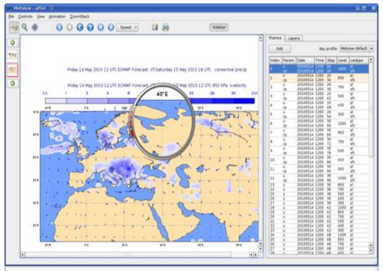
Data Analysis and Visualisation Using Metview

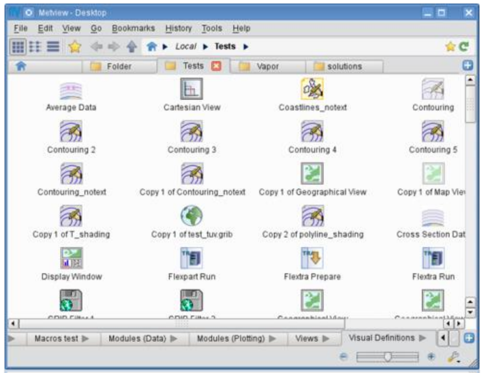
Computer User Training Course 2015

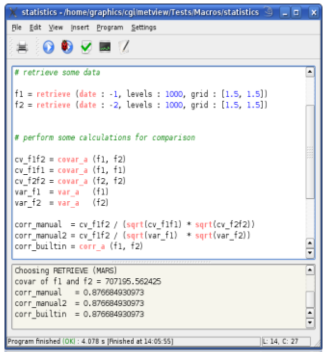
Iain Russell, Fernando Ii, Sándor Kertész, Stephan Siemen

Development Section

metview@ecmwf.int






statistics - /home/graphics/cgi/metview/Tests/Macros/statistics
File Edit View Insert Program Settings

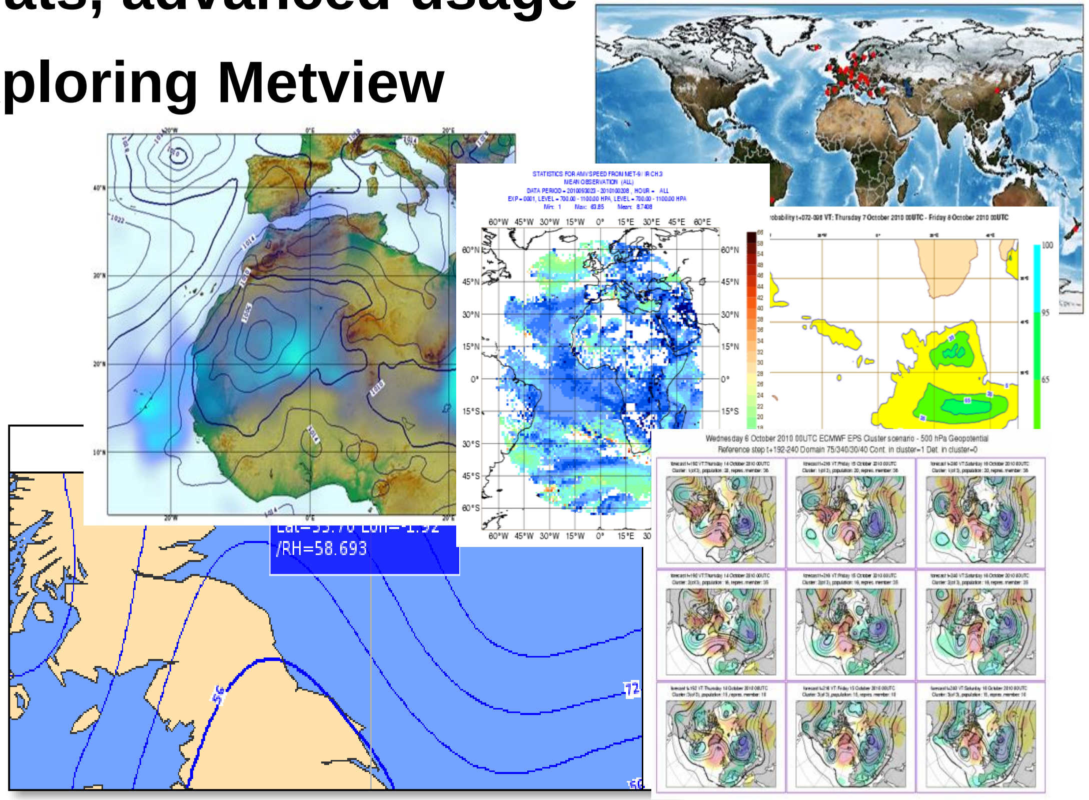
```
# retrieve some data

f1 = retrieve (date : -1, levels : 1000, grid : [1.5, 1.5])
f2 = retrieve (date : -2, levels : 1000, grid : [1.5, 1.5])


# perform some calculations for comparison

cv_f1f2 = covar_a (f1, f2)
cv_f1f1 = covar_a (f1, f1)
cv_f2f2 = covar_a (f2, f2)
var_f1  = var_a   (f1)
var_f2  = var_a   (f2)

corr_manual  = cv_f1f2 / (sqrt(cv_f1f1) * sqrt(cv_f2f2))
corr_manual2 = cv_f1f2 / (sqrt(var_f1)  * sqrt(var_f2))
corr_builtin = corr_a (f1, f2)
```

```
Choosing RETRIEVE (MARS)
covar of f1 and f2 = 707195.562425
corr_manual   = 0.876684930973
corr_manual2  = 0.876684930973
corr_builtin  = 0.876684930973
```

```
Program finished (OK) : 4.078 s [Finished at 14:05:55]    L: 14, C: 27
```

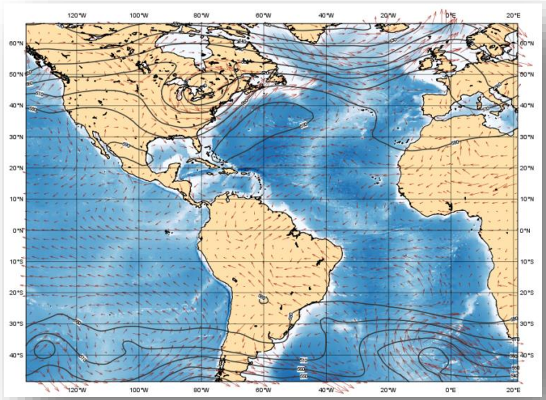
Outline

- Day 1: Introduction, main features
- Day 2: Data (1) and processing
- Day 3: Data (2), time and graphs
- Day 4: Graphics formats, advanced usage
- Day 5: Batch jobs, exploring Metview



Metview: meteorological workstation

- **Retrieve/manipulate/visualise meteorological data**
- **Working environment for operational and research meteorologists**
- **Allows analysts and researchers to easily build products interactively and run them in batch mode**



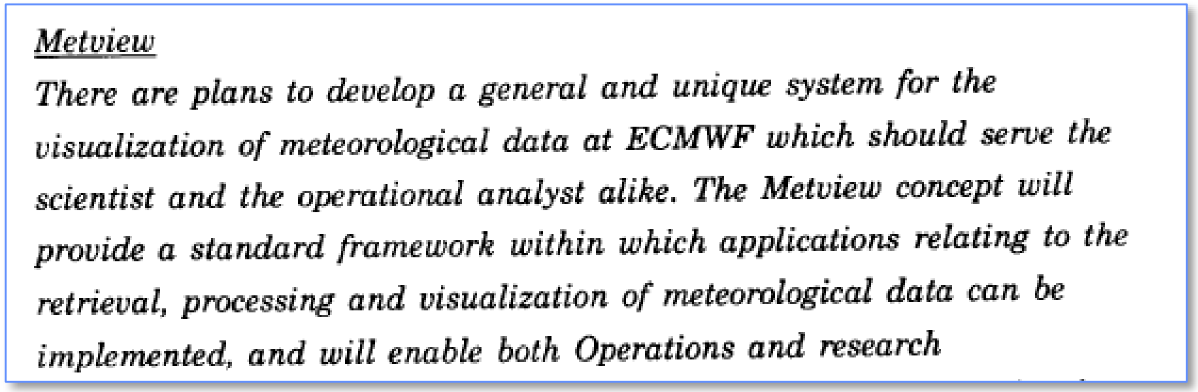
Built on core ECMWF technologies:

MARS, GRIB_API, Magics, ODB, Emoslib

- **Open Source under Apache Licence 2.0**
  - ***Increased interest from research community***
- **Metview is a co-operation project with INPE (Brazil)**

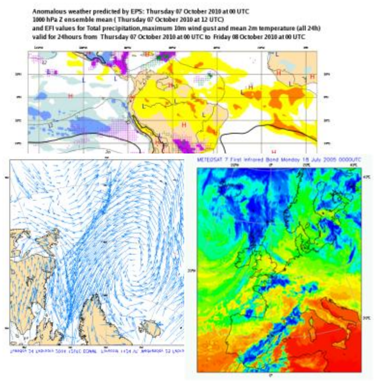
Metview history

- **Announced at first EGOWS in June 1990 (Oslo)**



Metview

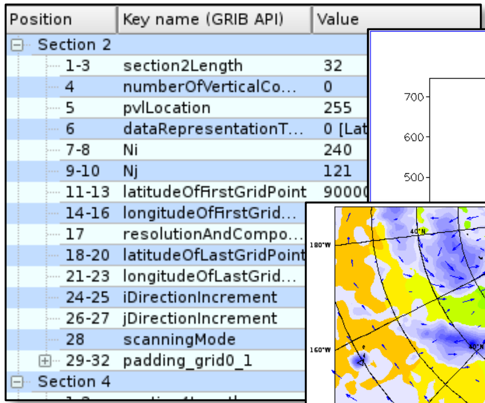
There are plans to develop a general and unique system for the visualization of meteorological data at ECMWF which should serve the scientist and the operational analyst alike. The Metview concept will provide a standard framework within which applications relating to the retrieval, processing and visualization of meteorological data can be implemented, and will enable both Operations and research



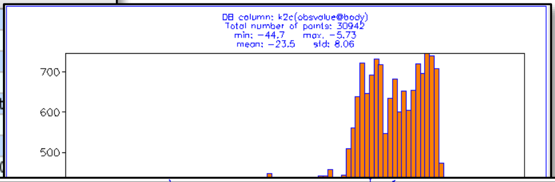
● **First prototype in 1991**	***INPE***
● **First operational version in 1993**	***Metview 1.0***
● **OpenGL graphics introduced in 1998**	***Metview 2.0***
● **New user interface in 2000**	***Metview 3.0***
● **Magics++ and Qt introduced in 2010**	***Metview 4.0***
● **New Qt Desktop introduced in 2014**	***Metview 4.5***

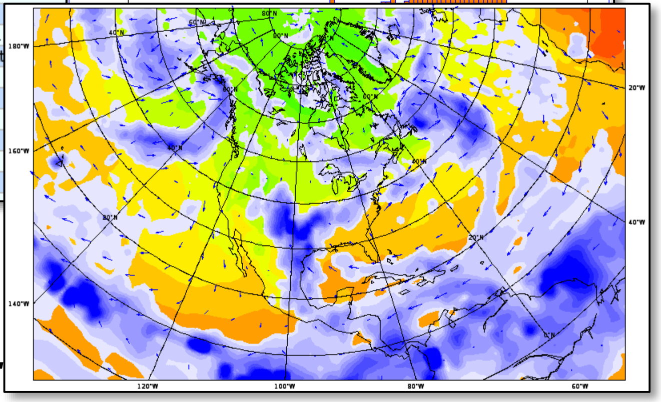
What can Metview do?

- **Data:**
  - **Access**
  - **Examine**
  - **Manipulate**
  - **Plot / Overlay**
    - **Generate graphics files:**
      ps, eps, kml, svg, png,…



Position	Key name (GRIB API)	Value
Section 2		
1-3	section2Length	32
4	numberOfVerticalCo…	0
5	pvlLocation	255
6	dataRepresentationT…	0 [Lat
7-8	Ni	240
9-10	Nj	121
11-13	latitudeOfFirstGridPoint	9000(
14-16	longitudeOfFirstGrid…	
17	resolutionAndCompo…	
18-20	latitudeOfLastGridPoint	
21-23	longitudeOfLastGrid…	
24-25	iDirectionIncrement	
26-27	jDirectionIncrement	
28	scanningMode	
29-32	padding_grid0_1	
Section 4		



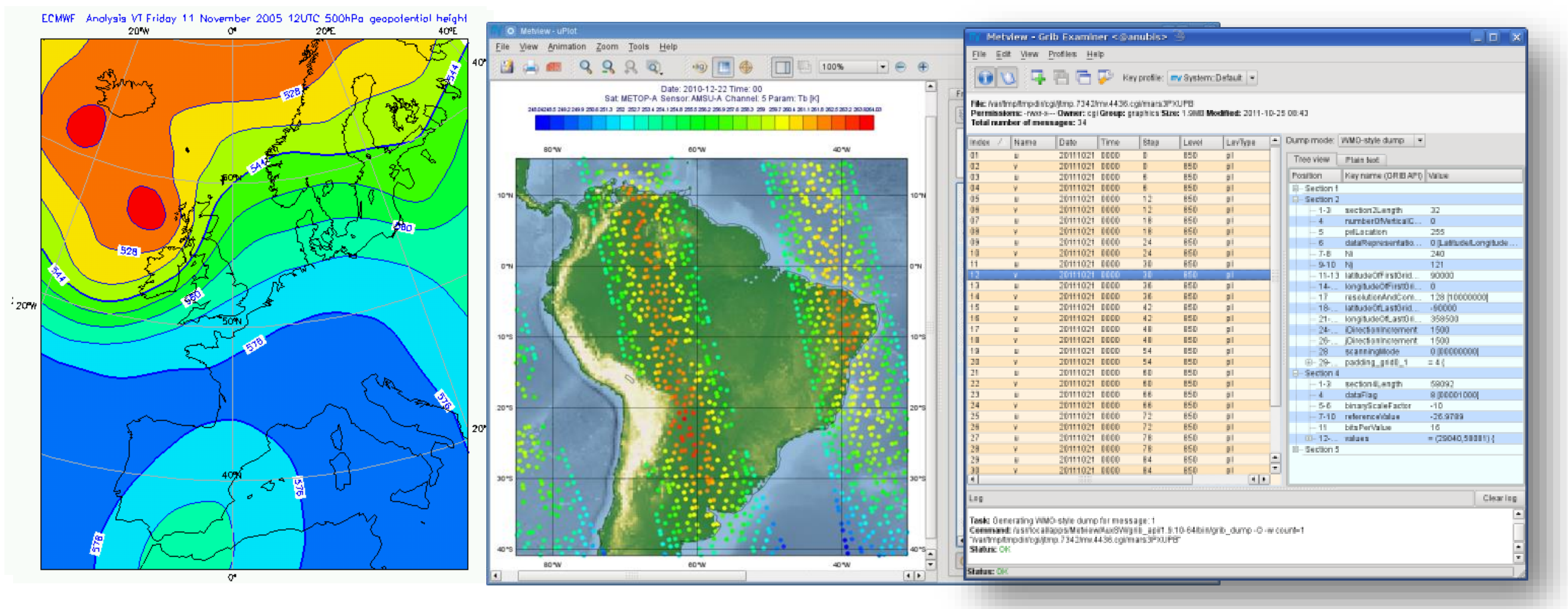


- **Can be run interactively or in batch**
- **Runs self-contained standalone**
  - **From laptops to supercomputers**
  - **No special data servers required (but easily connected to MARS or local databases)**

Main features

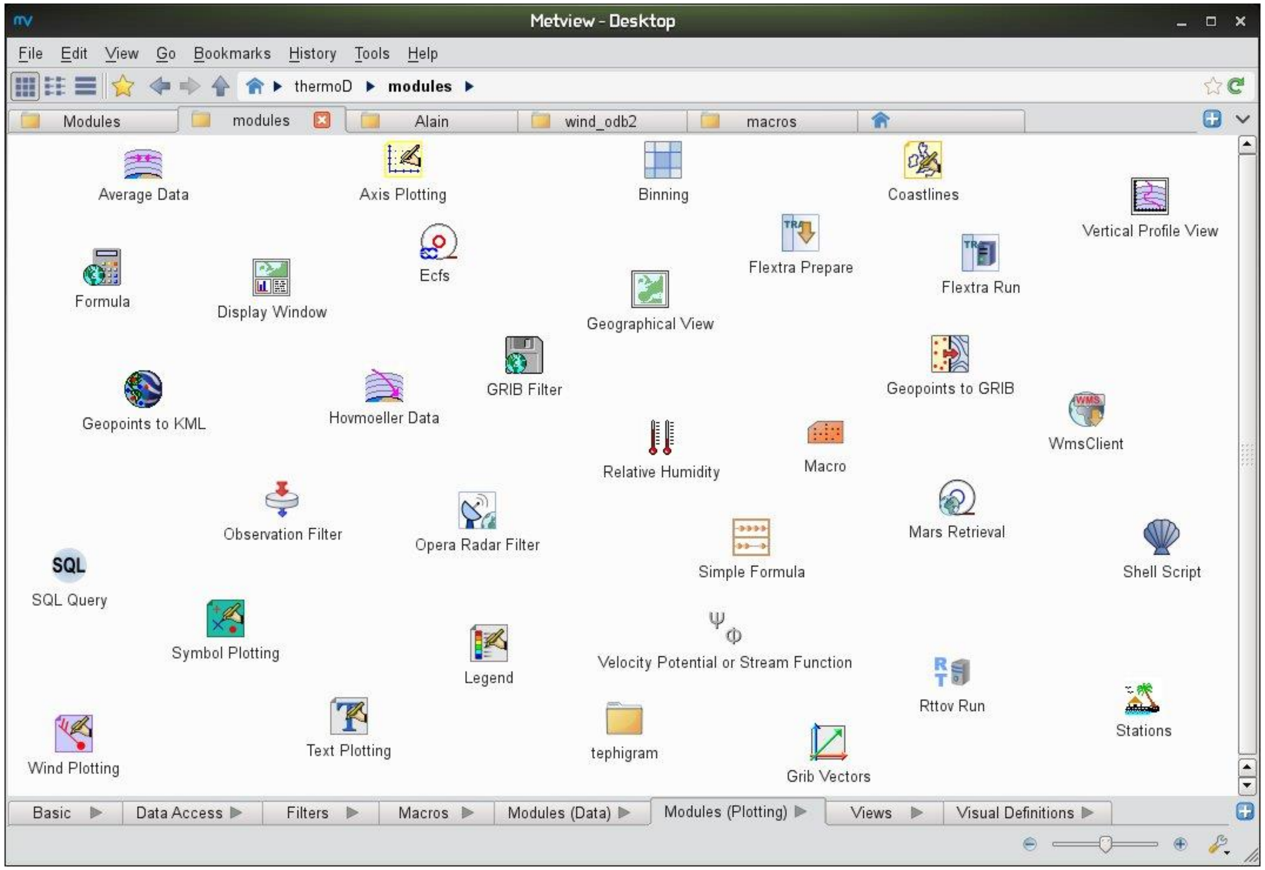
1) **Data handling**

- **Supports a variety of data types (meteorological and non-meteorological)**
- **Rich set of modules and functions for data manipulation**


ECMWF Analysis VT Friday 11 November 2005 12UTC 500hPa geopotential height
Metview - uPlot
Date: 2010-12-22 Time: 00
Sat: METOP-A Sensor: AMSU-A Channel: 5 Param: Tb [K]
Metview - Grib Examiner

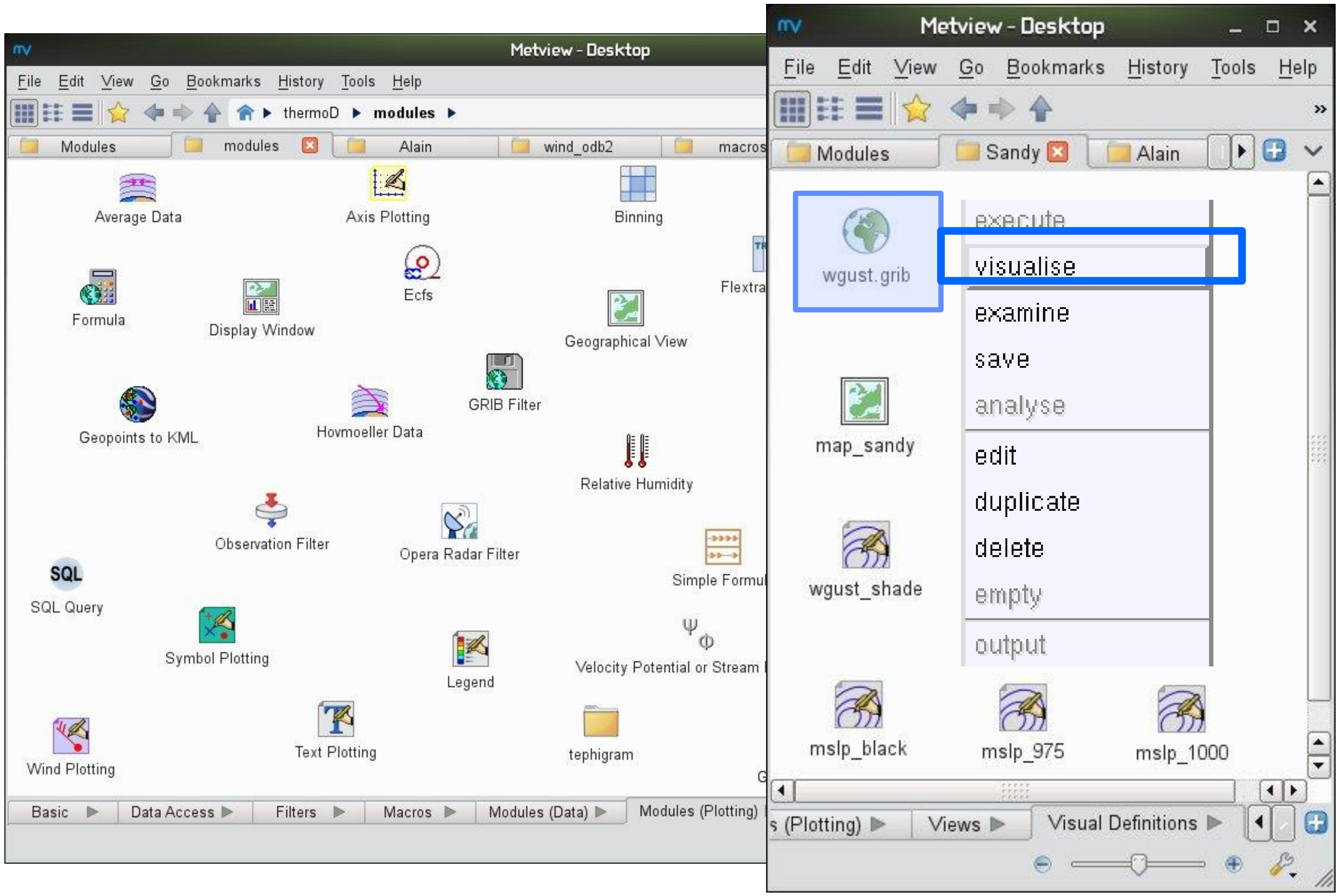
Main features

2) Icon-based interface



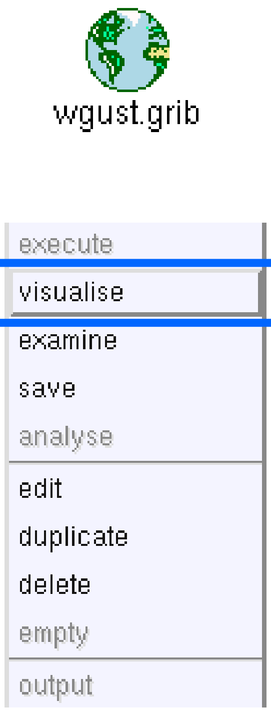
Main features

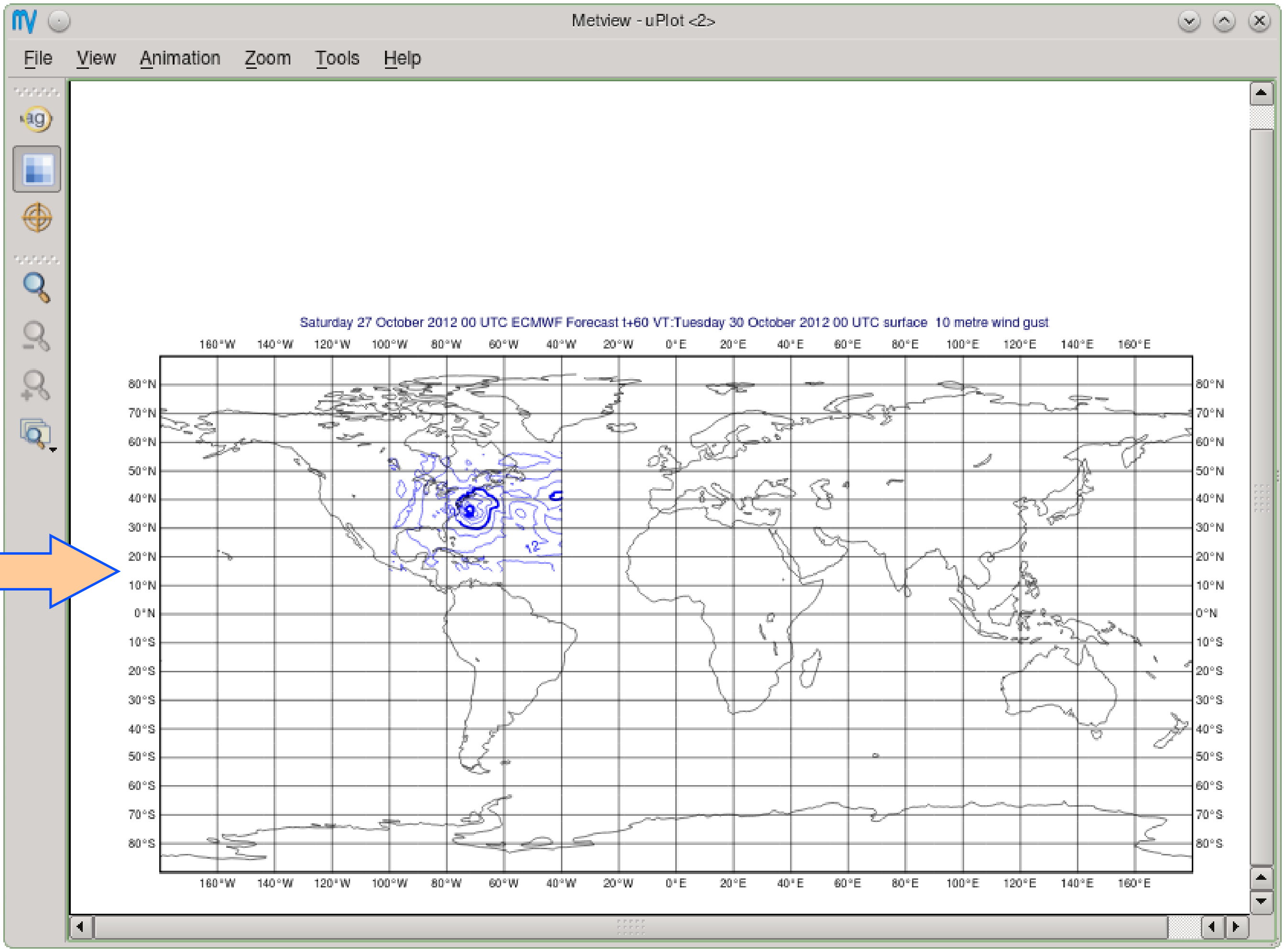
3) Drag and Drop support



Visualisation

GRIB file



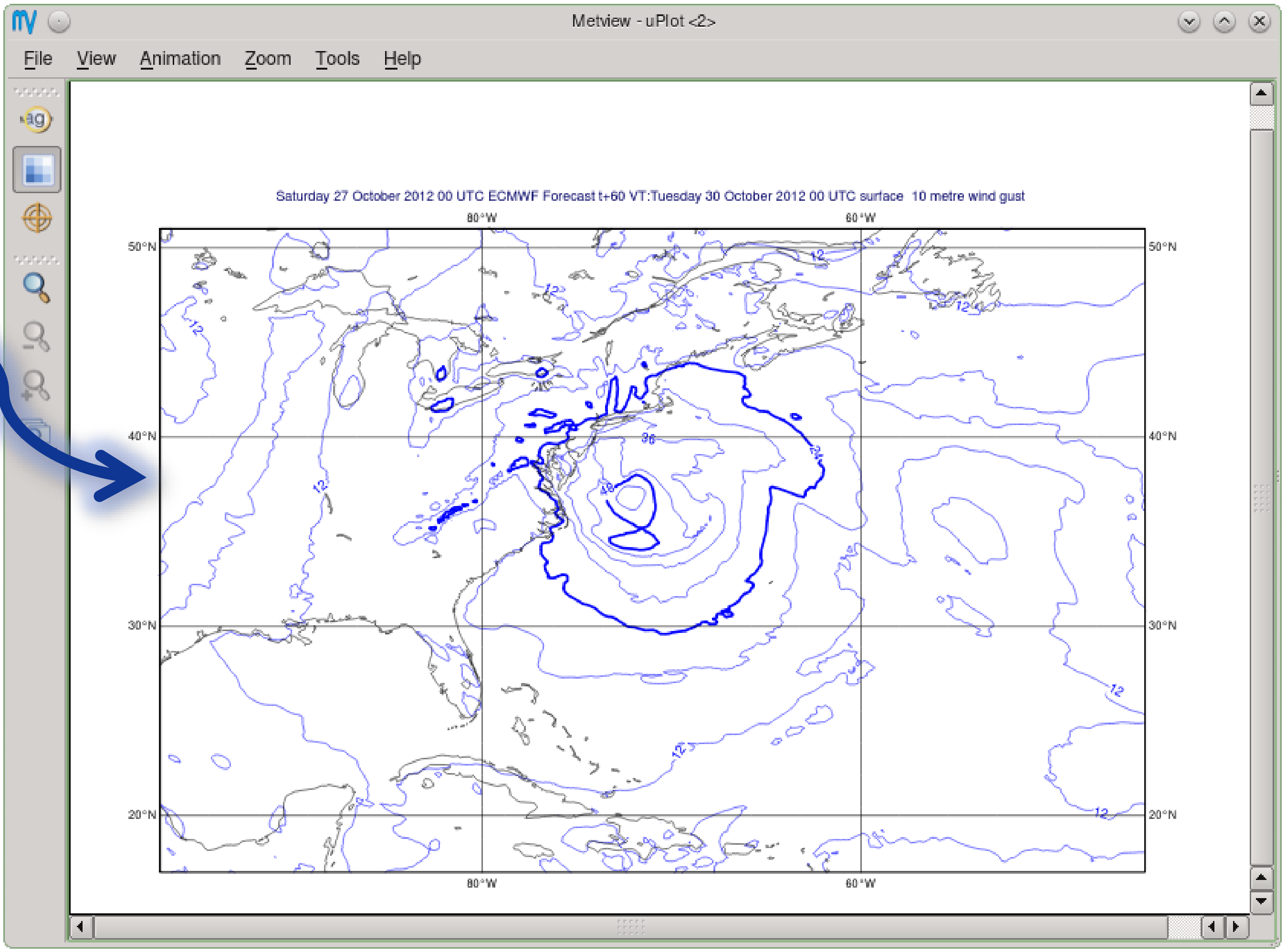


Saturday 27 October 2012 00 UTC ECMWF Forecast t+60 VT:Tuesday 30 October 2012 00 UTC surface 10 metre wind gust

Drag and Drop

Map view

map_sandy

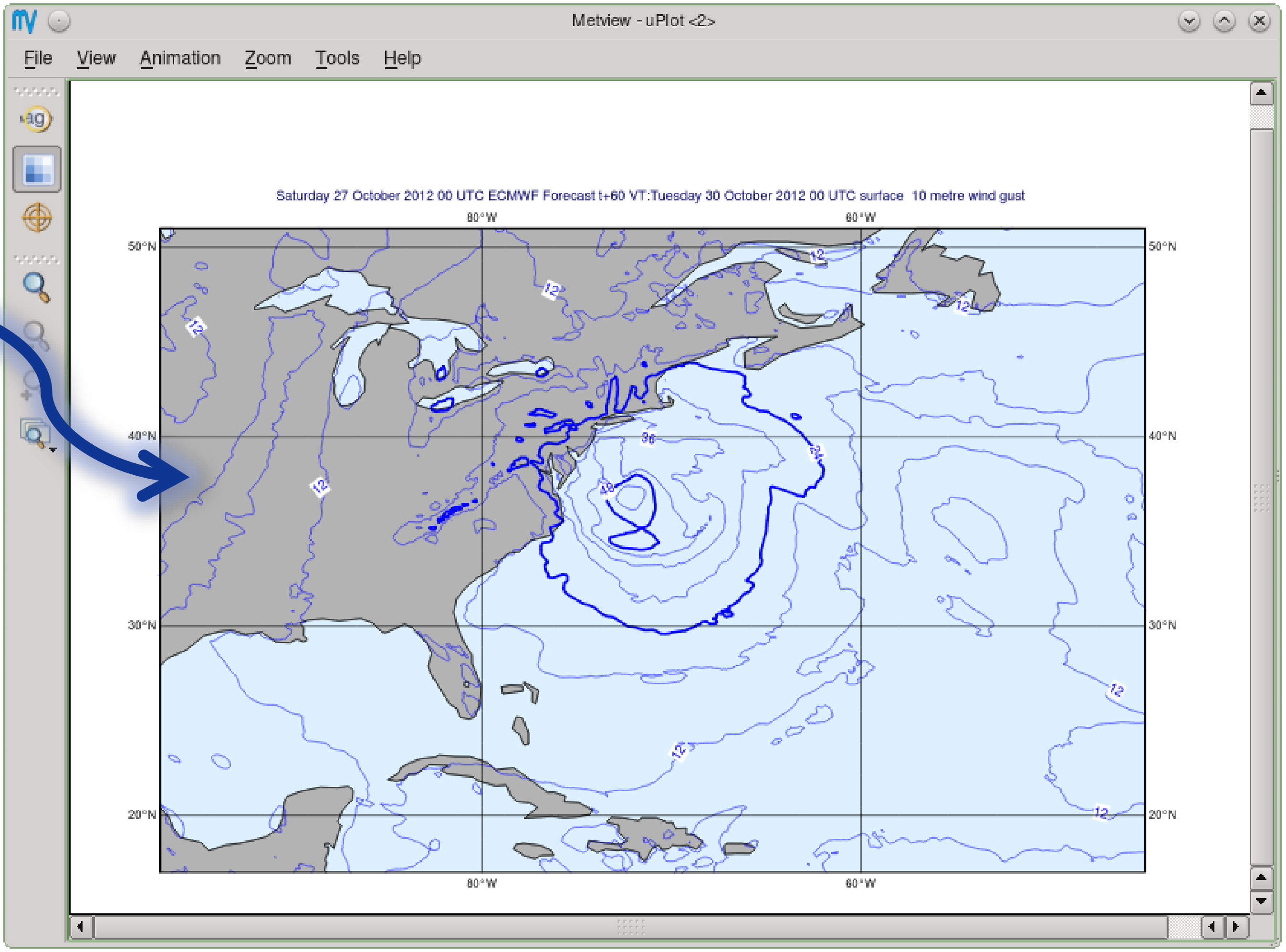


Saturday 27 October 2012 00 UTC ECMWF Forecast t+60 VT:Tuesday 30 October 2012 00 UTC surface 10 metre wind gust

Drag and Drop

Coastlines

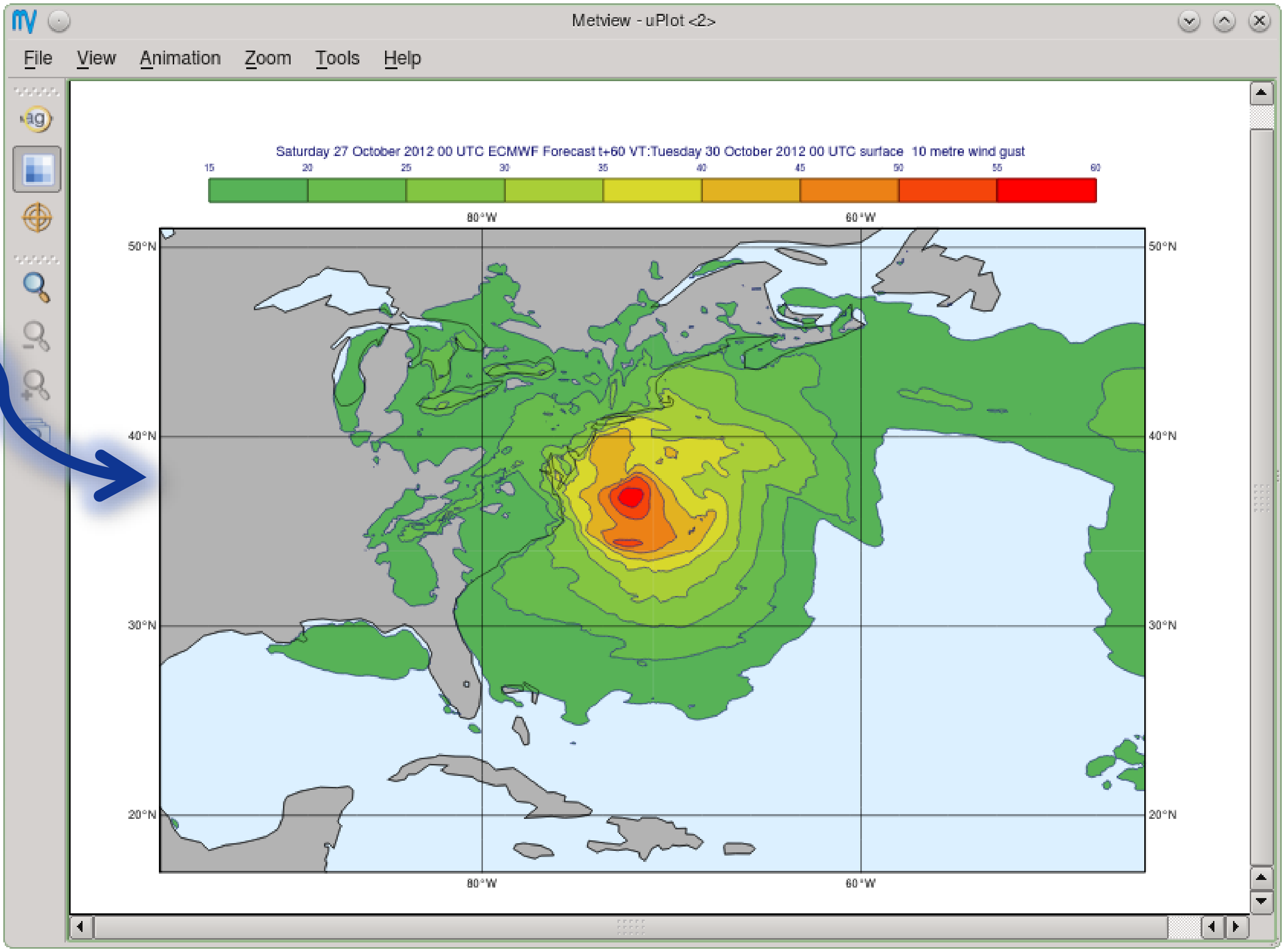
coast_grey_light


Metview - uPlot <2>
File View Animation Zoom Tools Help
Saturday 27 October 2012 00 UTC ECMWF Forecast t+60 VT:Tuesday 30 October 2012 00 UTC surface 10 metre wind gust
80°W
60°W
50°N
40°N
30°N
20°N

Drag and Drop

Contour shading

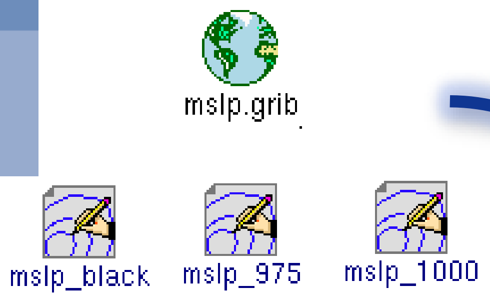
wgust_shade


Metview - uPlot <2>
File View Animation Zoom Tools Help
Saturday 27 October 2012 00 UTC ECMWF Forecast t+60 VT:Tuesday 30 October 2012 00 UTC surface 10 metre wind gust
15 20 25 30 35 40 45 50 55 60
80°W 60°W
50°N 40°N 30°N 20°N

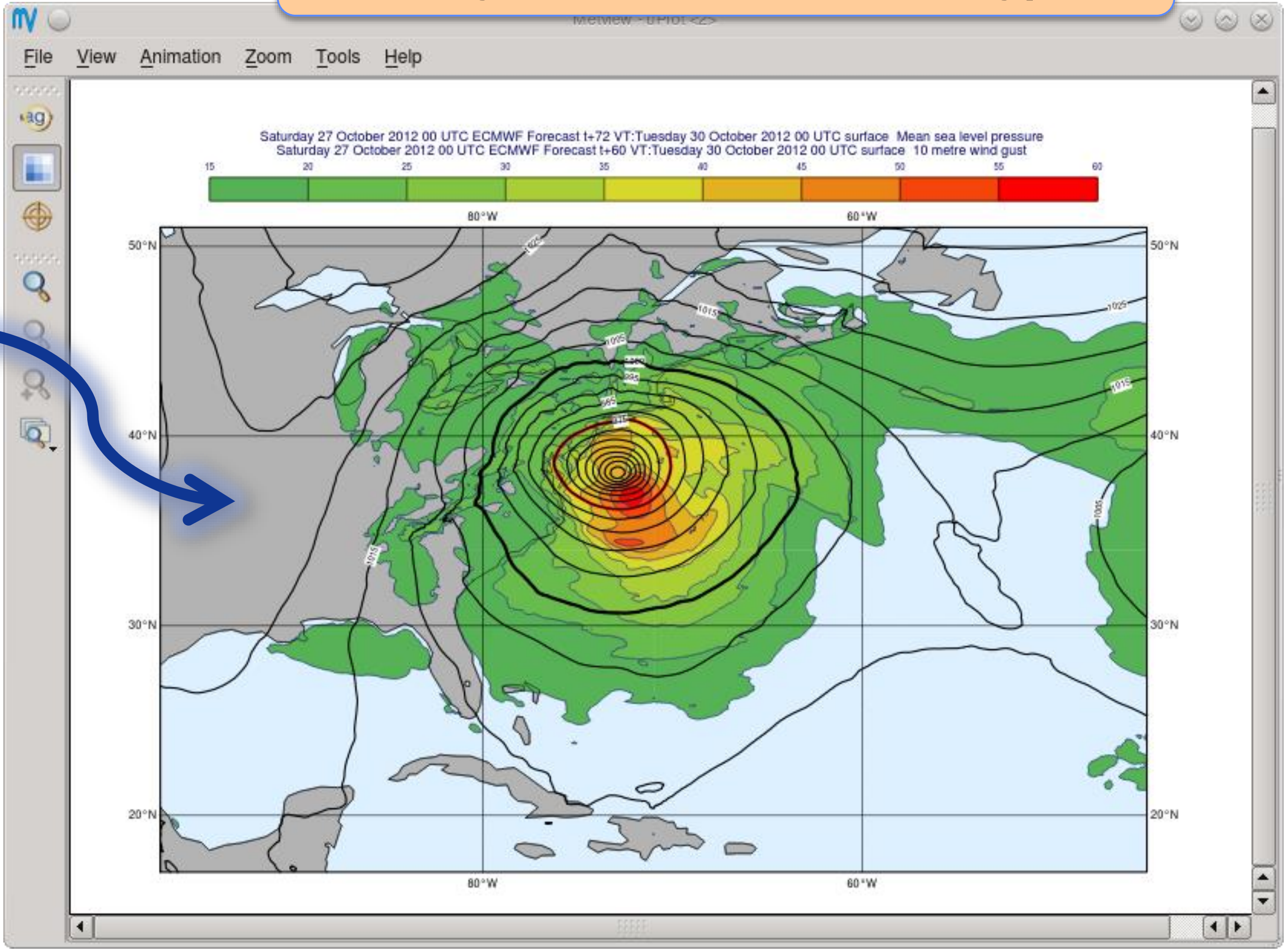
Drag and Drop - Overlay

Overlay works for all the data types!

MSLP (GRIB)



Contouring



Saturday 27 October 2012 00 UTC ECMWF Forecast t+72 VT:Tuesday 30 October 2012 00 UTC surface Mean sea level pressure
Saturday 27 October 2012 00 UTC ECMWF Forecast t+60 VT:Tuesday 30 October 2012 00 UTC surface 10 metre wind gust

Main features

4) Macro language

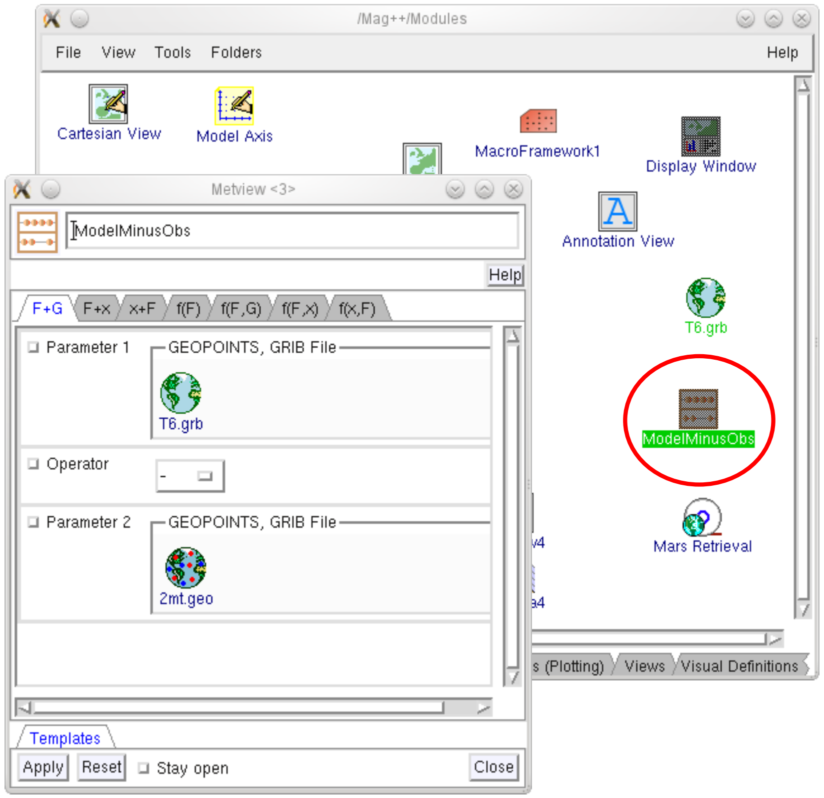
- Powerful meteorologically oriented language
- Simple script language + modern computer language
- Extensive list of functions
- Interfaces with Fortran/C/C++ code
- Outputs:
  - Derived data
  - Interactive plotting window
  - Multiple plots
- Customised editor
- Run in batch or interactive modes

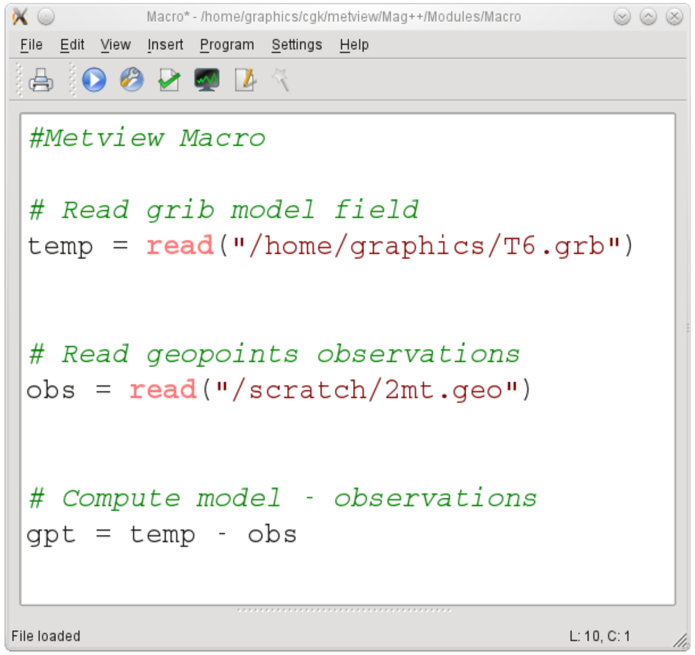
```
# Read a grib file
temp = read ( “/home/graphics/temp.grb” )

# Re-scaling field
if  threshold > 0  then
     temp = temp – 273.5
     a = integrate ( temp )
end if

# Compute the gradient
q = gradientb ( temp )

# Save field
write ( "/home/graphics/gradient.grb“ , q )

# Plot field
plot ( [ps,svg], q )
```

Main features

5) Strong synergy between Icons & Macros

- **Every icon can be translated into a Macro command**



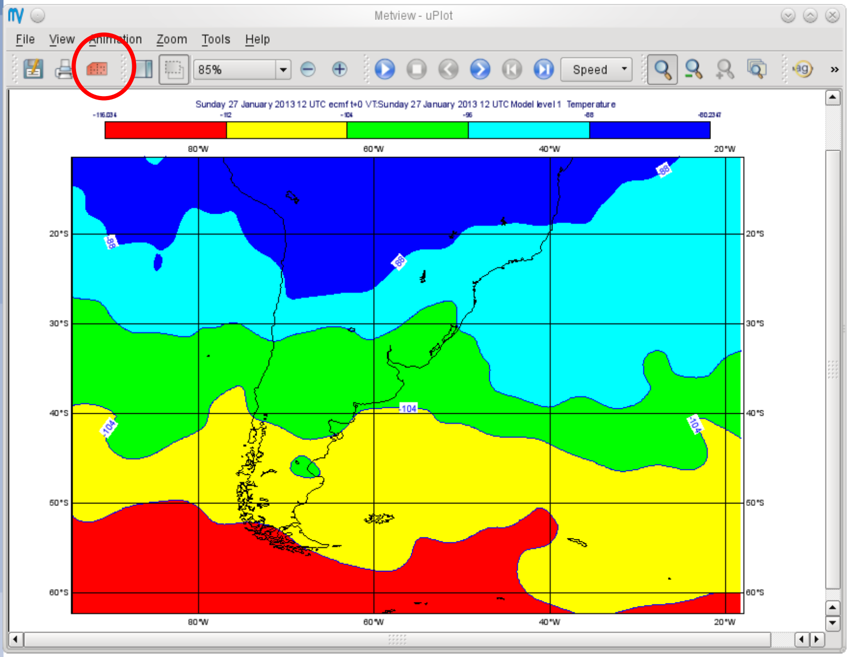

Macro* - /home/graphics/cgk/metview/Mag++/Modules/Macro
File Edit View Insert Program Settings Help
File loaded
L: 10, C: 1

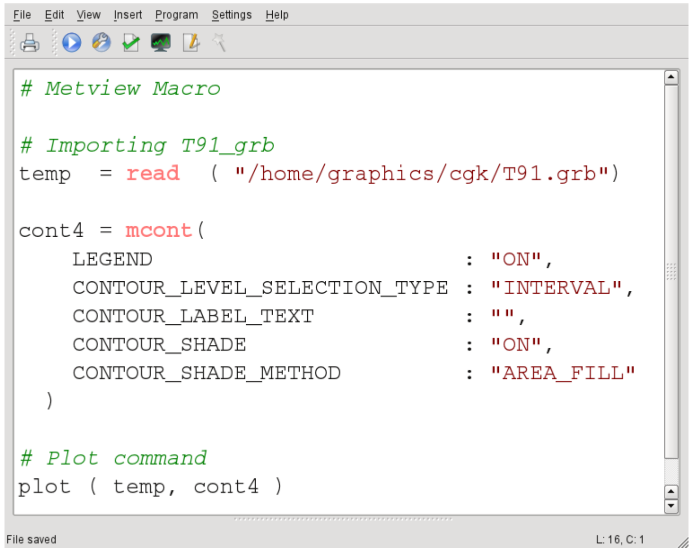
```
#Metview Macro

# Read grib model field
temp = read("/home/graphics/T6.grb")


# Read geopoints observations
obs = read("/scratch/2mt.geo")


# Compute model - observations
gpt = temp - obs
```

Main features

5) Strong synergy between Icons & Macros

- Plots can be translated into a Macro program


Metview - uPlot
File
View
Animation
Zoom
Tools
Help
85%
Speed
Sunday 27 January 2013 12 UTC ecmf t+0 VT:Sunday 27 January 2013 12 UTC Model level 1 Temperature
80°W
60°W
40°W
20°W
20°S
30°S
40°S
50°S
60°S



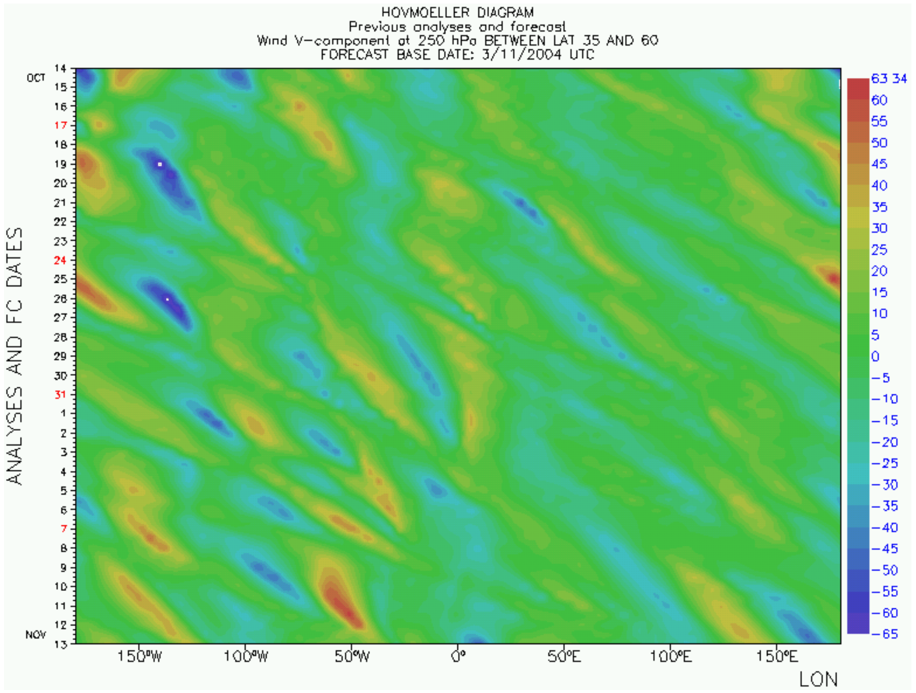
```
# Metview Macro

# Importing T91_grb
temp  = read  ( "/home/graphics/cgk/T91.grb")

cont4 = mcont(
    LEGEND                         : "ON",
    CONTOUR_LEVEL_SELECTION_TYPE : "INTERVAL",
    CONTOUR_LABEL_TEXT           : "",
    CONTOUR_SHADE                : "ON",
    CONTOUR_SHADE_METHOD         : "AREA_FILL"
  )

# Plot command
plot ( temp, cont4 )
```

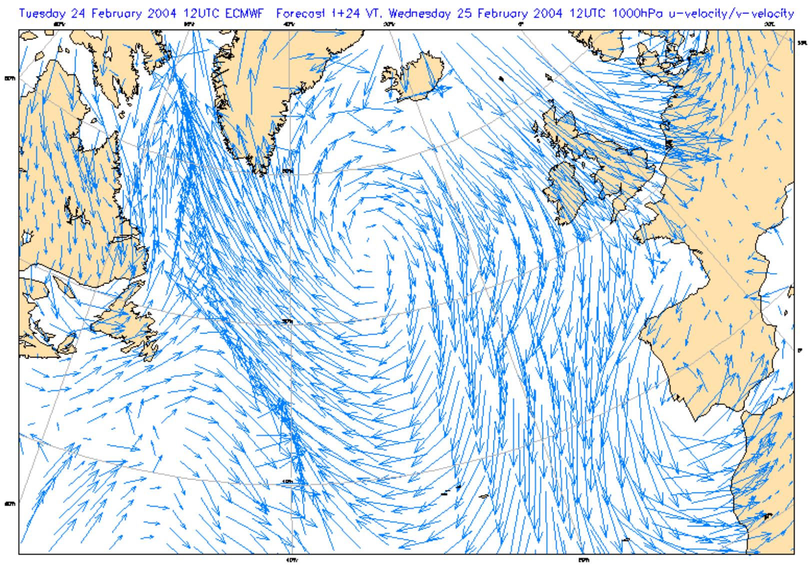
File saved   L: 16, C: 1

Main features

6) **Can produce a variety of meteorological charts**

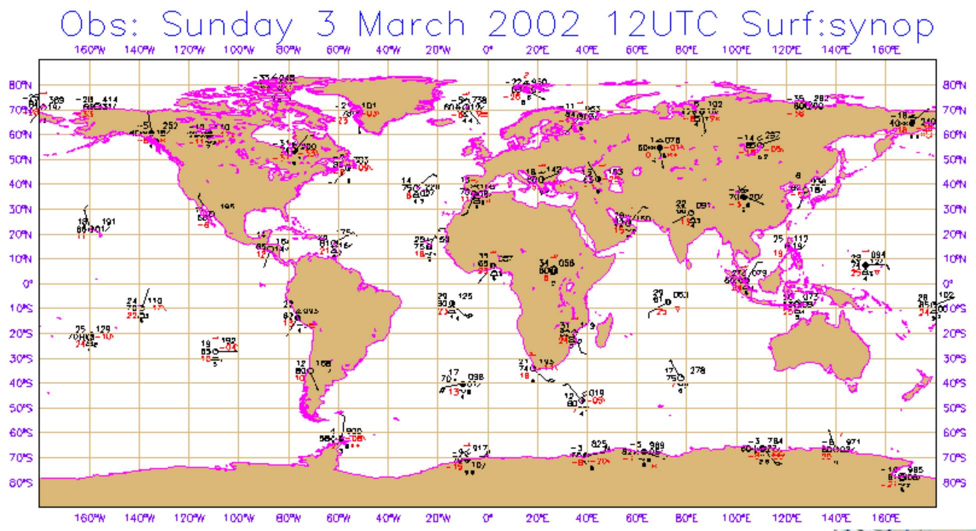
► **Rich set of visualisation attributes**

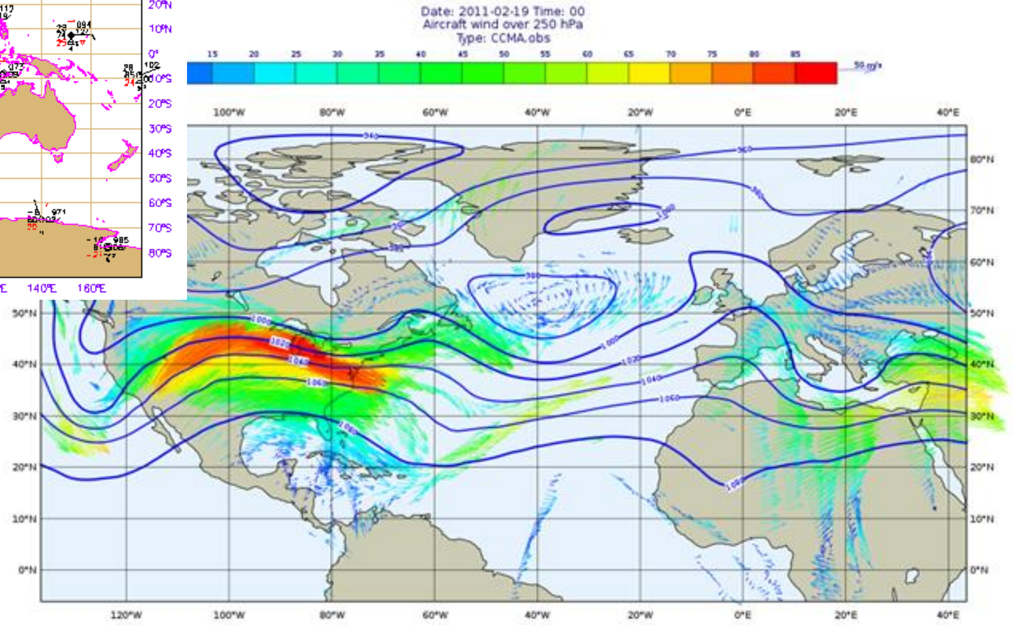



Tuesday 24 February 2004 12UTC ECMWF Forecast t+24 VT. Wednesday 25 February 2004 12UTC 1000hPa u-velocity/v-velocity

Main features

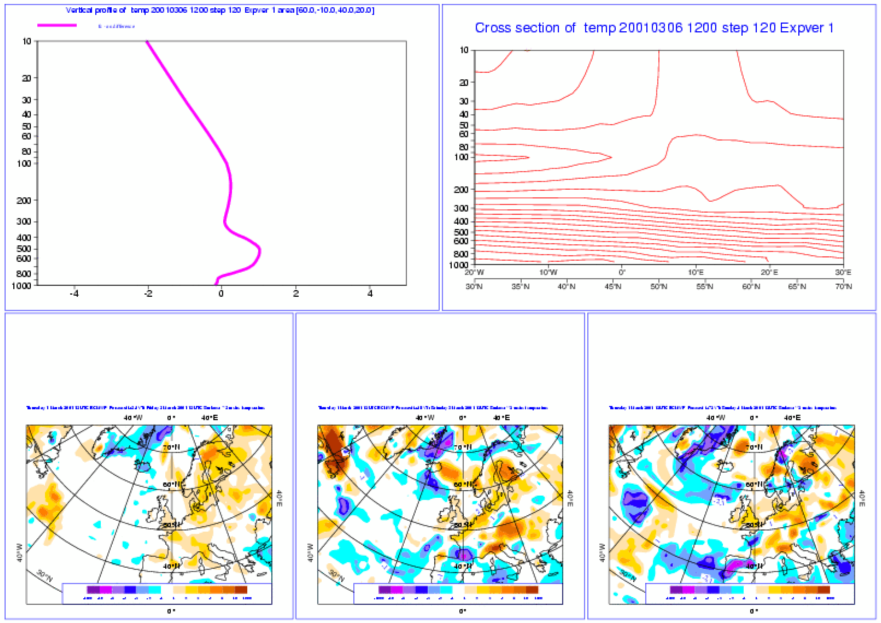
6) Can produce a variety of meteorological charts

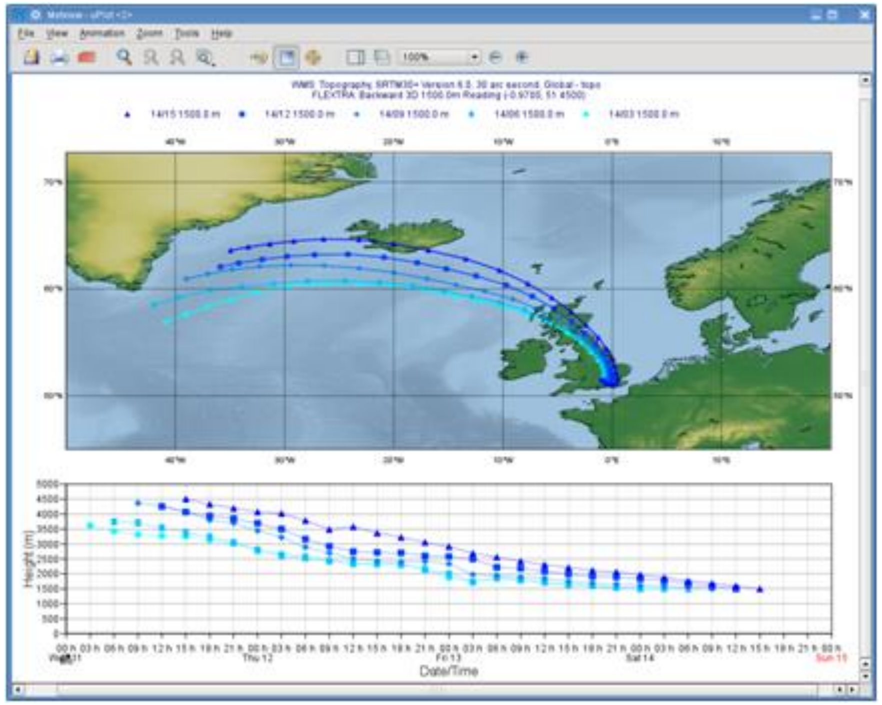




Main features

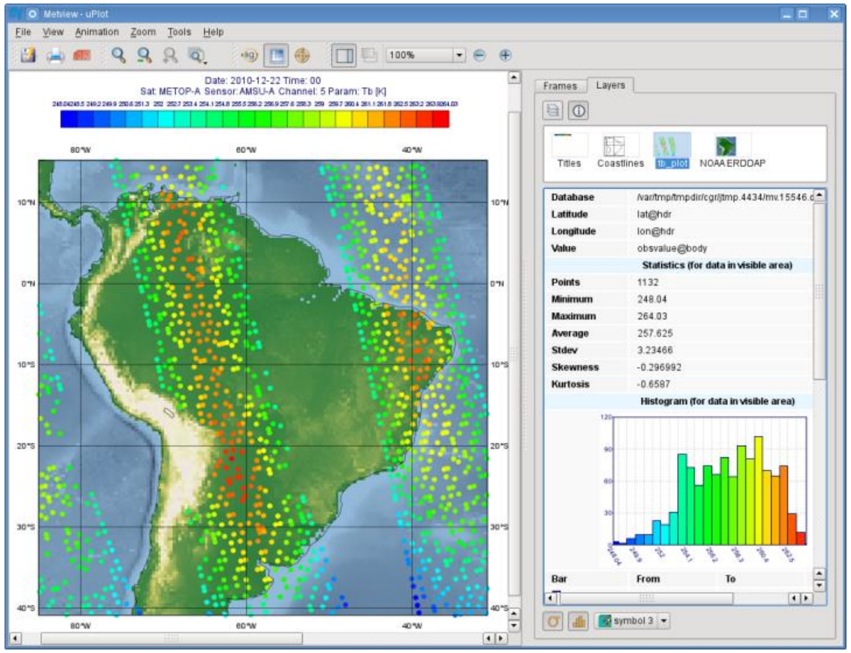
6) Can produce a variety of meteorological charts



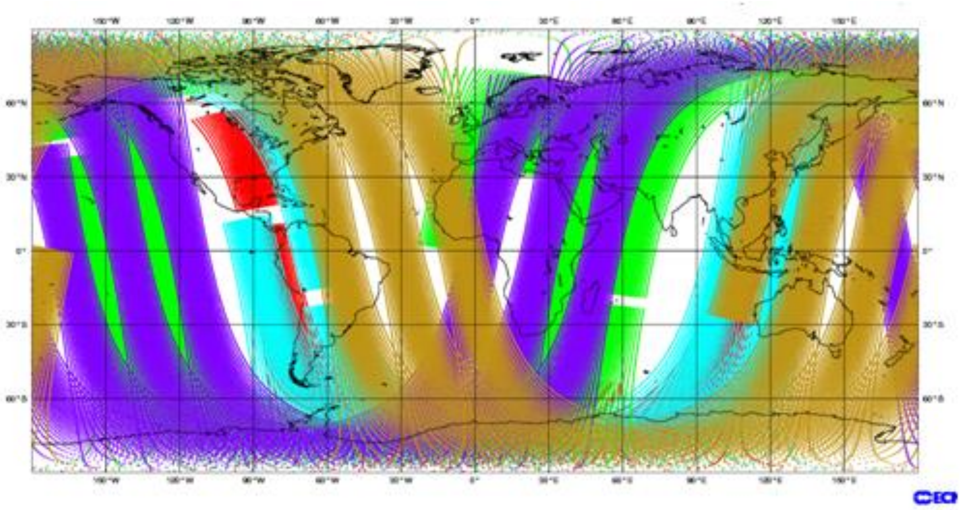

WMS Topography SRTM30+ Version 6.0 30 arc second Global - topo
FLEXTRA Backward 3D 1500.0m Reading (-0.9785, 51.4500)
Height (m)
Date/Time

Main features

6) Can produce a variety of meteorological charts

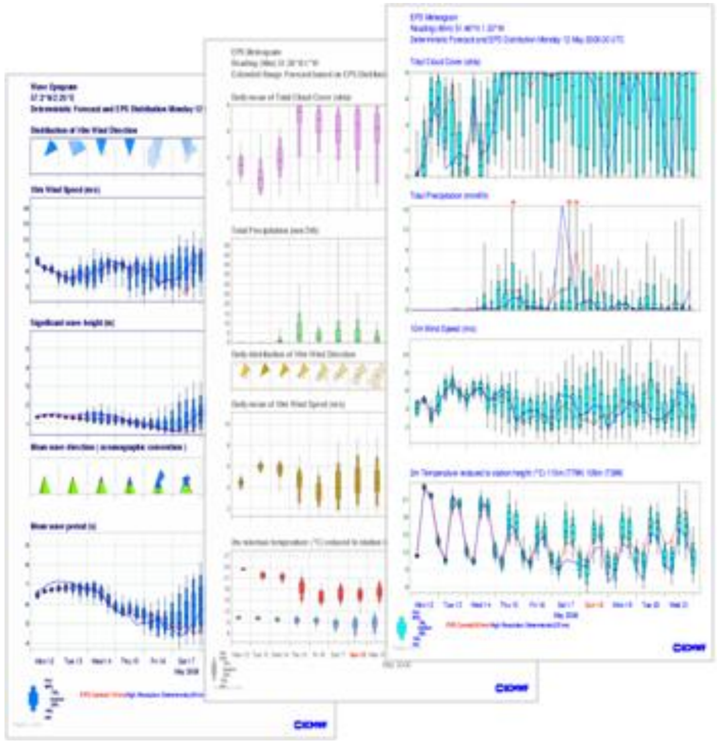


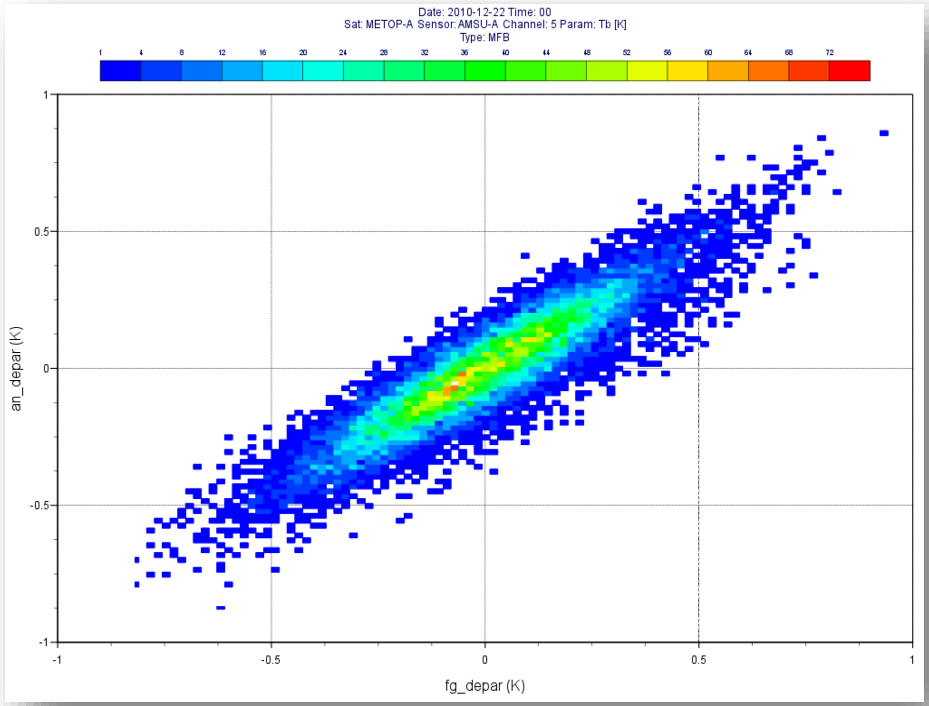
Database	/var/tmp/tmpdir/cgr/jtmp.4434/mv.15546.d
Latitude	lat@hdr
Longitude	lon@hdr
Value	obsvalue@body
	Statistics (for data in visible area)
Points	1132
Minimum	248.04
Maximum	264.03
Average	257.625
Stdev	3.23466
Skewness	-0.296992
Kurtosis	-0.6587
	Histogram (for data in visible area)



Main features

6) **Can produce a variety of meteorological charts**

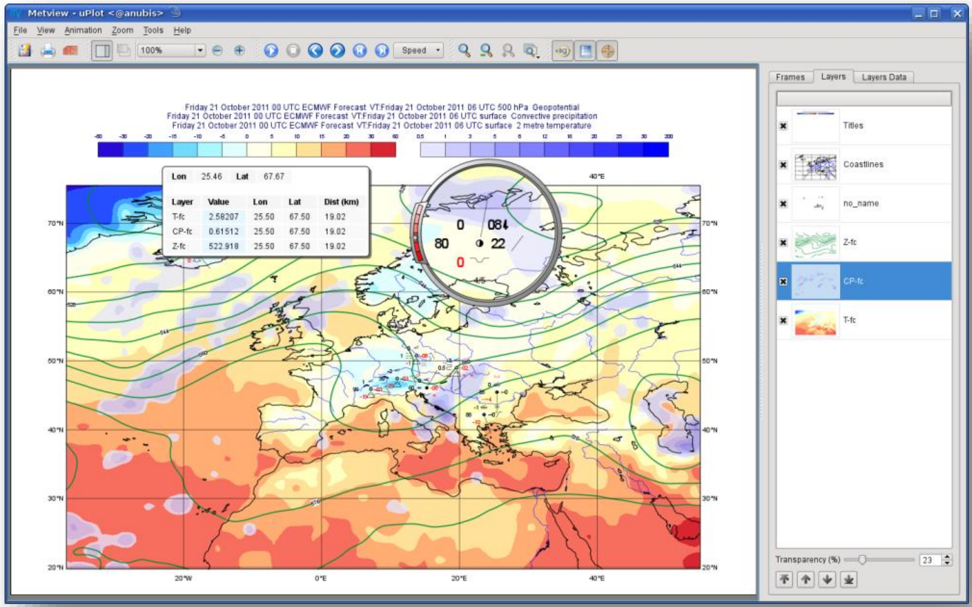




Main features

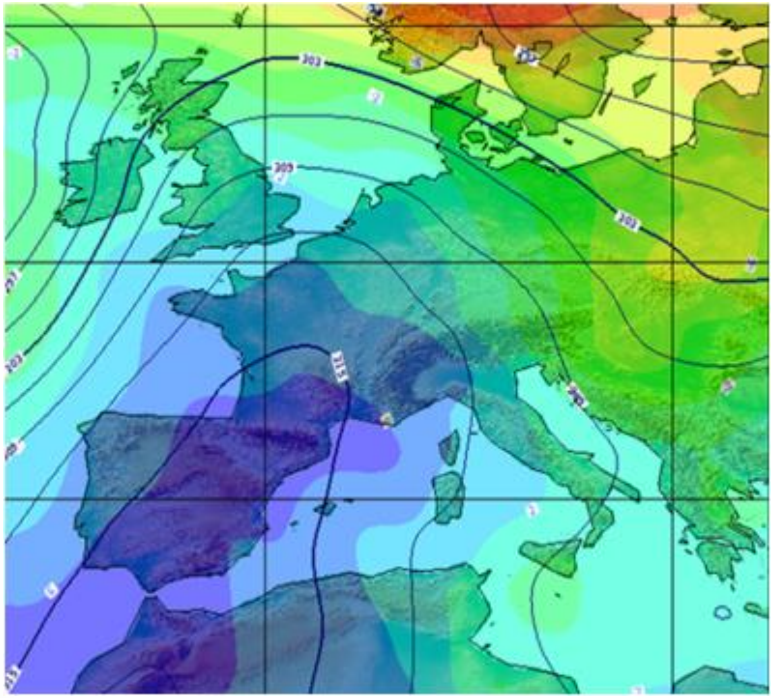
6) **Can produce a variety of meteorological charts**

► **Easy to overlay different data sets**


Metview - uPlot <@anubis>
File View Animation Zoom Tools Help
100%
Speed
Frames Layers Layers Data
Titles
Coastlines
no_name
Z-fc
CP-fc
T-fc
Transparency (%) 23

Friday 21 October 2011 00 UTC ECMWF Forecast VT:Friday 21 October 2011 06 UTC 500 hPa Geopotential
Friday 21 October 2011 00 UTC ECMWF Forecast VT:Friday 21 October 2011 06 UTC surface Convective precipitation
Friday 21 October 2011 00 UTC ECMWF Forecast VT:Friday 21 October 2011 06 UTC surface 2 metre temperature

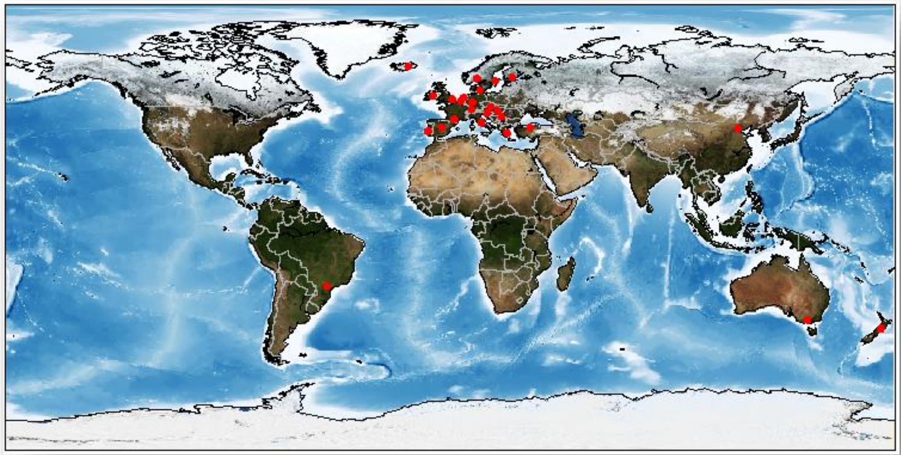
Lon	25.46	Lat	67.67	
Layer	**Value**	**Lon**	**Lat**	**Dist (km)**
T-fc	2.58207	25.50	67.50	19.02
CP-fc	0.61512	25.50	67.50	19.02
Z-fc	522.918	25.50	67.50	19.02



Who uses Metview?

- **Used internally at ECMWF by researchers and operational analysts**
  - **To assess the quality of Observations/Forecast**
  - **To develop new (graphical) products**
  - **For general research activities**

- **Member States (local installations and remotely on our *ecgate* server)**



- **Other national weather services and Universities**
- **Commercial customers of ECMWF products**

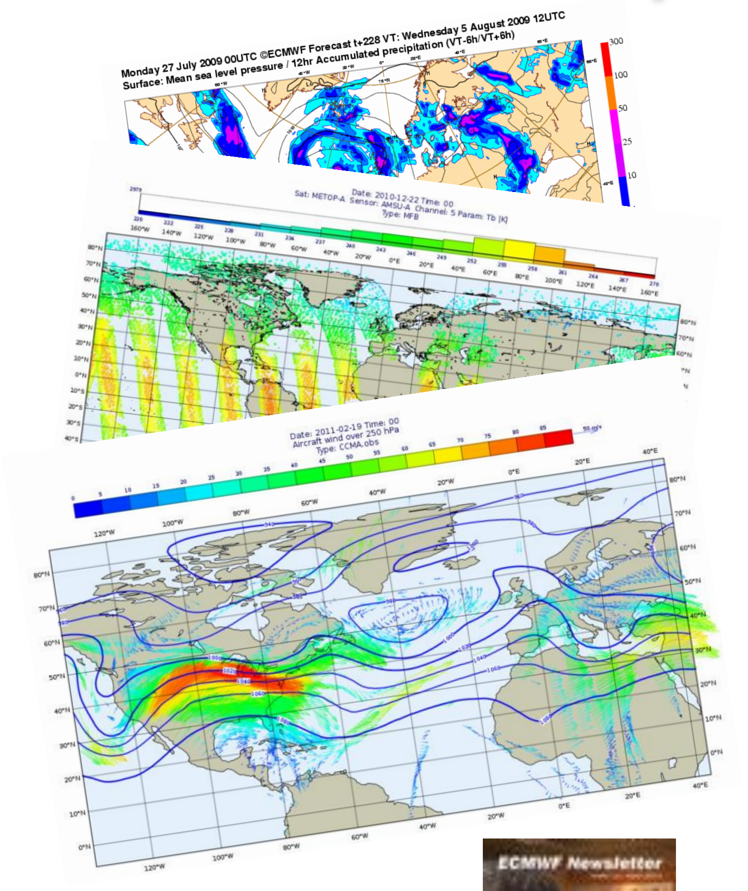
For more information …

email us:

- **Metview: metview@ecmwf.int**

visit our web pages:

- **https://software.ecmwf.int/metview**
- **Download**
  - **Source code, virtual machine**
- **Documentation and tutorials available**
- **Metview articles in recent ECMWF newsletters**


Monday 27 July 2009 00UTC ©ECMWF Forecast t+228 VT: Wednesday 5 August 2009 12UTC
Surface: Mean sea level pressure / 12hr Accumulated precipitation (VT-6h/VT+6h)
Date: 2010-12-22 Time: 00
Sat: METOP-A Sensor: AMSU-A Channel: 5 Param: Tb [K]
Type: MFB
Date: 2011-02-19 Time: 00
Aircraft wind over 250 hPa
Type: CCMA.obs

A quick tour of Metview

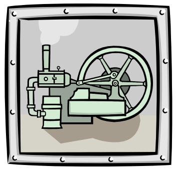
Fernando Ii

Software Applications Team

Metview Principles

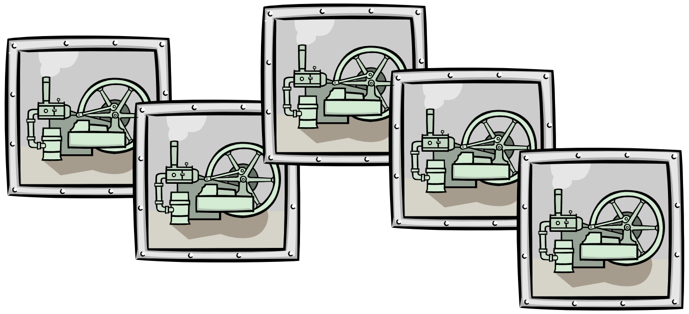
- **First Metview Principle:**

  "Everything in Metview is an Icon"



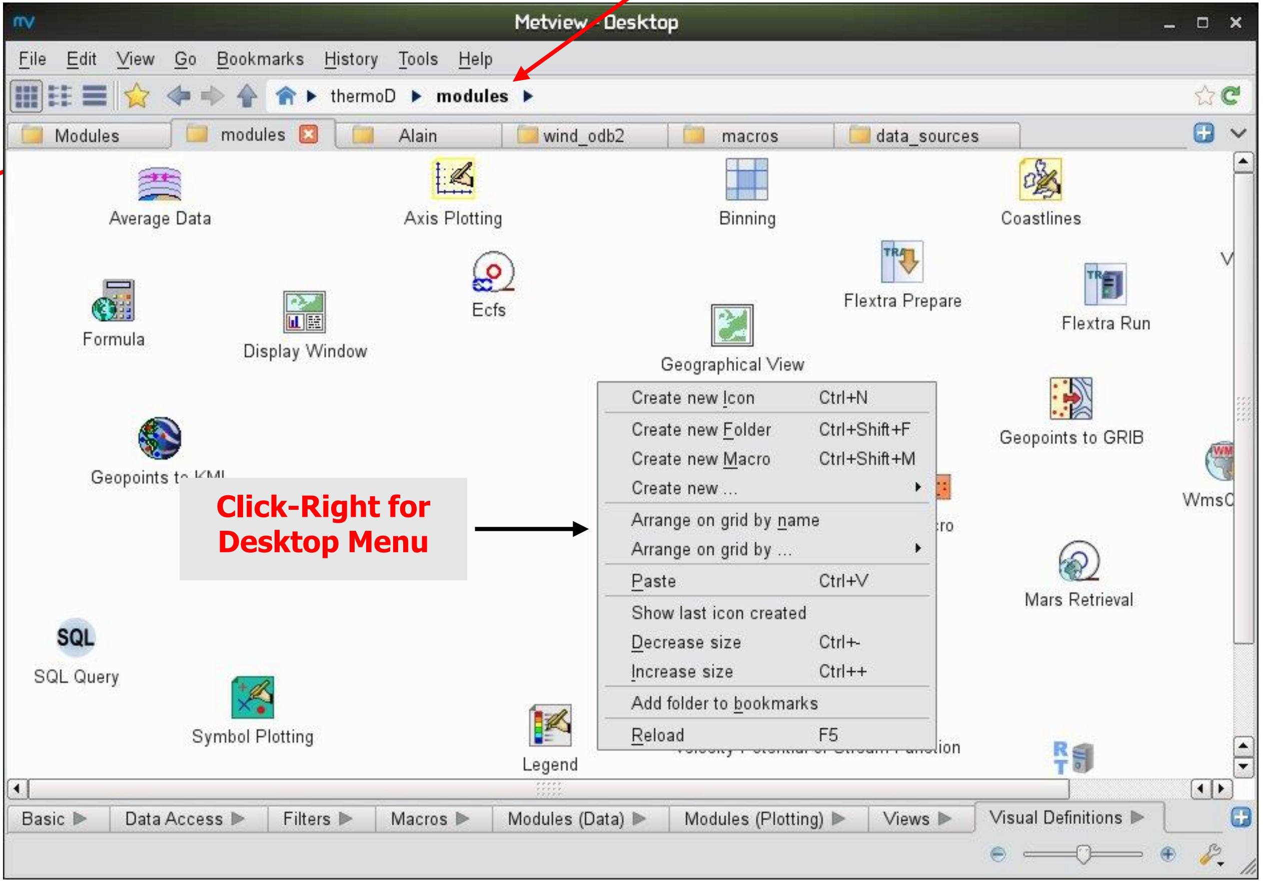
- **Second Metview Principle:**

  "Every Metview Task is a sequence of actions on icons"



Metview Desktop

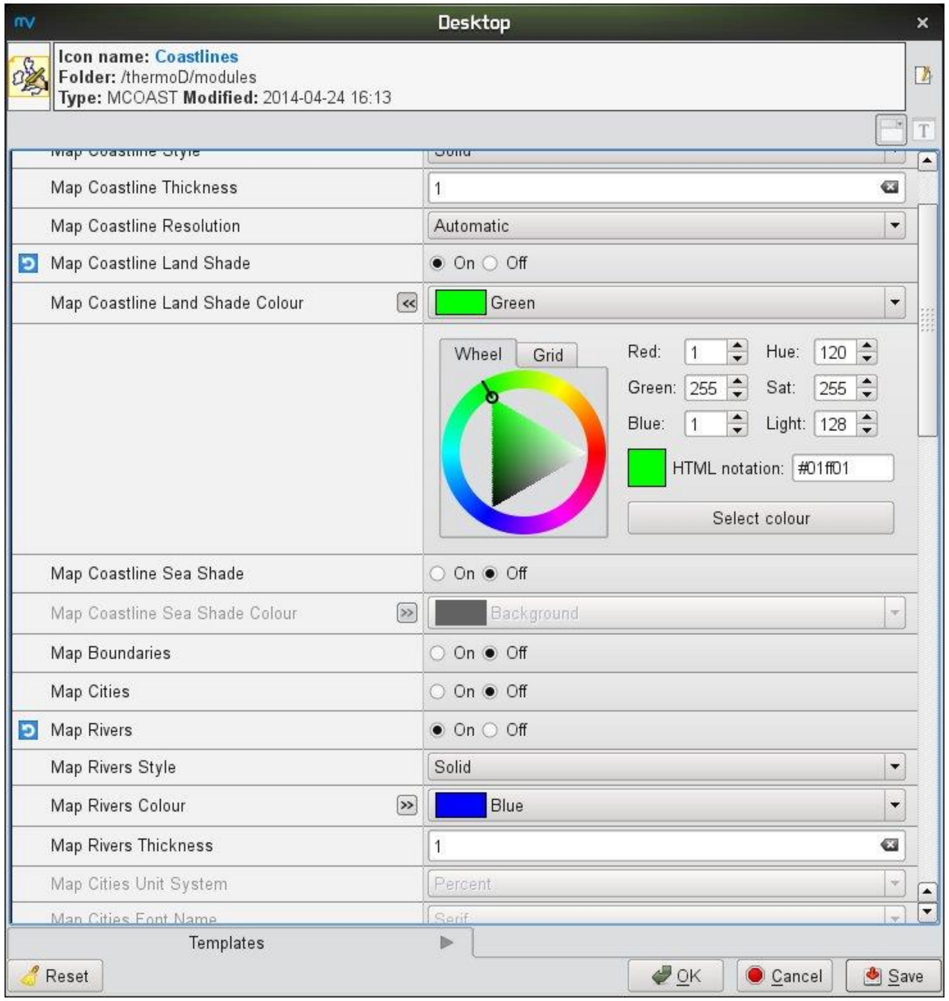
New user interface in version 4.5!



Icon Standard Editor

Input area

Save/Exit area


Desktop

Icon name: Coastlines
Folder: /thermoD/modules
Type: MCOAST Modified: 2014-04-24 16:13

Map Coastline Style	Solid
Map Coastline Thickness	1
Map Coastline Resolution	Automatic
Map Coastline Land Shade	[✓ On] Off
Map Coastline Land Shade Colour	🟩 Green
	Wheel Grid · Red: 1 · Green: 255 · Blue: 1 · Hue: 120 · Sat: 255 · Light: 128 · HTML notation: #01ff01 · Select colour
Map Coastline Sea Shade	On [✓ Off]
Map Coastline Sea Shade Colour	Background
Map Boundaries	On [✓ Off]
Map Cities	On [✓ Off]
Map Rivers	[✓ On] Off
Map Rivers Style	Solid
Map Rivers Colour	🟦 Blue
Map Rivers Thickness	1
Map Cities Unit System	Percent
Map Cities Font Name	Serif

Input element: Alphanumeric Field

Input element: Colour Menu

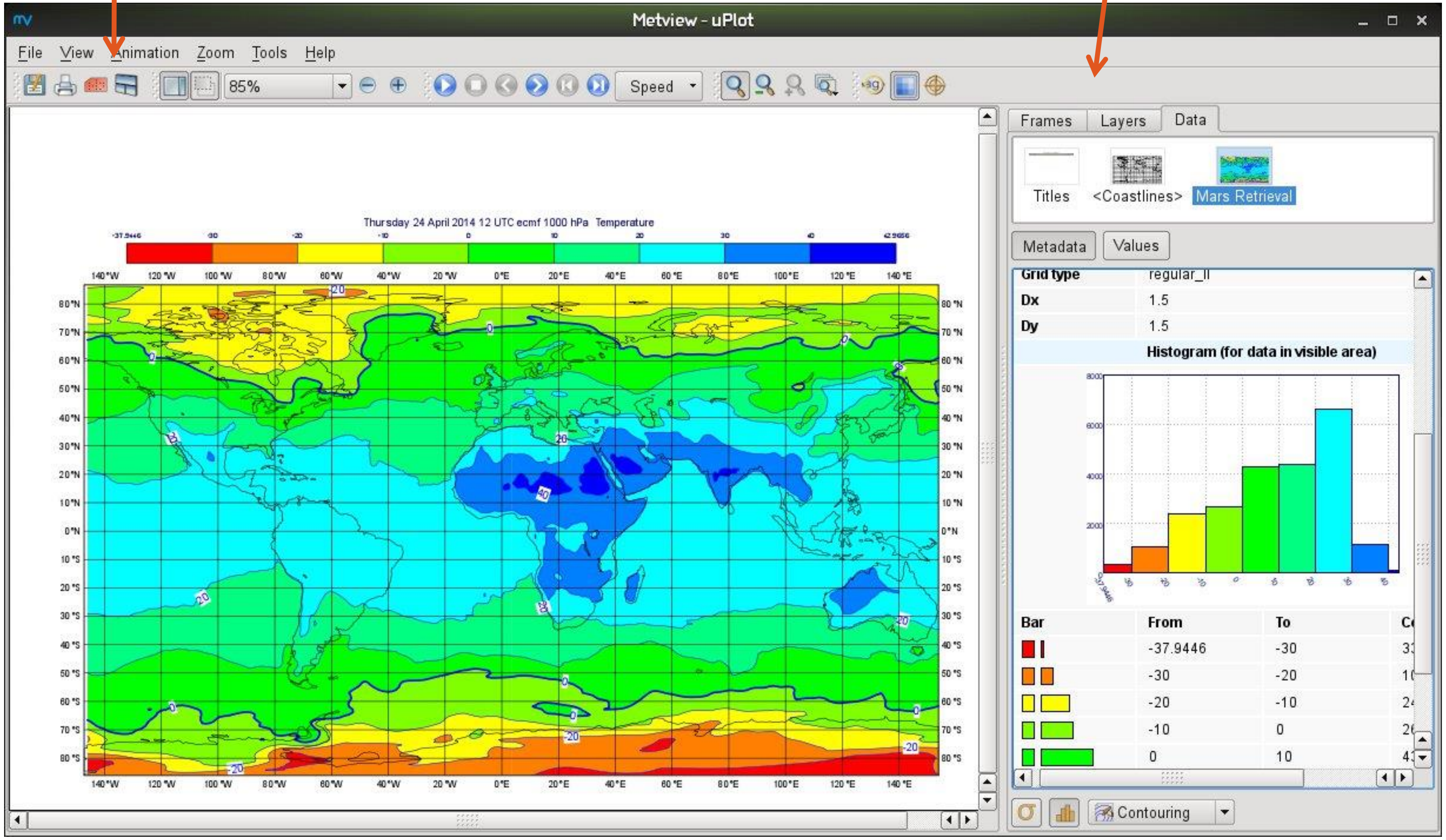
Input element: Toggle option

Input element: Option Menu

Display Window

Controls

Metadata



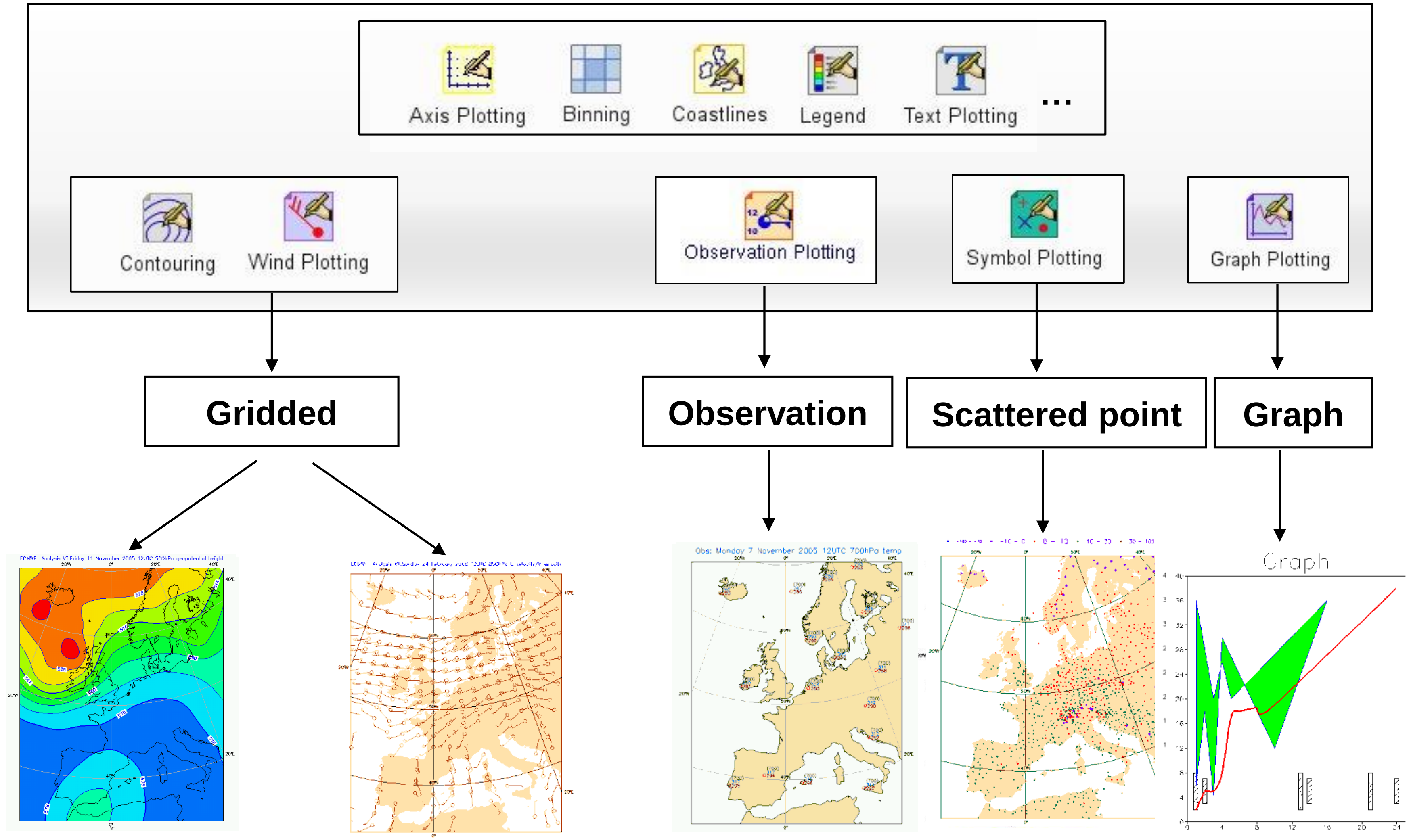
Metview - uPlot
File View Animation Zoom Tools Help
85%
Speed
Thursday 24 April 2014 12 UTC ecmf 1000 hPa Temperature
Frames Layers Data
Titles <Coastlines> Mars Retrieval
Metadata Values
Contouring

Grid type	regular_ll
Dx	1.5
Dy	1.5

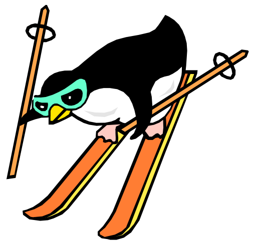
Histogram (for data in visible area)

Bar	From	To	C
🟥	-37.9446	-30	3
🟧	-30	-20	1
🟨	-20	-10	2
🟩	-10	0	2
🟩	0	10	4

Visual Definition (*visdef*)



Desktop Behaviour (1)

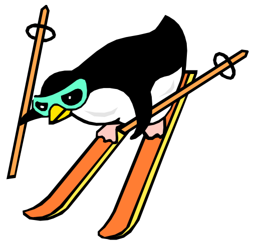


- **KDE settings relevant to Metview: (personal preference)**

1) Change the window behaviour

- **KDE menu (icon at bottom-left)**
- **System Settings**
- **Window behaviour**
- **Window behaviour**
- **Set *Focus stealing prevention level* to "*None*"**
- **Set *Policy* to "Focus Follows Mouse"**
- **Disable *Click raises active window***
- **Apply and close the dialog**

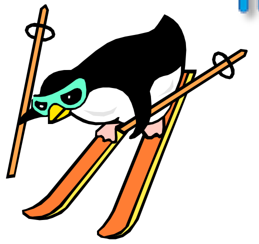
Desktop Behaviour (2)



2) Change the desktop behaviour

- **KDE menu (icon at bottom-left)**
- **System Settings**
- **Desktop**
- **Screen Edges**
- **Disable the settings**
  - ***Maximise windows by dragging…***
  - ***Tile windows by dragging....***
- **Apply and close the dialog**

Starting Metview



- **To start Metview, please type the following command from an *xterm*:**

```
metview_new &
```

- **Please minimise the *xterm* but do not close it**

Metview Tutorial: A Quick Tour

Get the data and icons for the day

- **From a command line type:**

```
~trx/mv_data/get_day_1.sh
```

- **A new folder called “training” will appear in your Metview desktop**
- **A new folder called “day_1” will appear in your “training” folder**
- **Please do exercise “A Quick Tour of Metview” in the provided sub-folder “a quick tour”**

Additional Notes

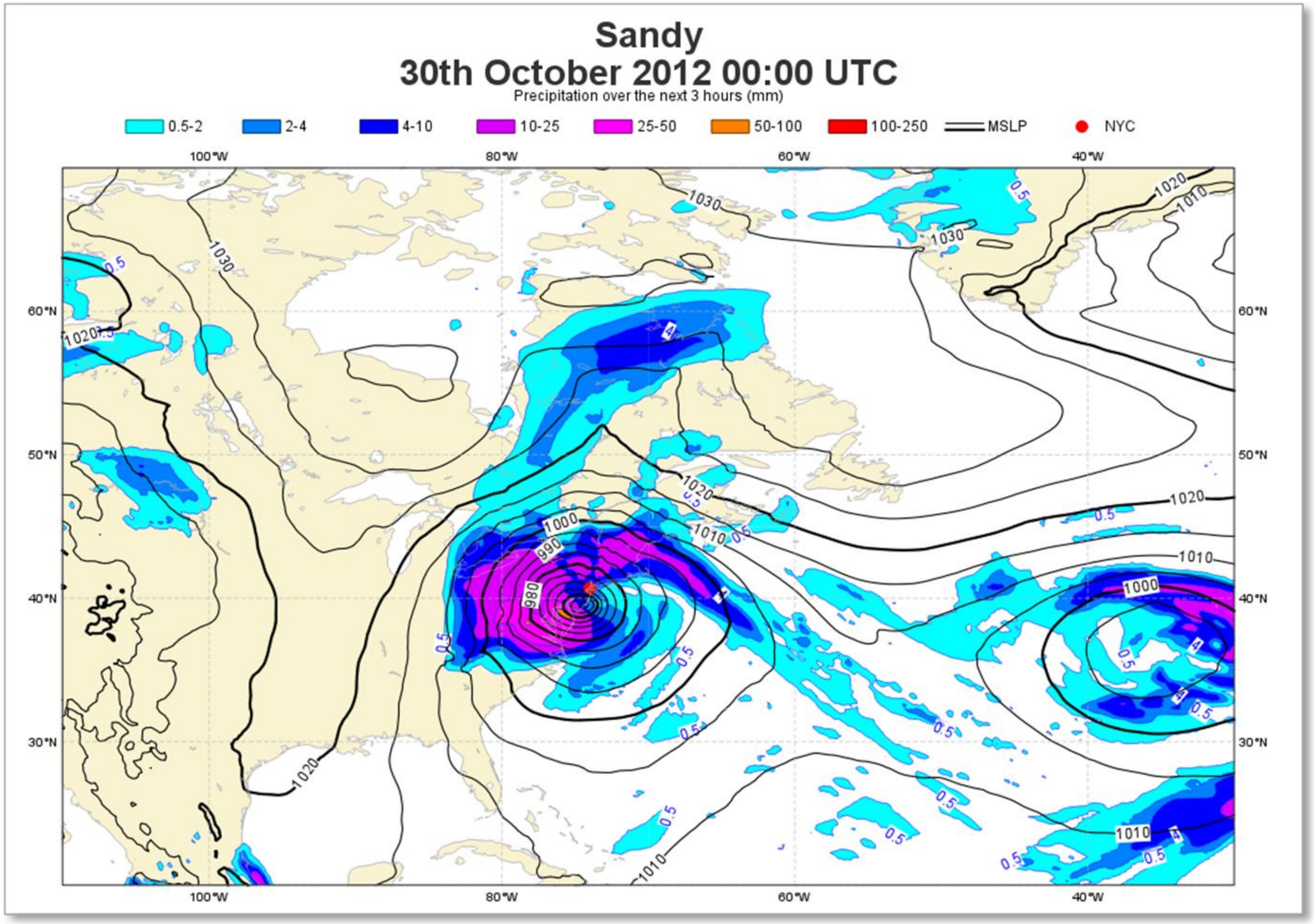
- **Metview scans its open folders for new files every 8 seconds**
- **‘View | Reload’ forces an immediate rescan (F5)**
- **Deleted icons go into the Wastebasket – right-click, Empty to finally delete icons from there**
- **Contouring often has automatic unit conversion – can be deactivated in the *Contour* icon**
- **Cursor data – shows both scaled and non-scaled values**
- **Layer meta-data reflects the selected area**

Case Study: Contouring Hurricane Sandy

Iain Russell

Software Applications Team

Metview Tutorial: Case Study – Contouring Sandy



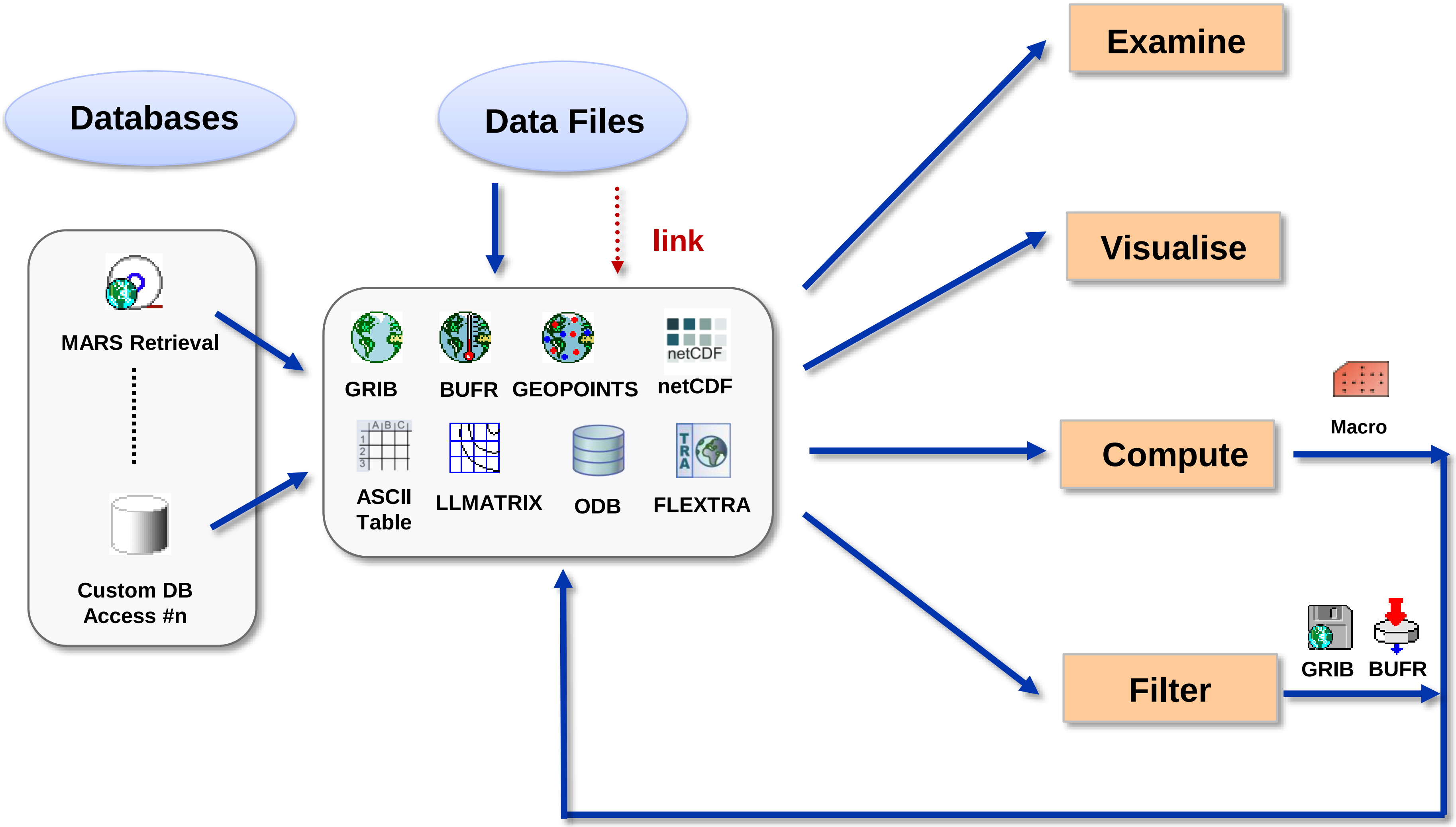
- **Please do exercise “Case Study: Plotting Hurricane Sandy on a Map” in the provided sub-folder “contouring sandy”**

Data in Metview – Part 1

Sándor Kertész

Software Applications Team

Data handling in Metview



Metview Tutorial: Data Part 1

Get the data and icons for the day

- **From a command line type:**

```
~trx/mv_data/get_day_2.sh
```

- **A new folder called “day_2” will appear in your “training” folder**

- **Please do exercise “Data Part 1” in the provided sub-folder “data 1”**

Additional Notes (1)

- **What data is stored in MARS?**
  - **WebMars catalogue: http://www.ecmwf.int/en/forecasts/datasets**

- **MARS access through Metview:**
  - **On ECMWF machines, including ecgate server**
  - **Can set up at local institute**
  - **MARS Web API**

- **MARS language syntax**
  - **List of values: 0/12/24/36/48**
  - **Range of values: 0/TO/48/BY/12**

- **MARS date format**
  - **Specific dates, e.g. 20090303**
  - **Relative dates, e.g. –1 (yesterday)**

Additional Notes (2)

- **Use action save from the icon menu to get a local copy of data files**
- **If an icon goes red, then check the message log (Ctrl-L)**
- **Icons can be input to other icons, thus forming a chain**

Processing data in Metview

Iain Russell

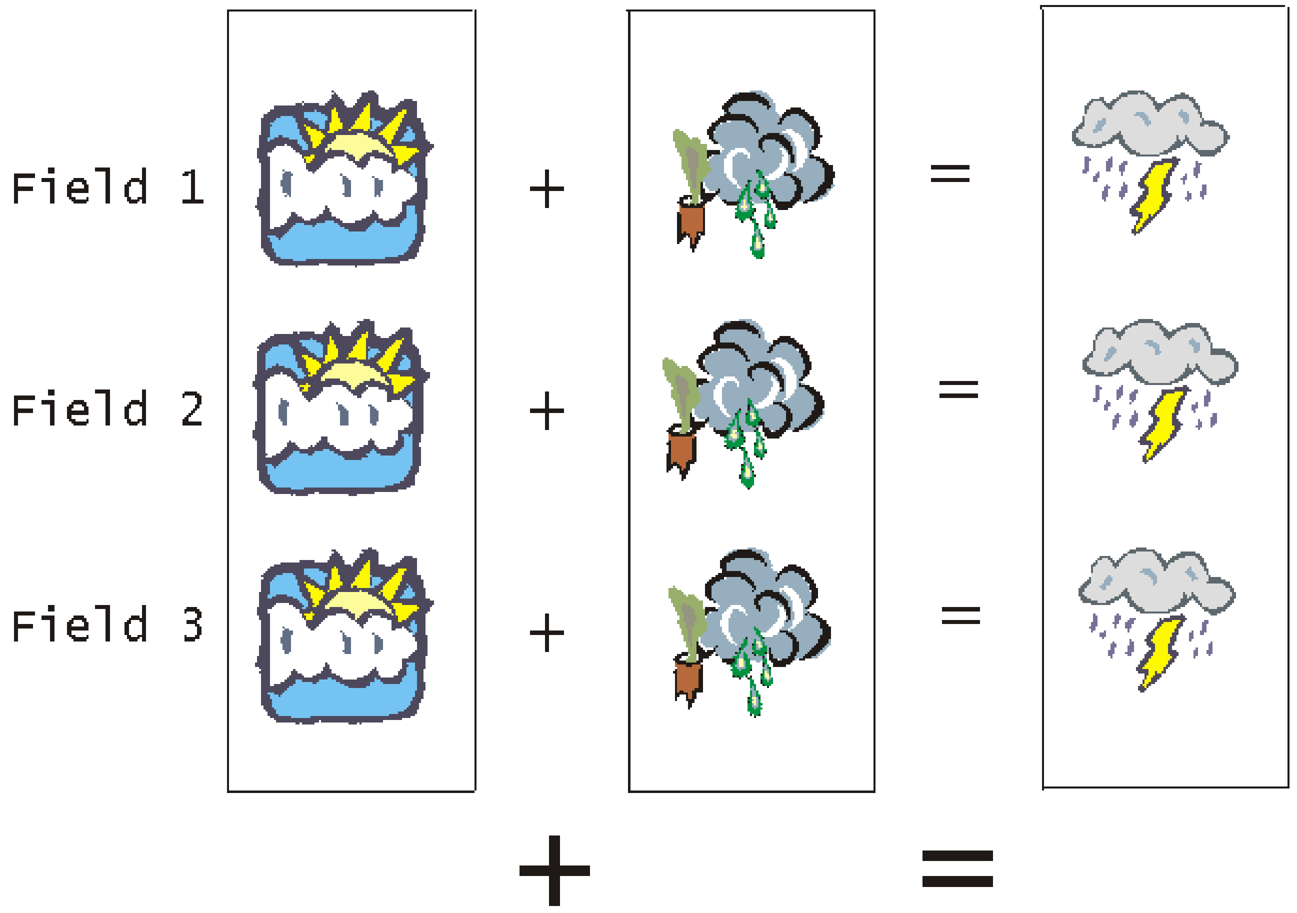
Software Applications Team

Processing Data - Fieldsets

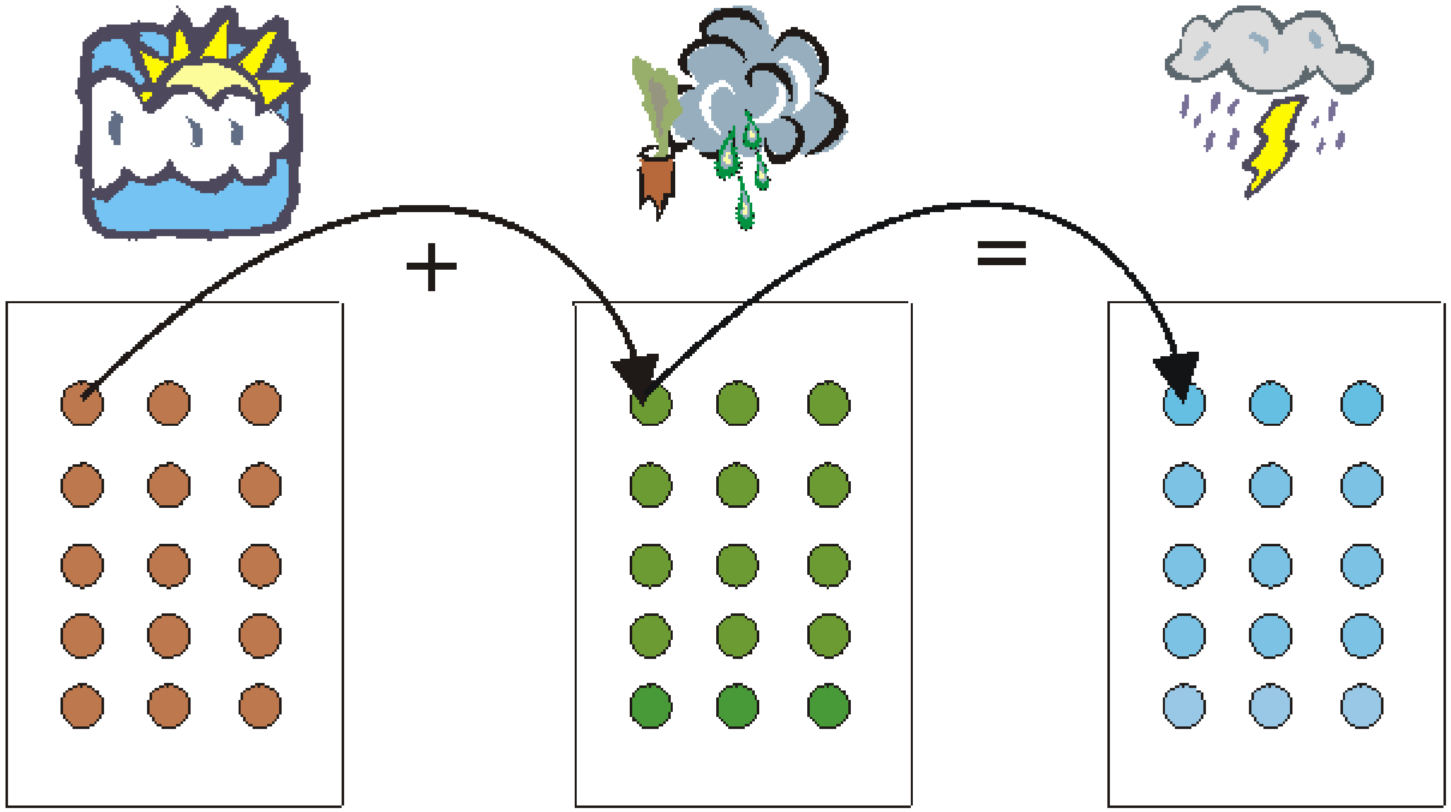
- **Definition**
  - **Entity composed of several meteorological fields, (e.g. output of a MARS retrieval).**
- **Operations and functions on fieldsets**
  - **Operations on two fieldsets are carried out between each pair of corresponding values within each pair of corresponding fields. The result is a new fieldset.**

```
result = fieldset_1 + fieldset_2
```

Processing Data - Fieldsets



Processing Data - Fieldsets

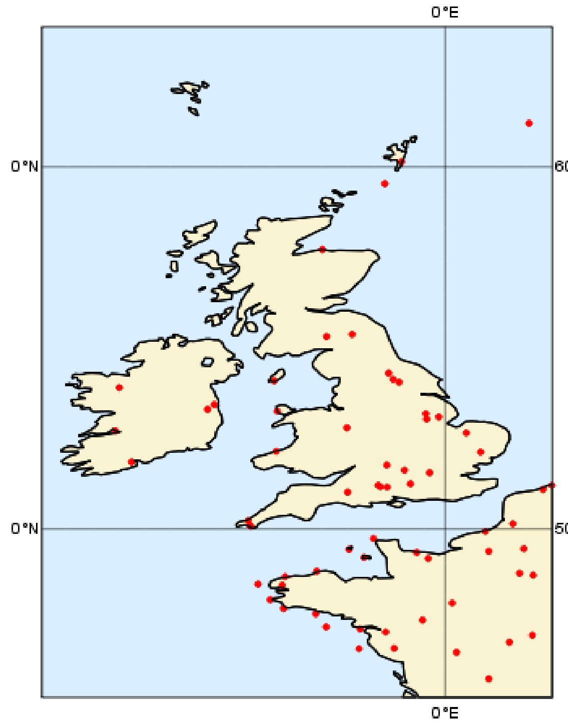


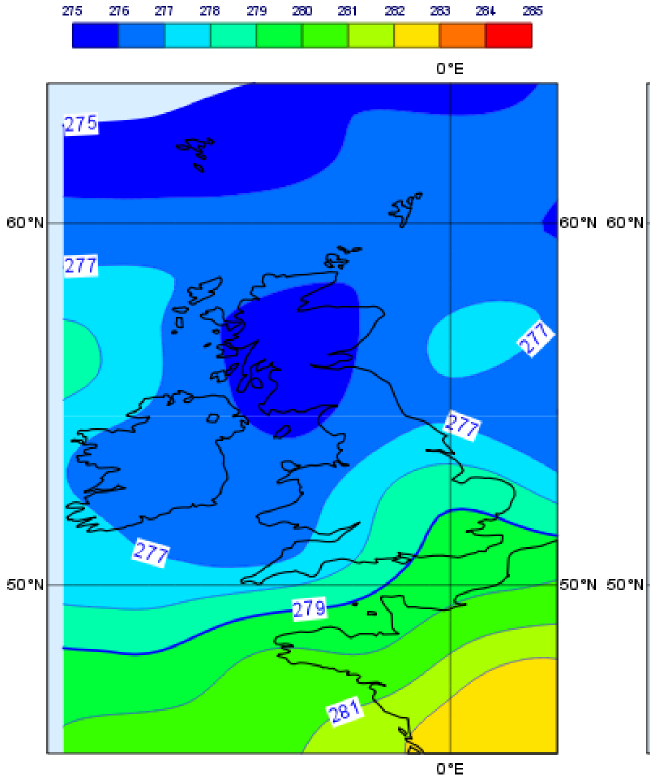
Processing Data - Fieldsets

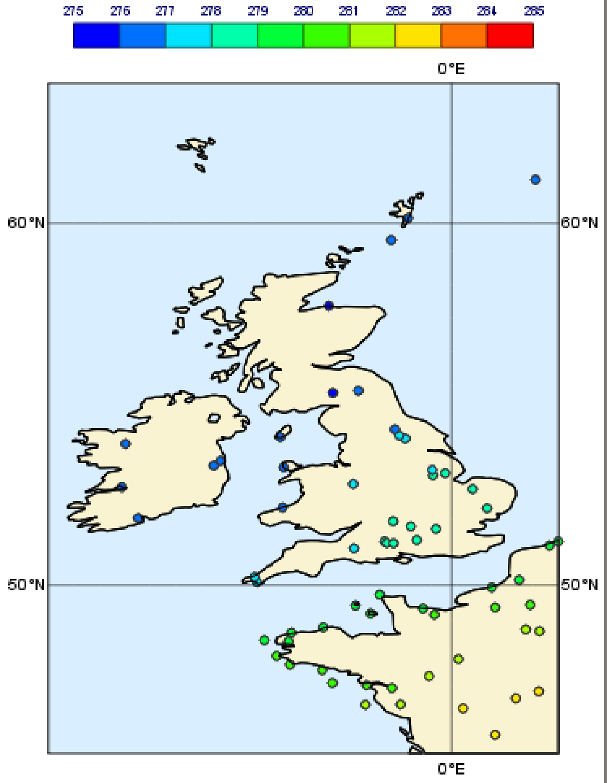
- **Lots more functionality available for fieldsets, for example:**
  - **Interpolation between grids**
  - **Conversion between fields and scattered points**
  - **Extraction of sub-areas**
  - **Computations**
  - **Filtering**

Processing Data – Fieldsets and Geopoints

- **Operations between fields and geopoints**
  - **First, the field values are interpolated onto the geopoints locations**
  - **Then computations are done in 'geopoints space'**







Metview Tutorial: Processing Data

- **Please do “Processing Data” in the provided sub-folder “processing data”**

Additional Notes

- **Extracting fields from fieldsets**
    - **fieldset [number]**
    - **fieldset [number,number]**
    - **fieldset [number,number,number]**
- **Examples :**

```
y = x[2]        # copies field 2 of x into y
y = x[3,8]      # copies fields 3,4,5,6,7 and 8
y = x[1,20,4] # copies fields 1, 5, 9, 13 and 17
```

- **Concatenating fields :**

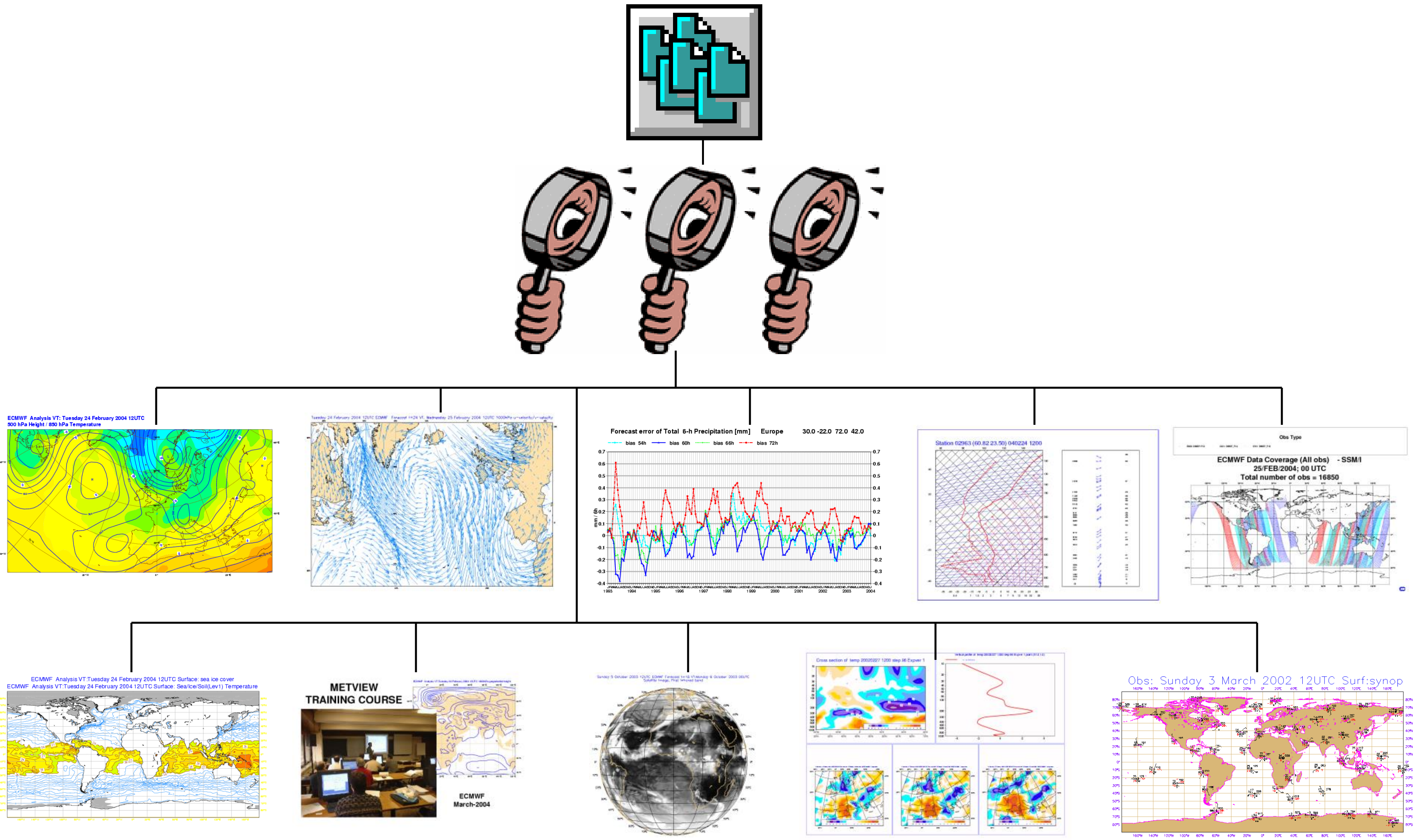
```
a = fs1 & fs2 & fs3
```

Views and layout

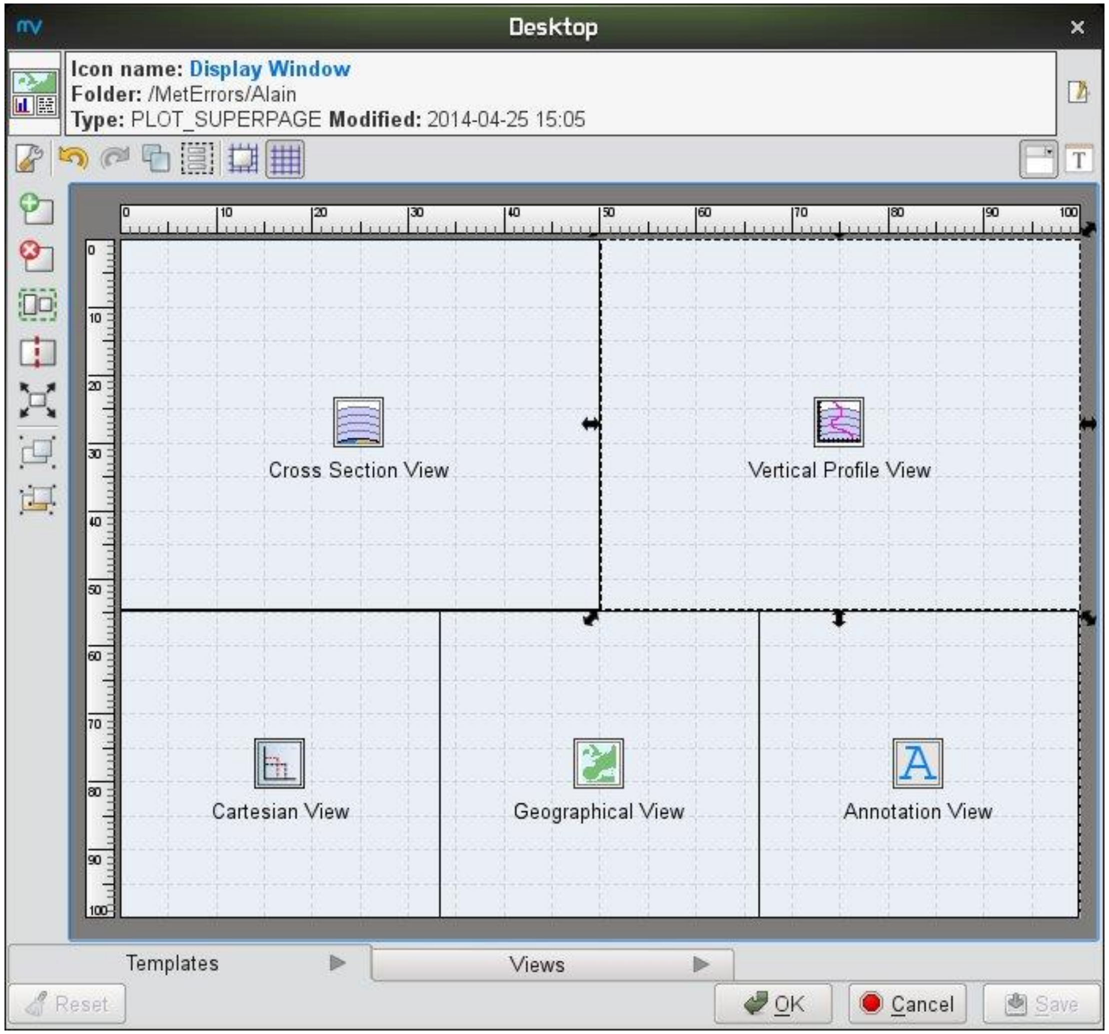
Fernando Ii

Software Applications Team

The VIEW concept


ECMWF Analysis VT: Tuesday 24 February 2004 12UTC
500 hPa Height / 850 hPa Temperature
Forecast error of Total 6-h Precipitation [mm] Europe 30.0 -22.0 72.0 42.0
bias 54h
bias 60h
bias 66h
bias 72h
Station 02963 (60.82 23.50) 040224 1200
Obs Type
ECMWF Data Coverage (All obs) - SSM/I
25/FEB/2004; 00 UTC
Total number of obs = 16850
ECMWF Analysis VT:Tuesday 24 February 2004 12UTC Surface: sea ice cover
ECMWF Analysis VT:Tuesday 24 February 2004 12UTC Surface: Sea/Ice/Soil(Lev1) Temperature
METVIEW
TRAINING COURSE
ECMWF
March-2004
Obs: Sunday 3 March 2002 12UTC Surf:synop

Display Window icon – layout editor



Metview Tutorial: Views and Layout

- **Please do exercise "Analysis Views" in the provided sub-folder "analysis views"**

- **Please do exercise "Layout in Metview", also in the provided sub-folder "analysis views"**

Part 4 – Additional Notes

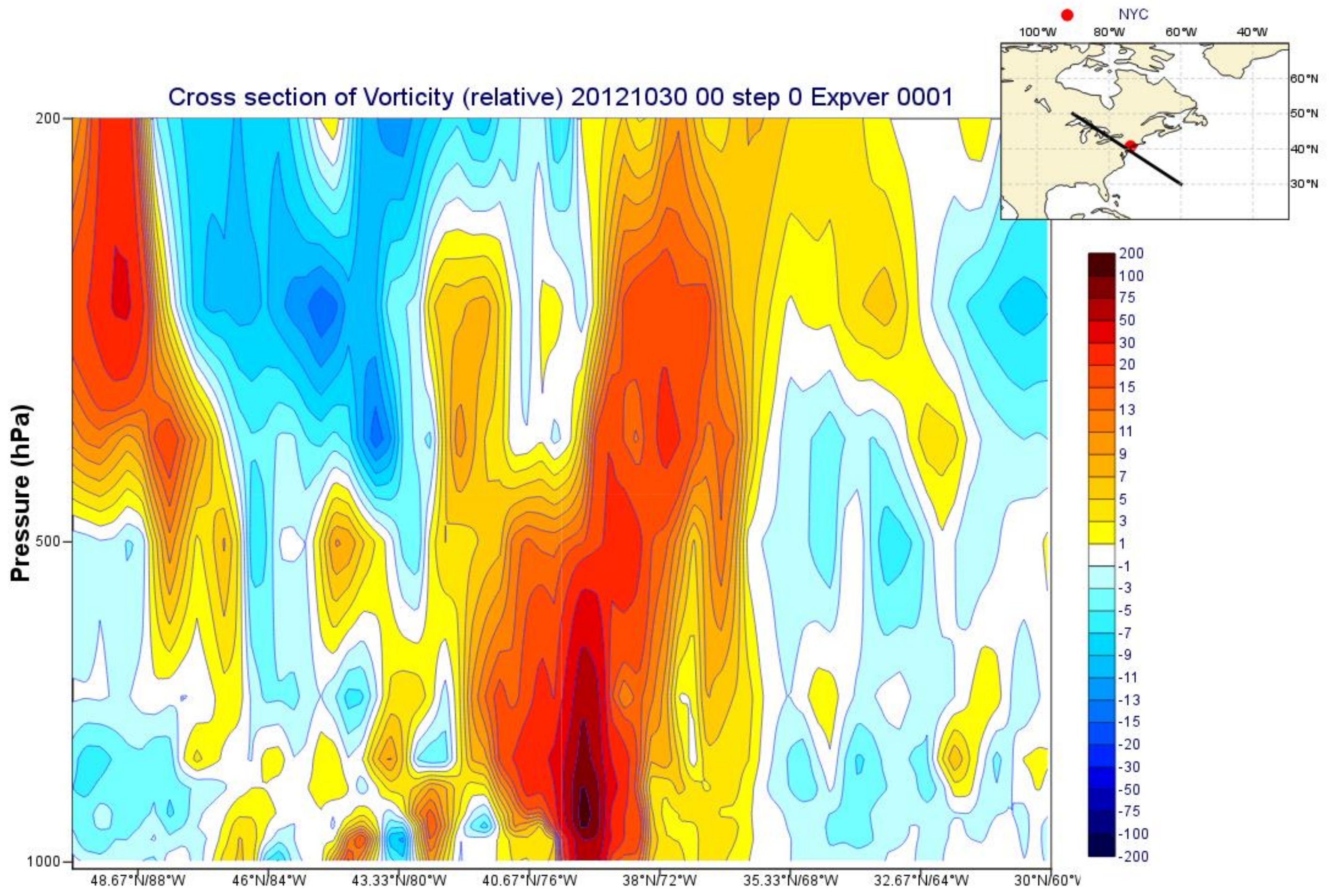
- **Many options are common to all views (position, …)**
- **Axis Plotting icons can be used to modify the look of the axes (e.g. fonts, titles, colours, etc)**

Case study: Cross section of Hurricane Sandy

Iain Russell

Software Applications Team

Metview Tutorial: Case Study – Cross Section of Sandy



- **Please do "Case Study: Cross Section of Sandy" in the provided sub-folder "sandy cross section"**

Data in Metview – Part 2

Iain Russell

Software Applications Team

More Data Formats

- **NetCDF**
  - **Multi-dimensional arrays (matrices, lines, points)**
- **Geopoints**
  - **Metview-specific**
  - **ASCII, scattered geographical data points**
- **ASCII tables**
  - **E.g. CSV – columns of values**
- **Other ASCII**
- **ODB (Observation Database, ECMWF)**

Visualisers

- GRIB is 'easy' to plot
  - Standardised meta-data – geographic coordinates, resolution, etc
- Some other formats (e.g. netCDF) are more versatile and can contain matrices, scattered points, multiple variables, etc
  - users need to tell Metview what to plot

→ visualiser icons

Metview Tutorial: Data Part 2

Get the data and icons for the day

- **From a command line type:**

```
~trx/mv_data/get_day_3.sh
```

- **A new folder called “day_3” will appear in your “training” folder**

- **Please do exercise “Data Part 2” in the provided sub-folder “data 2”**

Additional Notes

- **Note the different plot types available in the Visualiser icons**
  - **Allow a range of ways to interpret and plot data, e.g. geographic, x/y, matrices, vector pairs, …**
  - **Choose the plot type before entering other parameters!**

Handling time in Metview

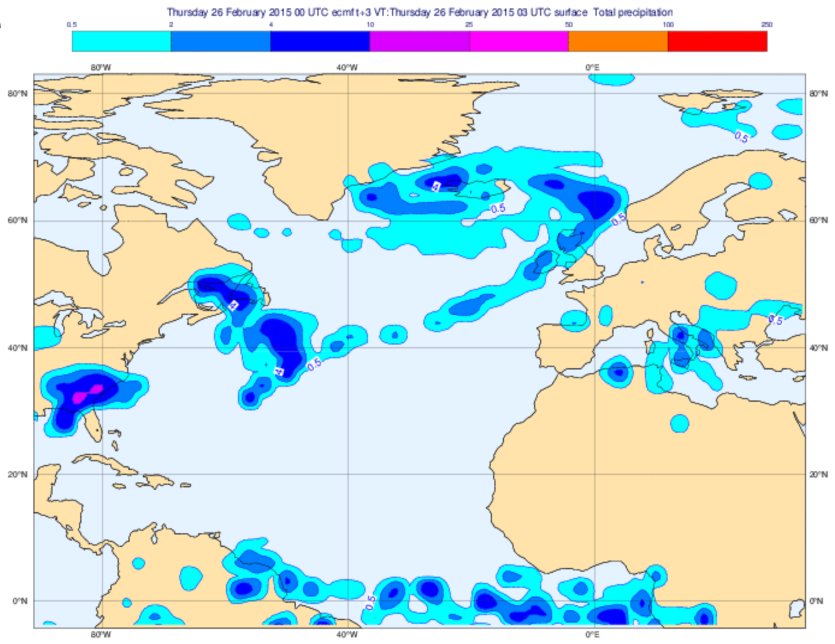
Iain Russell

Software Applications Team

Handling Time in Metview

- **Note that time itself is multi-dimensional!**
  - **Run time (base time) and forecast step**
  - **Run time + step = valid time**
  - **Also analysis data which has step=0**

- **Can extract time from most data types – in Macro , we have the *date* variable type**


Thursday 26 February 2015 00 UTC ecmf t+3 VT:Thursday 26 February 2015 03 UTC surface Total precipitation
0.5
2
4
10
25
50
100
250
80°W
40°W
0°E
80°N
60°N
40°N
20°N
0°N
0.5

Handling Time in Metview

- Dates defined as a built-in type - year, month, day, hour, minute and second.
- Dates can be created as literals using :
  - yyyy-mm-dd
  - yyyy-DDD
  - where : yr, yyyy - 4 digit yr, mm - 2 digit month, dd - 2 digit day, DDD - 3 digit Julian day.
- The time can be added using :
  - HH:MM   or   HH:MM:SS
  - E.g.

```
start_date = 2003-03-20 12:01
```

Handling Time in Metview

- Date arithmetic works with ‘1' being a day

```
d1 = 2015-12-31
d2 = d1 + 1
print (d2)
   2016-01-01 00:00:00
```

Handling Time in Metview

- **Function `date()` creates dates from numbers:**

```
d1        = date(20080129)
today     = date(0)
yesterday = date(-1)
```

- **Hour, minute and second components are zero.**
- **To create a full date, use decimal dates:**

```
d = date(20080129.5)
```

or

```
d = 2008-01-29 + 0.5
```

or

```
d = 2008-01-29 + hour(12)
```

Handling Time in Metview

- **Note that numbers passed to Metview modules are automatically converted to dates:**

```
r = retrieve(date : -1, ...)

r = retrieve(date : 20070101, ...)
```

Handling Time in Metview

- **Loops on dates using a for loop:**

```
 for d = 2007-01-01 to 2007-03-01 do
      ... # each step is 1 day
 end for

for d = 2007-01-01 to 2007-03-01 by 2 do
      ... # each step is 2 days
end for

for d = 2007-01-01 to 2007-03-01 by hour(6) do
      print(d)
       ... # each step is 6 hours
end for
```

Data Overlay

- **Multi-data visualisations, e.g. T+Z,…**
  - **When are different data overlaid in the same plot?**
- **Default data overlay rules**
- **Need more control? – Use the Data Overlay Setting**

Metview Tutorial: Handling Time

- **Please do exercise “Handling Time in Metview” in the provided sub-folder “time”**

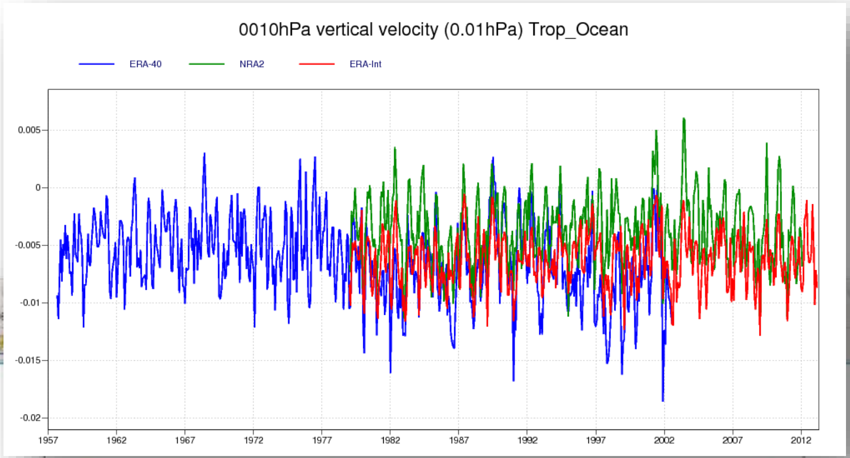
Graph Plotting

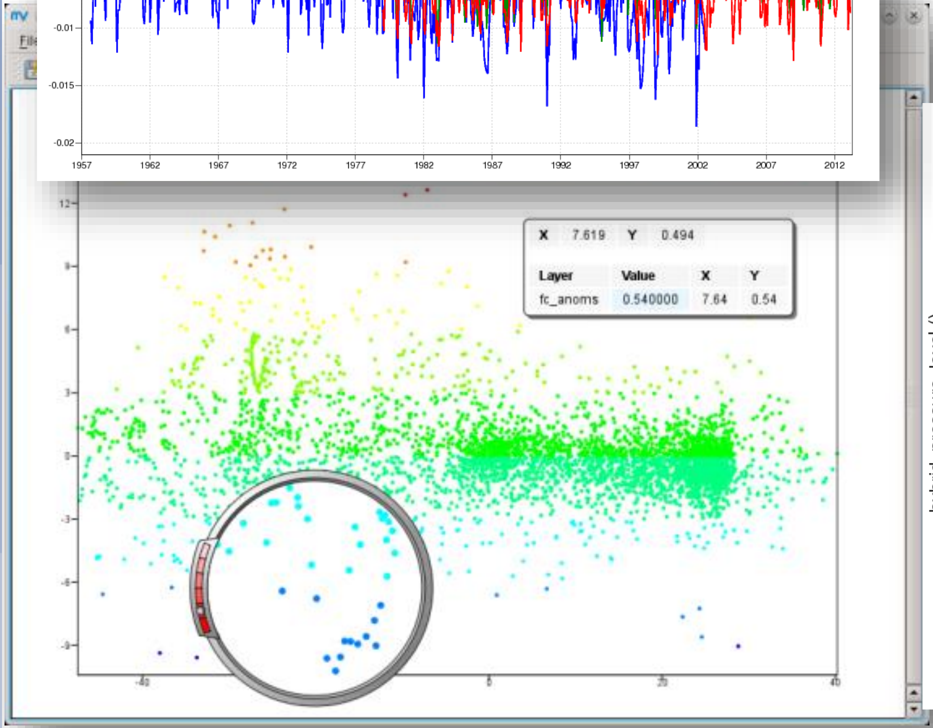
Iain Russell

Software Applications Team

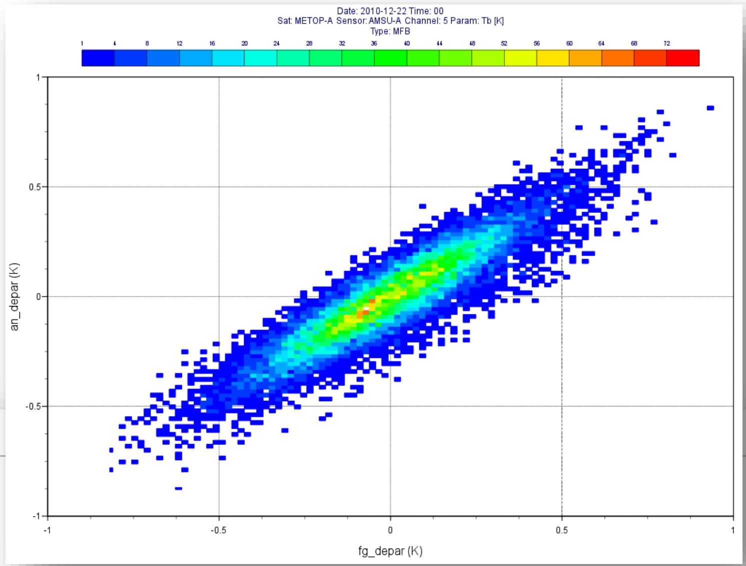
Graph Plotting

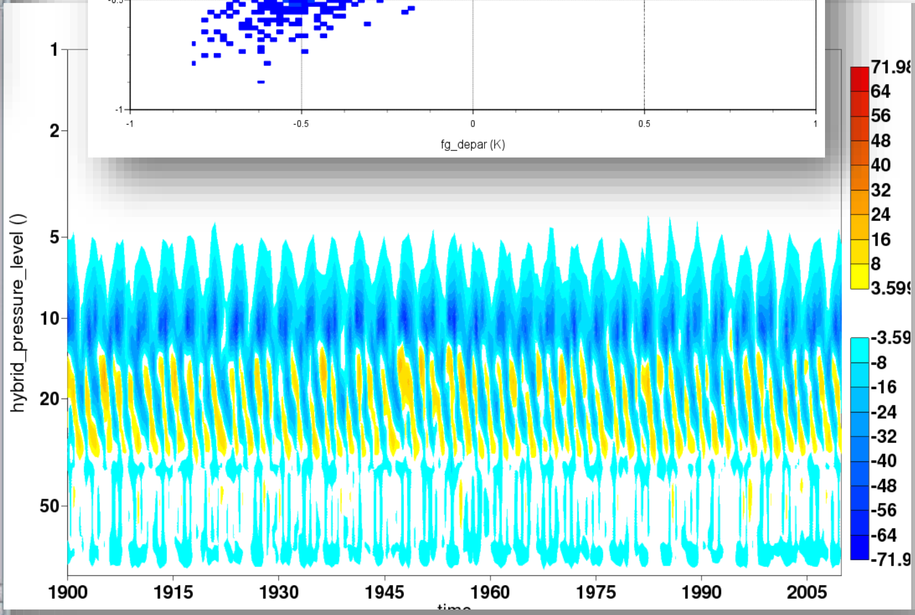
- **It's not only geographical plots!**




X 7.619 Y 0.494

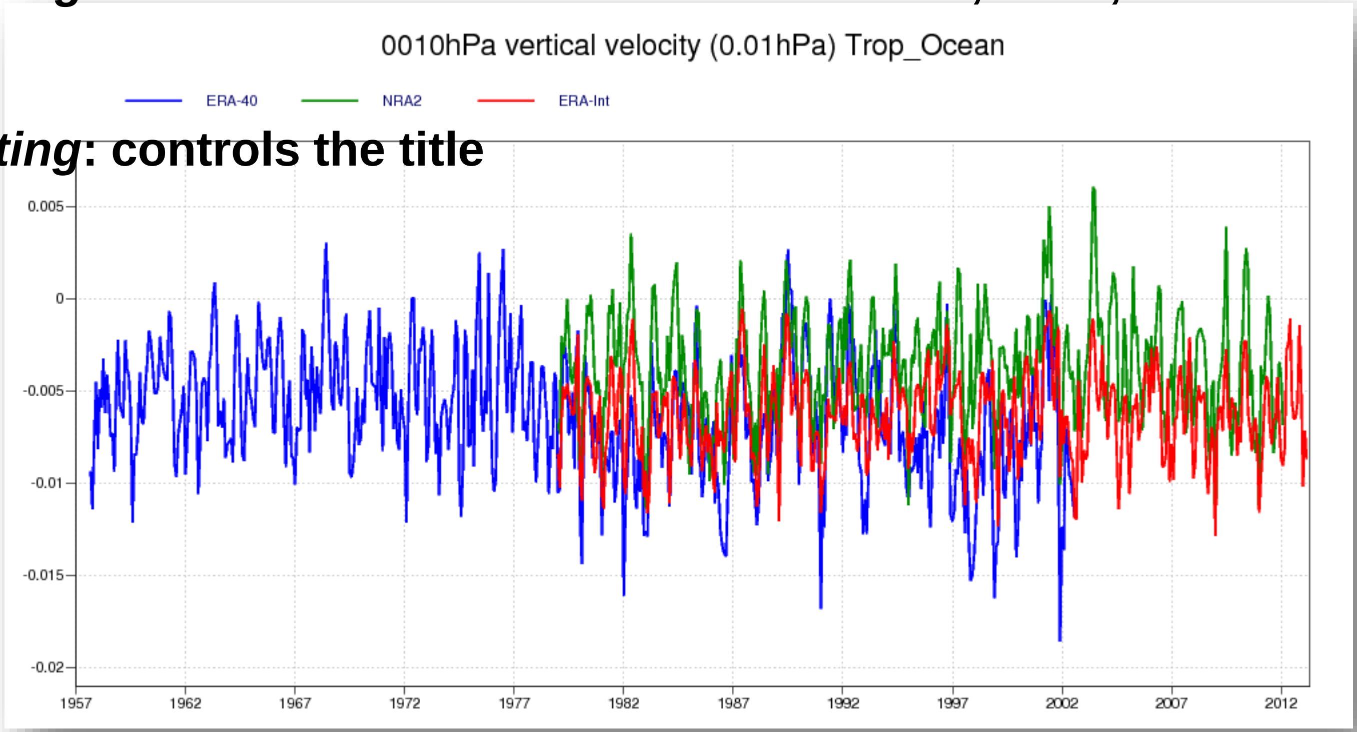
Layer	Value	X	Y
fc_anoms	0.540000	7.64	0.54





Graph Plotting: The Elements

- **Graph plots consist of:**
  - ***Cartesian View*: defines the coordinate system**
  - **Data!**
  - **Visdefs for the data**
  - ***Axis Plotting*: defines the look of the axes – colours, titles, etc.**
  - ***Legend***
  - ***Text Plotting*: controls the title**

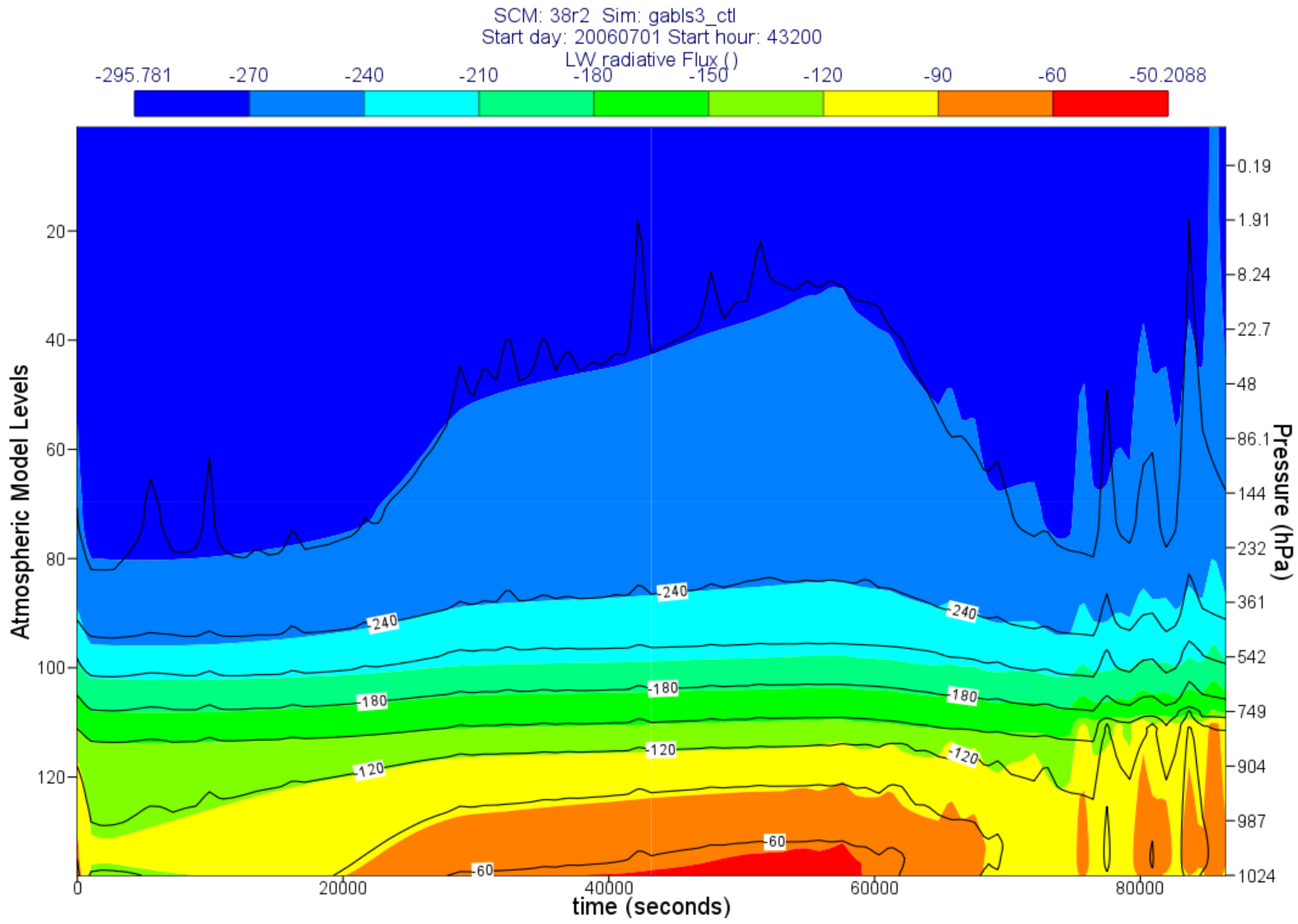


Metview Tutorial: Graph Plotting

- **Please do exercise “Graph Plotting in Metview” in the provided sub-folder “graph”**

Additional Notes

- **It is possible to use two Cartesian View icons to produce a ‘double’ axis**

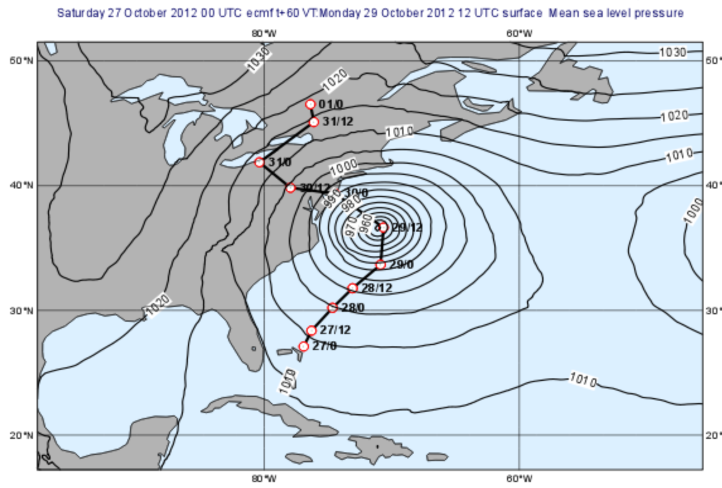


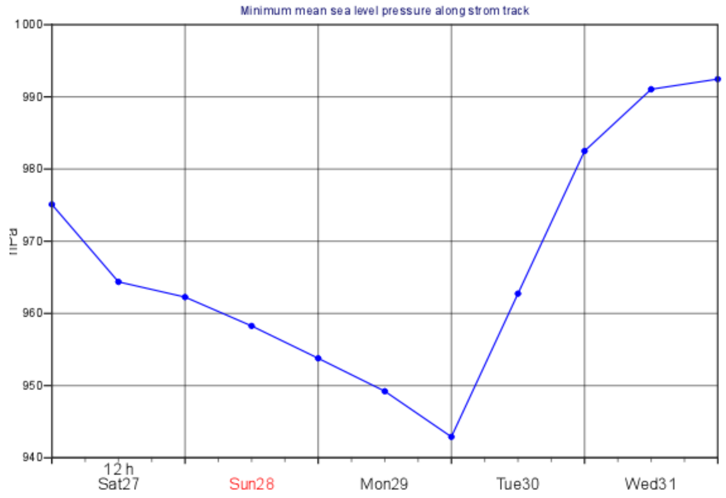
Case Study: Plotting the track of Hurricane Sandy

Sándor Kertész

Software Applications Team

Metview Tutorial: Case Study – Plotting the track of Hurricane Sandy


Saturday 27 October 2012 00 UTC ecmf t+60 VT:Monday 29 October 2012 12 UTC surface Mean sea level pressure
01/0
31/12
31/0
30/12
30/0
29/12
29/0
28/12
28/0
27/12
27/0
80°W
60°W
50°N
40°N
30°N
20°N
1030
1020
1010
1000



- **Please do exercise "Case Study: Plotting the track of Hurricane Sandy" in the provided sub-folder "sandy track"**

Publishing graphical output

Stephan Siemen

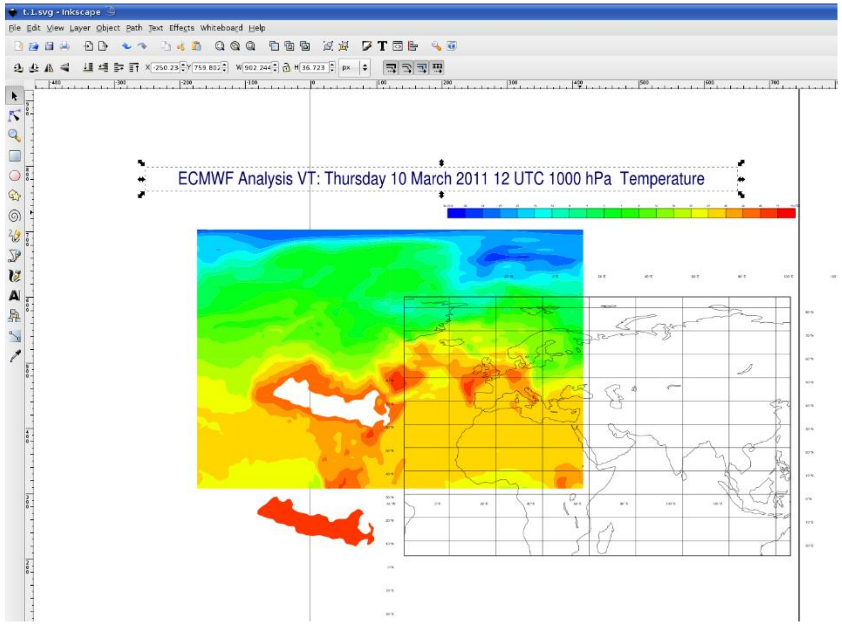
Magics developer

The role of Magics in Metview

- **Magics is ECMWF's plotting package**
  - **Tuned to handle meteorological data**
  - **GRIB, BUFR, ODB, CSV**
  - **APIs: Python, Fortran and Metview**
  - **Used also for the web (ecCharts, Metgrams)**
- **Everything graphical in Metview is done through Magics**

Output formats

- **PostScript & EPS**
  - **Vector format for printing**
- **PDF**
  - **Static vector format for web & archive**
- **PNG**
  - **Raster format for web**
- **SVG**
  - **Vector format for web**
- **KML/KMZ**
  - **Format for Google Earth & Maps**
- **Qt**
  - **Used by Metview only**


t.1.svg - Inkscape
File Edit View Layer Object Path Text Effects Whiteboard Help
ECMWF Analysis VT: Thursday 10 March 2011 12 UTC 1000 hPa Temperature

Metview Tutorial: Graphical Output Formats

Get the data and icons for the day

- **From a command line type:**

```
~trx/mv_data/get_day_4.sh
```

- **A new folder called “day_4” will appear in your “training” folder**

- **Please do exercise “Working with graphical output” in the provided sub-folder “graphics formats”**

Macro in-depth

Iain Russell

Software Applications Team

Macro Essentials - Variables

- **No need for declaration**
- **Dynamic typing**

```
a = 1          # type(a) = 'number'
a = 'hello'    # type(a) = 'string'
a = [4, 5]     # type(a) = 'list'
a = |7, 8|     # type(a) = 'vector'
```

Macro Essentials - Variables

- **Scope and Visibility**
  - **Variables inside functions are local**
- **Functions cannot see ‘outside’ variables**

```
x = 9                # cannot see y here

function func

     y = 10          # cannot see x here

end func

                     # cannot see y here
```

Macro Essentials - Variables

- **Scope and Visibility**
  - **… unless a variable is defined to be ‘global’**

```
global g1 = 9              # cannot see y1 here
function func
     y1 = 10 + g1          # can see g1 here
end func
                           # cannot see y1 here
```

Macro Essentials - Variables

- **Scope and Visibility**
  - **… a better solution is to pass a parameter**
  - **… that way, the function can be reused in other macros**

```
x = 9


func(x)      # x is passed as a parameter


function func (t : number) #t adopts value of x
    y1 = 10 + t            # y1 = 10 + 9
end func
```

Macro Essentials - Variables

- **Destroying variables automatically**
  - **When they go out of scope**

```
function plot_a
    a = retrieve(...)
    plot(a)
end plot_a


# Main routine
plot_a()   # a is created and destroyed
```

Macro Essentials - Variables

- **Destroying variables manually**
  - **Set to zero**
  - **(Variables can ‘hold’ lots of data, either in memory or in temporary files)**

```
a = retrieve(...)
plot(a) # we have finished with ’a’ now
a = 0
b = retrieve(...)
plot(b)
```

Macro Essentials - Strings

- `'Hello'` is the same as `"Hello"`

- Concatenate strings with strings, numbers and dates using the `'&'` operator

  eg. `"part1_" & "part2_" & 3`

  produces `"part1_part2_3"`

- Obtain substrings with `substring()`

  e.g. `substring ("Metview", 2, 4)`

  produces `"etv"`

  first last

Macro Essentials - Strings

- **Split a string into parts using `parse()`**
- **Creates a list of substrings**

```
n = parse("z500.grib", ".")
print ("name = ", n[1], " extension = ",
n[2])
```

- **prints the following string :**

```
name = z500 extension = grib
```

Macro Essentials - Lists

- **Ordered, heterogeneous collection of values. Not limited in length. List elements can be of any type, including lists. List are built using square brackets, and can be initialised with `nil`:**

```
l = [3,4,"foo","bar"]


l = nil
l = l & [2,3,[3,4]]
l = l & ["str1"] & ["str2"]
europe = [35,-12.5,75,42.5]    # S, W, N, E
```

Macro Essentials - Lists

- **Accessing List Elements**
- **Indexes start at 1**

```
mylist = [10,20,30,40]
a = mylist[1]      # a = 10
b = mylist[2,4]    # b = [20,30,40] (m to n)
c = mylist[1,4,2] # c = [10,30] (step 2)
```

Macro Essentials - Lists

- **Useful List Functions**

```
num_elements = count (mylist)
sorted = sort (mylist)
  # can provide custom sorting function

 if (2 in mylist) then
    …
 end if
```

Macro Essentials - Lists

- **Useful List Functions**

```
mylist = ['b', 'a', 'a', 'c']

# find occurrences of 'a' in list
 index   = find(mylist, 'a') # 2
 indexes = find(mylist, 'a', 'all') # [2,3]

 # return list of unique members
 reduced = unique(mylist) # ['b', 'a', 'c']
```

Macro Essentials - Lists

- **List Operations**
- **Operators acting on lists will act on each list element, returning a list of results**

-
  ```
  a = [3, 4]
  b = a + 5   # b is now [8, 9]
  c = a * b   # c is now [24, 36]
  ```
- **Lists are general-purpose, and are not recommended for handling large amounts (thousands) of numbers – for that, use *vectors* (see later)**

Macro Essentials - Vectors

- **Ordered, array of numbers. Much more efficient than lists for high volumes of numeric data. Vectors are built using the vertical bar symbol, and can be initialised with `nil`:**

```
v = |7, 8, 9|


v = nil # start from nil and append
v = v & |4.4, 5.5, 3.14| & |8, 9|


v = vector(10000) # pre-allocate space
v[1] = 4 # assign values to indexes
```

Macro Essentials - Vectors

- **Can extract data arrays from most data types into vector variables:**

```
v = values(fieldset)
v = values(geopoints)
v = values(netcdf) # takes from current variable
v = values(table, ‘col2’)
v = values(odb, ‘precip’)
```

Macro Essentials - Vectors

- **Assigning/replacing a range of values at once:**

```
v = |10,20,30,40|
v[2] = |99,99| # v is now |10,99,99,40|
```

Macro Essentials - Vectors

- **Operations and functions are applied to each element:**

```
x = |3, 4, 5|
y = x + 10 # y is now |13, 14, 15|

c = cos(x)

u = |7.3, 4.2, 3.6|
v = |-4.4, 1.1, -2.1|
spd = sqrt((u*u) + (v*v))
```

Macro Essentials - Vectors

- **Accessing vector elements**
- **Indexes start at 1**

```
v = |10,20,30,40|
a = v[1]         # a = 10
b = v[2,4]       # b = |20,30,40| (m to n)
c = v[1,4,2]     # c = |10,30| (step 2)
d = v[1,4,2,2]   # d = |10,20,30,40|
                 # (take 2 at each step)
```

Macro Essentials – Loops, Tests & Functions

- **The for, while, repeat, loop statements**
  - **See ‘Metview Macro Syntax’ handout**
- **The if/else, when, case statements**
  - **See ‘Metview Macro Syntax’ handout**
- **Function declarations**
  - **See ‘Metview Macro Syntax’ handout**

Macro Essentials - Functions

- **Multiple versions**
  - **Can declare multiple functions with the same name, but with different parameter number/types.**

    ```
    function fn_test ()

    function fn_test (param1: string)

    function fn_test (param1: number)
    ```

  - **Correct one will be chosen according to the supplied parameters**

Macro Essentials - Definitions

- **A collection of named items (members)**
- **Eg**

```
a = (x : 1, y : 2) # create definition

c = a.x            # get value of ’x’
  or
c = a[”x”]
```

- **Like a struct in ‘C’ or a dictionary in Python**

Macro Essentials - Definitions

- **Icon-functions take definitions:**

```
acoast = mcoast(
     map_coastline_resolution        :    "high",
     map_coastline_colour            :    "red",
     map_grid_colour                 :    "grey",
     map_grid_longitude_increment    :    10,
     map_label_colour                :    "grey",
     map_coastline_land_shade        :    "on",
     map_coastline_land_shade_colour:     "cream"
     )
```

Macro Essentials - Definitions

```
param_def = ( param : "Z",
              type  : "FC",
              date  : -1,
              step  : 24 )

# retrieve as LL grid or not according to user
# choice
if (use_LL = "yes") then
  param_def.grid = [1.5,1.5]
end if

Z_ret = retrieve (param_def)
```

Macro Essentials - Definitions

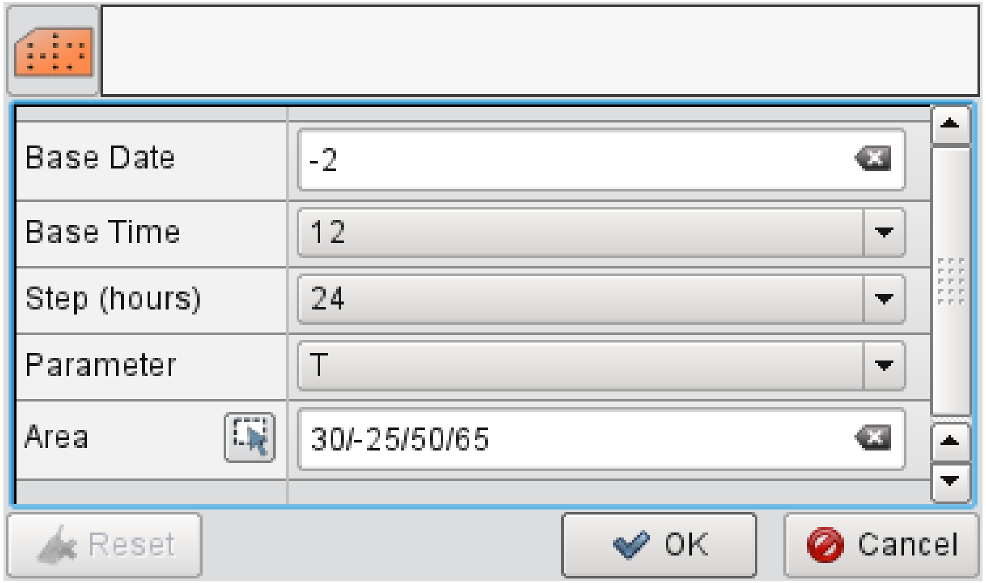
```
common_input = ( levtype : "PL",
                 levelist : 850,
                 time : 12,
                 grid : [2.5,2.5],
                 type : "AN" )

Uan = retrieve ( common_input,
                 date  : -1,
                 param : "U" )

Van = retrieve ( common_input,
                 date  : -2,
                 param : "V" )
```

Fortran and C in Macro - Introduction

- Users can write their own Macro functions in Fortran or C/C++, extending the Macro language
- Used in tasks which cannot be achieved by macro functions. Or use existing FORTRAN/C code to save time.
- FORTRAN/C-Metview macro interfaces support input data of types GRIB, number, string and vector. BUFR, images and matrices are waiting implementation.
- See examples in solutions folder

User Interfaces in Macro - Introduction

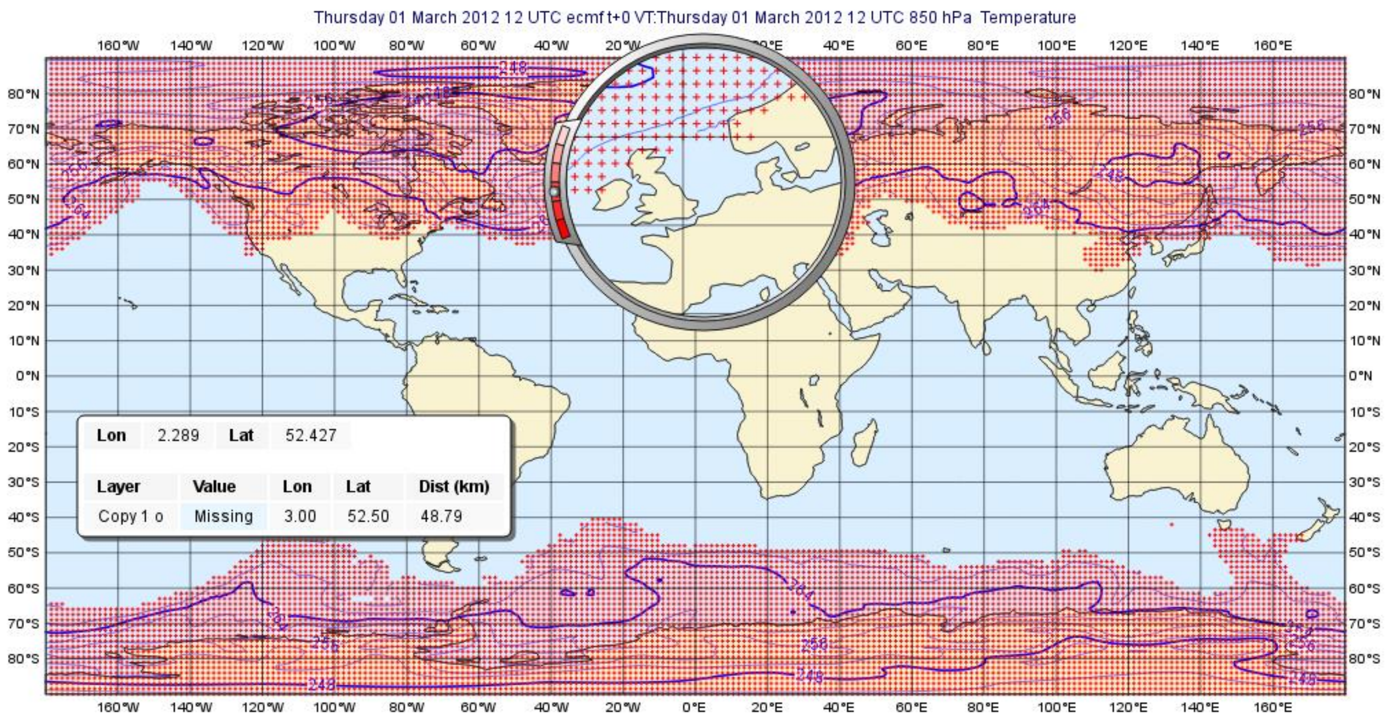
- **Users can write their own simple user interfaces in Macro**
- **One way to allow other users to call a macro and change its parameters without modifying code or passing command-line arguments**
- **See example in solutions folder**



Base Date	-2
Base Time	12
Step (hours)	24
Parameter	T
Area	30/-25/50/65

Missing Values

- **Fields and other data can have missing values**
- **Be aware of this!**
- **They can also be used to mask data, returning a specific subset of points**


Thursday 01 March 2012 12 UTC ecmf t+0 VT:Thursday 01 March 2012 12 UTC 850 hPa Temperature

Lon	2.289	Lat	52.427	
Layer	**Value**	**Lon**	**Lat**	**Dist (km)**
Copy 1 o	Missing	3.00	52.50	48.79

Metview Tutorial: Macro

- **Please do exercise “Organising Macros” in the provided sub-folder “organising macros”**
- **Please do exercise “Missing Values and Masks” in the provided sub-folder “missing values”**

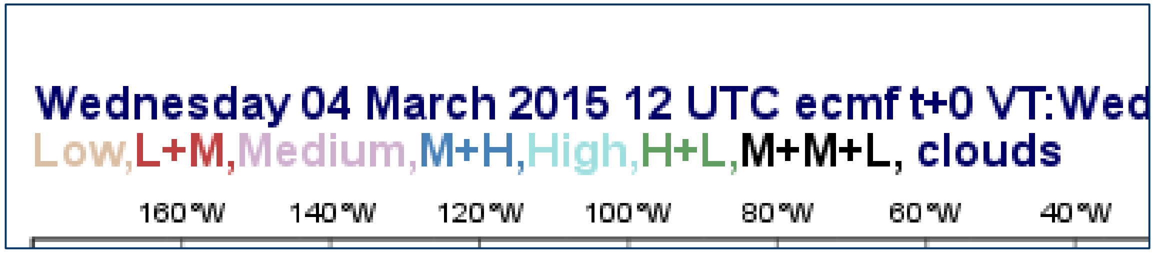
Titles and workflow

Iain Russell

Software Applications Team

Titles

- Automatic text for fields
- User-defined text
  - With or without elements extracted from the data
  - Styled using HTML notation


Wednesday 04 March 2015 12 UTC ecmf t+0 VT:Wed
Low,L+M,Medium,M+H,High,H+L,M+M+L, clouds

Optimising Your Workflow

- **How to store icons for easy re-use**
- **Changing the defaults**

Metview Tutorial: Titles and Workflow

- **Please do exercise “Optimising Your Workflow” in the provided sub-folder “workflow”**

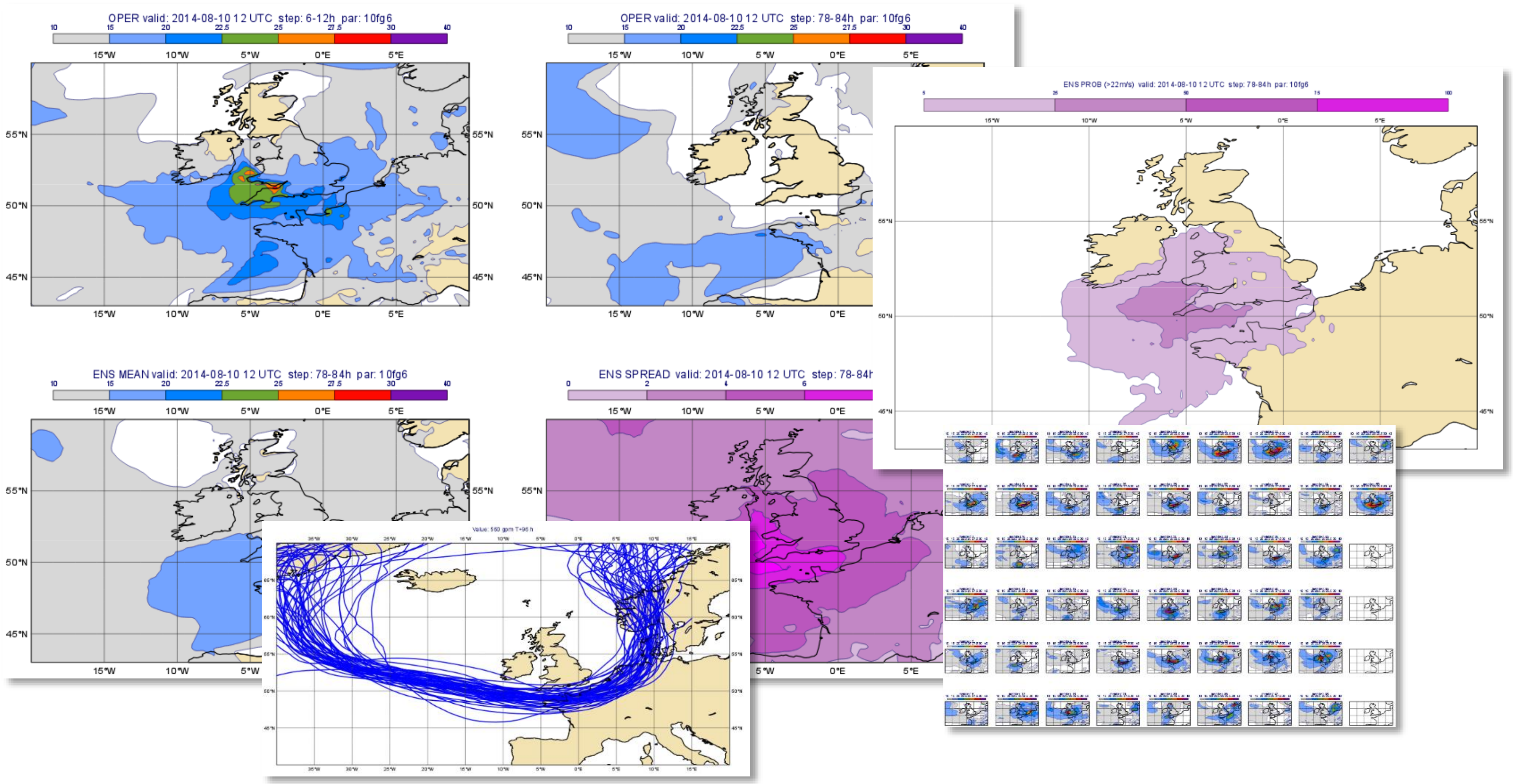
- **Please do exercise “Customising Your Plot Title” in the provided sub-folder “titles”**

Case Study: Ensemble forecast

Sándor Kertész

Software Applications Team

Metview Tutorial: Case Study – Ensemble Forecast


OPER valid: 2014-08-10 12 UTC step: 6-12h par: 10fg6
OPER valid: 2014-08-10 12 UTC step: 78-84h par: 10fg6
ENS PROB (>22m/s) valid: 2014-08-10 12 UTC step: 78-84h par: 10fg6
ENS MEAN valid: 2014-08-10 12 UTC step: 78-84h par: 10fg6
ENS SPREAD valid: 2014-08-10 12 UTC step: 78-84h
Value: 560 gpm T+96 h
15°W
10°W
5°W
0°E
5°E
55°N
50°N
45°N

- **Please do exercise "Case Study: Ensemble Forecast" in the provided sub-folder "ensemble forecast"**

Running Metview in batch mode

Iain Russell

Software Applications Team

Running Metview in Batch Mode

- **Can run macros directly from the command line:**
  - **metview –b <macro_name> [arg1 arg2 …]**

- **Arguments passed after the name of the macro to run**
- **Can also get environment variables**

- **The macro can detect whether it was run from the command line or the user interface**
  - **Can do different things in each case**

Running Metview in Batch Mode

Get the data and icons for the day

- **From a command line type:**

```
~trx/mv_data/get_day_5.sh
```

- **A new folder called “day_5” will appear in your “training” folder**

- **Please do exercise “Running Metview in Batch Mode” in the provided sub-folder “batch”**

Submitting Batch jobs on ecgate (separate presentation)

Xavi Abellan

User Support

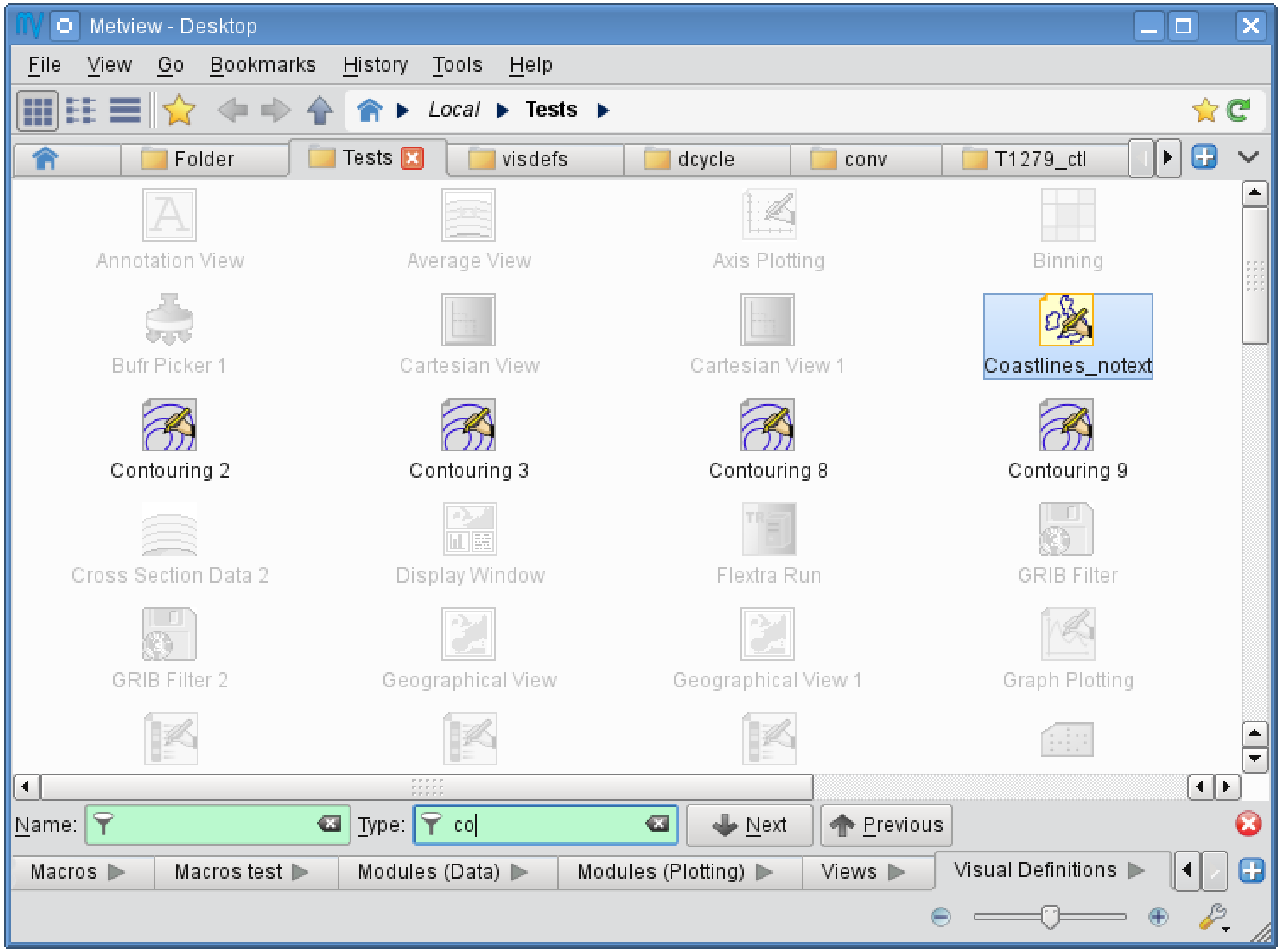
Exploring Metview

Iain Russell

Software Applications Team

Exploring Metview (1)

- **User interface**
  - **More advanced features**

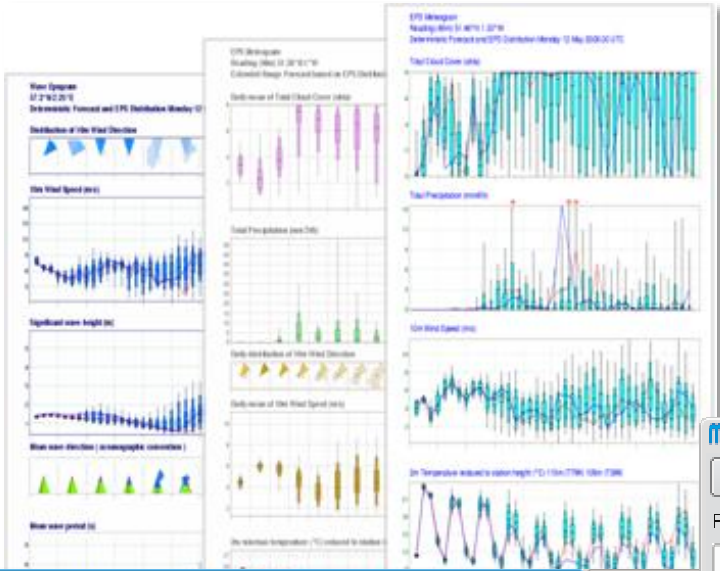


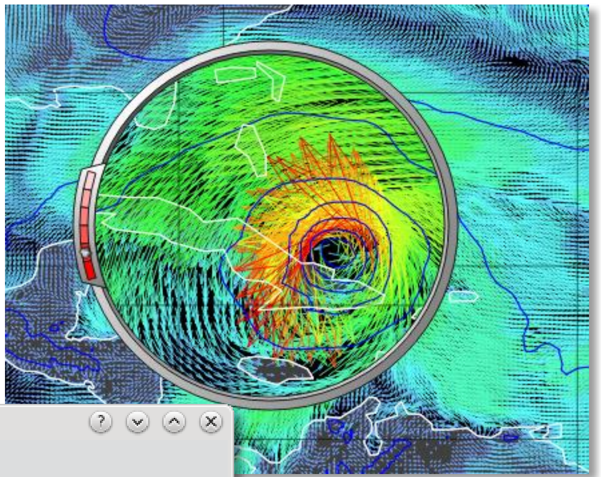
Exploring Metview (2)

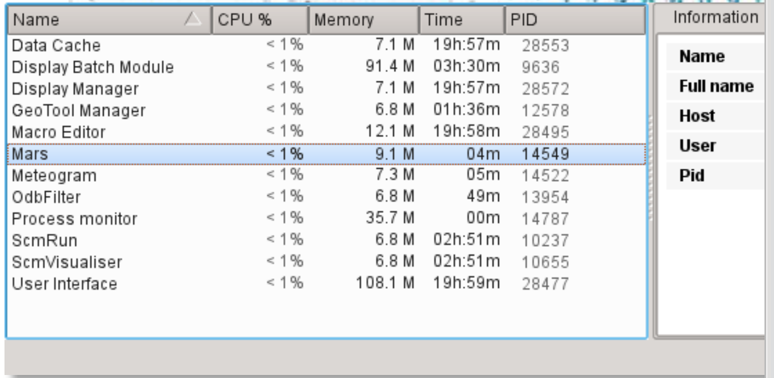
- **Mail** **-** exchange icons by email
- **Archive** - convert a group of icons into an archive
- **Monitor – to monitor and control tasks**
  - **Check the progress of long tasks**
  - **Abort a misbehaving Metview process**
- **Station – search Station Database**
  - **Access Metview database of 10,000 WMO stations**

Exploring Metview (3)

- **More applications, plot types and tools**



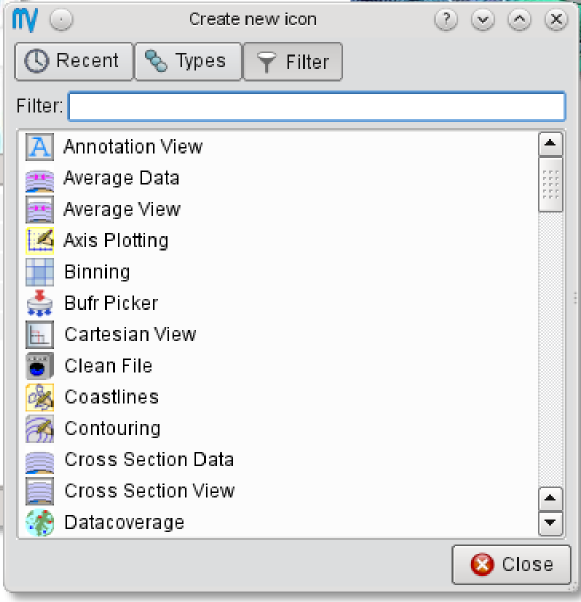




Name	CPU %	Memory	Time	PID
Data Cache	< 1%	7.1 M	19h:57m	28553
Display Batch Module	< 1%	91.4 M	03h:30m	9636
Display Manager	< 1%	7.1 M	19h:57m	28572
GeoTool Manager	< 1%	6.8 M	01h:36m	12578
Macro Editor	< 1%	12.1 M	19h:58m	28495
Mars	< 1%	9.1 M	04m	14549
Meteogram	< 1%	7.3 M	05m	14522
OdbFilter	< 1%	6.8 M	49m	13954
Process monitor	< 1%	35.7 M	00m	14787
ScmRun	< 1%	6.8 M	02h:51m	10237
ScmVisualiser	< 1%	6.8 M	02h:51m	10655
User Interface	< 1%	108.1 M	19h:59m	28477

Information

Name
Full name
Host
User
Pid

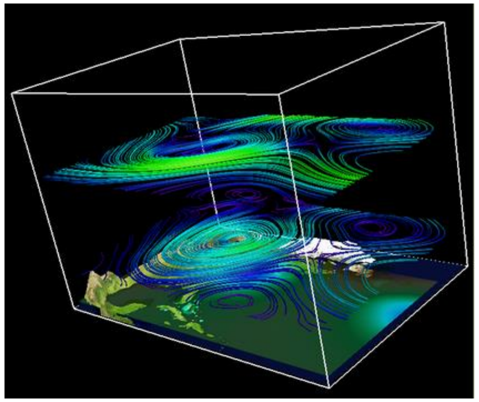


- Annotation View
- Average Data
- Average View
- Axis Plotting
- Binning
- Bufr Picker
- Cartesian View
- Clean File
- Coastlines
- Contouring
- Cross Section Data
- Cross Section View
- Datacoverage

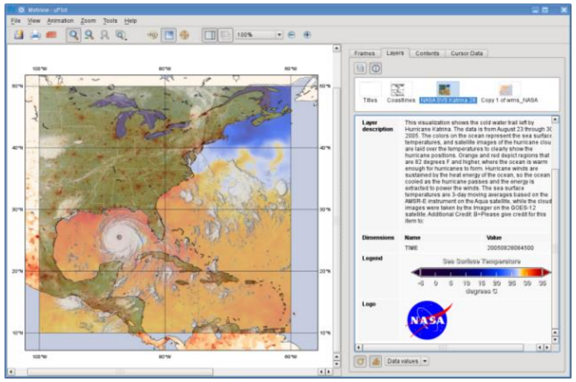
Exploring Metview: Tutorials

- **Many other tutorials available, including:**

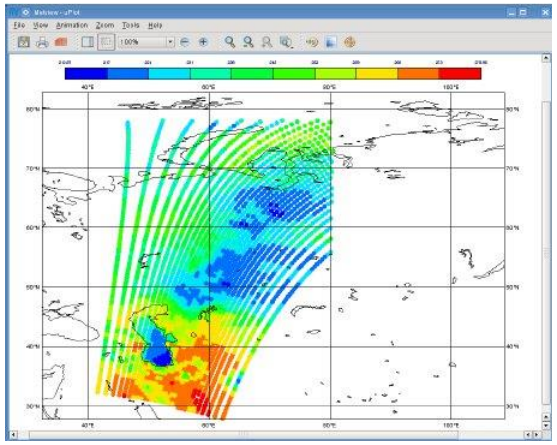
VAPOR

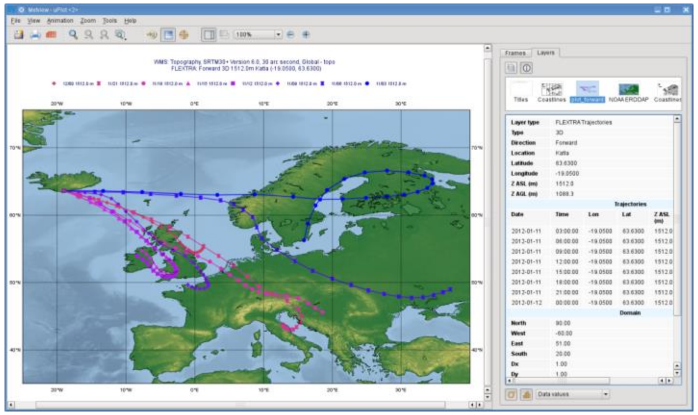


WMS



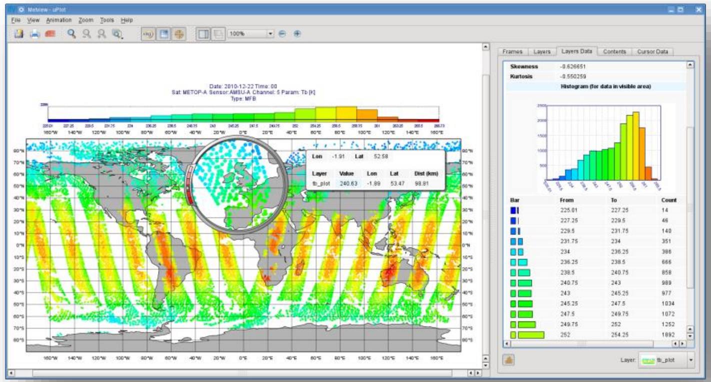
BUFR





FLEXTRA

ODB



https://software.ecmwf.int/metview

Metview Tutorial: Titles and Workflow

- **Please do exercise “Working with Folders and Icons” in the provided sub-folder “explore”**
- **Please do exercise “Exploring Metview” in the provided sub-folder “explore”**

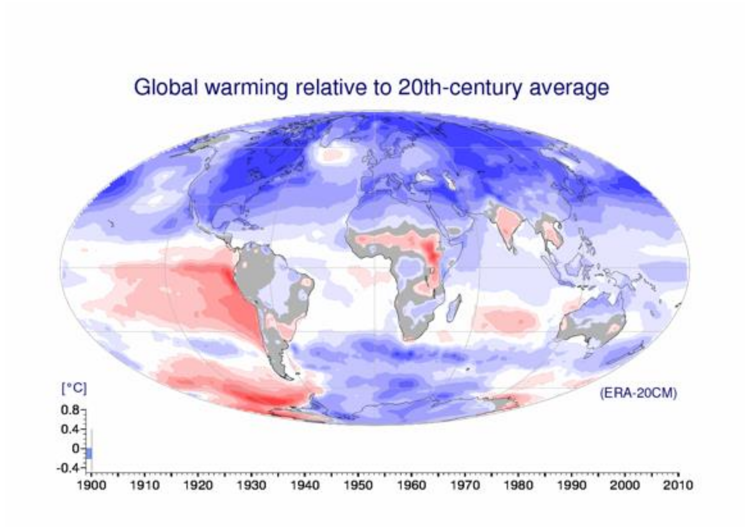
Closing remarks

Deploying Metview

- **Compilation from source**
  - The installation has been made much easier over recent years
    - CMake based build system is now the same for all ECMWF packages
    - Support for Mac OS X 10.10 was added
- **Used of pre-built versions**
  - More and more communities start building binary packages of Metview
    - *Opensuse Build Service (OBS)* offers RPMs for OpenSuse, SLES, Redhat, CentOS and Fedora
    - Ubuntu 15.04 will have Metview in its Science repository
  - Virtual machines are available to start quickly with Metview
    - On *SuseStudio* this can be used to build appliances for cloud services
  - There are plans to provide a Metview Docker image

Keep in touch ….

…. also when things work ;-)



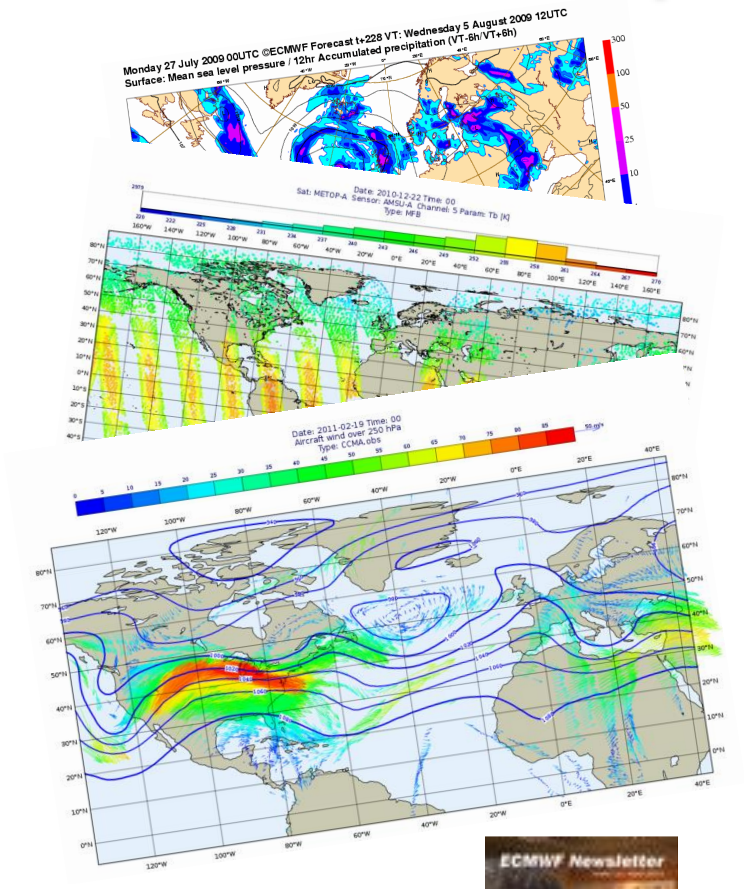
For more information …

email us:

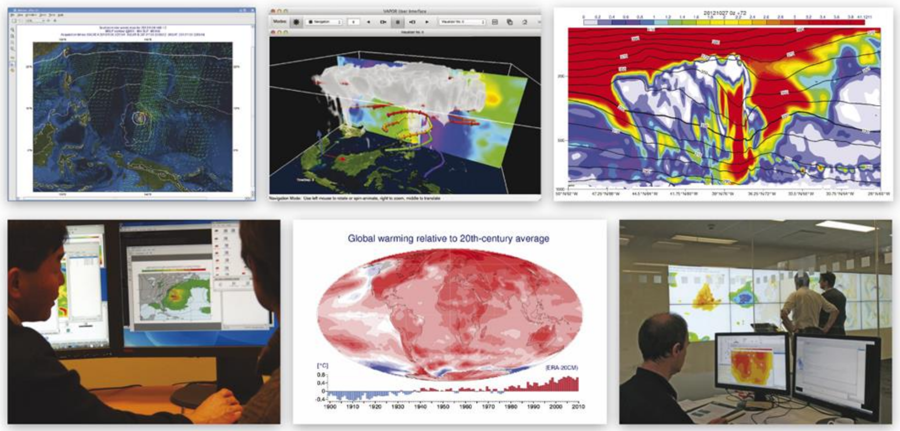
- **Metview: metview@ecmwf.int**

visit our web pages:

- **https://software.ecmwf.int/metview**
- **Download**
  - **Source code, virtual machine**
- **Documentation and tutorials available**
- **Metview articles in recent ECMWF newsletters**


Monday 27 July 2009 00UTC ©ECMWF Forecast t+228 VT: Wednesday 5 August 2009 12UTC
Surface: Mean sea level pressure / 12hr Accumulated precipitation (VT-6h/VT+6h)
Date: 2010-12-22 Time: 00
Sat: METOP-A Sensor: AMSU-A Channel: 5 Param: Tb [K]
Type: MFB
Date: 2011-02-19 Time: 00
Aircraft wind over 250 hPa
Type: CCMA.obs

Hope to see you again!



- **Using ECMWF Forecast meeting, 8-10 June**
  - **This years focus on the communication (visualisation) of uncertainties**

- **Visualisation in Meteorology week, 28 Sep – 2 Oct**
  - **Workshop on Meteorological Operational Systems**
  - **EGOWS – European Working Group on Operational meteorological Workstation Systems**
  - **OGC PlugFest**

Hope you enjoyed the course and found it useful!

Please leave us your feedback ☺