output filename, L for landscape orientation (P portrait) and a4 for the paper size.

exit

Cross example 1

The most important differences to the earlier version of the MetgraF cross section program are in the GRIB access and in the user interface. The internal interpolations and calculations are unchanged and a facility for differences has been added.

User interface

A batch job, where a namelist and some environment variables are set, is run to produce a postscript file. This may be displayed or printed or converted to gif as required.

Sample script

It is described below with explanations in *italic* Get the example: cross

```
#!/usr/bin/ksh
#
rm *.key namelist
ulimit -c 0
set -kx
#
EXP=FC44
MLEVDIR_1=/local_disk/hirlam.2.7.15/hl_arc/ulf
                                                   ; export MLEVDIR_1
MLEVKEY_1=$PWD
                                                    ; export MLEVKEY_1
PLEVDIR_1=/local_disk/hirlam.2.7.15/hl_arc/ulf
                                                    ; export PLEVDIR_1
PLEVKEY_1=$PWD
                                                    ; export PLEVKEY_1
#
```

This sets the path to the input files and the table of contents files, the .key files.

```
YEAR1='date +"%Y"'
YR1='date +"%y"'
MO1='date +"%m"'
DA1='date +"%d"'
HO1=00
```

Gives todays date. Note that year for cross should be given in YYYY but the Hirlam filenames are given in YY.

for LN1 in 12 do LONO=30 LANO=60 LOSO=00 LASO=50

Gives the endpoints of the cross section.

```
#
MLEVFIL1=fc${YR1}${MO1}${DA1}${HO1}${LN1}
PLEVFIL1=fc${YR1}${MO1}${DA1}${HO1}${LN1}pp
#
```

File names are build both for model and pressure level files.

```
cat > namelist << *EOINFILE
&namsec
    label ='Hirlam OPS.'
    lulog =6,
    logfile ='cross.log',
    print =.true.,</pre>
```

Log parameters are given here. In this case extra information are given on output 6 (the screen). If log_unit is set to 7 the output will be written to cross.log

mlevfile_1	='\${MLEVFIL1}'
plevfile_1	='\${PLEVFIL1}'
rlono	=\$LONO.,
rlano	=\$LANO.,
rloso	=\$LOSO.,
rlaso	=\$LASO.,

File, date and cross section given. As you can see no date and time has to be given to cross, it is only used to create filenames.

dxplot	=18.,
dyplot	=26.,

Size of the output map in cm.

pmin	=200,
pmax	=1050,

Max and min pressure to be drawn.

par	=11,33,34,39	, 13
information	typ =	109,109,109,100,105,
diff	=.f.,.f.,.f.	,.f.,.f.,
scale	=1.,1.,1,.1,	1.,
subtract	=0.,0.,0.,0.	,0.,
contours	=.f.,.t.,.f.	,.f.,.t.,
con_int	=5., 2., 2.,	2.,5.,
con_col	=004,001,002	,003,007,
con_thi	=4, 3, 3,	3,2,

con_lin	=1, 1, 2, 1,2,
shading	=.f.,.f.,.f.,.f.,.f.,
sha_num	=7,
sha_lis	=0.1,0.5,1.0,2.0,3.0,5.0,10.,25.,
sha_col	= 019,021,023,025,027,029,031,
sha_pat	=4,4,4,4,4,4,4,4,4,4,4,
arrows	=.f.,.f.,.f.,.f.,
arr_len	= 1.0,
arr_col	= 002,

Parameters for plotting. See namelists for maps if more information wanted.

=001,
=.f.,
=.t.,
=.t.,
=1,
=0.2,
=0.2,

Switches for model levels, mountains and others.

&end
*EOINFILE
#
/local_disk/MetgraF_2/bin/cross_b namelist cross.ps P a4
#

Executes Cross with the namelist input, the output filename, L for landscape orientation (P portrait) and a4 for the paper size.

rm namelist done exit

Cross example 2 A cross section script that also gives a map with a line that shows the position of the script. Look at: crossmap