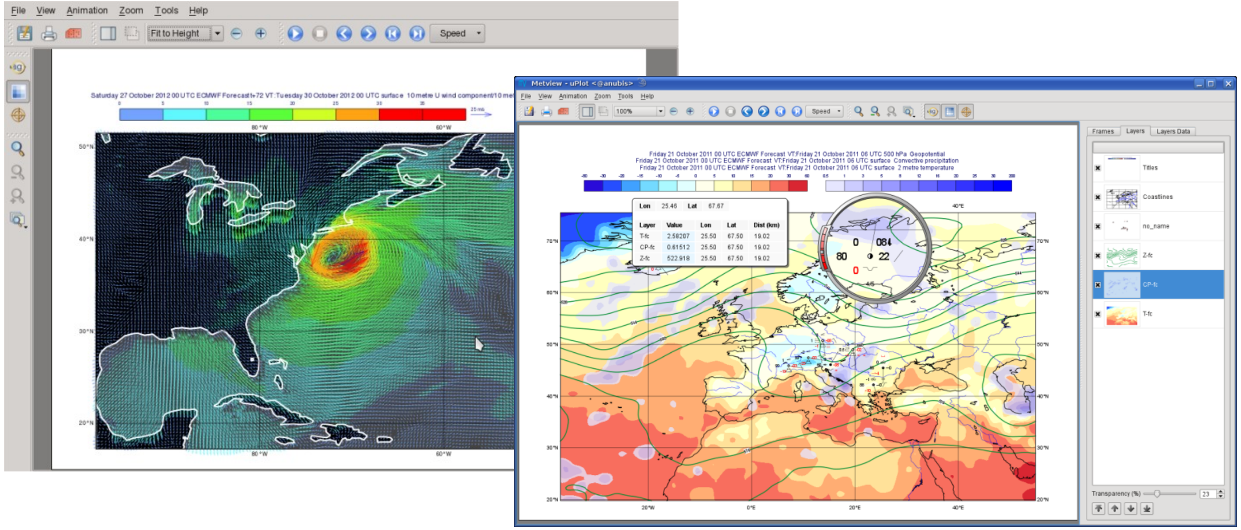
Data Handling with Metview

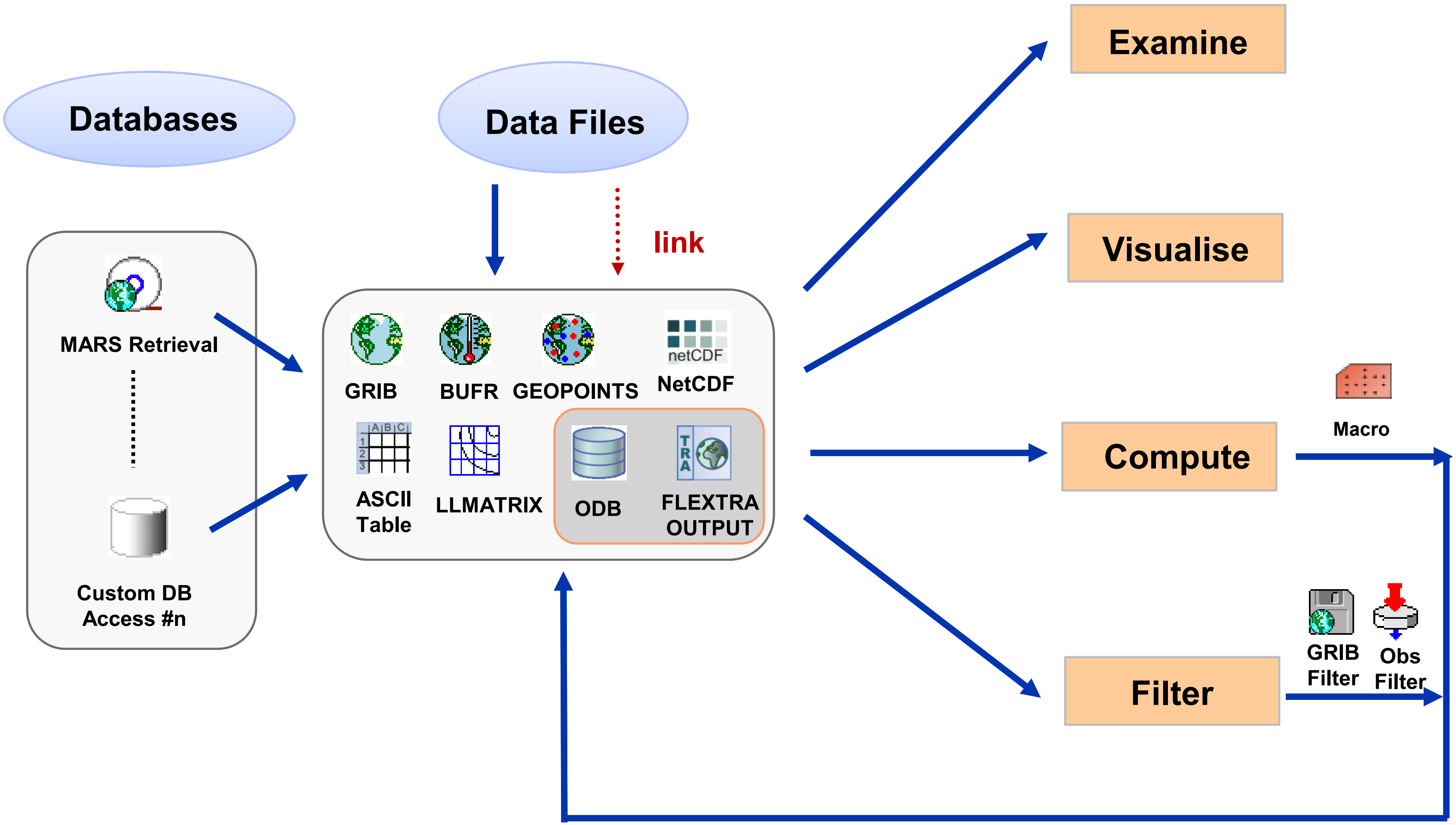

File View Animation Zoom Tools Help
Fit to Height
Speed
Saturday 27 October 2012 00 UTC ECMWF Forecast t+72 VT:Tuesday 30 October 2012 00 UTC surface 10 metre U wind component/10 me
Metview - uPlot <@anubis>
File View Animation Zoom Tools Help
100%
Speed
Friday 21 October 2011 00 UTC ECMWF Forecast VT:Friday 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
Frames
Layers
Layers Data
Titles
Coastlines
no_name
Z-fc
CP-fc
T-fc
Transparency (%)
23

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

Sándor Kertész

Development Section
ECMWF

Data handling in Metview

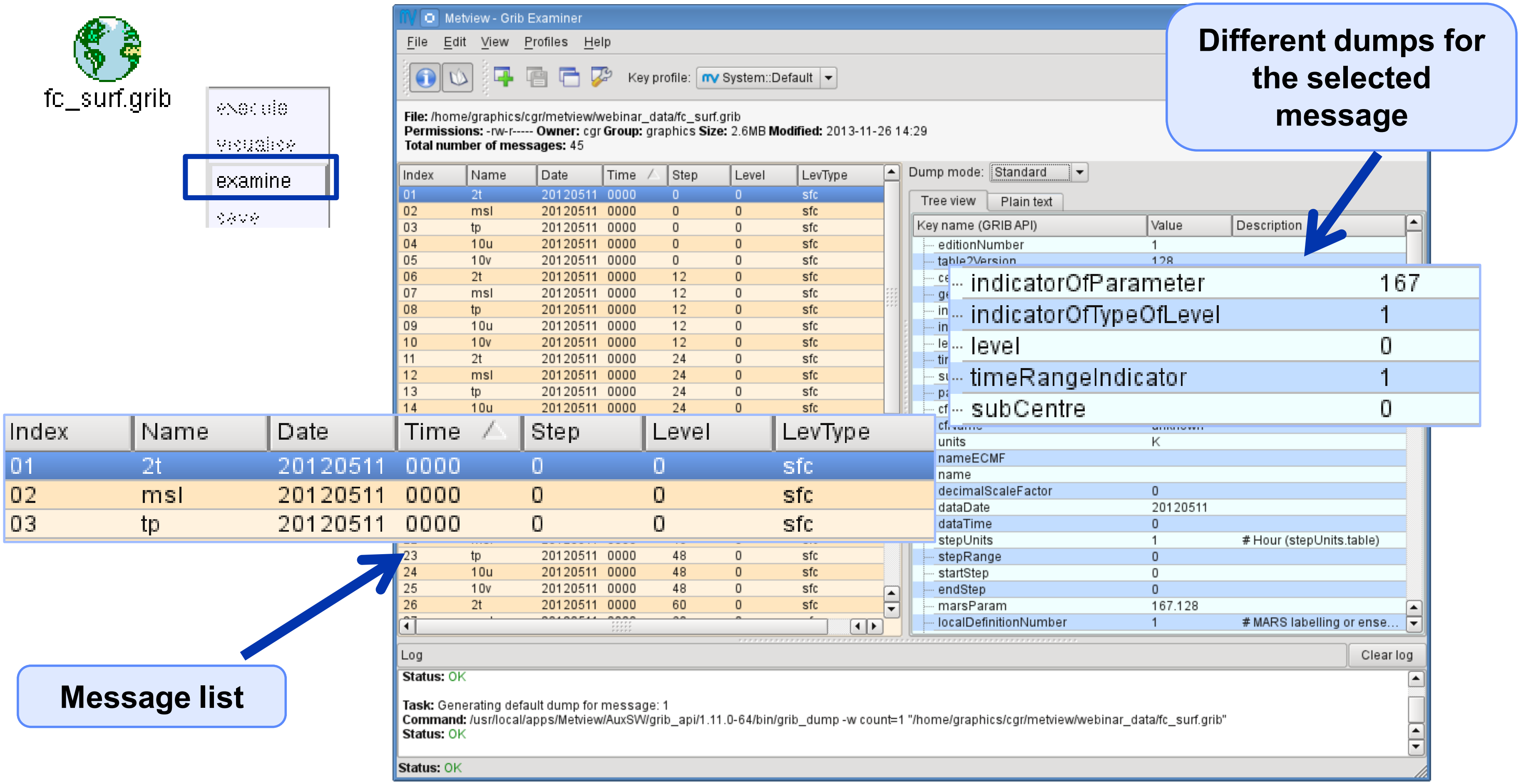


GRIB

- WMO’s binary format for gridded data
- The Metview interface is based on GRIB API
- Access to both Edition 1 and 2 files

GRIB Examiner

- GRIBs contents can be checked with the GRIB Examiner



File: /home/graphics/cgr/metview/webinar_data/fc_surf.grib
Permissions: -rw-r----- **Owner:** cgr **Group:** graphics **Size:** 2.6MB **Modified:** 2013-11-26 14:29
Total number of messages: 45

Index	Name	Date	Time	Step	Level	LevType
01	2t	20120511	0000	0	0	sfc
02	msl	20120511	0000	0	0	sfc
03	tp	20120511	0000	0	0	sfc
04	10u	20120511	0000	0	0	sfc
05	10v	20120511	0000	0	0	sfc
06	2t	20120511	0000	12	0	sfc
07	msl	20120511	0000	12	0	sfc
08	tp	20120511	0000	12	0	sfc
09	10u	20120511	0000	12	0	sfc
10	10v	20120511	0000	12	0	sfc
11	2t	20120511	0000	24	0	sfc
12	msl	20120511	0000	24	0	sfc
13	tp	20120511	0000	24	0	sfc
14	10u	20120511	0000	24	0	sfc
23	tp	20120511	0000	48	0	sfc
24	10u	20120511	0000	48	0	sfc
25	10v	20120511	0000	48	0	sfc
26	2t	20120511	0000	60	0	sfc

Key name (GRIB API)	Value	Description
editionNumber	1	
table2Version	128	
indicatorOfParameter	167	
indicatorOfTypeOfLevel	1	
level	0	
timeRangeIndicator	1	
subCentre	0	
units	K	
nameECMF		
name		
decimalScaleFactor	0	
dataDate	20120511	
dataTime	0	
stepUnits	1	# Hour (stepUnits.table)
stepRange	0	
startStep	0	
endStep	0	
marsParam	167.128	
localDefinitionNumber	1	# MARS labelling or ense...

```
Status: OK

Task: Generating default dump for message: 1
Command: /usr/local/apps/Metview/AuxSW/grib_api/1.11.0-64/bin/grib_dump -w count=1 "/home/graphics/cgr/metview/webinar_data/fc_surf.grib"
Status: OK
```

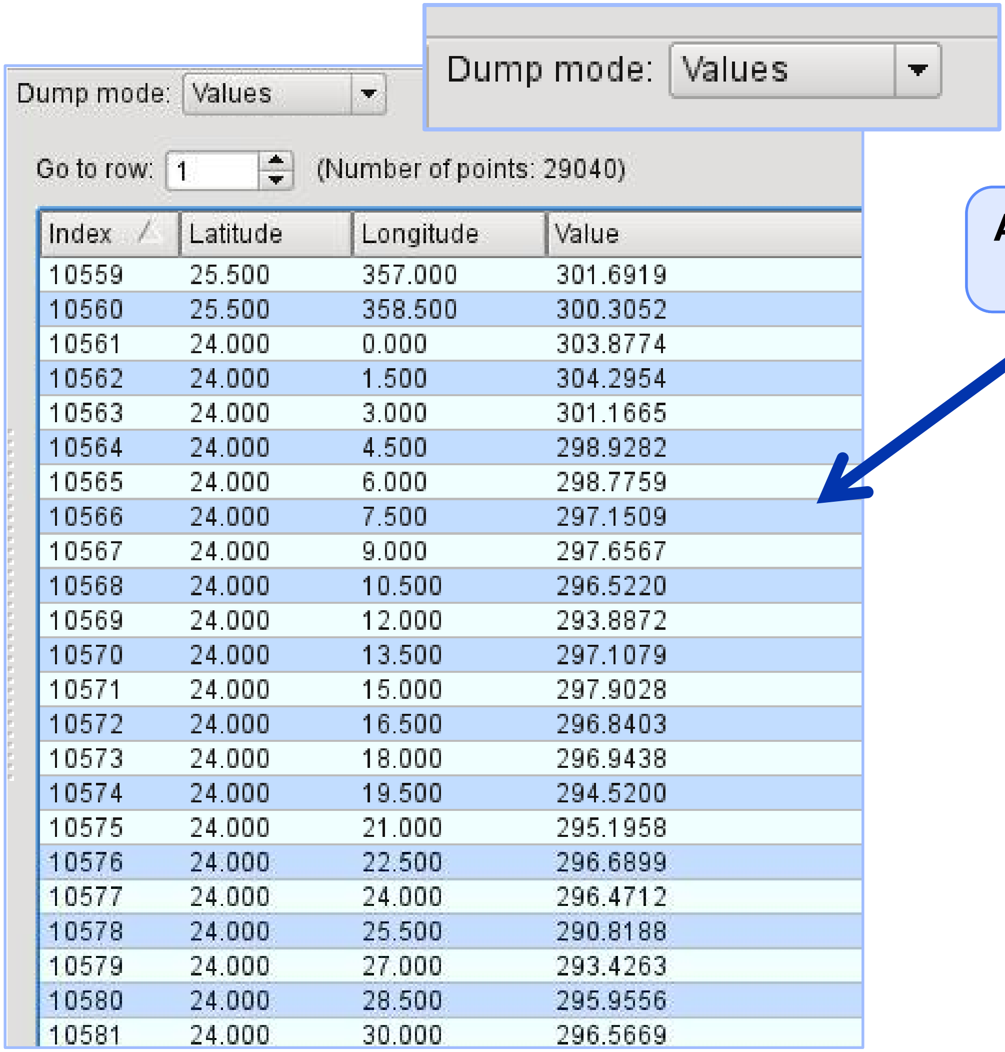
Status: OK

Index	Name	Date	Time	Step	Level	LevType
01	2t	20120511	0000	0	0	sfc
02	msl	20120511	0000	0	0	sfc
03	tp	20120511	0000	0	0	sfc

Different dumps for the selected message

Message list

GRIB Examiner – Values dump



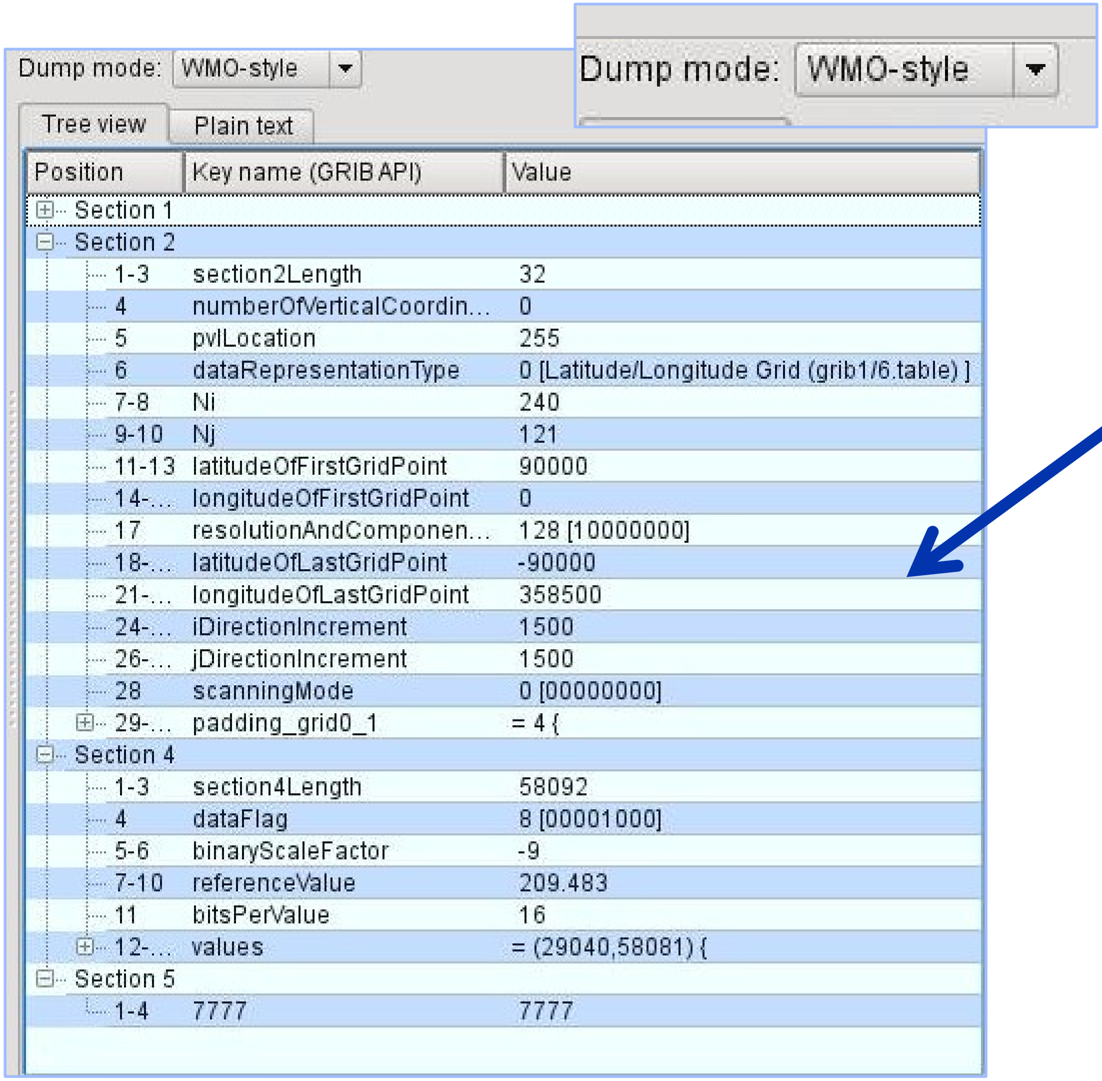
Dump mode: Values

Go to row: 1 (Number of points: 29040)

Index	Latitude	Longitude	Value
10559	25.500	357.000	301.6919
10560	25.500	358.500	300.3052
10561	24.000	0.000	303.8774
10562	24.000	1.500	304.2954
10563	24.000	3.000	301.1665
10564	24.000	4.500	298.9282
10565	24.000	6.000	298.7759
10566	24.000	7.500	297.1509
10567	24.000	9.000	297.6567
10568	24.000	10.500	296.5220
10569	24.000	12.000	293.8872
10570	24.000	13.500	297.1079
10571	24.000	15.000	297.9028
10572	24.000	16.500	296.8403
10573	24.000	18.000	296.9438
10574	24.000	19.500	294.5200
10575	24.000	21.000	295.1958
10576	24.000	22.500	296.6899
10577	24.000	24.000	296.4712
10578	24.000	25.500	290.8188
10579	24.000	27.000	293.4263
10580	24.000	28.500	295.9556
10581	24.000	30.000	296.5669

All the values for the selected message

GRIB Examiner – WMO-style dump



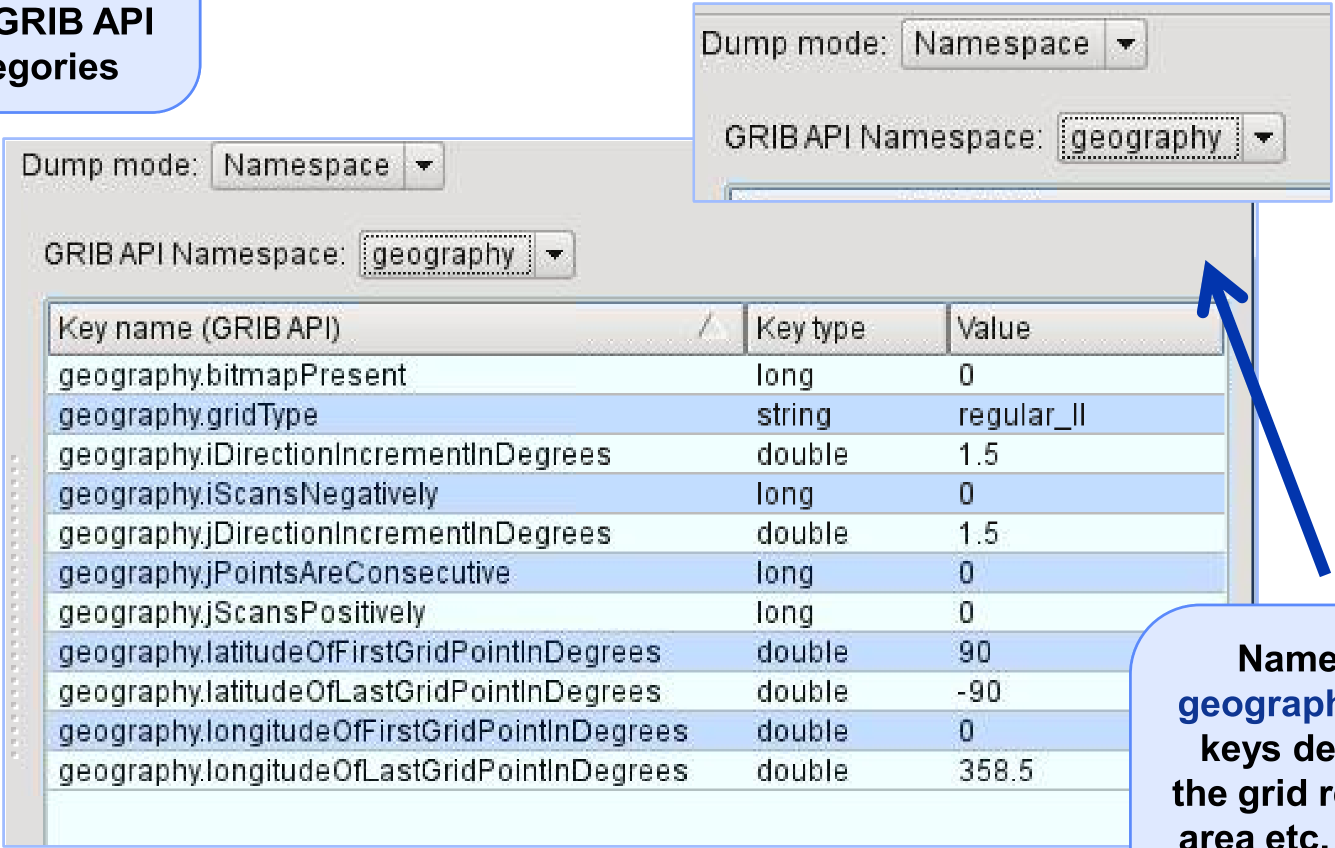
Dump mode: WMO-style

Position	Key name (GRIB API)	Value
Section 1		
Section 2		
1-3	section2Length	32
4	numberOfVerticalCoordin...	0
5	pvlLocation	255
6	dataRepresentationType	0 [Latitude/Longitude Grid (grib1/6.table)]
7-8	Ni	240
9-10	Nj	121
11-13	latitudeOfFirstGridPoint	90000
14-...	longitudeOfFirstGridPoint	0
17	resolutionAndComponen...	128 [10000000]
18-...	latitudeOfLastGridPoint	-90000
21-...	longitudeOfLastGridPoint	358500
24-...	iDirectionIncrement	1500
26-...	jDirectionIncrement	1500
28	scanningMode	0 [00000000]
29-...	padding_grid0_1	= 4 {
Section 4		
1-3	section4Length	58092
4	dataFlag	8 [00001000]
5-6	binaryScaleFactor	-9
7-10	referenceValue	209.483
11	bitsPerValue	16
12-...	values	= (29040,58081) {
Section 5		
1-4	7777	7777

Each section of the GRIB message is shown in a tree view

GRIB Examiner – Namespace dump

Namespace is a GRIB API concept to define GRIB API key categories



Dump mode: Namespace

GRIB API Namespace: geography

Key name (GRIB API)	Key type	Value
geography.bitmapPresent	long	0
geography.gridType	string	regular_ll
geography.iDirectionIncrementInDegrees	double	1.5
geography.iScansNegatively	long	0
geography.jDirectionIncrementInDegrees	double	1.5
geography.jPointsAreConsecutive	long	0
geography.jScansPositively	long	0
geography.latitudeOfFirstGridPointInDegrees	double	90
geography.latitudeOfLastGridPointInDegrees	double	-90
geography.longitudeOfFirstGridPointInDegrees	double	0
geography.longitudeOfLastGridPointInDegrees	double	358.5

Dump mode: Namespace

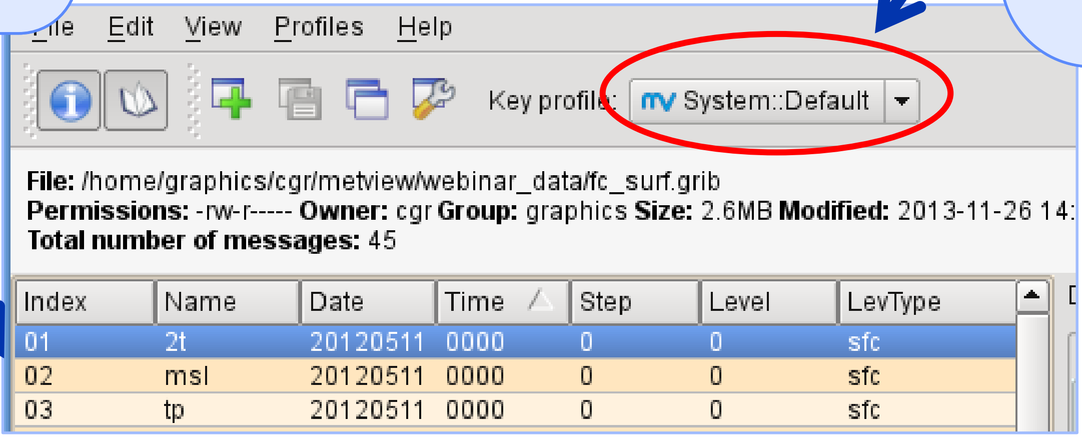
GRIB API Namespace: geography

Namespace geography groups keys describing the grid resolution, area etc. GRIB API concept

GRIB Examiner – Key profiles

The message list is presented using a set of GRIB API keys. A group of these keys is called a **key profile**.

The default profile is called **System::Default**. It is read-only! However you can create any number of additional profiles.

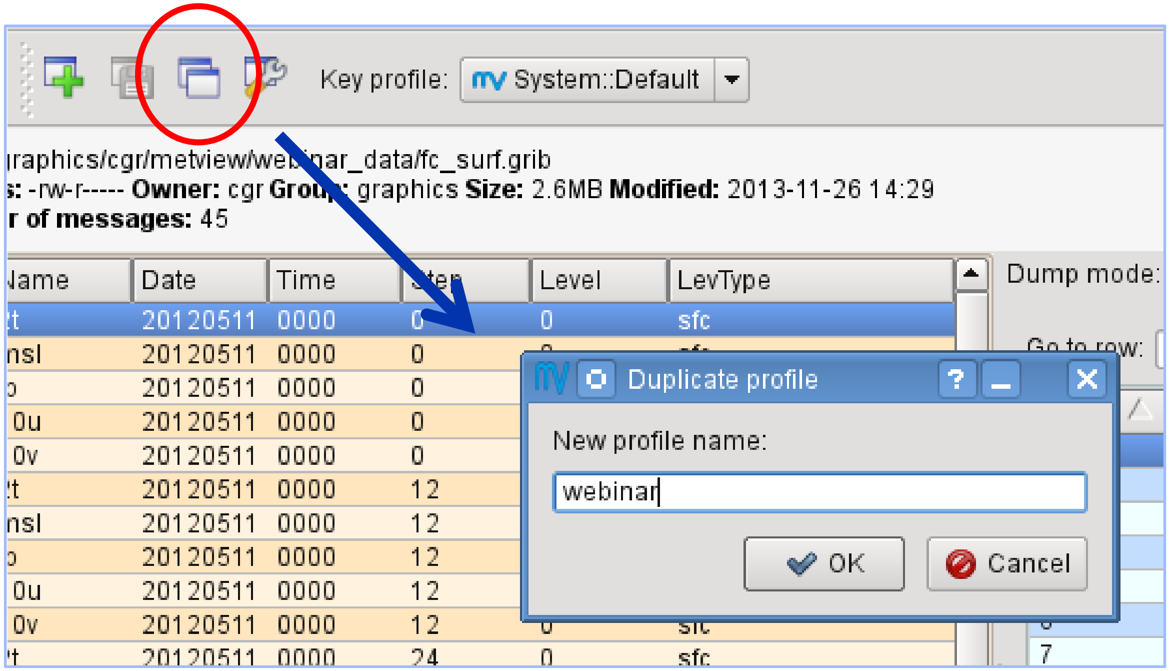


Key profile: System::Default

File: /home/graphics/cgr/metview/webinar_data/fc_surf.grib
Permissions: -rw-r----- **Owner:** cgr **Group:** graphics **Size:** 2.6MB **Modified:** 2013-11-26 14:
Total number of messages: 45

Index	Name	Date	Time	Step	Level	LevType
01	2t	20120511	0000	0	0	sfc
02	msl	20120511	0000	0	0	sfc
03	tp	20120511	0000	0	0	sfc

GRIB Examiner – Create a new key profile



raphics/cgr/metview/webinar_data/fc_surf.grib
s: -rw-r----- **Owner:** cgr **Group:** graphics **Size:** 2.6MB **Modified:** 2013-11-26 14:29
r of messages: 45

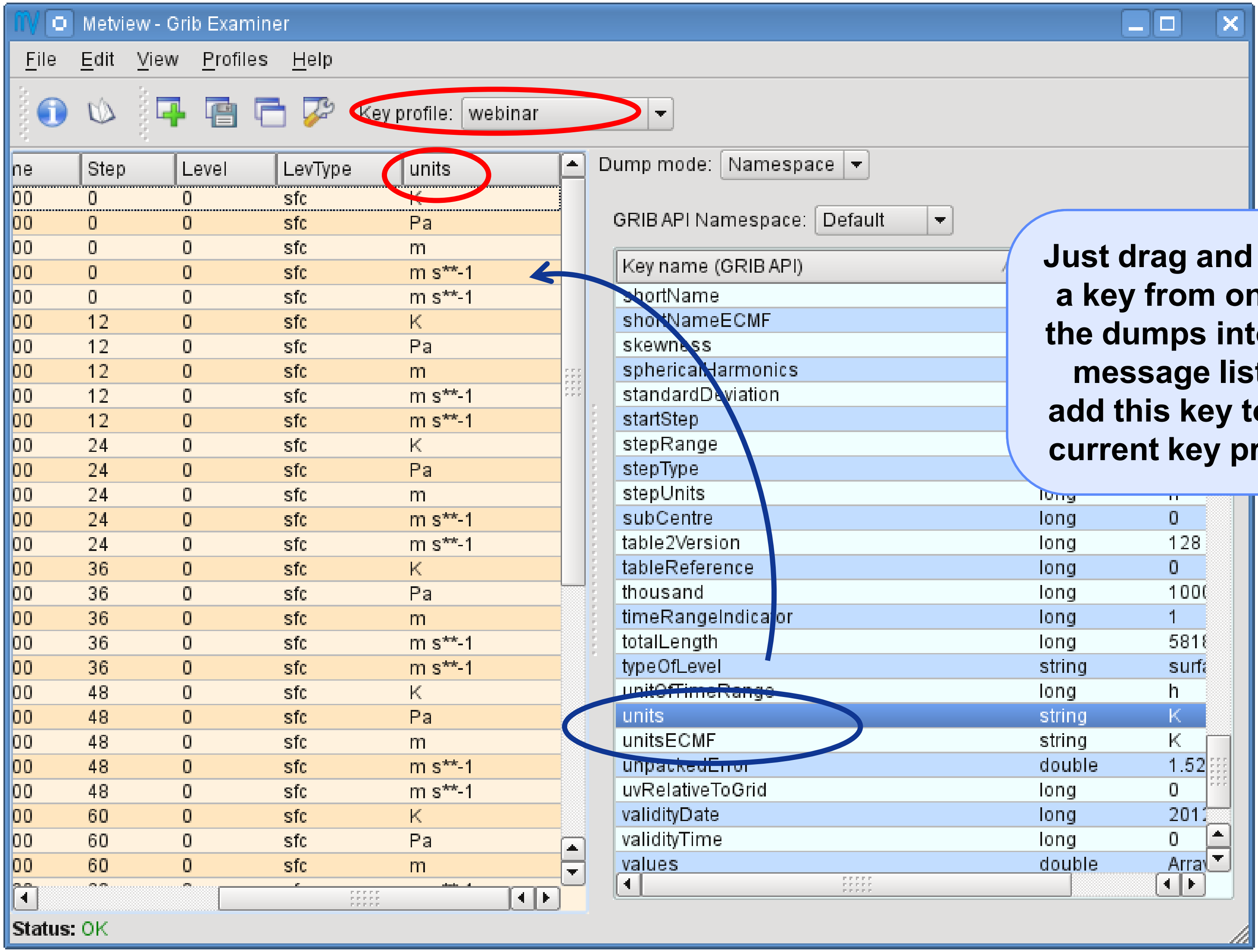
Name	Date	Time	Step	Level	LevType
't	20120511	0000	0	0	sfc
nsl	20120511	0000	0	0	sfc
)	20120511	0000	0		
0u	20120511	0000	0		
0v	20120511	0000	0		
't	20120511	0000	12		
nsl	20120511	0000	12		
)	20120511	0000	12		
0u	20120511	0000	12		
0v	20120511	0000	12	0	sfc
't	20120511	0000	24	0	sfc

Duplicate profile

New profile name: webinar

The easiest way to create a new key profile is to duplicate an existing one

GRIB Examiner – Populate key profiles



Key profile: webinar

ne	Step	Level	LevType	units
00	0	0	sfc	K
00	0	0	sfc	Pa
00	0	0	sfc	m
00	0	0	sfc	m s**-1
00	0	0	sfc	m s**-1
00	12	0	sfc	K
00	12	0	sfc	Pa
00	12	0	sfc	m
00	12	0	sfc	m s**-1
00	12	0	sfc	m s**-1
00	24	0	sfc	K
00	24	0	sfc	Pa
00	24	0	sfc	m
00	24	0	sfc	m s**-1
00	24	0	sfc	m s**-1
00	36	0	sfc	K
00	36	0	sfc	Pa
00	36	0	sfc	m
00	36	0	sfc	m s**-1
00	36	0	sfc	m s**-1
00	48	0	sfc	K
00	48	0	sfc	Pa
00	48	0	sfc	m
00	48	0	sfc	m s**-1
00	48	0	sfc	m s**-1
00	60	0	sfc	K
00	60	0	sfc	Pa
00	60	0	sfc	m

Dump mode: Namespace

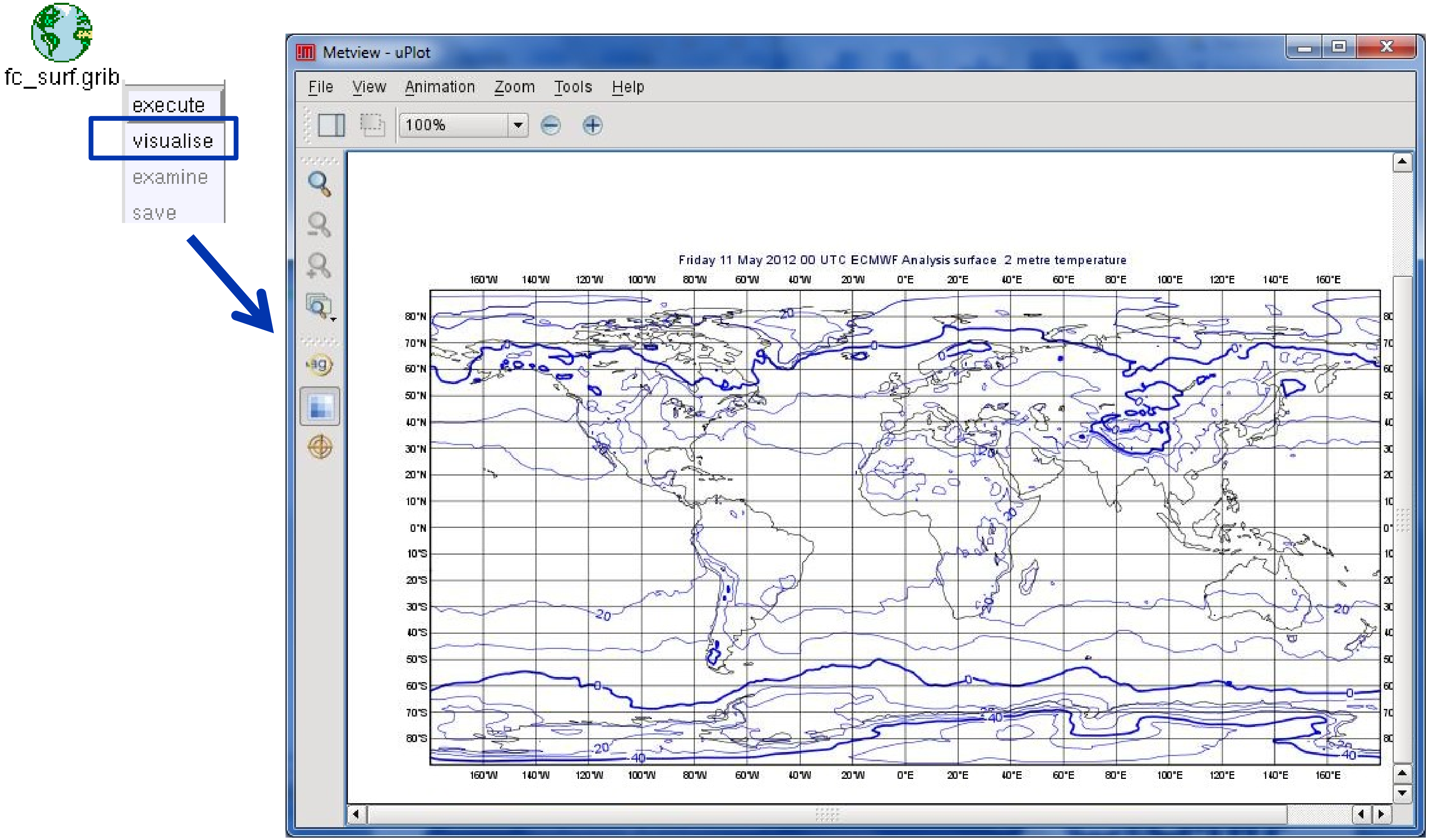
GRIB API Namespace: Default

Key name (GRIB API)		
shortName		
shortNameECMF		
skewness		
sphericalHarmonics		
standardDeviation		
startStep		
stepRange		
stepType		
stepUnits	long	
subCentre	long	0
table2Version	long	128
tableReference	long	0
thousand	long	100
timeRangeIndicator	long	1
totalLength	long	581
typeOfLevel	string	surf
unitOfTimeRange	long	h
units	string	K
unitsECMF	string	K
unpackedError	double	1.52
uvRelativeToGrid	long	0
validityDate	long	201
validityTime	long	0
values	double	Arra

Status: OK

Just drag and drop a key from one of the dumps into the message list to add this key to the current key profile

GRIB plotting



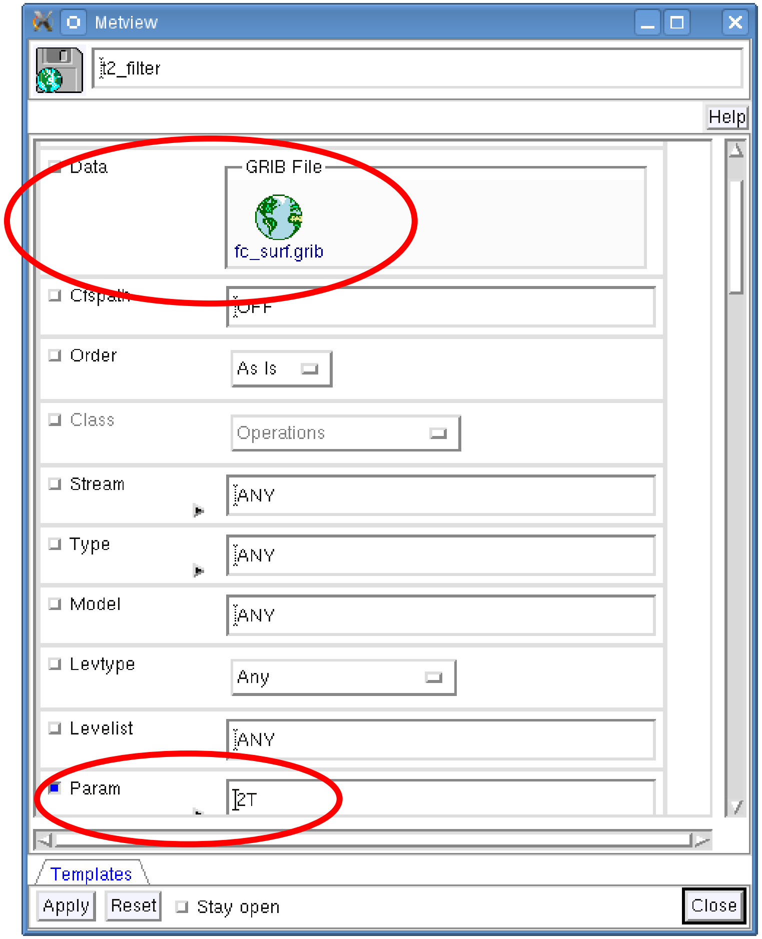
Overlaying fields from the same GRIB file

Example: overlay T2 and MSLP forecasts from file fc_surf.grib

- We need to filter out each parameter into a separate file
- We will use the GRIB Filter icon

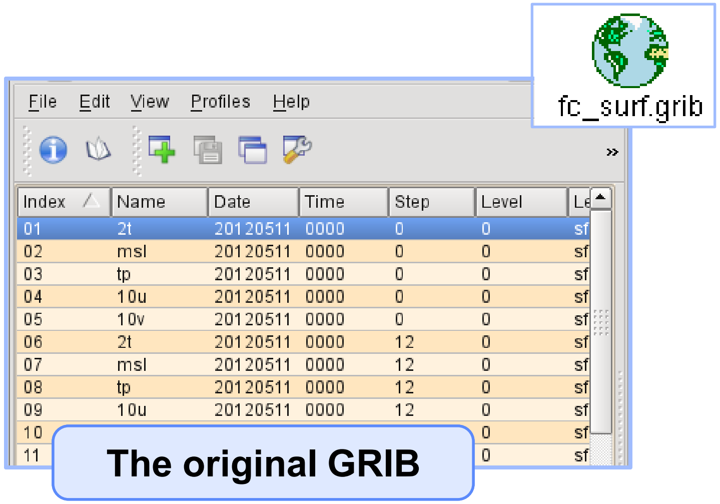
- It allows filtering according to parameter, date, time, level etc.
- It caches the results (name turns green) and can be used directly in the same way as GRIB icon

GRIB Filter: Parameter selection



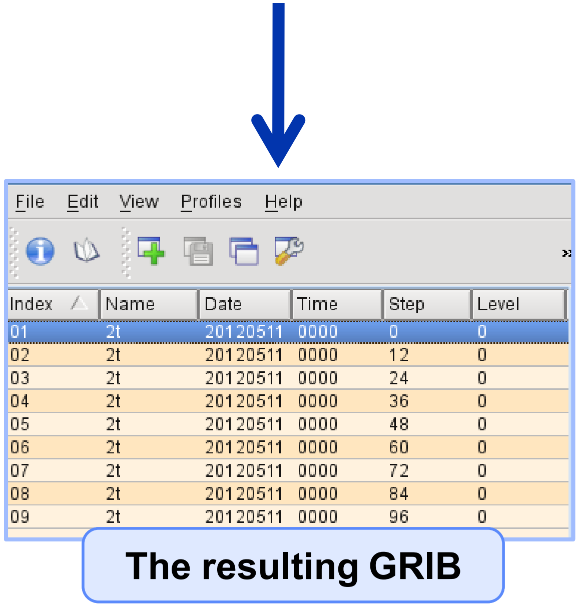
t2_filter

Data	GRIB File fc_surf.grib
□ Cfspath	OFF
□ Order	As Is
□ Class	Operations
□ Stream	ANY
□ Type	ANY
□ Model	ANY
□ Levtype	Any
□ Levelist	ANY
■ Param	2T



Index	Name	Date	Time	Step	Level	Le
01	2t	20120511	0000	0	0	sf
02	msl	20120511	0000	0	0	sf
03	tp	20120511	0000	0	0	sf
04	10u	20120511	0000	0	0	sf
05	10v	20120511	0000	0	0	sf
06	2t	20120511	0000	12	0	sf
07	msl	20120511	0000	12	0	sf
08	tp	20120511	0000	12	0	sf
09	10u	20120511	0000	12	0	sf
10					0	sf
11					0	sf

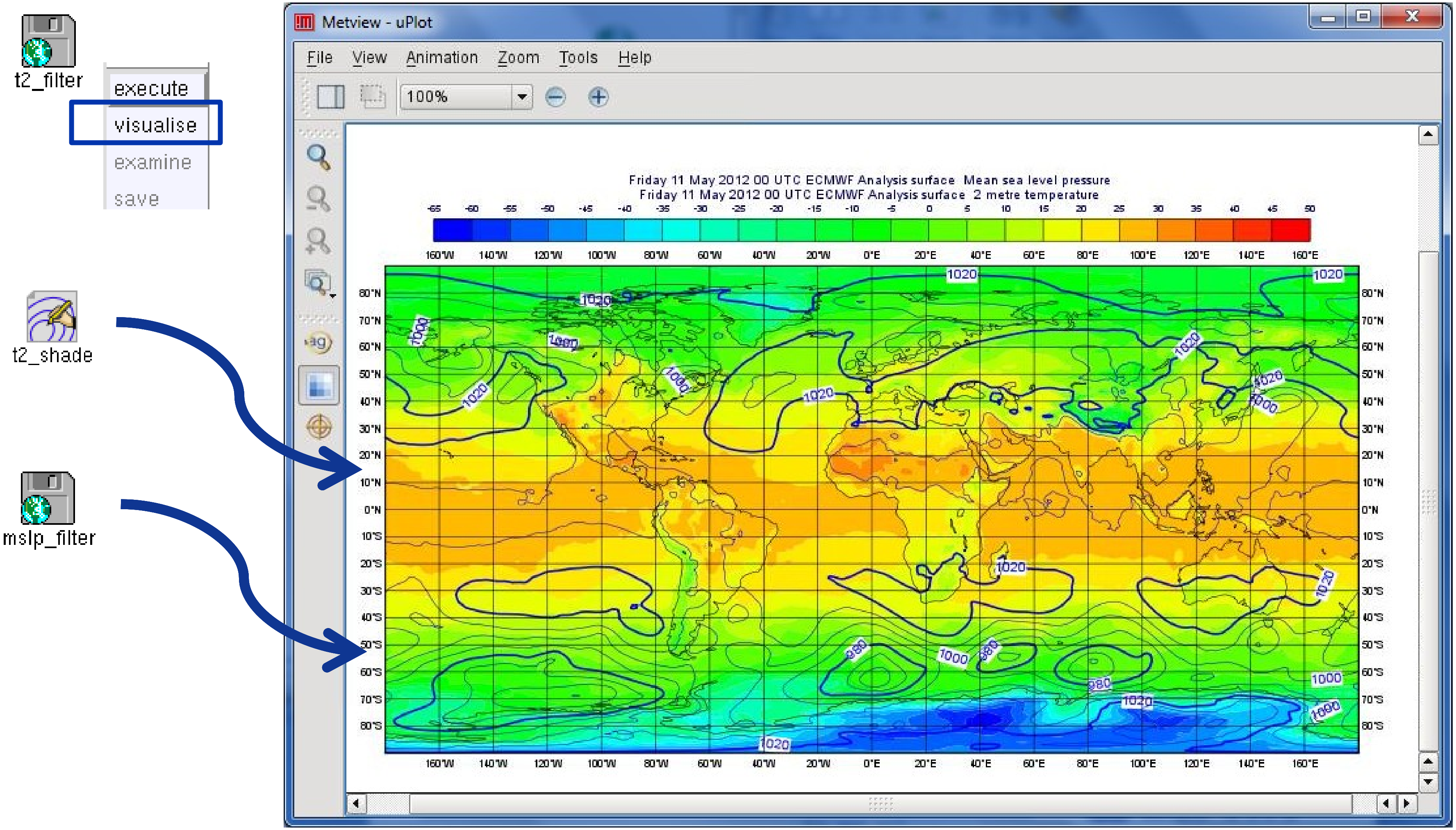
The original GRIB



Index	Name	Date	Time	Step	Level
01	2t	20120511	0000	0	0
02	2t	20120511	0000	12	0
03	2t	20120511	0000	24	0
04	2t	20120511	0000	36	0
05	2t	20120511	0000	48	0
06	2t	20120511	0000	60	0
07	2t	20120511	0000	72	0
08	2t	20120511	0000	84	0
09	2t	20120511	0000	96	0

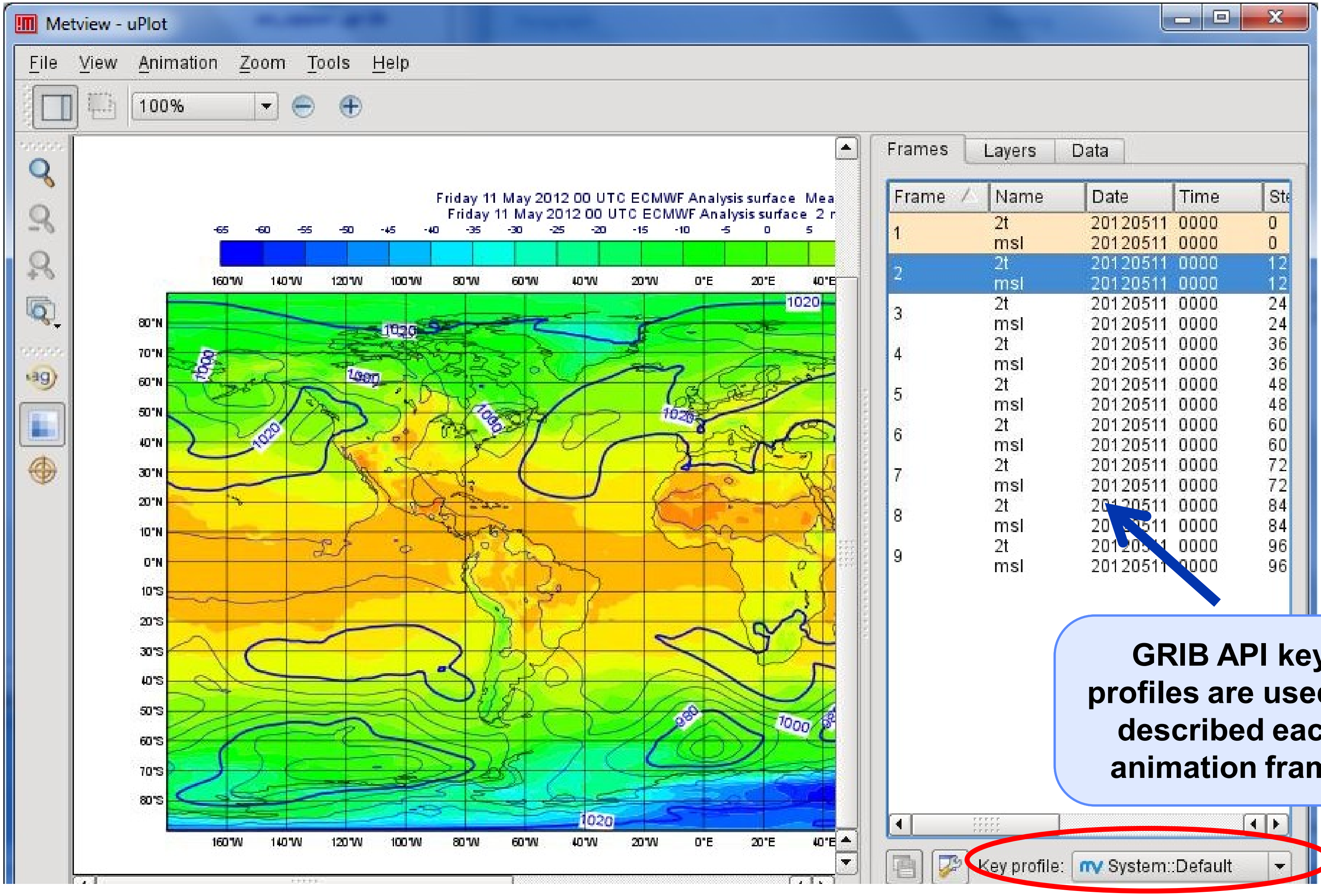
The resulting GRIB

Overlaying GRIB fields



Friday 11 May 2012 00 UTC ECMWF Analysis surface Mean sea level pressure
Friday 11 May 2012 00 UTC ECMWF Analysis surface 2 metre temperature

Overlaying GRIB fields

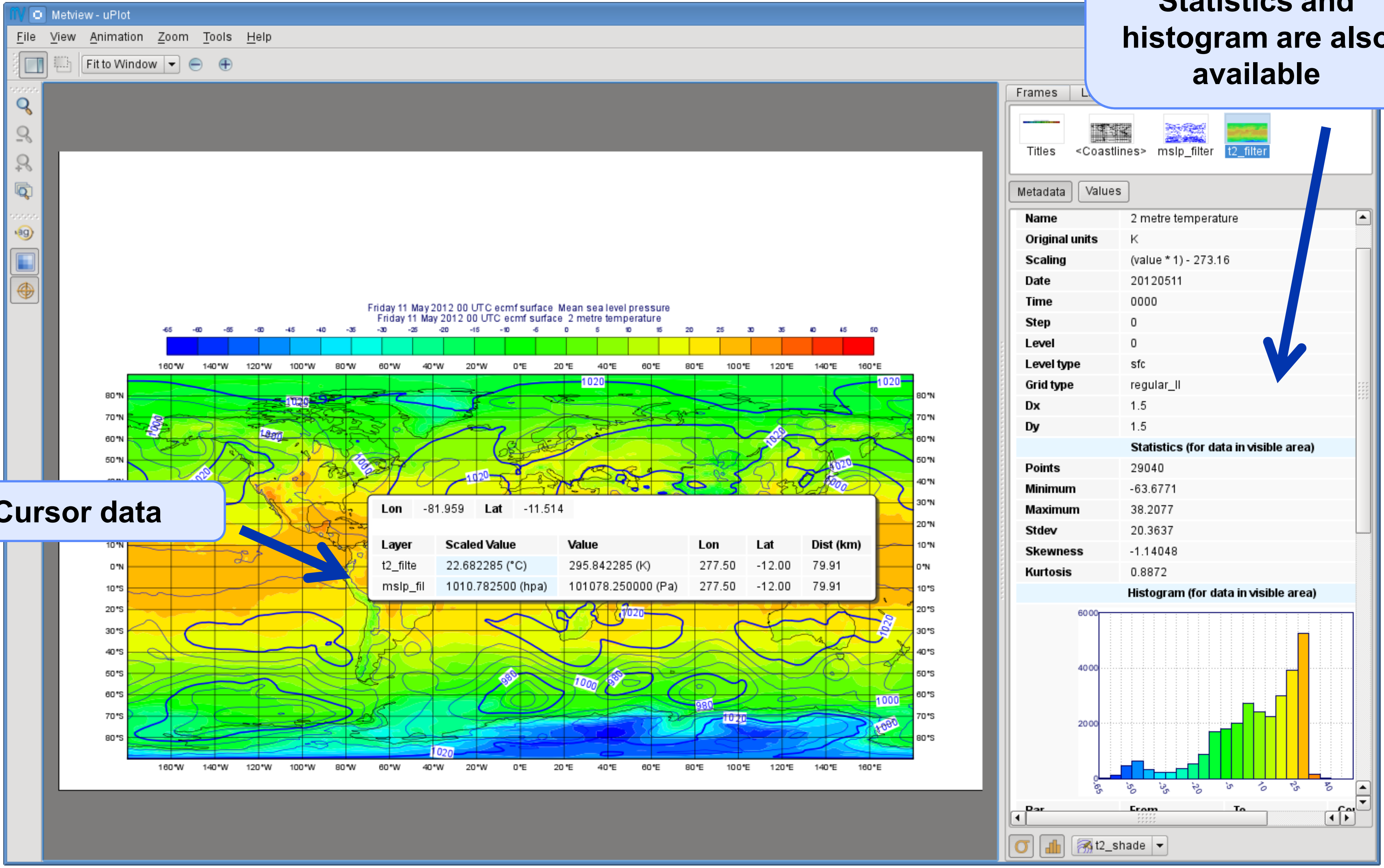


Frame	Name	Date	Time	Ste
1	2t	20120511	0000	0
	msl	20120511	0000	0
2	2t	20120511	0000	12
	msl	20120511	0000	12
3	2t	20120511	0000	24
	msl	20120511	0000	24
4	2t	20120511	0000	36
	msl	20120511	0000	36
5	2t	20120511	0000	48
	msl	20120511	0000	48
6	2t	20120511	0000	60
	msl	20120511	0000	60
7	2t	20120511	0000	72
	msl	20120511	0000	72
8	2t	20120511	0000	84
	msl	20120511	0000	84
9	2t	20120511	0000	96
	msl	20120511	0000	96

Key profile: System::Default

GRIB API key profiles are used to described each animation frame

GRIB data inspection


Metview - uPlot
File View Animation Zoom Tools Help
Fit to Window
Frames
Titles <Coastlines> mslp_filter t2_filter
Metadata Values
Friday 11 May 2012 00 UTC ecmf surface Mean sea level pressure
Friday 11 May 2012 00 UTC ecmf surface 2 metre temperature
160°W 140°W 120°W 100°W 80°W 60°W 40°W 20°W 0°E 20°E 40°E 60°E 80°E 100°E 120°E 140°E 160°E
80°N 70°N 60°N 50°N 40°N 30°N 20°N 10°N 0°N 10°S 20°S 30°S 40°S 50°S 60°S 70°S 80°S
1020 1000 980
6000 4000 2000 0
-65 -50 -35 -20 -5 10 25 40
Par From To Col
t2_shade

Lon	-81.959	Lat	-11.514

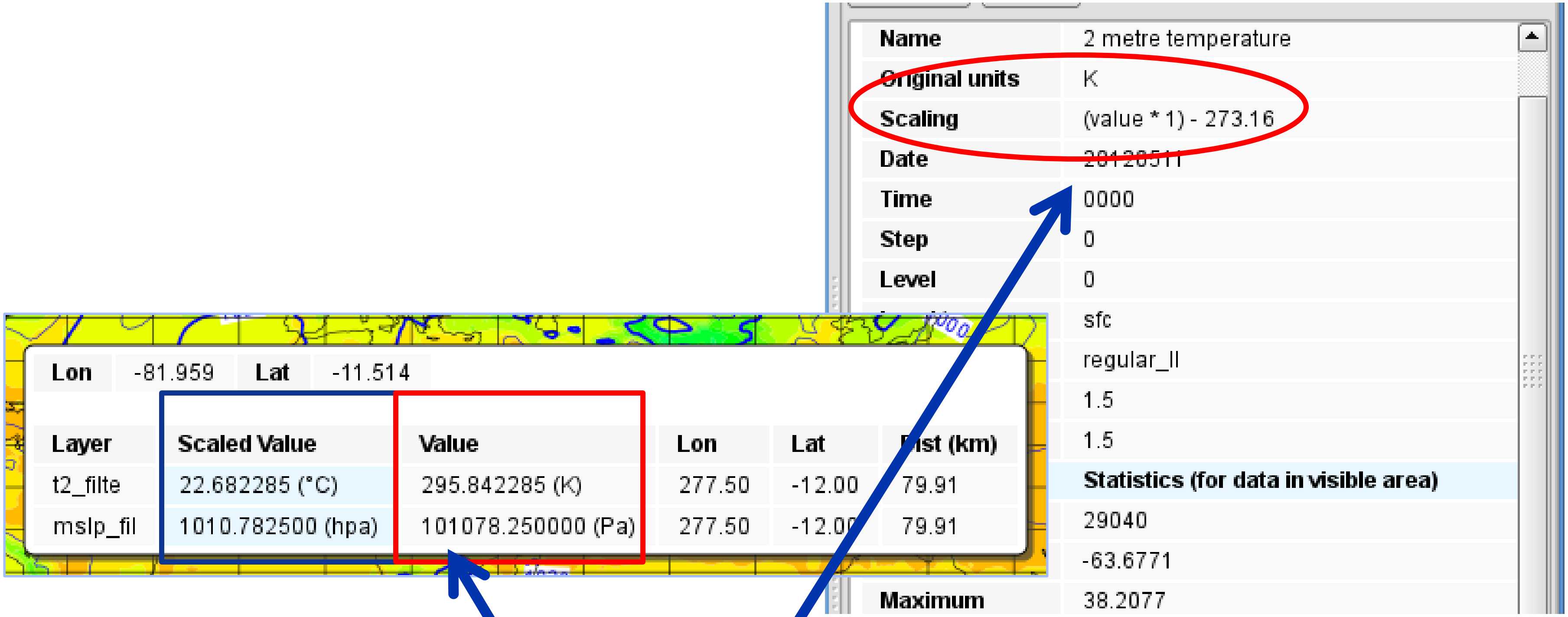
Layer	Scaled Value	Value	Lon	Lat	Dist (km)
t2_filte	22.682285 (°C)	295.842285 (K)	277.50	-12.00	79.91
mslp_fil	1010.782500 (hpa)	101078.250000 (Pa)	277.50	-12.00	79.91

Name	2 metre temperature
Original units	K
Scaling	(value * 1) - 273.16
Date	20120511
Time	0000
Step	0
Level	0
Level type	sfc
Grid type	regular_ll
Dx	1.5
Dy	1.5
	Statistics (for data in visible area)
Points	29040
Minimum	-63.6771
Maximum	38.2077
Stdev	20.3637
Skewness	-1.14048
Kurtosis	0.8872
	Histogram (for data in visible area)

Statistics and histogram are also available

Cursor data

GRIB scaling for plotting



Name	2 metre temperature
Original units	K
Scaling	(value * 1) - 273.16
Date	20120511
Time	0000
Step	0
Level	0
	sfc
	regular_ll
	1.5
	1.5
	Statistics (for data in visible area)
	29040
	-63.6771
Maximum	38.2077

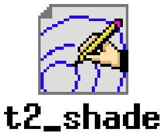
Lon	-81.959	Lat	-11.514

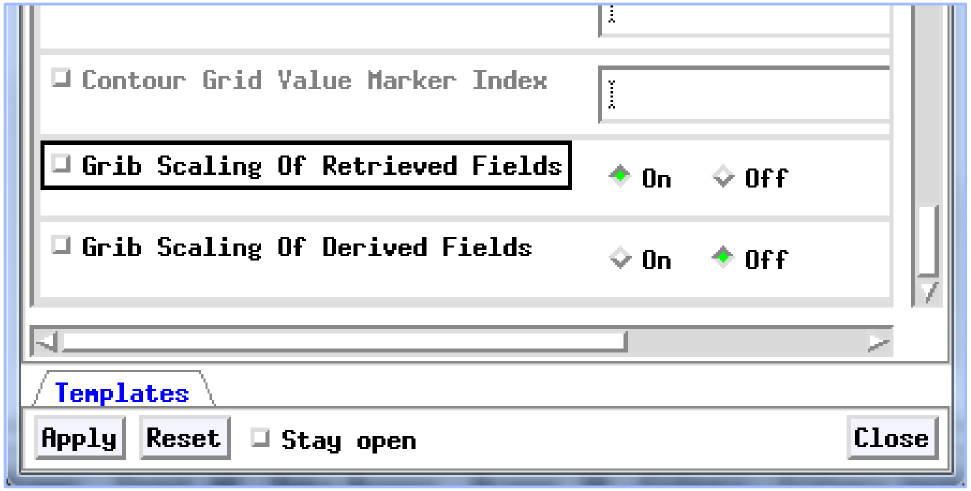
Layer	Scaled Value	Value	Lon	Lat	Dist (km)
t2_filte	22.682285 (°C)	295.842285 (K)	277.50	-12.00	79.91
mslp_fil	1010.782500 (hpa)	101078.250000 (Pa)	277.50	-12.00	79.91

Both T2 and MSLP are scaled

GRIB scaling for plotting

This parameter tells Metview to apply scaling for certain fields for contouring





□ Contour Grid Value Marker Index

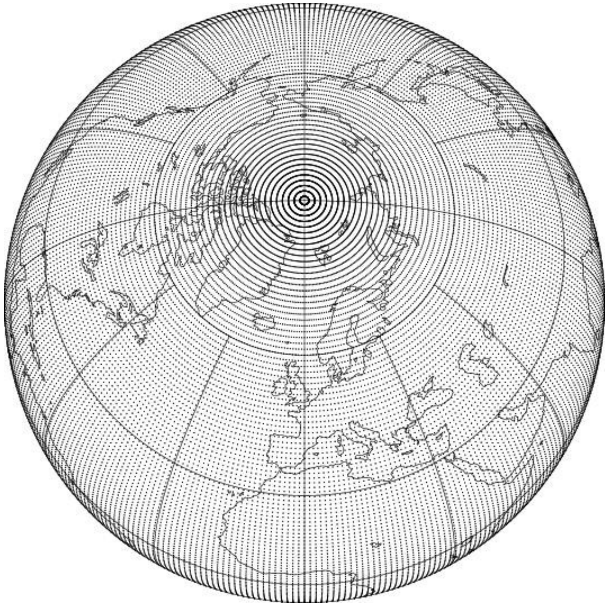
□ Grib Scaling Of Retrieved Fields ● On ○ Off

□ Grib Scaling Of Derived Fields ○ On ● Off

Other usage of GRIB Filter: interpolation

- Spherical harmonics to gridpoint transformation
- Interpolation between different grids
  - Regular Gaussian grid
  - Reduced Gaussian grid
  - lat-lon grids etc.
- Currently it is based on EMOS lib

GRIB Filter



How to use the interpolation?

Example: compute the difference between two different resolution T500 fields

fc_upper.grib

Forecast
1.5x1.5
global grid

an_upper.grb

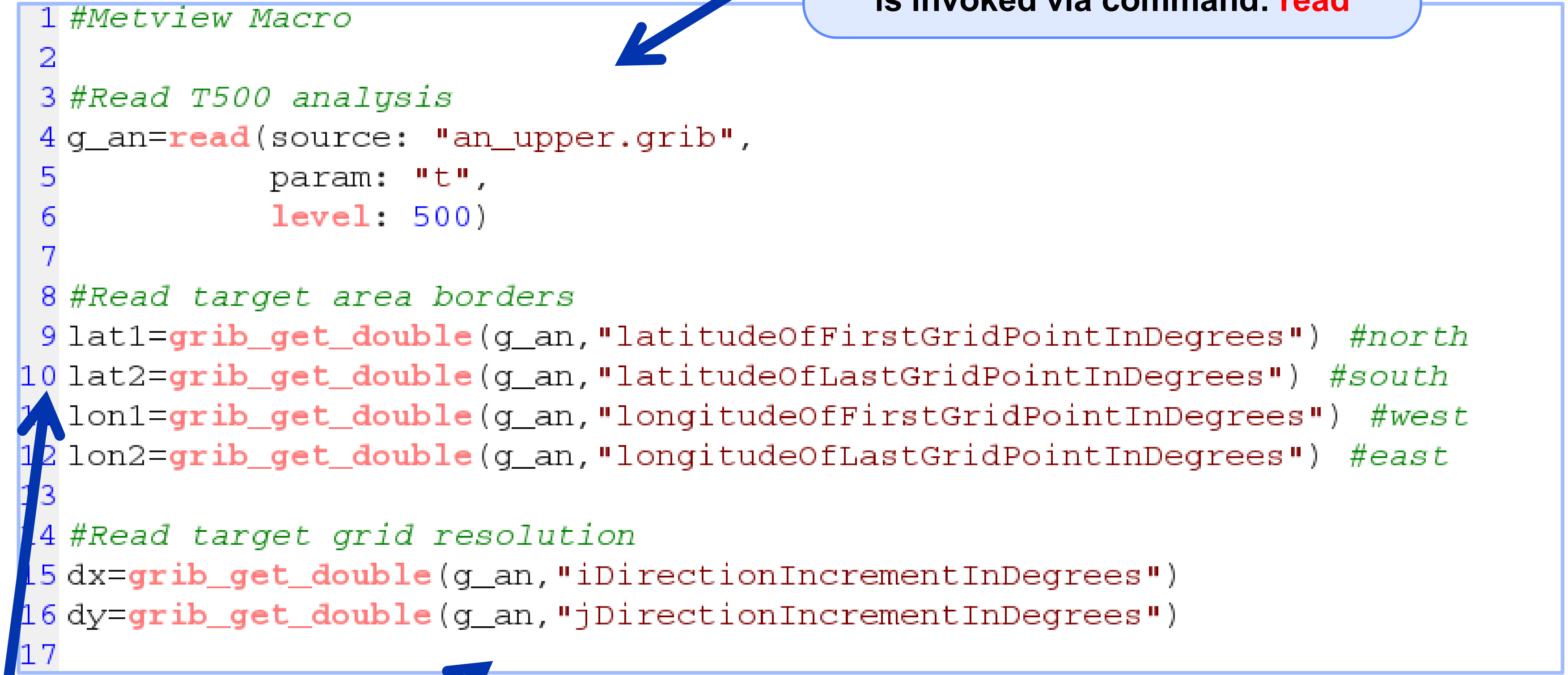
Analysis at D4
0.25x0.25 limited
area grid

- The steps involved:
  1. Filter T500 for the matching date and time
  2. Interpolate the global field to the LAM grid
  3. Compute the difference

We will write a macro!

Macro: Compute difference #1

Filter **parameter** and **level** from analysis. In Macro the GRIB Filter is invoked via command: **read**



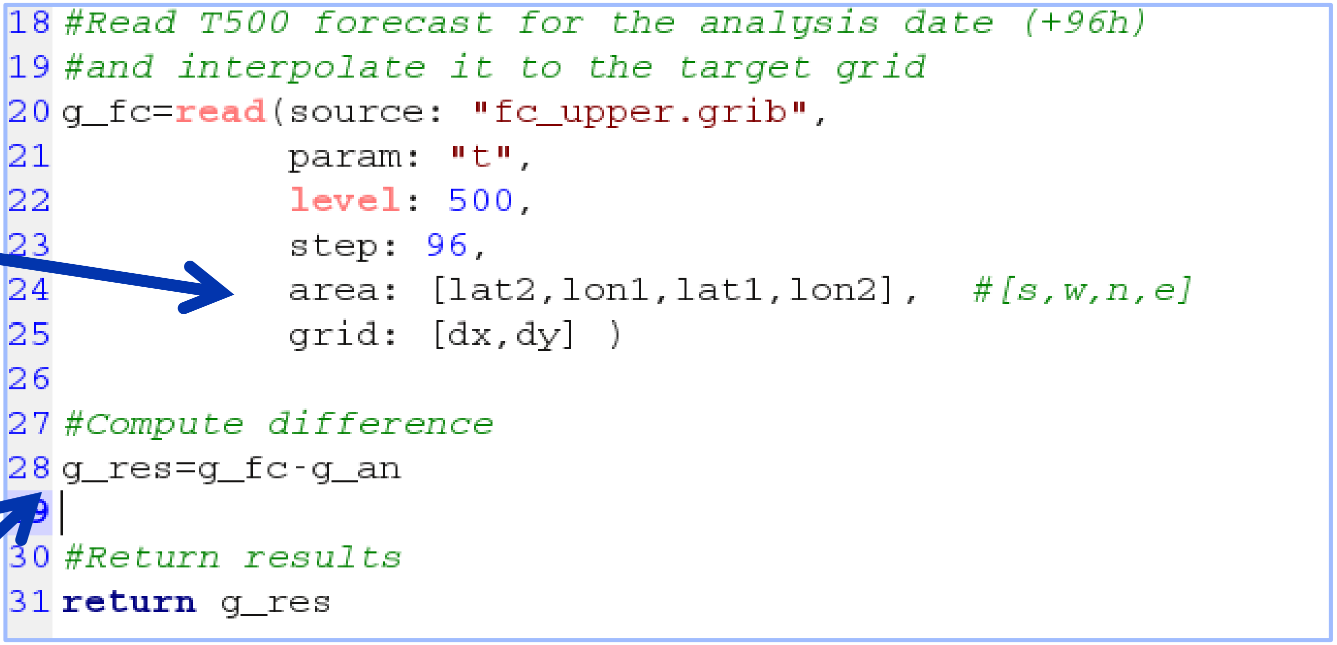
```
#Metview Macro

#Read T500 analysis
g_an=read(source: "an_upper.grib",
          param: "t",
          level: 500)

#Read target area borders
lat1=grib_get_double(g_an,"latitudeOfFirstGridPointInDegrees") #north
lat2=grib_get_double(g_an,"latitudeOfLastGridPointInDegrees") #south
lon1=grib_get_double(g_an,"longitudeOfFirstGridPointInDegrees") #west
lon2=grib_get_double(g_an,"longitudeOfLastGridPointInDegrees") #east

#Read target grid resolution
dx=grib_get_double(g_an,"iDirectionIncrementInDegrees")
dy=grib_get_double(g_an,"jDirectionIncrementInDegrees")

```

Here we read a set of GRIB API keys

Macro: Compute difference #2

Filter parameter and level and step from forecast

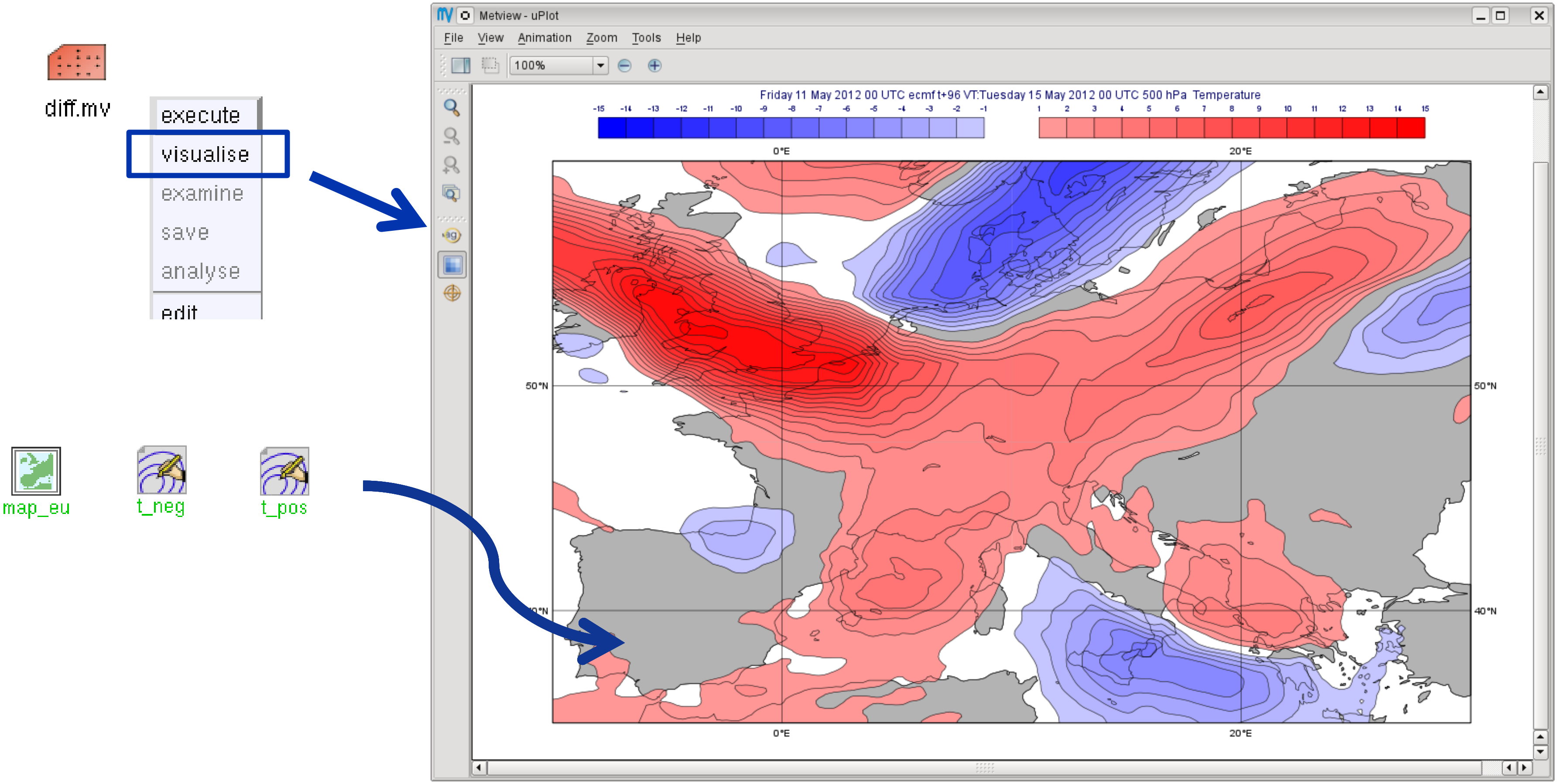
Interpolation



```
#Read T500 forecast for the analysis date (+96h)
#and interpolate it to the target grid
g_fc=read(source: "fc_upper.grib",
          param: "t",
          level: 500,
          step: 96,
          area: [lat2,lon1,lat1,lon2],   #[s,w,n,e]
          grid: [dx,dy] )

#Compute difference
g_res=g_fc-g_an

#Return results
return g_res
```

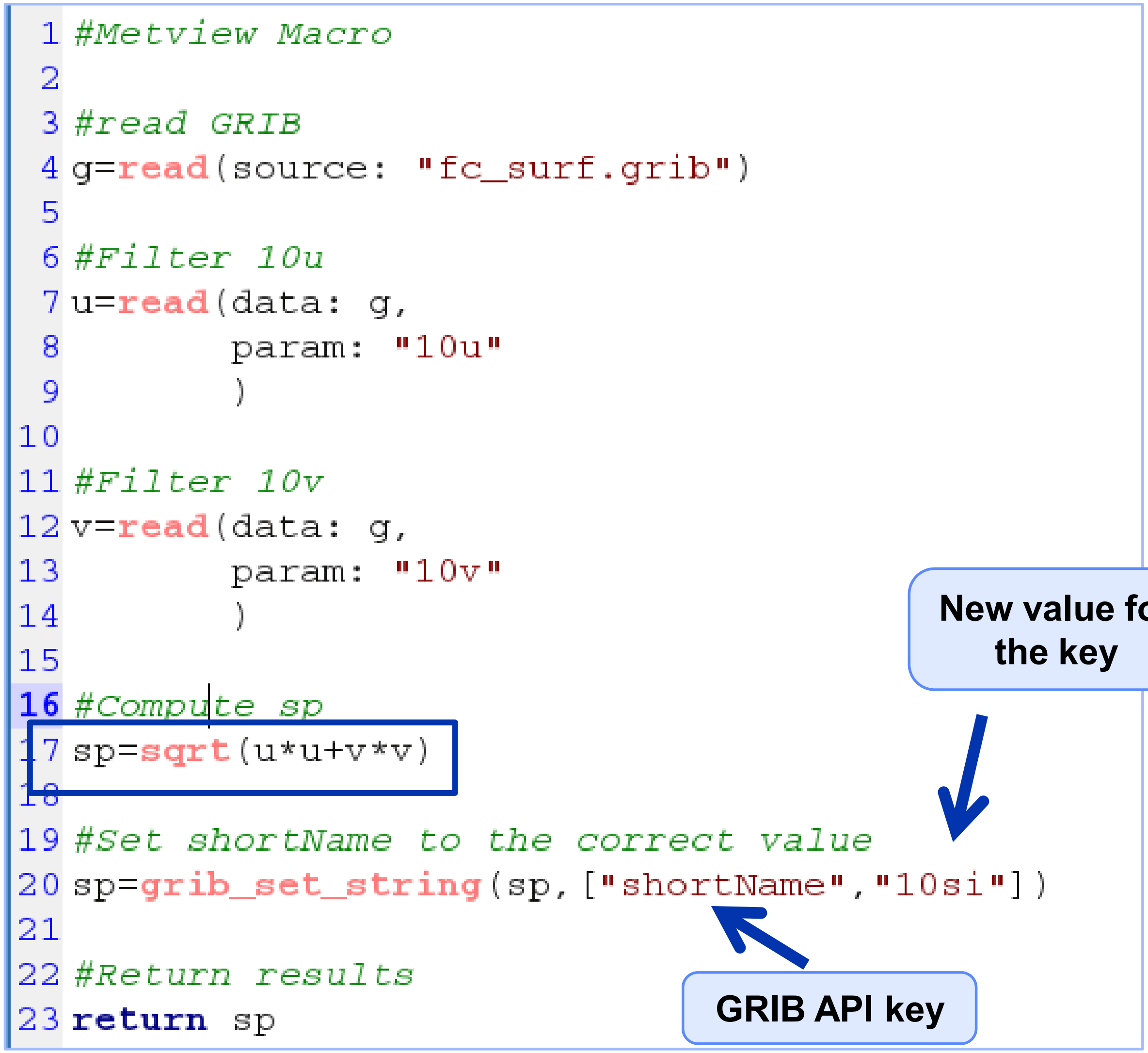
Difference operator only works between grids with the same number of points

Macro: Compute difference #2



Macro usage: compute wind speed

Example: compute 10 windspeed from u and v components



```
#Metview Macro

#read GRIB
g=read(source: "fc_surf.grib")

#Filter 10u
u=read(data: g,
       param: "10u"
       )

#Filter 10v
v=read(data: g,
       param: "10v"
       )

#Compute sp
sp=sqrt(u*u+v*v)

#Set shortName to the correct value
sp=grib_set_string(sp,["shortName","10si"])

#Return results
return sp
```

Fieldset operation

Here we set the GRIB header

New value for the key

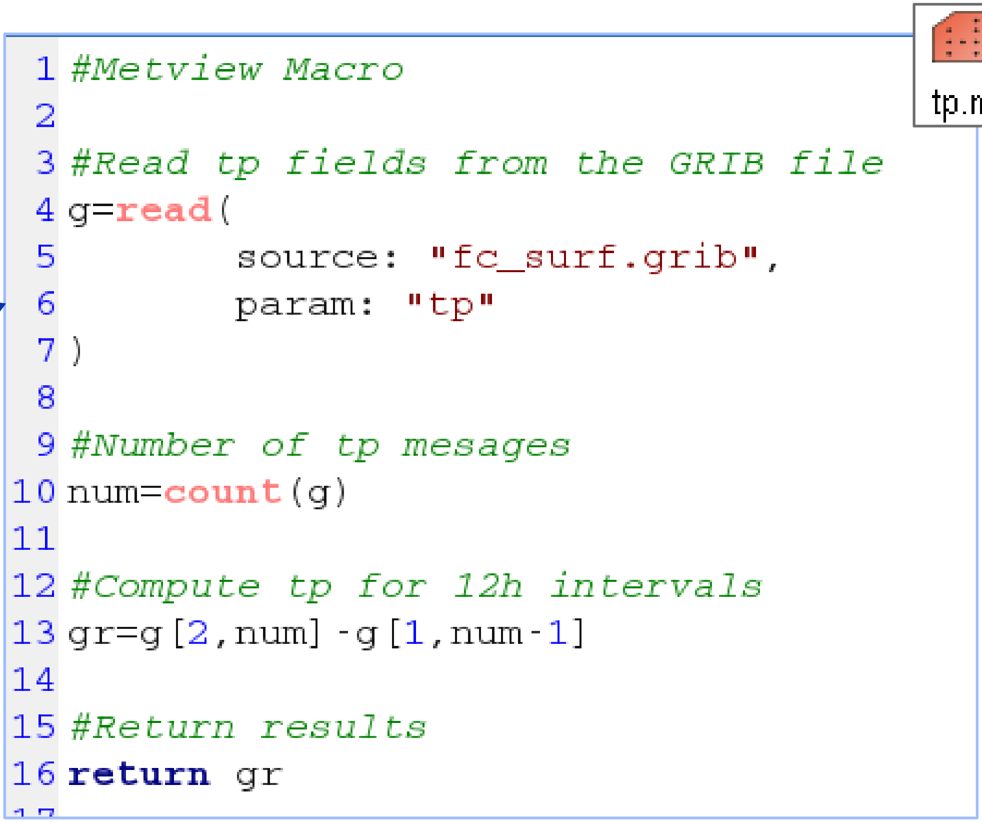
GRIB API key

Macro usage: compute precipitation for intervals

- Precipitation is often stored as an accumulated quantity
- We want to see precipitation for a given interval (e.g. 12h, 24h)

Example: compute precipitation for 12 h intervals from file fc_surf.grib

tp.mv

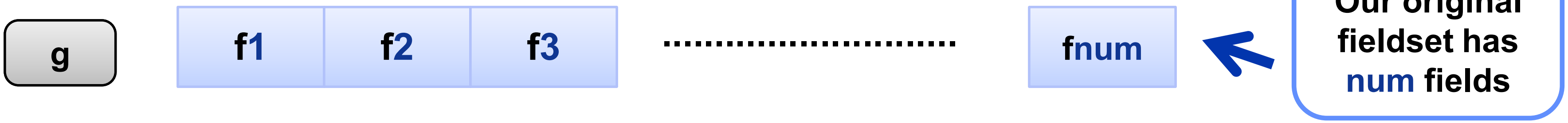


```
#Metview Macro

#Read tp fields from the GRIB file
g=read(
        source: "fc_surf.grib",
        param: "tp"
)

#Number of tp mesages
num=count(g)

#Compute tp for 12h intervals
gr=g[2,num]-g[1,num-1]

#Return results
return gr
```

We filter tp

Operation between slices of fieldsets: we compute the difference of adjacent fields

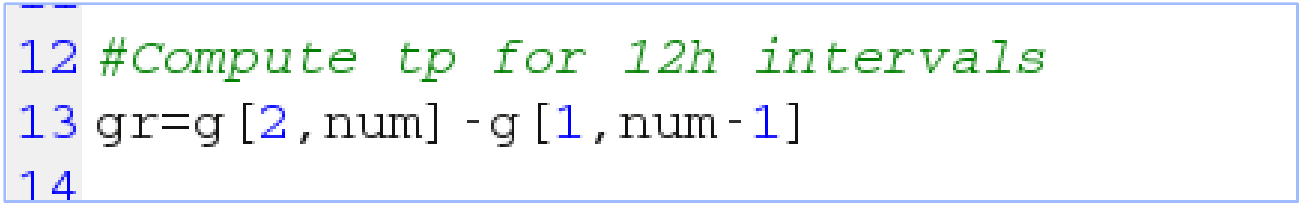
Precipitation computation explained



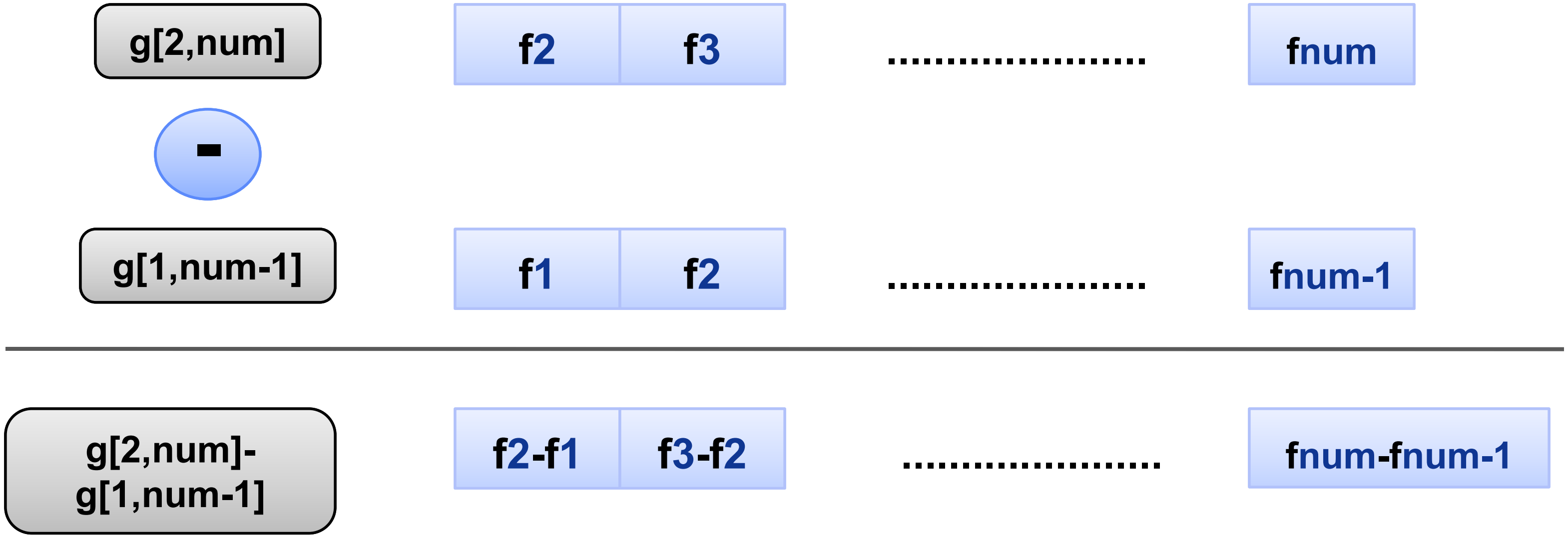
g

f1 f2 f3 .... fnum

Our original fieldset has num fields



```
#Compute tp for 12h intervals
gr=g[2,num]-g[1,num-1]
```



g[2,num]

f2 f3 .... fnum

-

g[1,num-1]

f1 f2 .... fnum-1

g[2,num]-g[1,num-1]

f2-f1 f3-f2 .... fnum-fnum-1

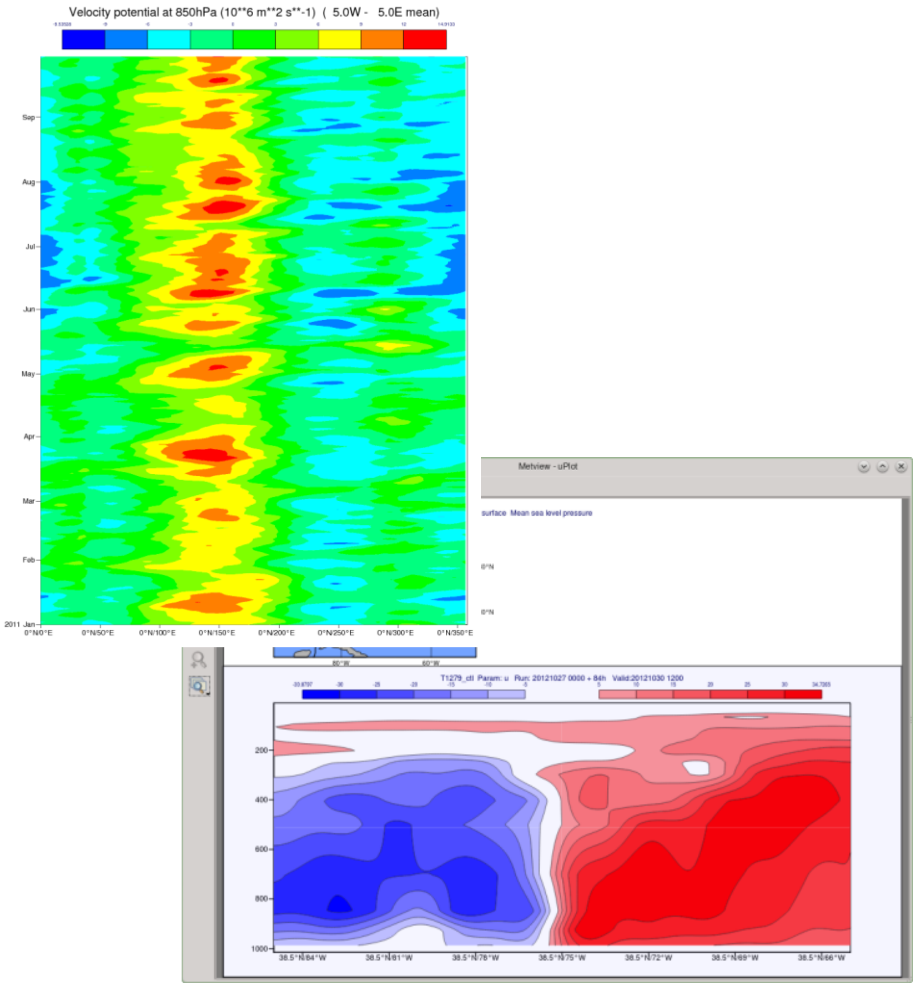
Macro usage: more functions

- A rich set of macro functions exists for GRIB. A few examples:
  - latitudes(), longitudes(), values(): read the latitudes, longitudes and values of a field into vectors (in-memory arrays)
  - average(): compute average
  - mask(): set field values to 0 or 1 using an area mask
  - bitmap(): assign missing values to a field using a mask
  - nobitmap(): replace missing values

See Macro Tutorial 3 for some elaborated examples, such as masking one field based on the values of another (e.g. apply a land sea mask to a field to remove (i.e. to bitmap) points over sea)

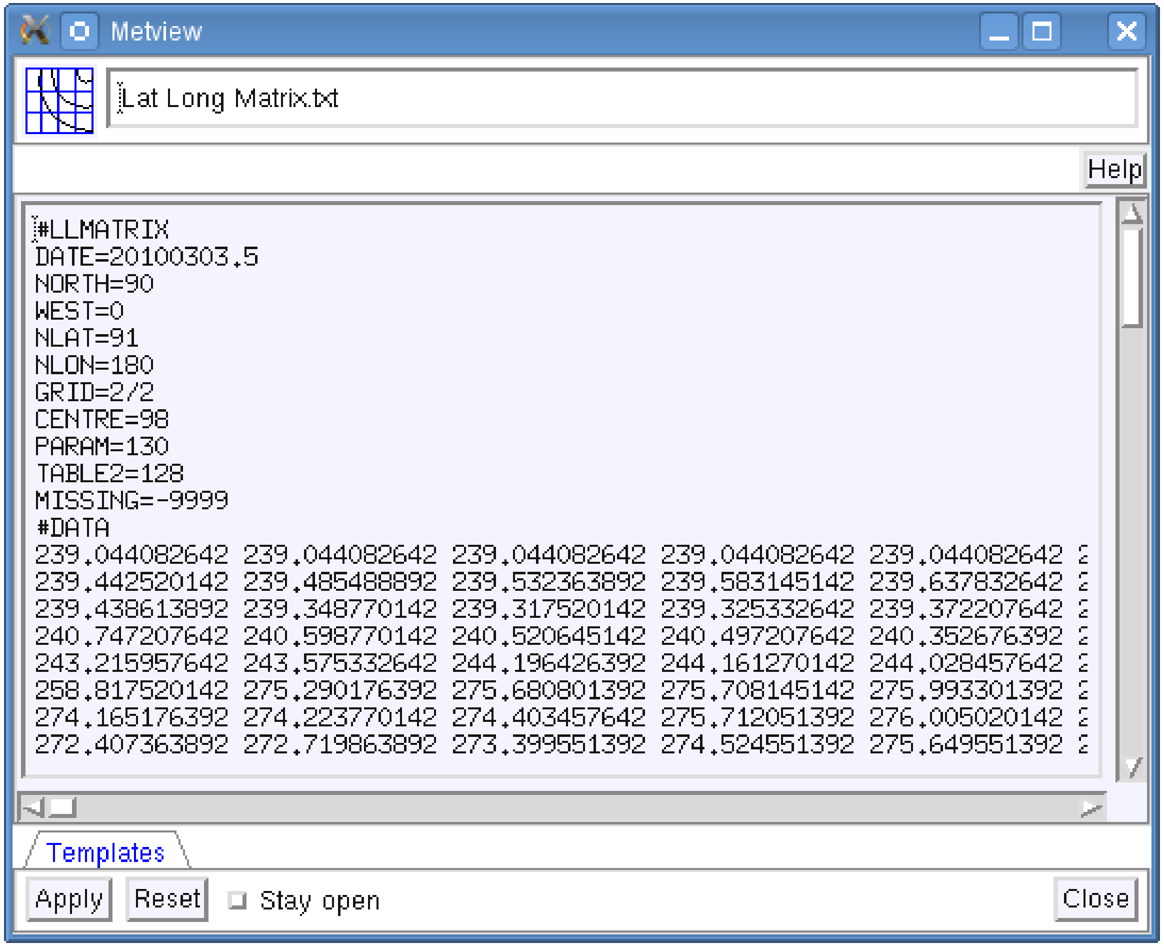
Complex plot types for GRIB

- These plots require data extraction from multiple fields and some computations as well
- There are a set of GRIB specific icons to generate:
  - Cross sections
  - Hovmøller diagrams
  - Zonal mean plots
  - Vertical profiles



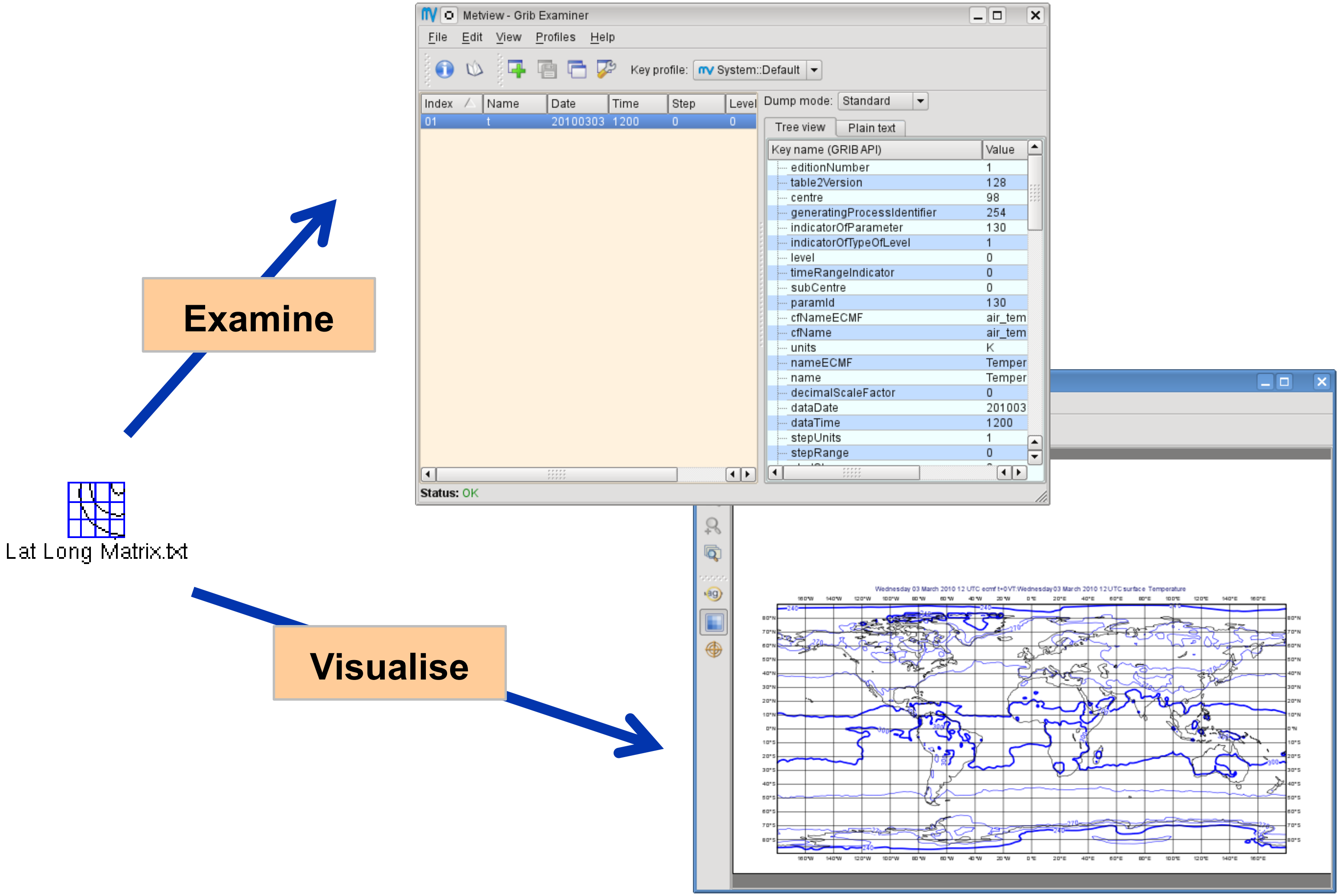
Lat Long Matrix

- Metview's ASCII format for gridded data
- Turned into GRIB internally
- Can be edited as a text file



```
#LLMATRIX
DATE=20100303.5
NORTH=90
WEST=0
NLAT=91
NLON=180
GRID=2/2
CENTRE=98
PARAM=130
TABLE2=128
MISSING=-9999
#DATA
239.044082642 239.044082642 239.044082642 239.044082642 239.044082642
239.442520142 239.485488892 239.532363892 239.583145142 239.637832642
239.438613892 239.348770142 239.317520142 239.325332642 239.372207642
240.747207642 240.598770142 240.520645142 240.497207642 240.352676392
243.215957642 243.575332642 244.196426392 244.161270142 244.028457642
258.817520142 275.290176392 275.680801392 275.708145142 275.993301392
274.165176392 274.223770142 274.403457642 275.712051392 276.005020142
272.407363892 272.719863892 273.399551392 274.524551392 275.649551392
```

Lat Long Matrix – Behaves like a GRIB



Examine

Visualise

Lat Long Matrix.txt

Metview - Grib Examiner

Key profile: System::Default

Index	Name	Date	Time	Step	Level
01	t	20100303	1200	0	0

Dump mode: Standard

Key name (GRIB API)	Value
editionNumber	1
table2Version	128
centre	98
generatingProcessIdentifier	254
indicatorOfParameter	130
indicatorOfTypeOfLevel	1
level	0
timeRangeIndicator	0
subCentre	0
paramId	130
cfNameECMF	air_tem
cfName	air_tem
units	K
nameECMF	Temper
name	Temper
decimalScaleFactor	0
dataDate	201003
dataTime	1200
stepUnits	1
stepRange	0

Status: OK

Wednesday 03 March 2010 12 UTC ecmf t+0 VT:Wednesday 03 March 2010 12 UTC surface Temperature

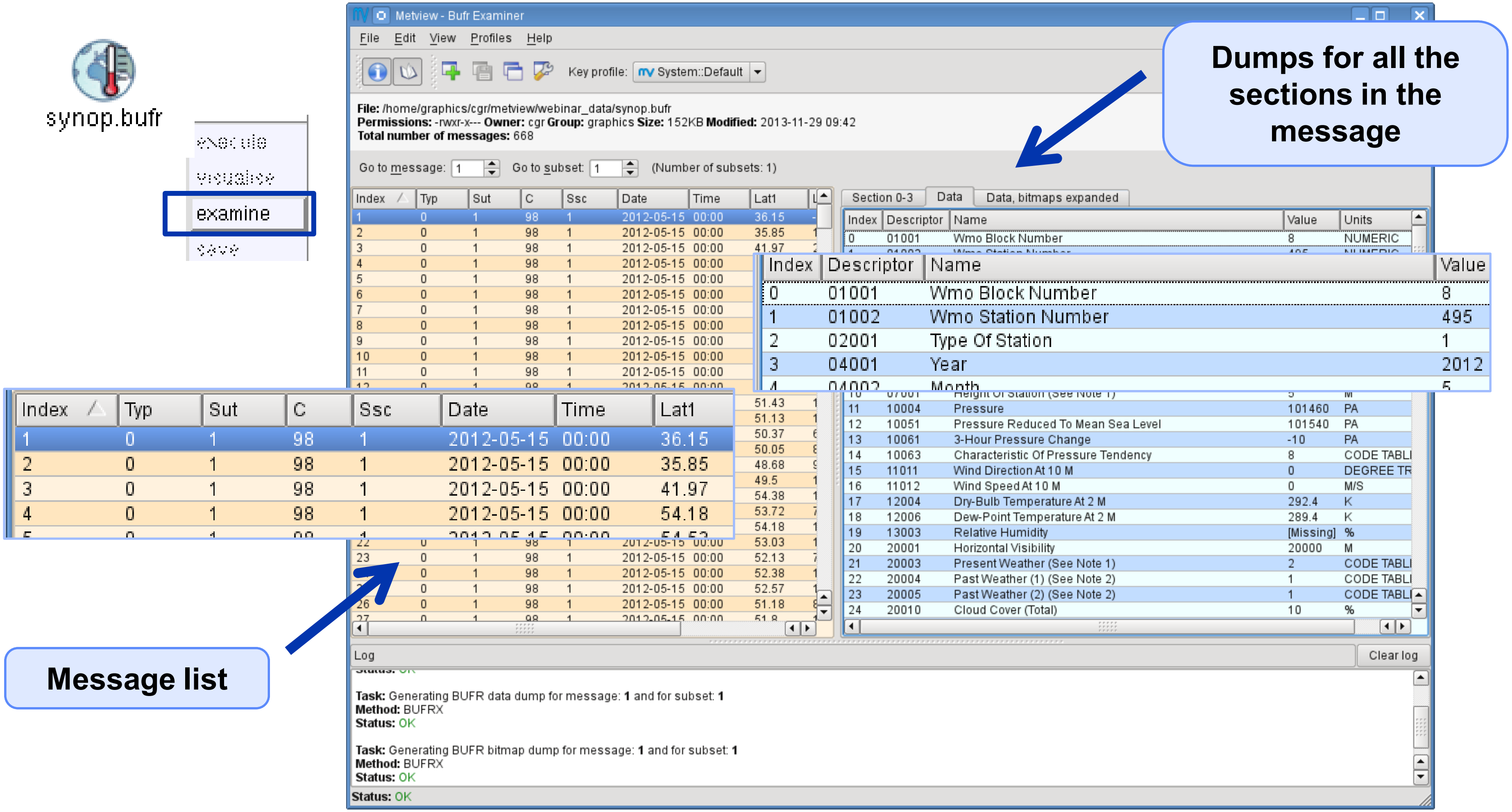
BUFR

- WMO's binary format for observation data
- Metview offers a high level interface to work with BUFR
- Internally we use BUFRDC (part of EMOS lib) to decode BUFR messages

There is a BUFR tutorial available on the Metview web page

BUFR Examiner

- BUFRs contents can be checked with the **BUFR Examiner**



Dumps for all the sections in the message

Message list

File: /home/graphics/cgr/metview/webinar_data/synop.bufr
Permissions: -rwxr-x--- **Owner:** cgr **Group:** graphics **Size:** 152KB **Modified:** 2013-11-29 09:42
Total number of messages: 668

Go to message: 1 Go to subset: 1 (Number of subsets: 1)

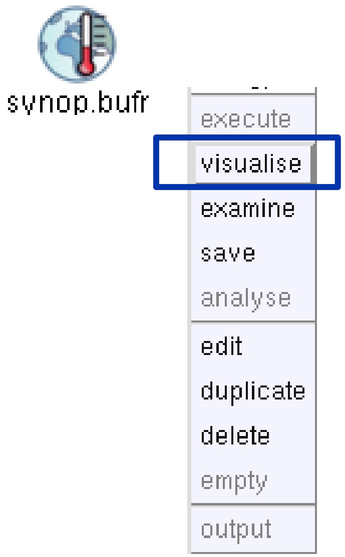
Index	Typ	Sut	C	Ssc	Date	Time	Lat1
1	0	1	98	1	2012-05-15	00:00	36.15
2	0	1	98	1	2012-05-15	00:00	35.85
3	0	1	98	1	2012-05-15	00:00	41.97
4	0	1	98	1	2012-05-15	00:00	54.18
5	0	1	98	1	2012-05-15	00:00	
6	0	1	98	1	2012-05-15	00:00	
7	0	1	98	1	2012-05-15	00:00	
8	0	1	98	1	2012-05-15	00:00	
9	0	1	98	1	2012-05-15	00:00	
10	0	1	98	1	2012-05-15	00:00	
11	0	1	98	1	2012-05-15	00:00	
							51.43
							51.13
							50.37
							50.05
							48.68
							49.5
							54.38
							53.72
							54.18
22	0	1	98	1	2012-05-15	00:00	53.03
23	0	1	98	1	2012-05-15	00:00	52.13
	0	1	98	1	2012-05-15	00:00	52.38
	0	1	98	1	2012-05-15	00:00	52.57
26	0	1	98	1	2012-05-15	00:00	51.18
27	0	1	98	1	2012-05-15	00:00	51.8

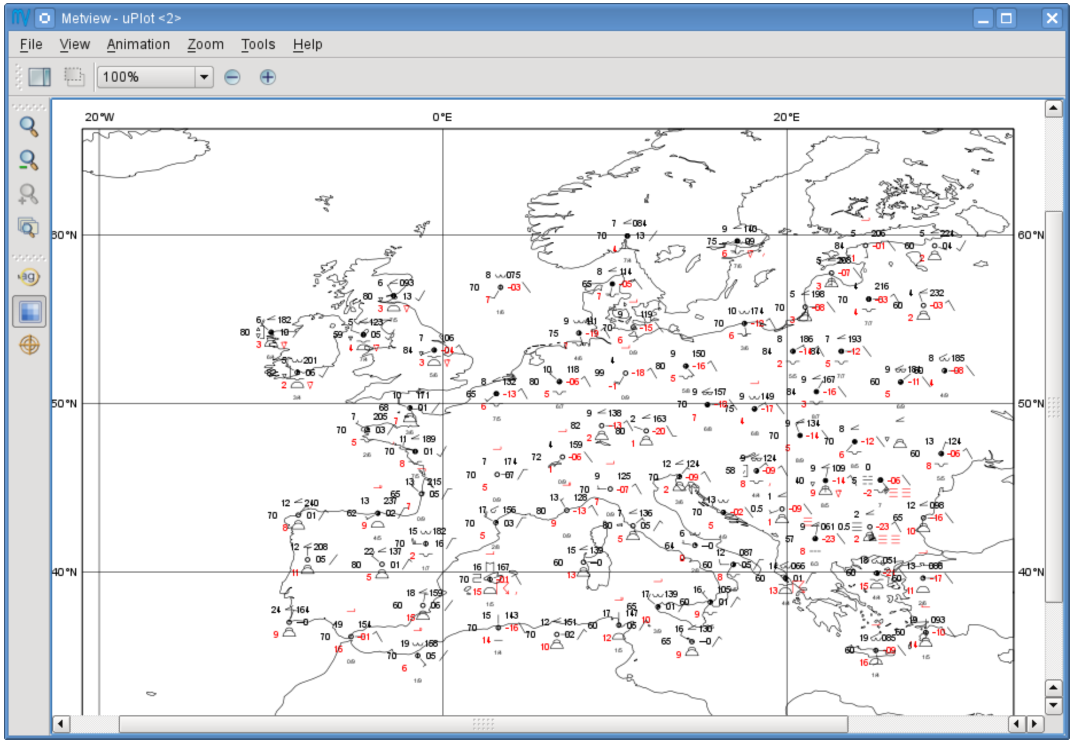
Index	Descriptor	Name	Value	Units
0	01001	Wmo Block Number	8	NUMERIC
1	01002	Wmo Station Number	495	NUMERIC
2	02001	Type Of Station	1	
3	04001	Year	2012	
4	04002	Month	5	
10	07001	Height Of Station (See Note 1)	5	M
11	10004	Pressure	101460	PA
12	10051	Pressure Reduced To Mean Sea Level	101540	PA
13	10061	3-Hour Pressure Change	-10	PA
14	10063	Characteristic Of Pressure Tendency	8	CODE TABLI
15	11011	Wind Direction At 10 M	0	DEGREE TR
16	11012	Wind Speed At 10 M	0	M/S
17	12004	Dry-Bulb Temperature At 2 M	292.4	K
18	12006	Dew-Point Temperature At 2 M	289.4	K
19	13003	Relative Humidity	[Missing]	%
20	20001	Horizontal Visibility	20000	M
21	20003	Present Weather (See Note 1)	2	CODE TABLI
22	20004	Past Weather (1) (See Note 2)	1	CODE TABLI
23	20005	Past Weather (2) (See Note 2)	1	CODE TABLI
24	20010	Cloud Cover (Total)	10	%

```
Status: OK

Task: Generating BUFR data dump for message: 1 and for subset: 1
Method: BUFRX
Status: OK

Task: Generating BUFR bitmap dump for message: 1 and for subset: 1
Method: BUFRX
Status: OK
Status: OK
```

BUFR Plotting

- We can directly visualise BUFR files with conventional observations (e.g. SYNOP)

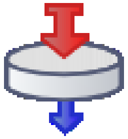




BUFR: Accessing data

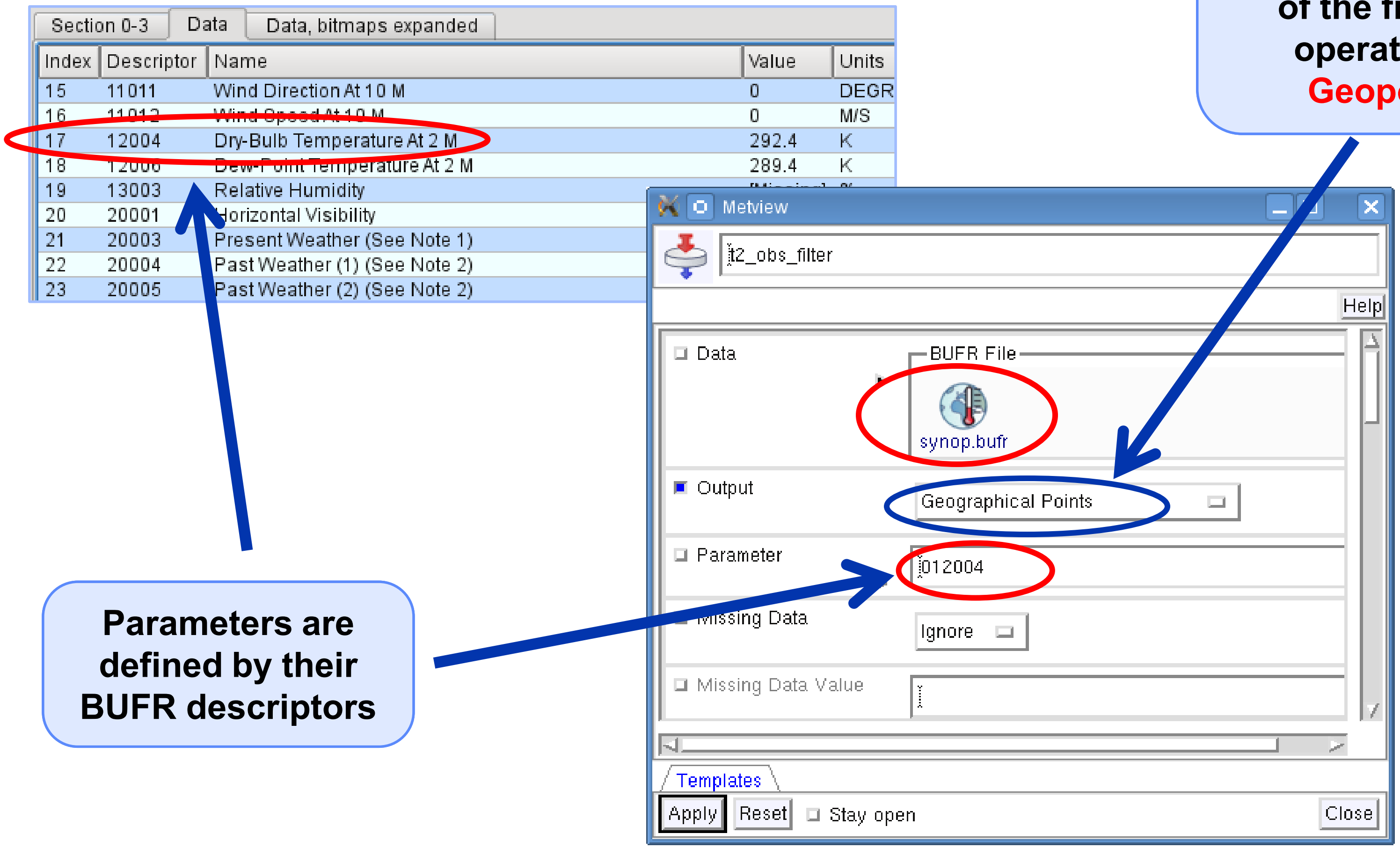
Example: extract and plot T2 with symbol plotting from file synop.bufr

- We need to use the Observation Filter icon



- It can perform filtering according to parameter, level, area, time, channel etc.

BUFR: Filtering



Section 0-3 | Data | Data, bitmaps expanded

Index	Descriptor	Name	Value	Units
15	11011	Wind Direction At 10 M	0	DEGR
16	11012	Wind Speed At 10 M	0	M/S
17	12004	Dry-Bulb Temperature At 2 M	292.4	K
18	12006	Dew-Point Temperature At 2 M	289.4	K
19	13003	Relative Humidity	[Missing]	%
20	20001	Horizontal Visibility		
21	20003	Present Weather (See Note 1)		
22	20004	Past Weather (1) (See Note 2)		
23	20005	Past Weather (2) (See Note 2)		

Metview

t2_obs_filter

□ Data — BUFR File: synop.bufr

■ Output: Geographical Points

□ Parameter: 012004

□ Missing Data: Ignore

□ Missing Data Value:

We set the output of the filtering operation to **Geopoints**

Parameters are defined by their BUFR descriptors

We can save the result into a file

t2.gpt

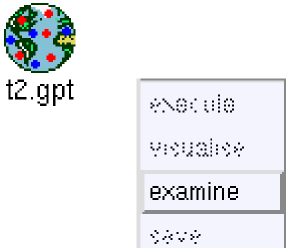
Geopoints

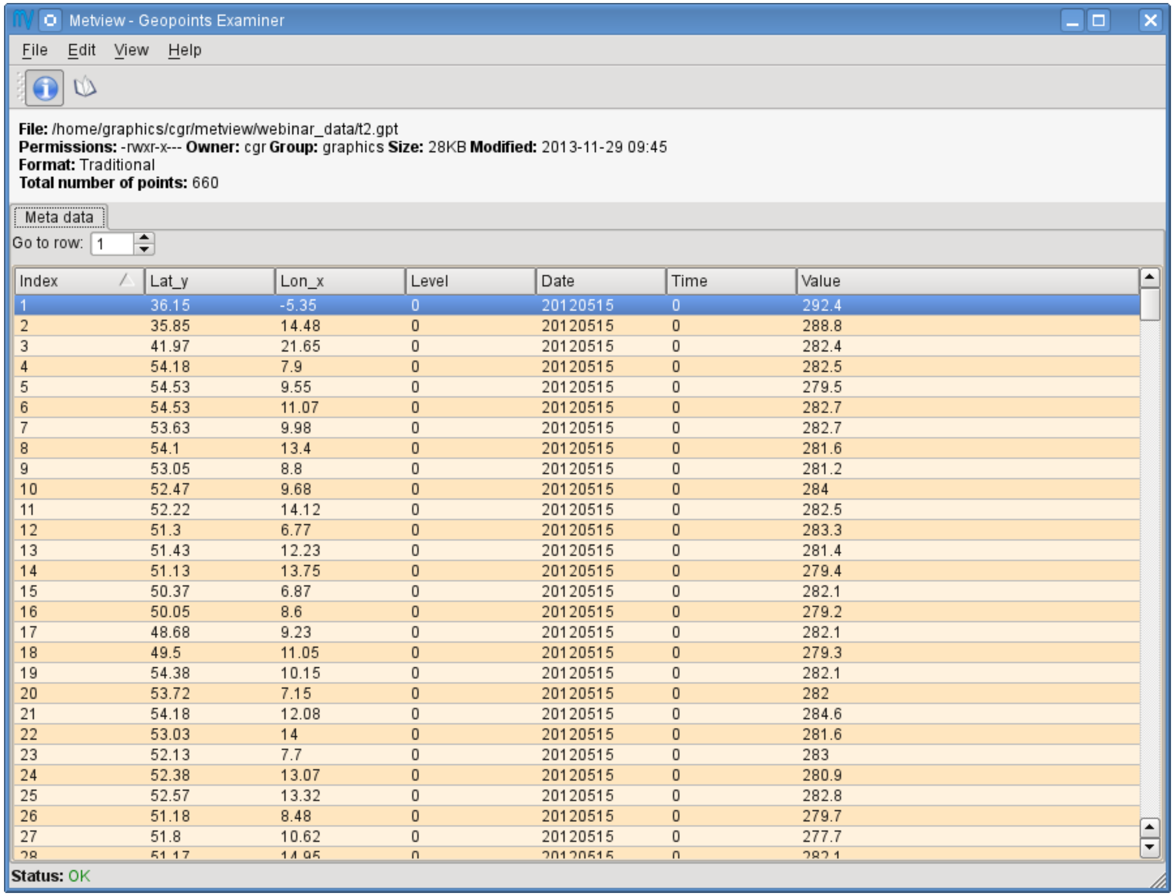
- Metview’s custom format to store scattered geo-referenced data
- ASCII files with 4 different types: The default is shown here:

```
#GEO
PARAMETER = 12004
lat      long     level    date       time     value
#DATA
36.15    -5.35    0        20120515   0000     292.4
35.85    14.48    0        20120515   0000     288.8
41.97    21.65    0        20120515   0000     282.4
```

Geopoints Examiner

- Geopoints contents can be checked with the Geopoints Examiner
- This is how the result of the BUFR filtering looks like





File: /home/graphics/cgr/metview/webinar_data/t2.gpt
Permissions: -rwxr-x--- **Owner:** cgr **Group:** graphics **Size:** 28KB **Modified:** 2013-11-29 09:45
Format: Traditional
Total number of points: 660

Go to row: 1

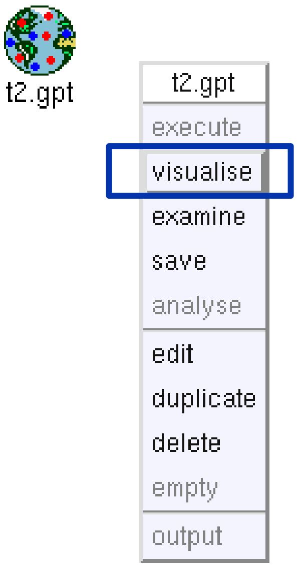
Index	Lat_y	Lon_x	Level	Date	Time	Value
1	36.15	-5.35	0	20120515	0	292.4
2	35.85	14.48	0	20120515	0	288.8
3	41.97	21.65	0	20120515	0	282.4
4	54.18	7.9	0	20120515	0	282.5
5	54.53	9.55	0	20120515	0	279.5
6	54.53	11.07	0	20120515	0	282.7
7	53.63	9.98	0	20120515	0	282.7
8	54.1	13.4	0	20120515	0	281.6
9	53.05	8.8	0	20120515	0	281.2
10	52.47	9.68	0	20120515	0	284
11	52.22	14.12	0	20120515	0	282.5
12	51.3	6.77	0	20120515	0	283.3
13	51.43	12.23	0	20120515	0	281.4
14	51.13	13.75	0	20120515	0	279.4
15	50.37	6.87	0	20120515	0	282.1
16	50.05	8.6	0	20120515	0	279.2
17	48.68	9.23	0	20120515	0	282.1
18	49.5	11.05	0	20120515	0	279.3
19	54.38	10.15	0	20120515	0	282.1
20	53.72	7.15	0	20120515	0	282
21	54.18	12.08	0	20120515	0	284.6
22	53.03	14	0	20120515	0	281.6
23	52.13	7.7	0	20120515	0	283
24	52.38	13.07	0	20120515	0	280.9
25	52.57	13.32	0	20120515	0	282.8
26	51.18	8.48	0	20120515	0	279.7
27	51.8	10.62	0	20120515	0	277.7
28	51.17	14.95	0	20120515	0	282.1

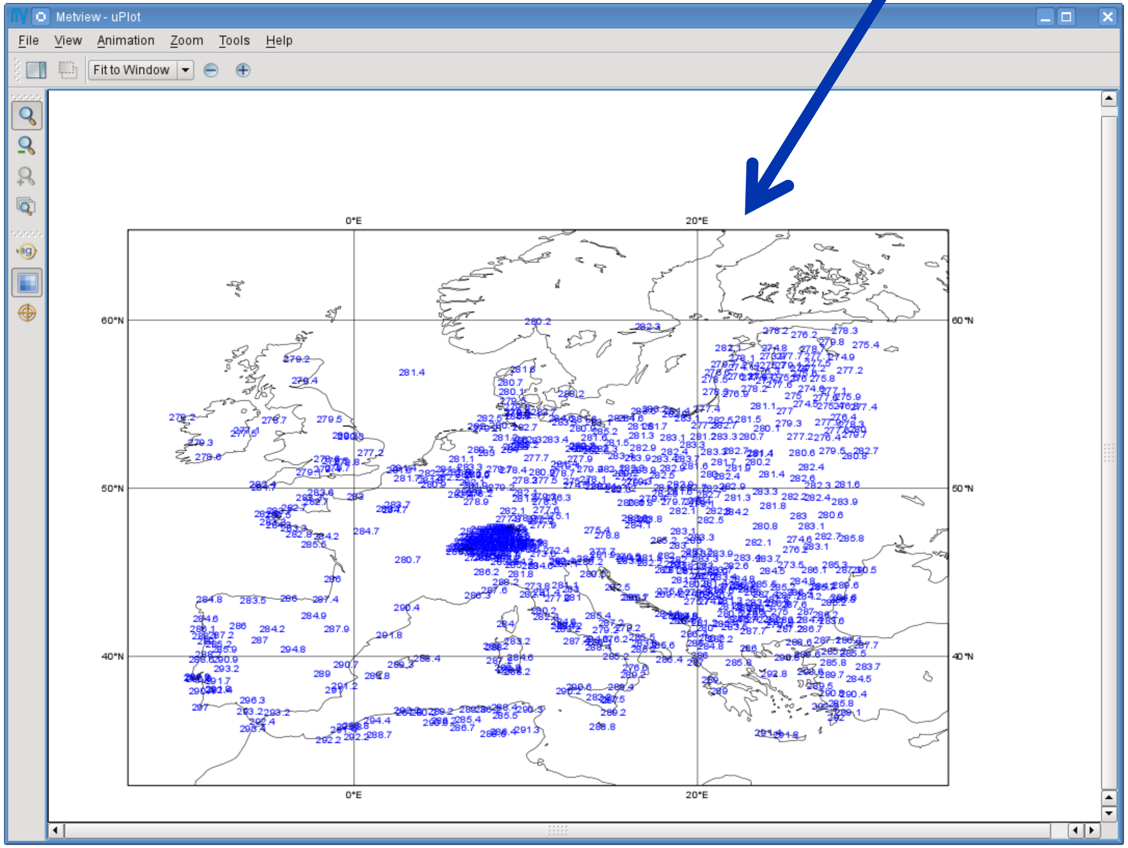
Status: OK

Geopoints Plotting

- We can directly visualise Geopoints
- It is based on symbol plotting

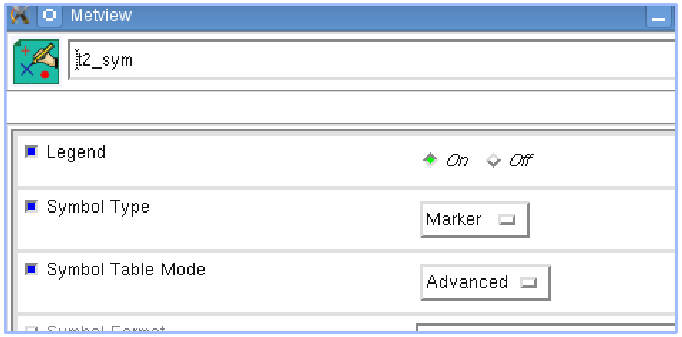
By default the numbers are plotted to the map





Customisation with Symbol Plotting

- The **Symbol Plotting** icon offers a large number of options for plot customisation
- We can use the **Advanced Table Mode** to define a nice colour palette between the min and max colours (just like for Contouring)



Metview

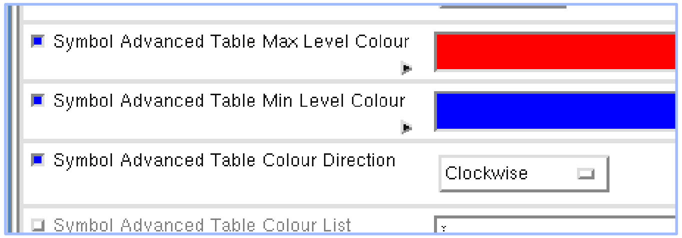
t2_sym

■ Legend ◆ On ◇ Off

■ Symbol Type: Marker

■ Symbol Table Mode: Advanced

□ Symbol Format

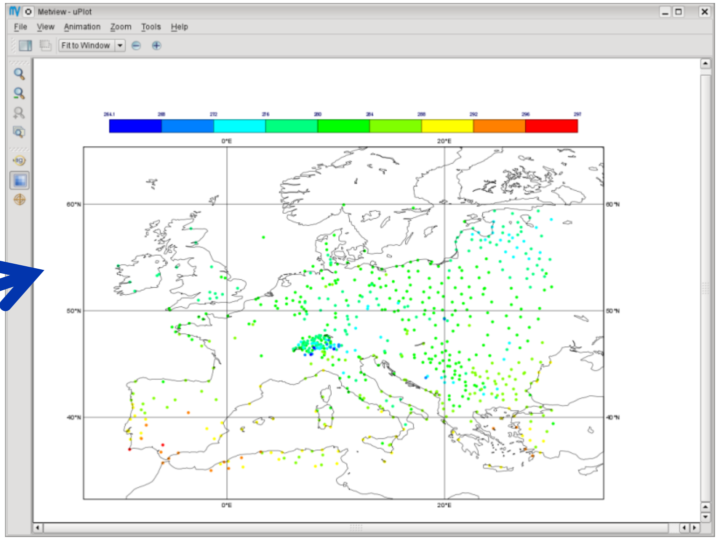


■ Symbol Advanced Table Max Level Colour: 🟥

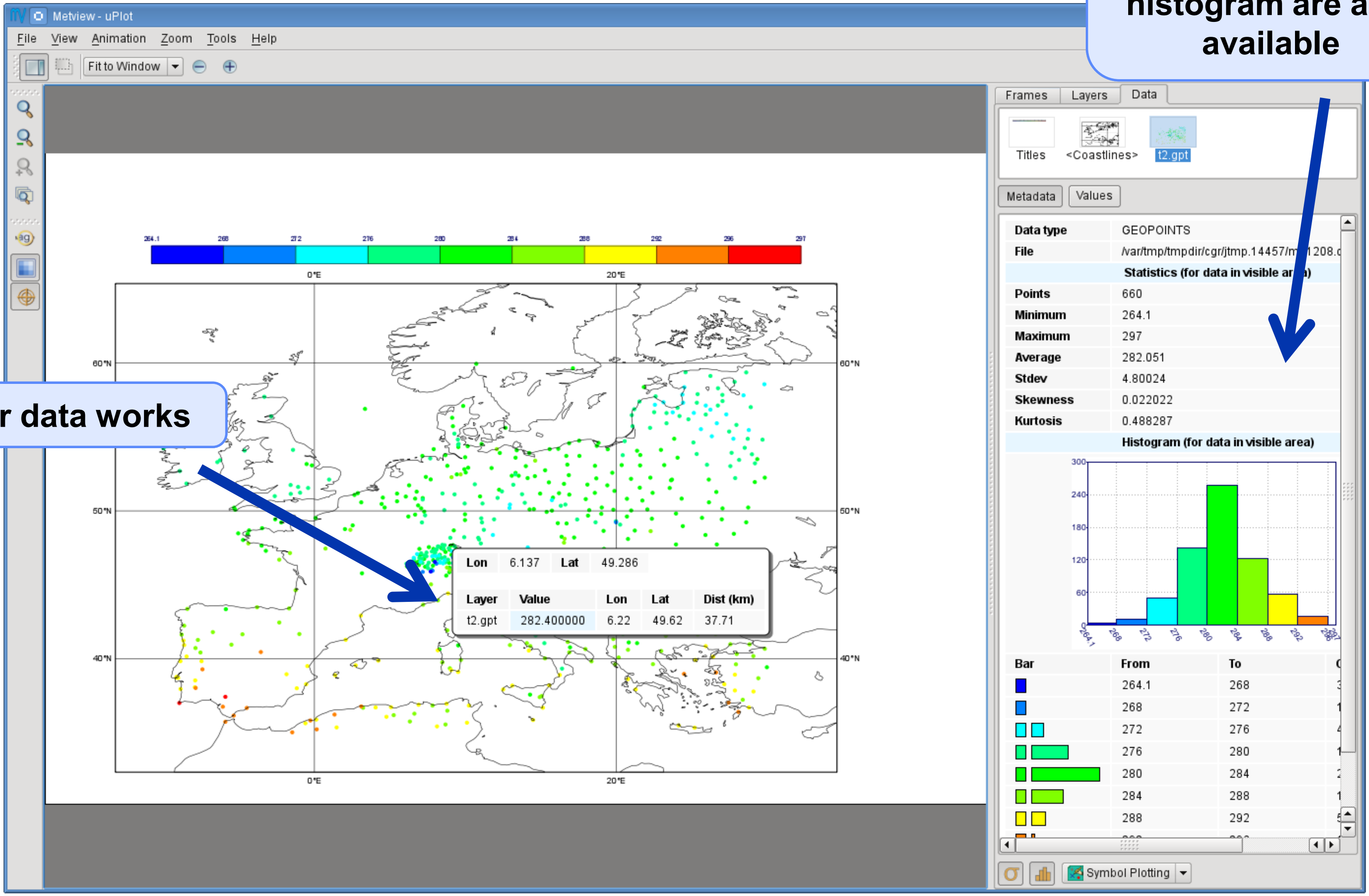
■ Symbol Advanced Table Min Level Colour: 🟦

■ Symbol Advanced Table Colour Direction: Clockwise

□ Symbol Advanced Table Colour List



Geopoints Plotting



Lon	6.137	Lat	49.286

Layer	Value	Lon	Lat	Dist (km)
t2.gpt	282.400000	6.22	49.62	37.71

Data type	GEOPOINTS
File	/var/tmp/tmpdir/cgr/jtmp.14457/m 1208.c
	Statistics (for data in visible area)
Points	660
Minimum	264.1
Maximum	297
Average	282.051
Stdev	4.80024
Skewness	0.022022
Kurtosis	0.488287
	Histogram (for data in visible area)

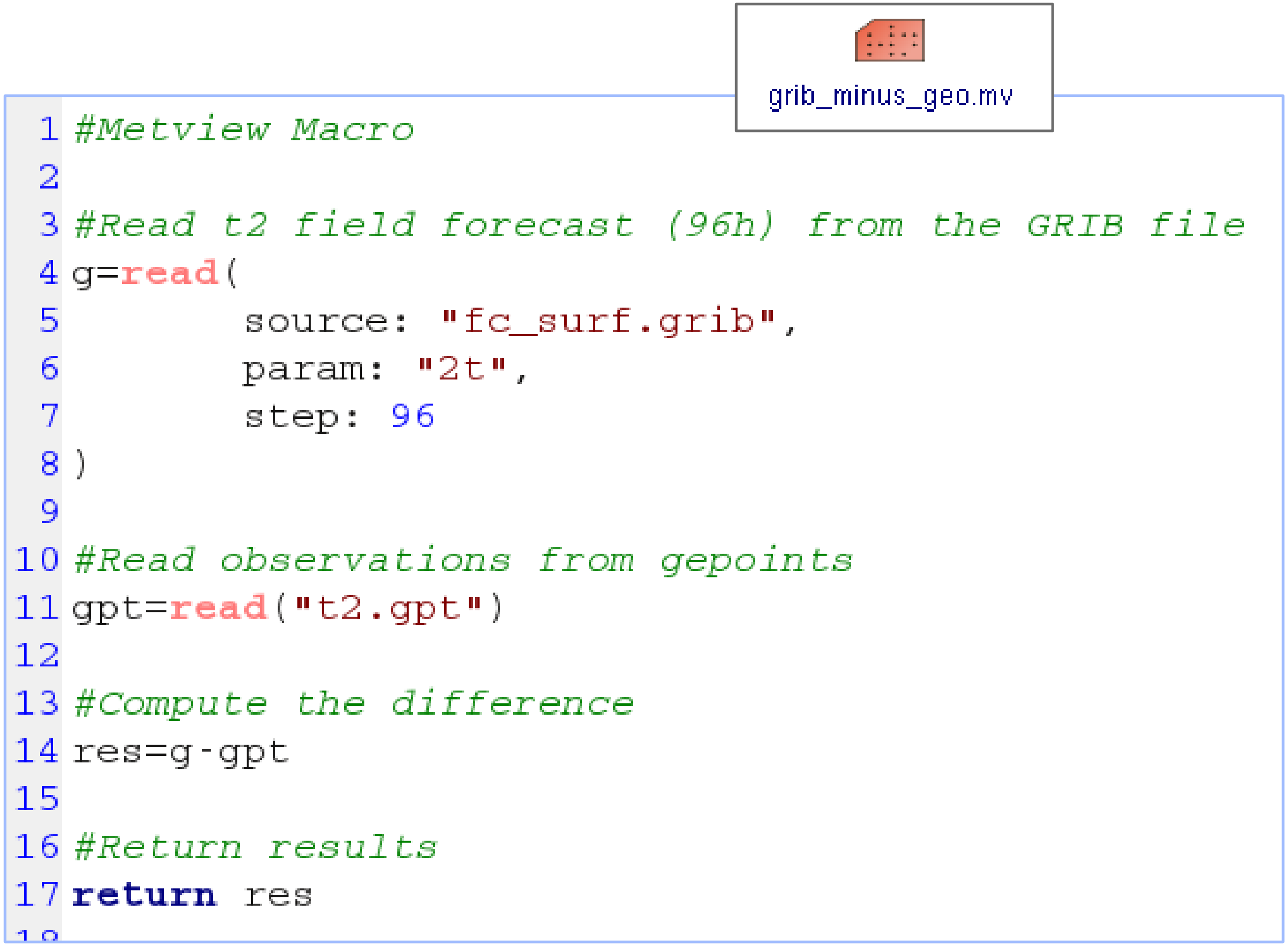
Bar	From	To	C
🟦	264.1	268	3
🟦	268	272	1
🟦	272	276	4
🟩	276	280	1
🟩	280	284	2
🟩	284	288	1
🟨	288	292	5

Statistics and histogram are also available

Cursor data works

Macro: difference between GRIB and Geopoints

Example:
compute the difference between the T2 forecast and observations



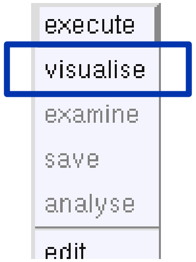
```
#Metview Macro

#Read t2 field forecast (96h) from the GRIB file
g=read(
        source: "fc_surf.grib",
        param: "2t",
        step: 96
)

#Read observations from gepoints
gpt=read("t2.gpt")

#Compute the difference
res=g-gpt

#Return results
return res
```

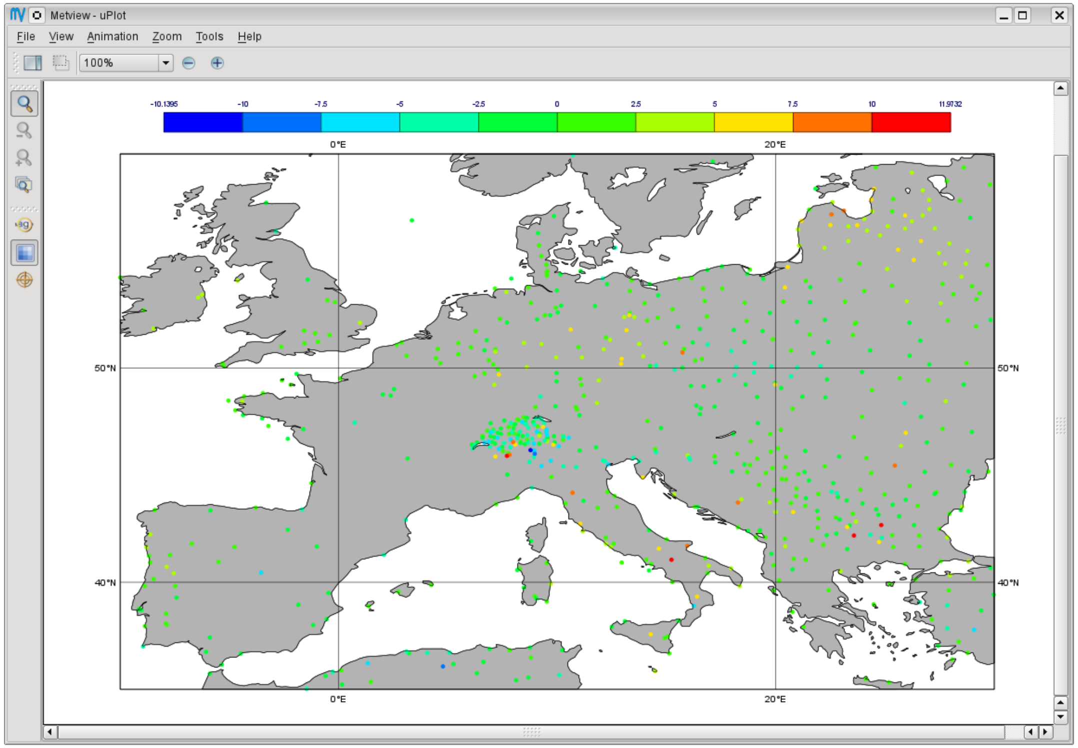
This step involves interpolation of the GRIB data to the Geopoints locations

The result is another Geopoints

Compute difference between GRIB and Geopoints

grib_minus_geo.mv





Forecast minus observation differences

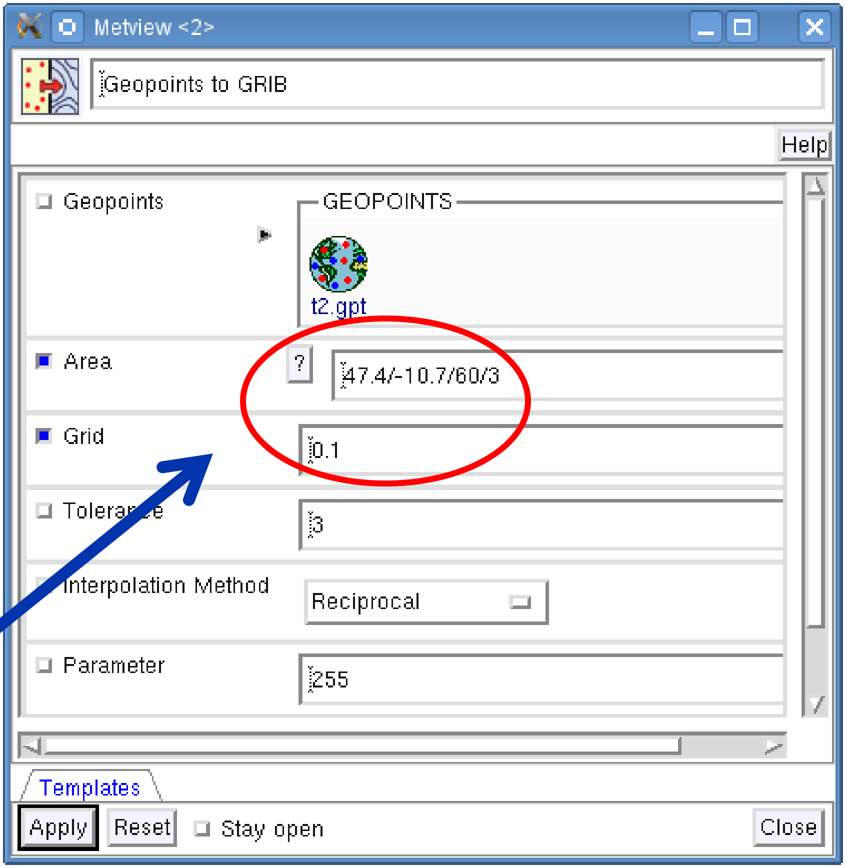
Geopoints to GRIB

Example: interpolate T2 observations onto a grid then apply contouring

- We need to use the **Geopoints to GRIB** icon

Geopoints to GRIB

- This icon interpolates Geopoints data onto a regular lat-lon grid and encodes it into GRIB



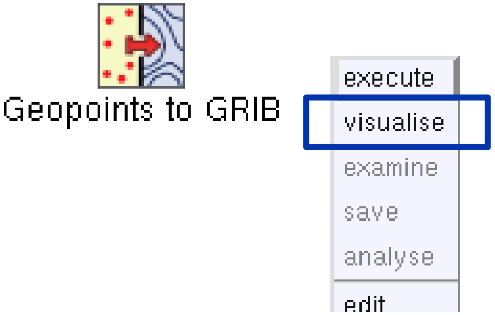
Geopoints to GRIB

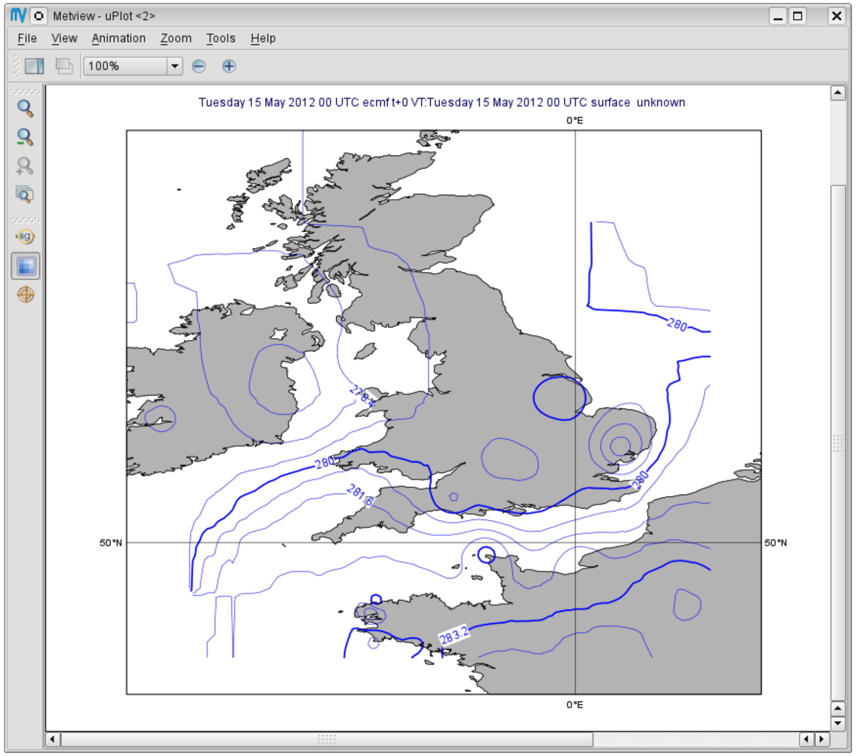
□ Geopoints	GEOPOINTS t2.gpt
■ Area	47.4/-10.7/60/3
■ Grid	0.1
□ Tolerance	3
□ Interpolation Method	Reciprocal
□ Parameter	255

□ Stay open

The grid definition

Geopoints to GRIB





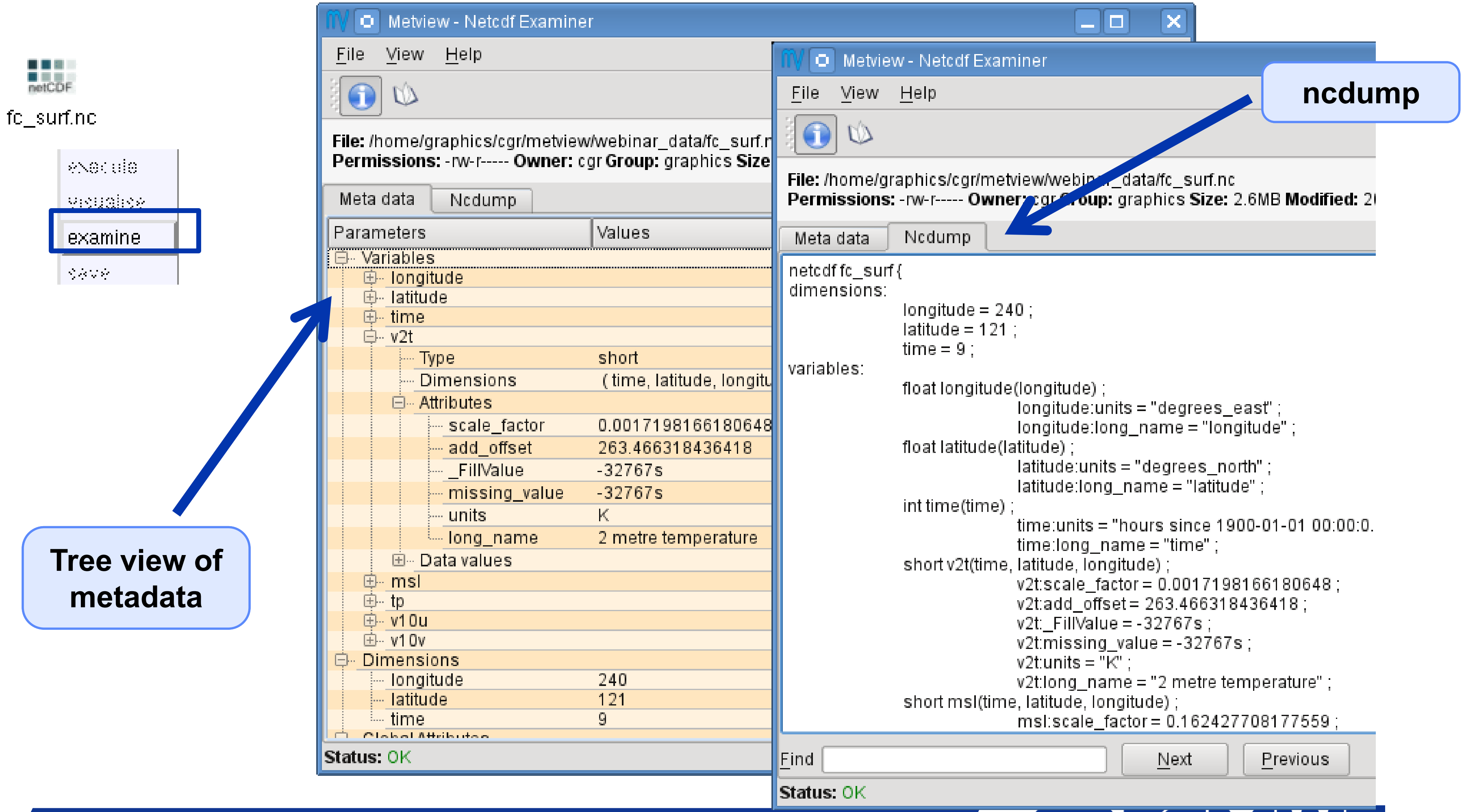
T2 observations interpolated onto a 0.1x0.1 grid

NetCDF

- UNIDATA’s binary format for multidimensional arrays
- Metview’s NetCDF plotting interface was added a few years ago

NetCDF Examiner

- NetCDF contents can be checked with the **NetCDF Examiner**



Tree view of metadata

ncdump

File: /home/graphics/cgr/metview/webinar_data/fc_surf.r
Permissions: -rw-r----- **Owner:** cgr **Group:** graphics **Size**

```
Parameters                  Values
⊟ Variables
    ⊞ longitude
    ⊞ latitude
    ⊞ time
    ⊟ v2t
        Type                short
        Dimensions          ( time, latitude, longitu
        ⊟ Attributes
            scale_factor    0.00171981661806480
            add_offset      263.466318436418
            _FillValue      -32767s
            missing_value   -32767s
            units           K
            long_name       2 metre temperature
        ⊞ Data values
    ⊞ msl
    ⊞ tp
    ⊞ v10u
    ⊞ v10v
⊟ Dimensions
    longitude               240
    latitude                121
    time                    9
```

Status: OK

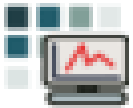
File: /home/graphics/cgr/metview/webinar_data/fc_surf.nc
Permissions: -rw-r----- **Owner:** cgr **Group:** graphics **Size:** 2.6MB **Modified:** 2

```
netcdf fc_surf {
dimensions:
        longitude = 240 ;
        latitude = 121 ;
        time = 9 ;
variables:
        float longitude(longitude) ;
                longitude:units = "degrees_east" ;
                longitude:long_name = "longitude" ;
        float latitude(latitude) ;
                latitude:units = "degrees_north" ;
                latitude:long_name = "latitude" ;
        int time(time) ;
                time:units = "hours since 1900-01-01 00:00:0.
                time:long_name = "time" ;
        short v2t(time, latitude, longitude) ;
                v2t:scale_factor = 0.00171981661806480 ;
                v2t:add_offset = 263.466318436418 ;
                v2t:_FillValue = -32767s ;
                v2t:missing_value = -32767s ;
                v2t:units = "K" ;
                v2t:long_name = "2 metre temperature" ;
        short msl(time, latitude, longitude) ;
                msl:scale_factor = 0.162427708177559 ;
```

Status: OK

NetCDF: How to plot it?

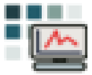
- NetCDF is so flexible it can contain almost any kind of data
- We need to use the NetCDF Visualiser icon



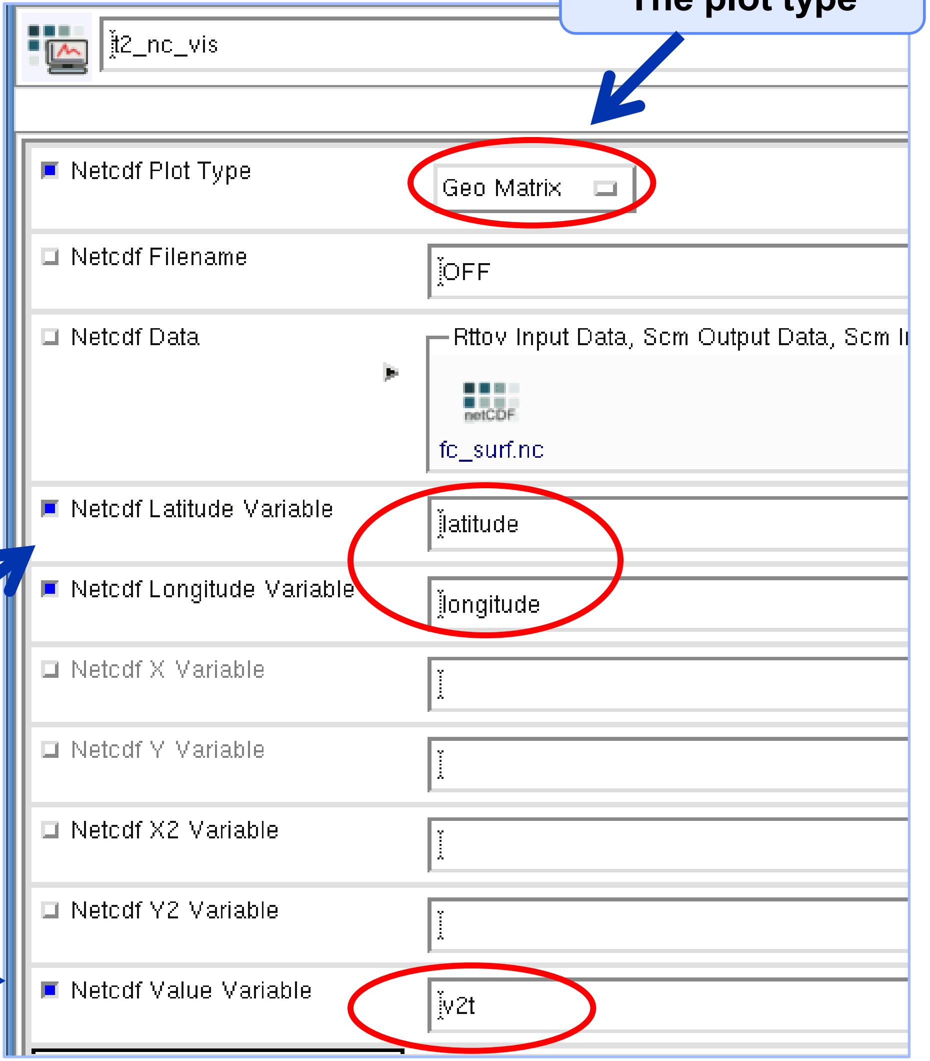
- It defines the way variables/dimensions are used for plotting

Plotting NetCDF data

Example:
plot T2 from file fc_surf.nc



t2_nc_vis



t2_nc_vis

■ Netcdf Plot Type: Geo Matrix

□ Netcdf Filename: OFF

□ Netcdf Data: Rttov Input Data, Scm Output Data, Scm I
fc_surf.nc

■ Netcdf Latitude Variable: latitude

■ Netcdf Longitude Variable: longitude

□ Netcdf X Variable:

□ Netcdf Y Variable:

□ Netcdf X2 Variable:

□ Netcdf Y2 Variable:

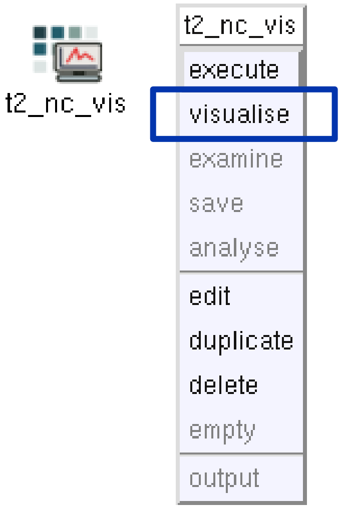
■ Netcdf Value Variable: v2t

The plot type

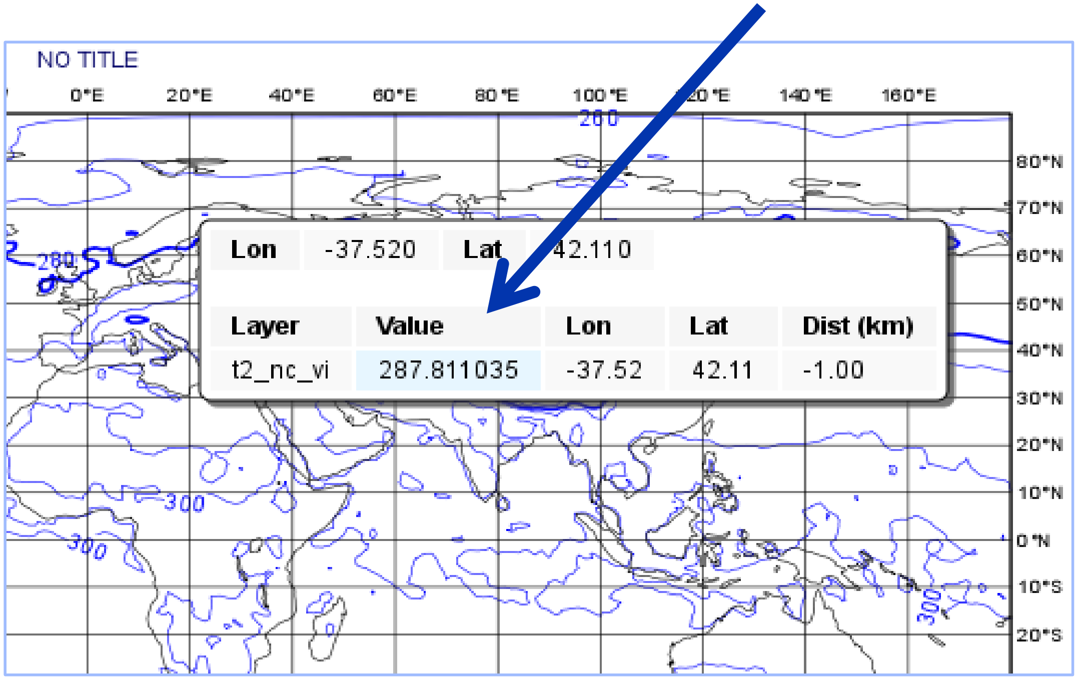
Latitude and longitude variables

The variable to plot (T2 is called **v2t** in our file)

NetCDF: Plotting



No scaling applies for NetCDF values: we have values in Kelvins



Lon	-37.520	Lat	42.110

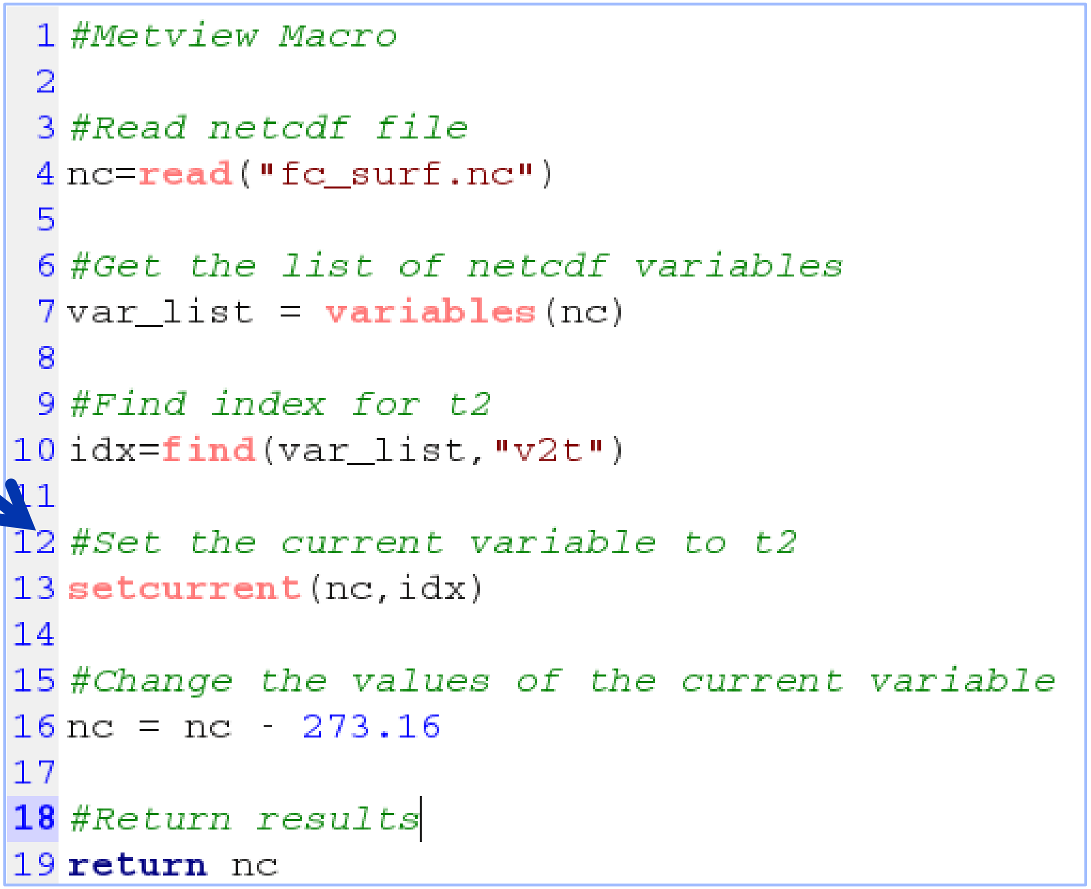
Layer	Value	Lon	Lat	Dist (km)
t2_nc_vi	287.811035	-37.52	42.11	-1.00

NetCDF: Macro Usage

Example: **convert values of T2 from Kelvin to Celsius**

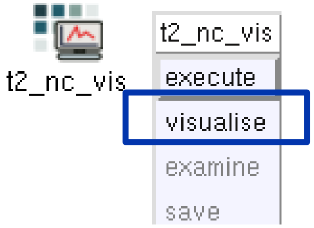
- **The NetCDF macro interface is based on the current variable concept: all operations are only valid to the currently selected NetCDF variable!**

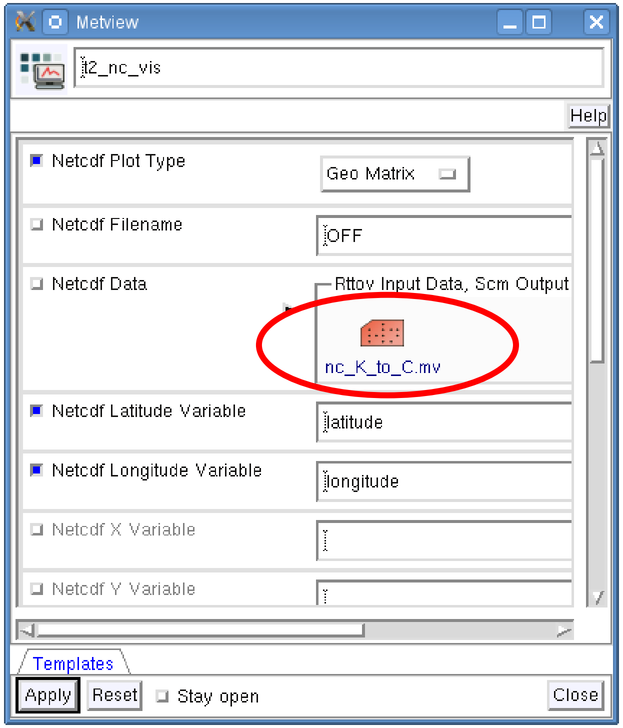
nc_K_to_C.mv



```
#Metview Macro

#Read netcdf file
nc=read("fc_surf.nc")

#Get the list of netcdf variables
var_list = variables(nc)

#Find index for t2
idx=find(var_list,"v2t")

#Set the current variable to t2
setcurrent(nc,idx)

#Change the values of the current variable
nc = nc - 273.16

#Return results
return nc
```

NetCDF: Plotting the modified data



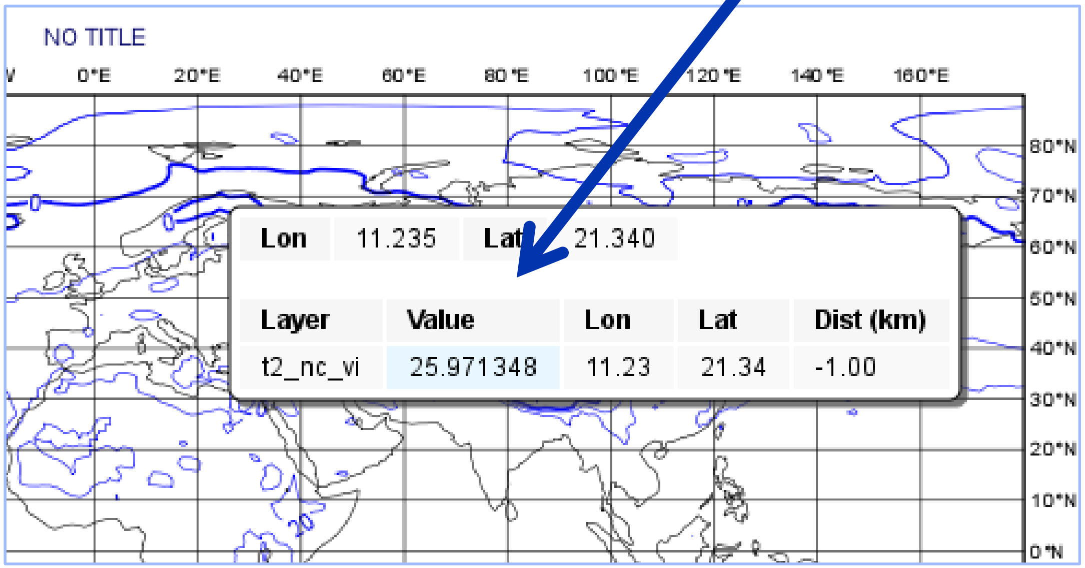


t2_nc_vis

- ■ Netcdf Plot Type: Geo Matrix
- □ Netcdf Filename: OFF
- □ Netcdf Data: Rttov Input Data, Scm Output — nc_K_to_C.mv
- ■ Netcdf Latitude Variable: latitude
- ■ Netcdf Longitude Variable: longitude
- □ Netcdf X Variable:
- □ Netcdf Y Variable:

□ Stay open

Now we have values in Celsius



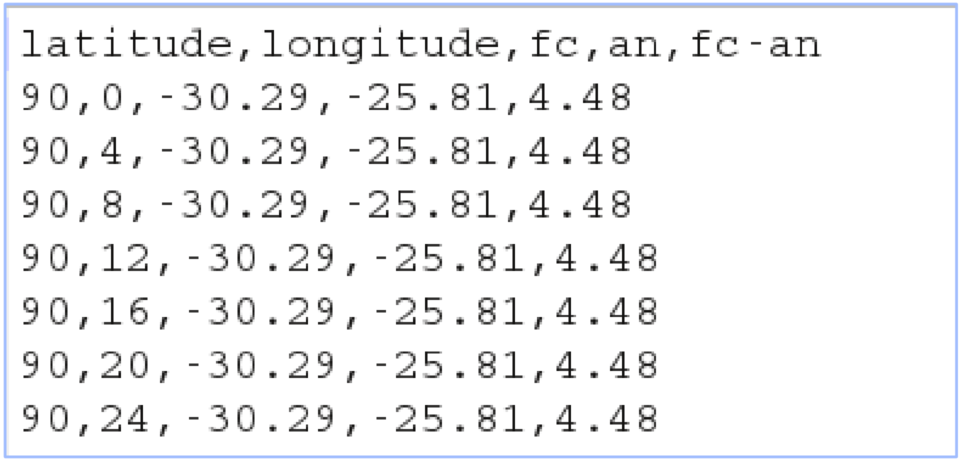
Lon	11.235	Lat	21.340

Layer	Value	Lon	Lat	Dist (km)
t2_nc_vi	25.971348	11.23	21.34	-1.00

ASCII Table Data

- ASCII file with data arranged with one variable per column
- Can contain a header
- CSV files can be handled as Table Data
- Geopoints files can be treated as Table Data as well

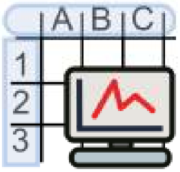
data.csv



```
latitude,longitude,fc,an,fc-an
90,0,-30.29,-25.81,4.48
90,4,-30.29,-25.81,4.48
90,8,-30.29,-25.81,4.48
90,12,-30.29,-25.81,4.48
90,16,-30.29,-25.81,4.48
90,20,-30.29,-25.81,4.48
90,24,-30.29,-25.81,4.48
```

Plotting Table Data

- Table Data plotting is based on the Table Visualiser icon



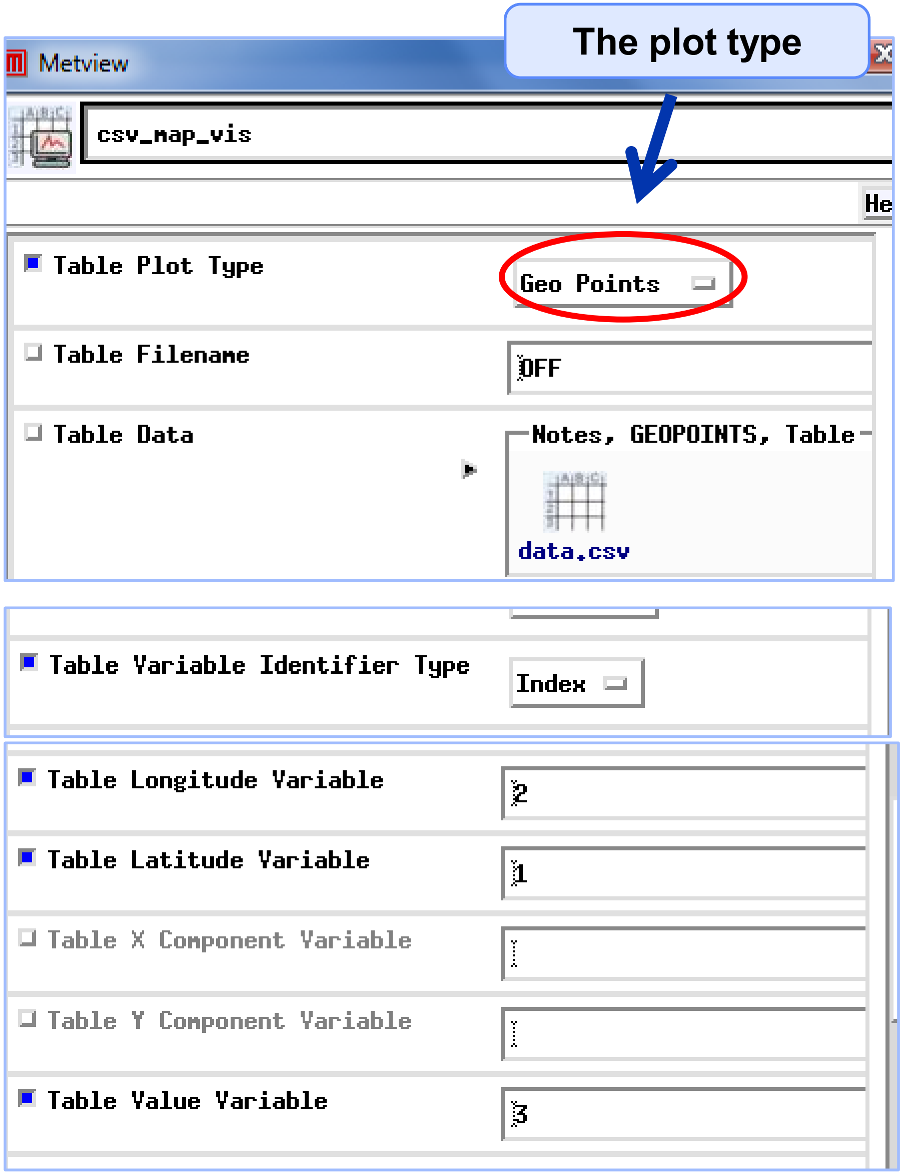
- It defines the way columns are used for plotting

Plotting Table Data

<u>Example:</u> plot the forecast values from file data.csv

csv_map_vis

The plot type



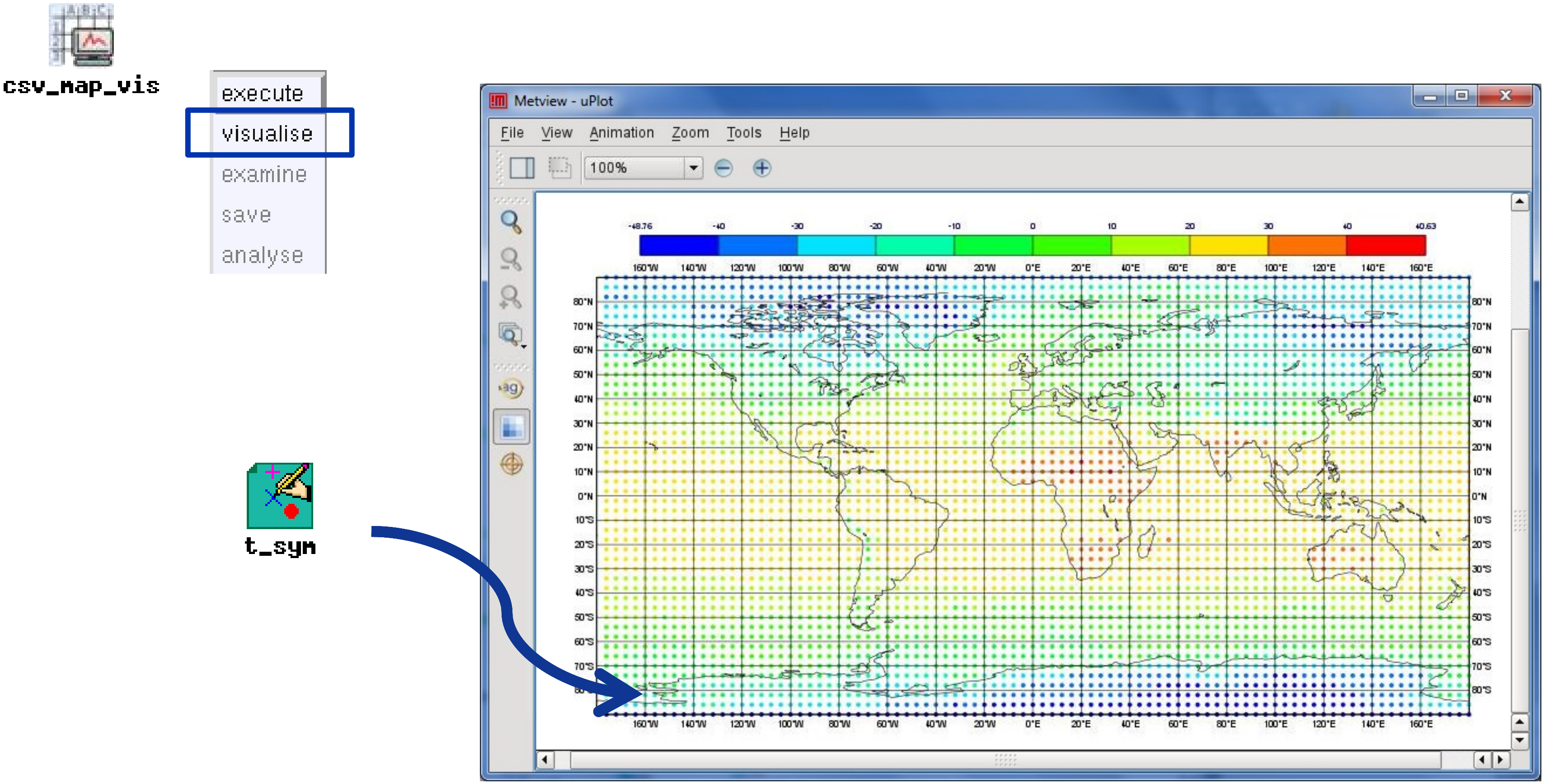
Metview

csv_map_vis

■ Table Plot Type	Geo Points
□ Table Filename	OFF
□ Table Data	Notes, GEOPOINTS, Table data.csv
■ Table Variable Identifier Type	Index
■ Table Longitude Variable	2
■ Table Latitude Variable	1
□ Table X Component Variable	
□ Table Y Component Variable	
■ Table Value Variable	3

We need to tell the visualiser which columns should be used from the file

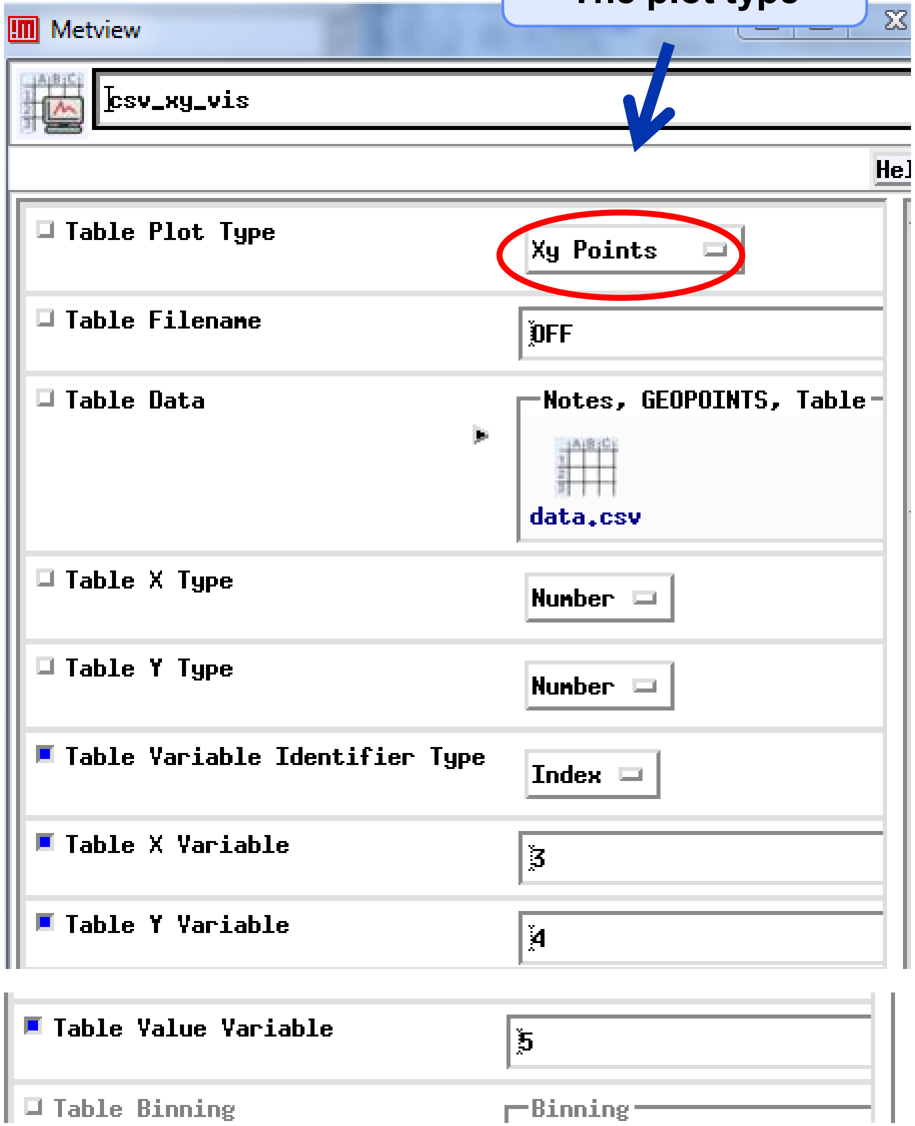
Plotting Table data


csv_map_vis
execute
visualise
examine
save
analyse
t_sym
Metview - uPlot
File View Animation Zoom Tools Help
100%
-48.76 -40 -30 -20 -10 0 10 20 30 40 40.63
160°W 140°W 120°W 100°W 80°W 60°W 40°W 20°W 0°E 20°E 40°E 60°E 80°E 100°E 120°E 140°E 160°E
80°N 70°N 60°N 50°N 40°N 30°N 20°N 10°N 0°N 10°S 20°S 30°S 40°S 50°S 60°S 70°S 80°S

Scatterplots from Table data

Example: generate a scatterplot from file data.csv with forecast in X axis and analysis in Y axis, and values (for colouring) taken from fc-an.

The plot type



Metview

csv_xy_vis

□ Table Plot Type: Xy Points

□ Table Filename: OFF

□ Table Data: Notes, GEOPOINTS, Table — data.csv

□ Table X Type: Number

□ Table Y Type: Number

■ Table Variable Identifier Type: Index

■ Table X Variable: 3

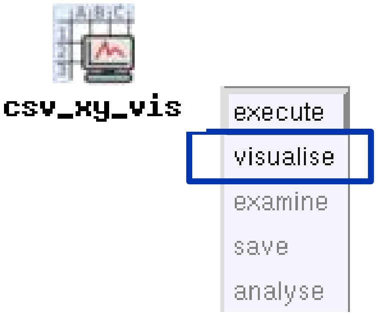
■ Table Y Variable: 4

■ Table Value Variable: 5

□ Table Binning: Binning

We need to tell the visualiser which columns should be used for X, Y and value

Scatterplots from Table data



t_diff

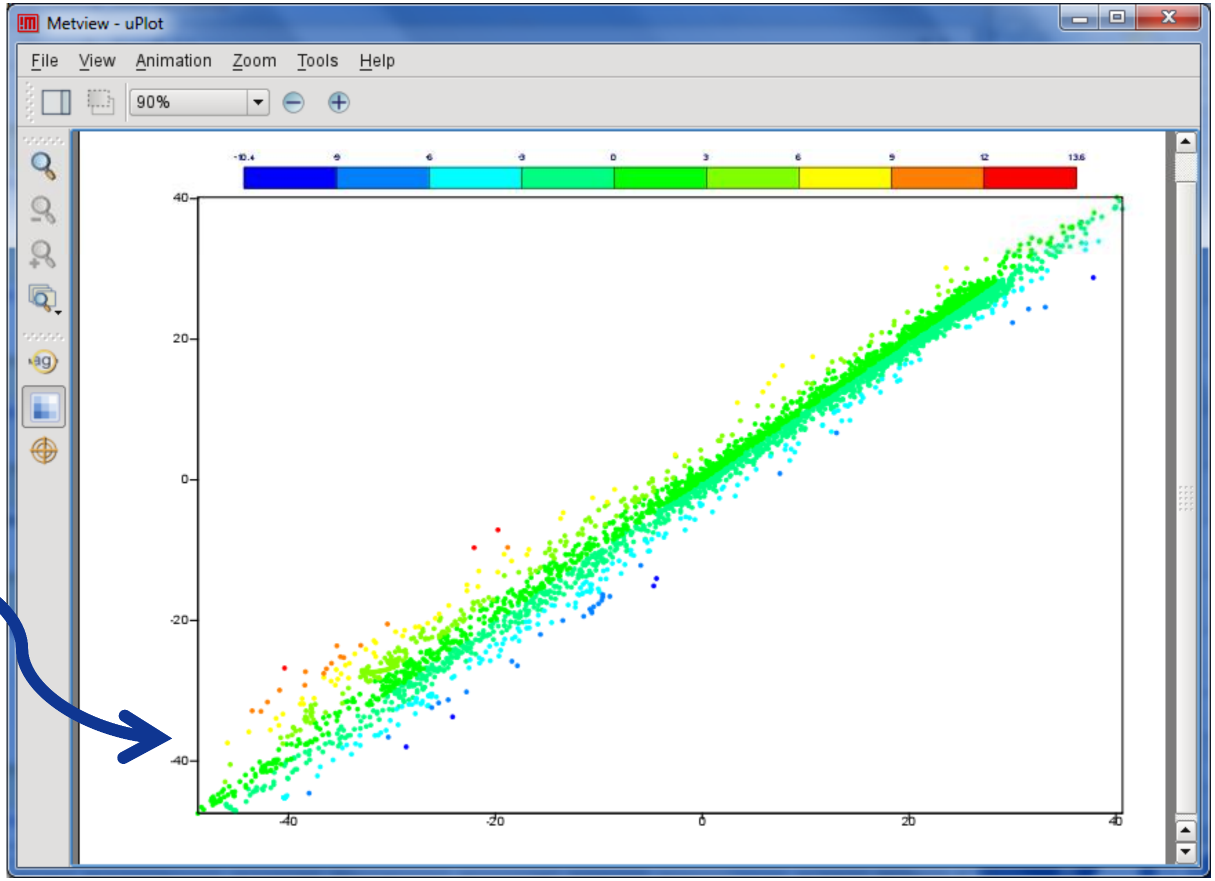
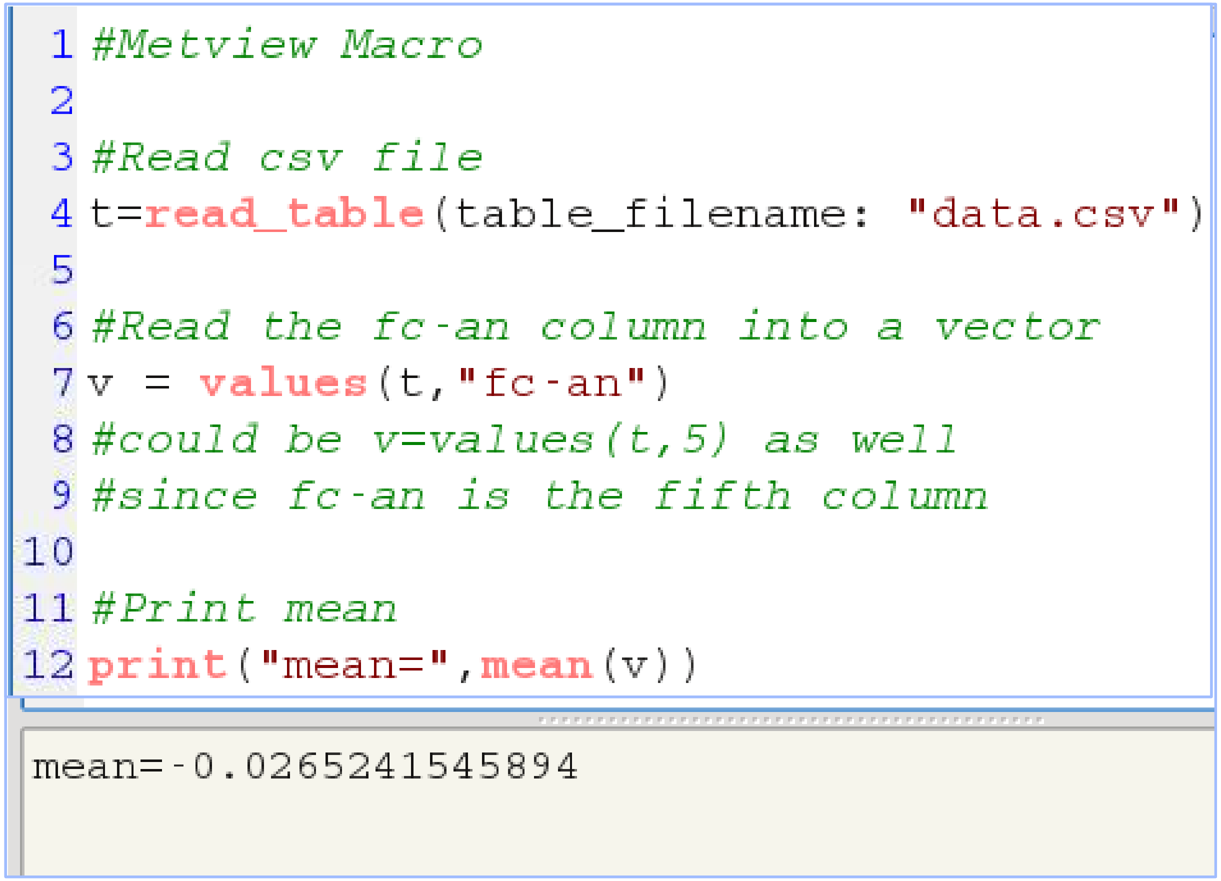


Table Data: macro usage

Example: compute the mean of the forecast-analysis values (5th column) from file data.csv



```
#Metview Macro

#Read csv file
t=read_table(table_filename: "data.csv")

#Read the fc-an column into a vector
v = values(t,"fc-an")
#could be v=values(t,5) as well
#since fc-an is the fifth column

#Print mean
print("mean=",mean(v))
```

The output of the macro →

```
mean=-0.0265241545894
```

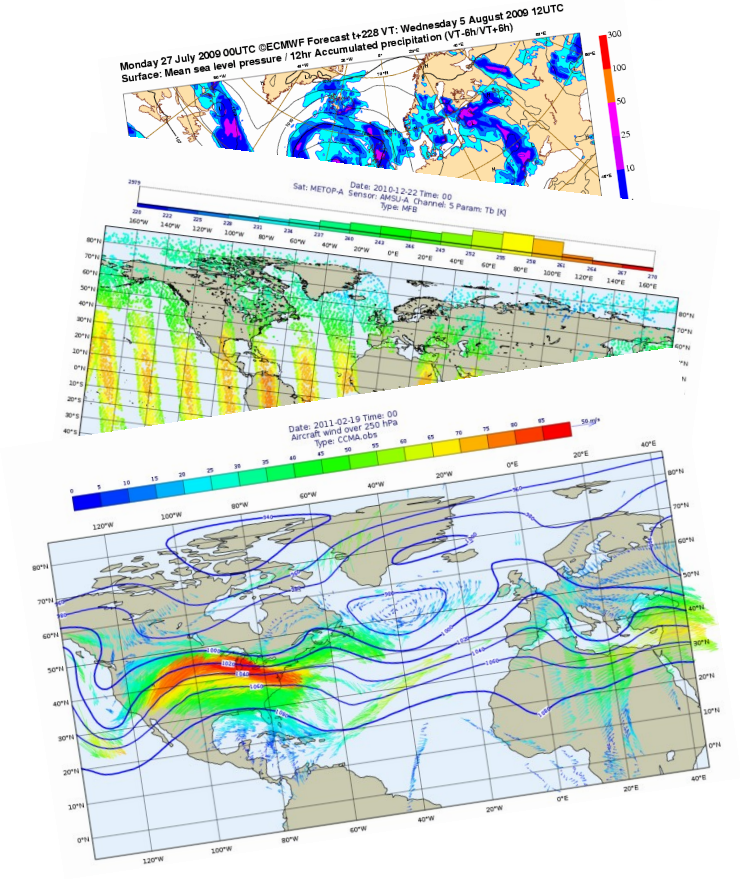
For more information …

email us:

- Metview: metview@ecmwf.int

visit our web pages:

- https://software.ecmwf.int/metview
- Documentation and tutorials
- Download the virtual machine


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

Thursday, 5th December, 9:30 AM UTC: Q&A

www.hipchat.com/gRuxxenlY