Migrating from Metview 3 to Metview 4

Introduction

The intention behind this page is not to advertise the new features of Metview 4 (see Complete Change History for this), but to help users migrate existing plots and macros from Metview 3 to Metview 4. In general, icons and macros which do no plotting should require very little, if any, modification. Many icons and macros used to produce plots will likewise require little or no modification, but some will need to be updated. These changes, plus some other points of interest are briefly described on this page.

System Settings, Macro libraries and Defaults

With the aim of allowing Metview 3 and Metview 4 to be run side-by-side, Metview 4 stores the system settings in a different folder. Everything which was previously stored in the `~/metview/Metview` folder is now stored in `~/metview/System`. This includes default icons, icon templates, the contents of icon drawers and Metview preferences. These will be initially set to the new defaults, so any icon drawers or defaults you have created yourself will not be there (although you can still copy your old icons across to the new folders). Note that any macros in your `Macros` folder will not be automatically copied over - these can be copied by hand.

Metview 3 icons greyed out

One of the aims of Metview 4 is to utilise the new features of Magics++. This requires some icons to be updated, specifically the visual definition icons such as *Contouring* and *Coastlines*. In order not to interfere with Metview 3's existing icons, Metview 4 defines a new set. These icons look the same as the old ones, but have some different parameters available. Metview 3's icons continue to work as normal (with some possible changes in behaviour), but they will appear slightly greyed out. The following screenshot shows this difference.

Metview 4 will not allow you to directly create new instances of the 'old' icons, but you will find the new ones in the desktop drawers and in the New Icon desktop menu.

From the point of view of macros, these icons have different icon-function names with the 'p' prefix replaced with an 'm'; instead of `pcoast`, we have `mcoas` , etc.

Although your old icons will still (mostly) work in Metview 4 (interactively and in Macro), you may wish to convert some of them to the new formats in order to use the new parameters available. As an example, suppose you have a *Contouring* icon from Metview 3 which you wish to convert into a Metview 4 *Cont ouring* icon. In Metview 4, first create a new contour icon. Edit this icon, and drop your old icon into the icon editor. The parameters which are valid in the new icon will be transferred; you can now click *Apply* to save the new icon.

Contouring

To make use of the new features of Magics, Metview 4 uses a new *Contouring* icon. One feature of Metview 3 which is not directly available in Metview 4 is Contour Splitting. This was a feature which enabled different isoline styles to be applied above, below and on a particular threshold (typically zero, for the purpose of examining the difference between two fields). To emulate this behaviour in Metview 4 requires either the use of the Rainbow contour settings, or the creation of three separate contour icons which can then be dropped together into the plot (or converted into three separate calls to `mcont`). See Migration: Split Contours for example Macro code and plots.

Automatic computation and plotting of **isotachs** is no longer supported in Magics. A small macro can be used to compute wind speed like this:

```
speed = sqrt(u*u + v*v) # where u and v are field(set)s of u and v wind components
plot(speed)
```

In Macro, `pcont` is replaced by `mcont`.

Coastlines

Metview 4's *Coastlines* icon has many more features than in Metview 3. One important thing to note is that in Metview 4 the coastline resolution has a much greater impact on the plot; a full-globe map with high resolution coastlines will be much too detailed, will take a long time to plot and will produce a very large (vector format) output file. The default resolution setting, *Automatic*, is intended to provide a reasonable balance between detail and efficiency. In Macro, `pcoast` is replaced by `mcoast`.

Map View

Metview's *Map View* has been replaced by *Geo View*, allowing access to the new projections of Metview 4. Note that there is now a new parameter with which you must specify whether you wish a global or non-global area (set **Map Area Definition** = **Corners** for a non-global area). Legacy *Map View* icons will still work.

Legends

Metview 4 has a new icon for handling legends - the *Legend* icon in the *Visual Definitions* drawer (Macro function `mlegend()`). This replaces the legend functionality that used to be in the *Text Plotting* and *Legend Entry* icons.

> Please note that because the Metview 3 *Text Plotting* icon (`ptext`) contains parameters for controlling the legend, using `ptext` disables the Metview 4 *Legend* (`mlegend`) functionality in order to avoid clashes. If you wish to use the new *Legend* functionality alongside text plotting specifications, you must use Metview 4's *Text Plotting* (`mtext`) icon instead of ptext.

Text Plotting

Metview 4 completely revised its handling of *Text, Annotation and Legend Plotting*. The mains differences are:

- **legend**: the *Text Plotting* icon no longer has any legend parameters - these are all now contained in the *Legend* icon.
- **font size**: Metview 3 (using MAGICS 6) automatically adjusted text font size in order to fit inside its containing box. A common technique was to define a large text size and let Metview scale it down automatically. Metview 4 (using Magics++) does not do this; it will use exactly the font size specified. Note that instead of *Text Reference Character Height*, Metview now uses *Text Font Size*.
- **user titles**: The way to specify user text in a title has changed a little. Metview 3 contained options to specify whether a title contained automatic text, user text or both; Metview 4 instead has a default title line which is `"<magics_title/>"`. Any text line with this string will have the automatic title; lines without it will not.
- **automatic titles**: whereas MAGICS 6 used to try to combine titles from different data onto a single line, Magics++ does not. So if a plot is the result of overlaying two GRIB fields, the automatic title will have two lines. The features mentioned below can be used to construct a semi-automatic title if this is not the desired behaviour.
- Magics++ also has features, documented here, to:
  - automatically add GRIB meta-data to a 'semi-automatic' plot title using GRIB_API keys. For instances:
    - where Metview 3 used `!PARAM!` Metview 4 would use `<grib_info key='name'/>`
    - provide two extra keys: base-date and valid-date
  - convert the date and time information according to a format string defined in C++ std::strftime (http://www.cplusplus.com/reference/ctime/strftime/)
- In Macro, *`ptext`* is replaced by *`mtext`*.

Examples:

```
str = "<grib_info key='base-date'/>  "
str = str & "<grib_info key='centre'/> t+<grib_info key='step'/>"
str = str & " VT:<grib_info key='valid-date'/>  "
str = str & "<grib_info key='level'/> hPa <grib_info key='name'/>"

title1 = mtext (
    text_line_count : 2,
    text_line_1     : "<magics_title/>",
    text_line_2     : str
                )

title2 = mtext (
    text_line_count : 4,
    text_line_1     : "<font colour='red' size='0.5'> My </font> <font colour='blue'> text </font>",
    text_line_2     : "<font colour='black' size='0.3'> <grib_info key='name'/> </font>",
    text_line_3     : "<grib_info key='base-date' format='%Y-%m-%d %H:%M:00'/>",
    text_line_4     : "Special C++ ctime characters: <grib_info key='valid-date' format='%A %B'/>"
                 )
```

See *Legends*, above, for information regarding the mixing of `ptext()` and `mlegend()`.

Text View, Empty View, Import View and Annotation

Metview 3 had an *Empty View* icon for reserving a blank area of a page. It also had a *Text View*, into which *Annotation* icons could be dropped. Metview 4 (from version 4.4 onwards) simplifies these functionalities by introducing a single view, the Annotation View, into which *Text Plotting* icons can be dropped. The *Import View* functionality will be available in the Annotation View in the future. An Empty View can be simulated by simply having an *Annotation View* w ith nothing in it. This means that the *Empty View*, *Text View*, *Import View* and *Annotation* icons from Metview 3 are no longer supported.

Curve Plotting

This is perhaps the area which has experienced the greatest change. Curve plotting is now achieved via one of the Visualiser icons. The *Curve View* is replaced by the more general-purpose *Cartesian View*. There is a Macro example which shows, side by side, Metview 3 and 4 code for plotting curves - Mi gration: Curve Plotting.

Axis Plotting

In Metview 3, the *Axis* icons determined the coordinate system for a non-geographical plot. In Metview 4, the *Cartesian View* icon defines this aspect of the plot, whereas the *Axis* icons only affect the plotting attributes of the axes. This is a cleaner concept, and more consistent with other View icons - the View icons define the projection and the Visual Definition icons determine the colour, line style, etc.

Symbol Plotting

The new *Symbol Plotting* icon provides access to a particularly useful feature of Magics - Advanced symbol plotting mode. This allows the automatic creation of colour scales in much the same way as is available for contouring. In Macro, `psymb` is replaced by `msymb`.

Graph Plotting

The new *Graph Plotting* icon is very similar to that in Metview 3. In Macro, `pgraph` is replaced by `mgraph`.

Vectors

Metview 3 had a *Vectors* icon which was used for combining two scalar fields and plotting them as a vector pair. It also allowed the use of a third field for the purpose of colouring the vector arrows. This functionality is now in a new icon, *Grib Vectors*.

Satellite View

Metview 4 no longer has a *Satellite View* icon, but something should be available in the future. For now, any satellite imagery should be reprojected to lat /long using Metview's Reprojection module.

ODB

Metview 4 completely revised its handling of ODB data and does not recognise any Metview 3 ODB icons. Consequently, such icons will not be visible in Metview 4. However, all the ODB functionality of Metview 3, plus a lot more, is available in Metview 4. Please see the Tutorials page for the ODB tutorial.

Tephigrams

The Tephigram icons have been replaced by the Thermo Data icon.

Overlay

In Metview 4, the default behaviour is to always overlay fields from different sources, even if their meta-data, such as date/time, do not match. This is different from Metview 3, whose default behaviour was to only overlay data whose timestamps 'match'. In Metview 3, there was a separate icon, the *Overlay Control* icon, which could be dropped into a View icon to change the overlay settings. In Metview 4 this is simpler, with a Map Overlay Control parameter directly in the *Geographical View* icon. See Data Overlay for more information.

Layer Ordering and Drawing Priority

Metview 3 used the *Drawing Priority* rules to determine which layers appeared on top of others (e.g. wind flags on top of shaded contouring). This was true even in the Macro `plot()` command, so that the order in which the layers were specified was not necessarily the order in which they were drawn. In Metview 4, the intention is that the `plot()` command honours the order in which the layers are specified, ignoring the old *Drawing Priority* rules - the first data layer supplied will appear as the bottom layer of the plot, etc. This is the case when using the new visdefs such as `mcont` and `mwind`, but when using the old ones such as `pcont` and `pwind`, the old *Drawing Priority* rules are applied as they were in Metview 3, allowing some backwards compatibility. This separation of behaviour breaks down when old and new visdefs are mixed in a single `plot()` command, e.g. `plot(t_data, pcont_def, wind_data, mwind_def)`. In this case, the two behaviours clash and the results might not be as desired. It is best to convert old visdefs to their new equivalents, and use the order in which they appear in the `plot()` command to determine the order in which they are rendered.

Cross Section Data, Average Data, Hovmoeller Data

These icons are currently restricted to returning a single frame (Metview 3 could, for instance, return a set of cross sections if there were multiple parameters or times in the input data). This feature is in the process of being implemented, and further documentation is available specifically for the 4.4 release of Metview, which contains further small revisions to the interface. See New Cross Section, Average, Vertical Profile and Hovmoeller modules in Metview 4.4.

Display Window Contents

The Display Window's *Contents Drawer* was an advanced feature which enabled the quick fine-tuning of a plot. This feature is not currently enabled in Metview 4, although it is planned to be in the future.

Macro

Metview's macro language now handles missing values in its data in a more consistent and useful way. Previously, functions such as `integrate()` returned a 'missing value indicator' if all its input values were missing. This was not easy to test for, and computations could use the result incorrectly without realising it. Now, all such functions return a `nil` variable when their inputs are invalid. Macros which do not test for this condition will fail if they try to use a `nil` variable in a computation. Example code:

```
a = integrate(data)
if (a = nil) then
  fail ('Integration failed')
end if
```

Another change in Macro is the expansion of the vector data type (see Vectors). Many Macro functions which previously returned lists of numbers will now return a vector (which is more efficient). This should mostly be transparent, since accessing a vector's elements is the same as for a list. However, user functions which explicitly test whether the returned variable is a list should be revised. See List of Operators and Functions.

Inline Fortran/C in Macro

Even within Metview 3, the GRIB_API-based inline Fortran/C routines (the 'MFI' routines) were available and preferred over the 'legacy' GRIBEX-based routines. Recent versions of Metview 4 have used newer versions of emoslib, which do not have GRIBEX support. It is thus now necessary to use the 'MFI' routines to pass parameters between Macro and inline C/Fortran. See Articles and find the PDF link at the top of the page for details on these interface routines.

Output Formats in Macro

Instead of a single `output` function, Metview 4 uses a set of functions to specify the desired output format, one for each type: `ps_output`, `png_output`, etc. It is possible to supply more than one of these to the `setoutput()` or the `plot()` commands. in order to obtain multiple formats in one go. One way to see the possible parameters for these functions is to type the appropriate function name in the Macro Editor and press F5 for an interactive help tool.

Limitations

There are some functionalities in Metview 3 which are not yet available in Metview 4, even with workarounds.

- The Metview-Vis5D interface has been replaced with an interface to the VAPOR software; see 3D visualisation with VAPOR

- Hovmoeller, cross section, zonal/meridional mean and vertical profile plots only generate a plot for the first set of data supplied. In Metview 3 for instance, you could supply a time series of vertical level data to the Cross Section module and receive one plot per time step; now, only the first time step will be plotted. The workaround is to write some Macro code to loop through the steps/parameters/etc and generate one plot at a time.
- The Satellite View is not available - the Geographical View allows the selection of the Geos projection, which simulates the view from a satellite; this is currently restricted to a sub-satellite point of zero degrees longitude, but this restriction may be lifted
- The Contents Drawer is not available from the Display Window

Spectra

Parameters *Title*, *Vertical Axis* and *Horizontal Axis* are no longer available in the user interface (Metview 4 compliant: they are not part of the data computation). If the icon action is *visualise* then default values for these parameters will be used. To customise the output title and axes, use icons Text Plotting and Cartesian View, respectively. Note that you might have to delete the existing *Spectra* icon from your *System/Defaults* folder if you cannot edit the new icon.