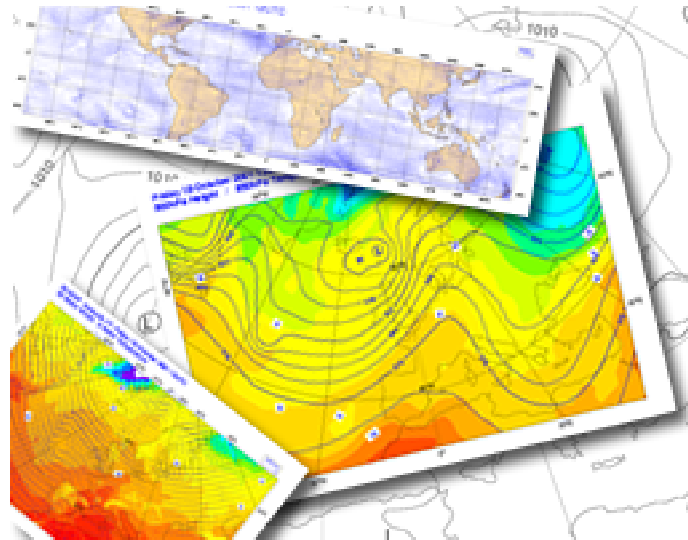


# Metview Developments



**Vesa Karhila**

# Contents

- **Metview background: past and present**
- **New applications**
- **Enhanced Macro Editor**
- **Macro library**
- **Metview availability and future**



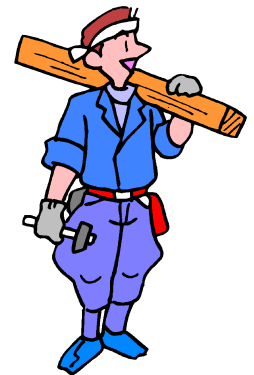
# Background

---

- **Developed at ECMWF**
  - ◆ **Co-operation w/ INPE (Brasil) and Meteo-France**
- **Open design**
  - ◆ **Easy to add new modules and features**
- **Uses ECMWF standard software:**
  - ◆ **MAGICS – graphics engine**
  - ◆ **MARS – database engine**
  - ◆ **EMOSLIB – data coding (GRIB, BUFR) and interpolation**
- **Metview Macro language**
  - ◆ **Powerful meteorologically oriented mature language**

# Current Developments

- **Mature software package**
  - ◆ **Continuous development**
    - Adapt to the changing environment
      - ◆ **New data types**
      - ◆ **New visualisation requirements**
      - ◆ **New computational requirements**
    - Add new features
    - Bug fixes
- **Runs on different platforms**
  - ◆ **Portability and installability**



# New Applications

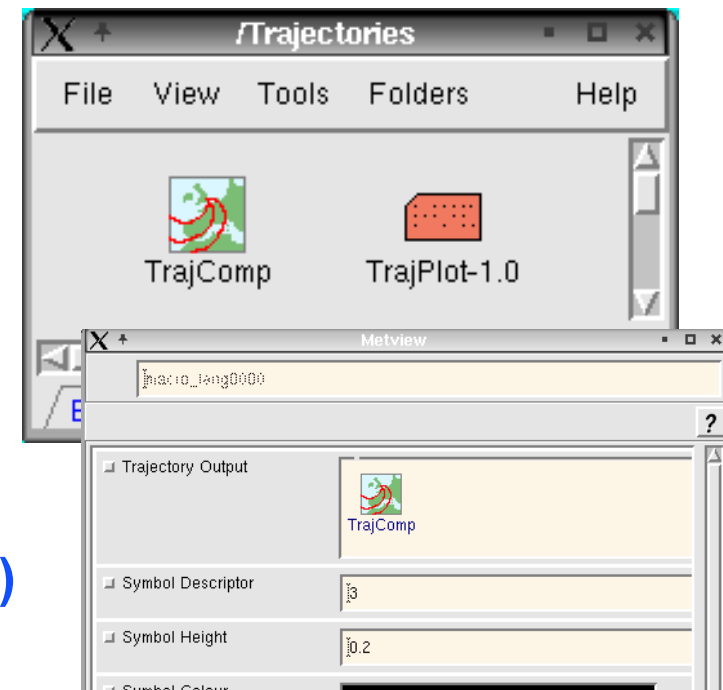
- **GRIB-to-Geopoints**

- ◆ Easy way to get grid point values and locations into ASCII file



- **Trajectory model**

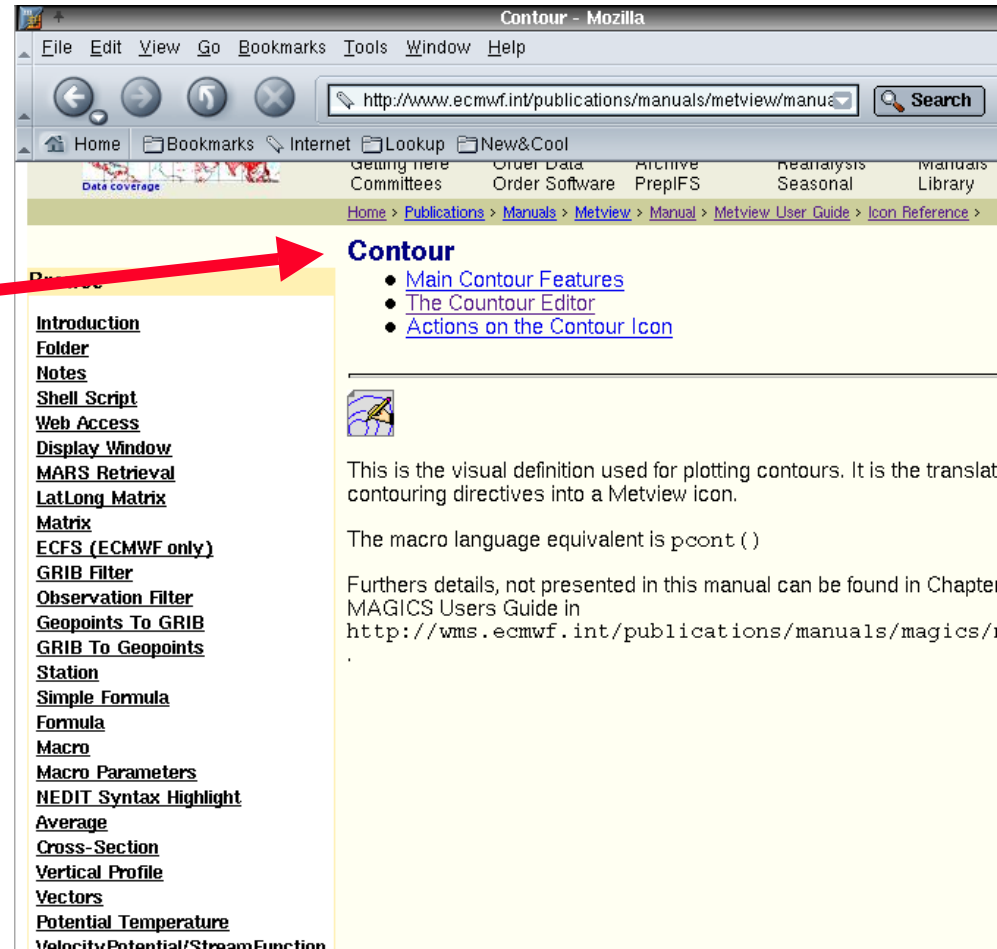
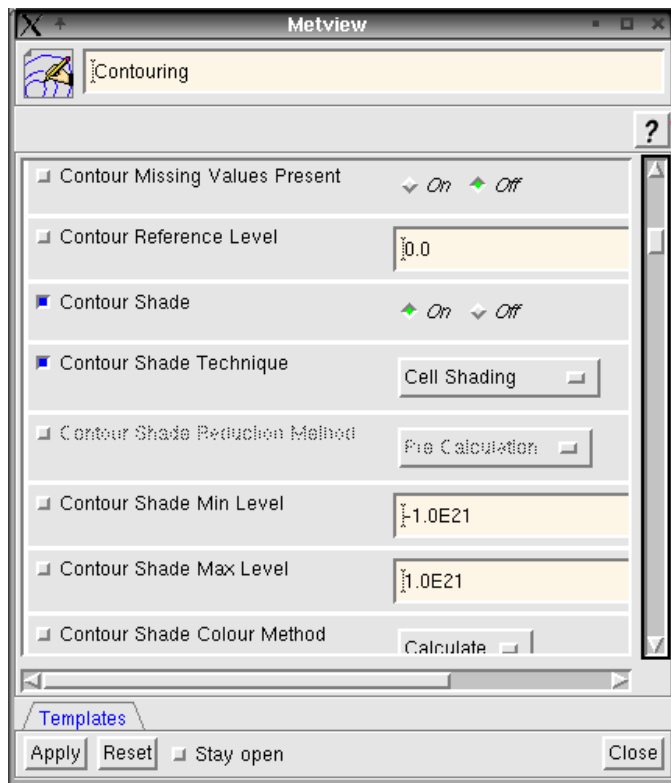
- ◆ Second generation, fully integrated
- ◆ Two icons
  - ➔ One to compute trajectories
  - ➔ One to visualise (a macro w/ UI)
- ◆ Real-time trajectory database available on *ecgate1* (-15...+10 days)
- ◆ Also from local data





# Help Connections

- Every editor has “?” button to access corresponding HTML manual page



# New Geographical Projections

- Older projections:

- ◆ Cylindrical Latitude-Longitude

- ◆ Polar Stereographic

- ◆ Mercator

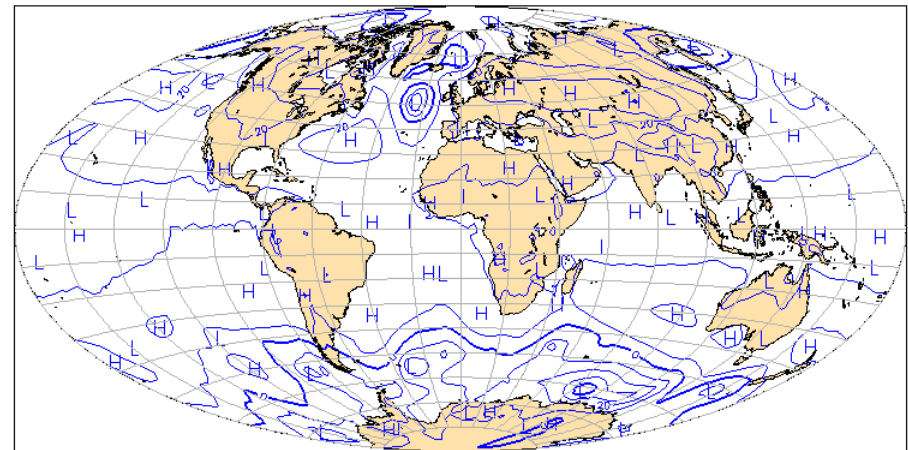
- Newer ones:

- ◆ “Ocean Cross Section”

- ◆ Lambert

- ◆ Aitoff

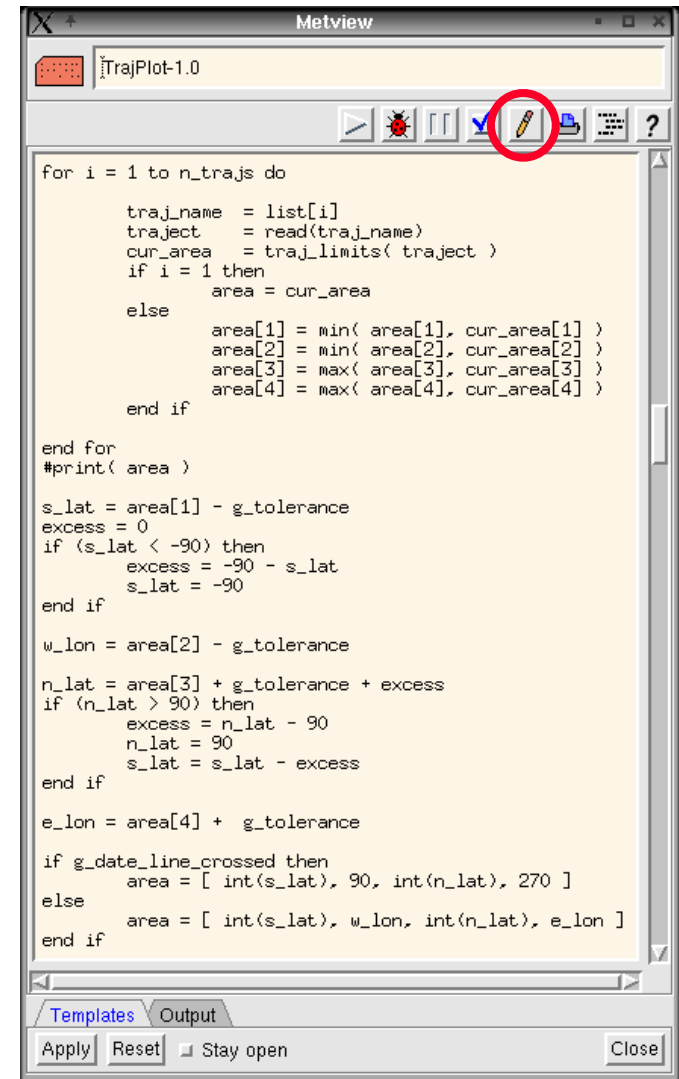
ECMWF Analysis VT:Thursday 6 November 2003 12UTC 1000hPa geopotential height





# Macro Editor(s)

- Based on a simple Motif widget
  - ◆ Limited editing functionality
    - ➔ Write and delete text
    - ➔ Copy – with mouse only
    - ➔ No line numbers
- Start an external editor
  - ◆ Editor defined by env.variable
  - ◆ Works with almost any text editor
    - ➔ vi, emacs, jot, ie, edit, nedit,...
- Some editors are easy to customise...



```
for i = 1 to n_trajs do
    traj_name = list[i]
    trajectory = read(traj_name)
    cur_area = traj_limits( trajectory )
    if i = 1 then
        area = cur_area
    else
        area[1] = min( area[1], cur_area[1] )
        area[2] = min( area[2], cur_area[2] )
        area[3] = max( area[3], cur_area[3] )
        area[4] = max( area[4], cur_area[4] )
    end if
end for
#print( area )

s_lat = area[1] - g_tolerance
excess = 0
if (s_lat < -90) then
    excess = -90 - s_lat
    s_lat = -90
end if

w_lon = area[2] - g_tolerance

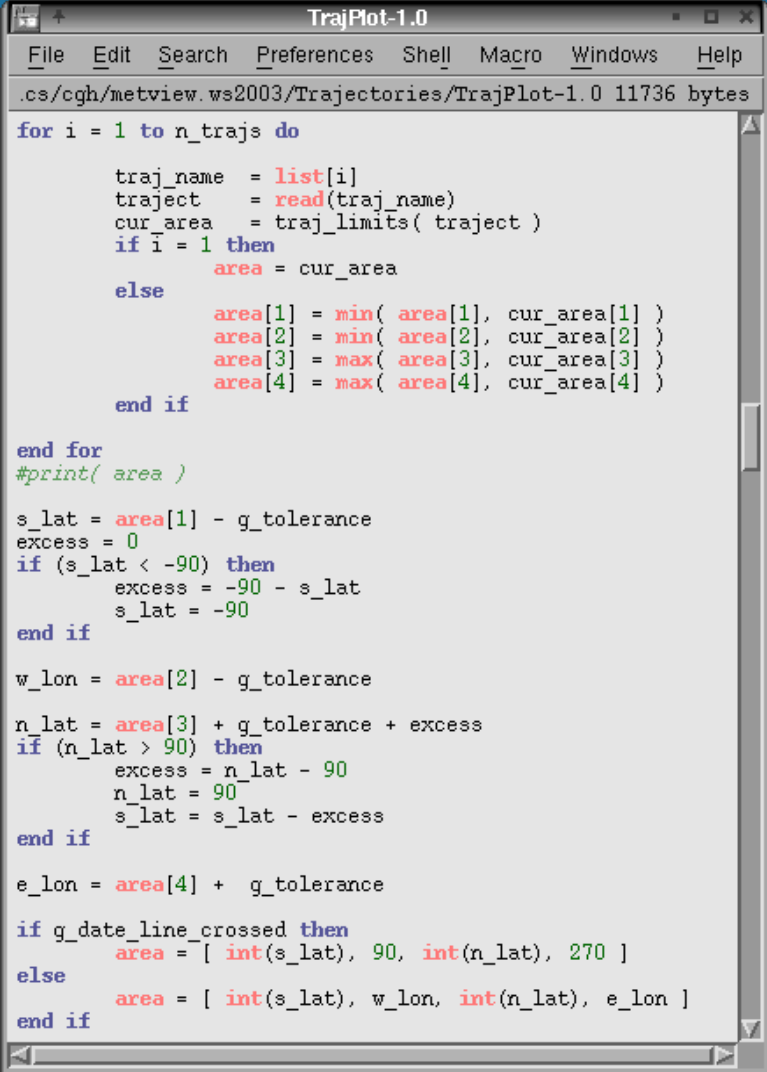
n_lat = area[3] + g_tolerance + excess
if (n_lat > 90) then
    excess = n_lat - 90
    n_lat = 90
    s_lat = s_lat - excess
end if

e_lon = area[4] + g_tolerance

if g_date_line_crossed then
    area = [ int(s_lat), 90, int(n_lat), 270 ]
else
    area = [ int(s_lat), w_lon, int(n_lat), e_lon ]
end if
```

# Enhanced External Macro Editor

- Based on *NEdit* (public domain)
- *NEdit* can be customised with a Metview Macro!
- Already implemented
  - ◆ Macro syntax highlighting
- We are working on ....



```
TrajPlot-1.0
File Edit Search Preferences Shell Macro Windows Help
.cs/cgh/metview.ws2003/Trajectories/TrajPlot-1.0 11736 bytes

for i = 1 to n_trajs do
    traj_name = list[i]
    traject   = read(traj_name)
    cur_area  = traj_limits( traject )
    if i = 1 then
        area = cur_area
    else
        area[1] = min( area[1], cur_area[1] )
        area[2] = min( area[2], cur_area[2] )
        area[3] = max( area[3], cur_area[3] )
        area[4] = max( area[4], cur_area[4] )
    end if
end for
#print( area )

s_lat = area[1] - g_tolerance
excess = 0
if (s_lat < -90) then
    excess = -90 - s_lat
    s_lat = -90
end if

w_lon = area[2] - g_tolerance

n_lat = area[3] + g_tolerance + excess
if (n_lat > 90) then
    excess = n_lat - 90
    n_lat = 90
    s_lat = s_lat - excess
end if

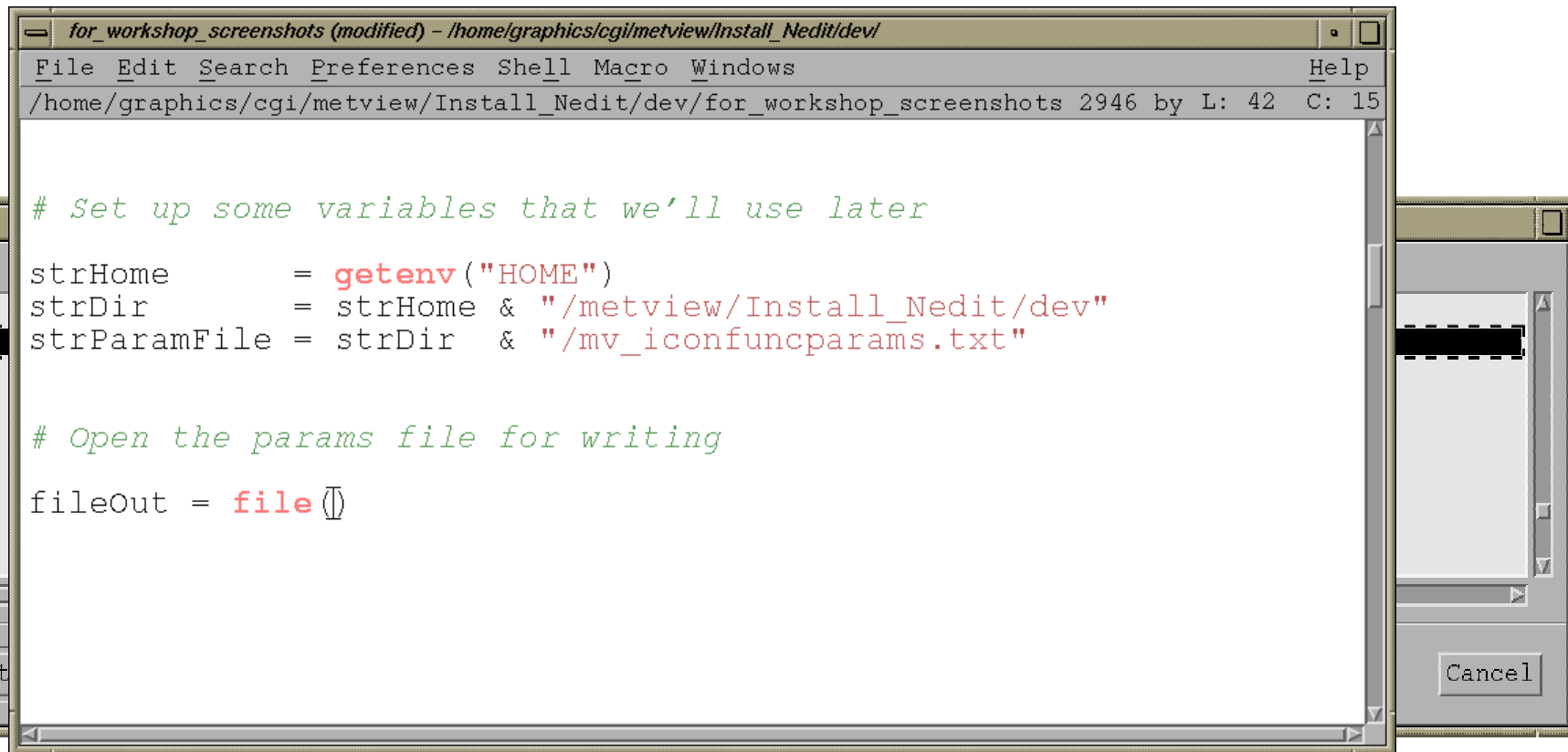
e_lon = area[4] + g_tolerance

if g_date_line_crossed then
    area = [ int(s_lat), 90, int(n_lat), 270 ]
else
    area = [ int(s_lat), w_lon, int(n_lat), e_lon ]
end if
```

# Macro Editor – Function Listing

- We are working on (I):

- ◆ Listing of all available library functions...



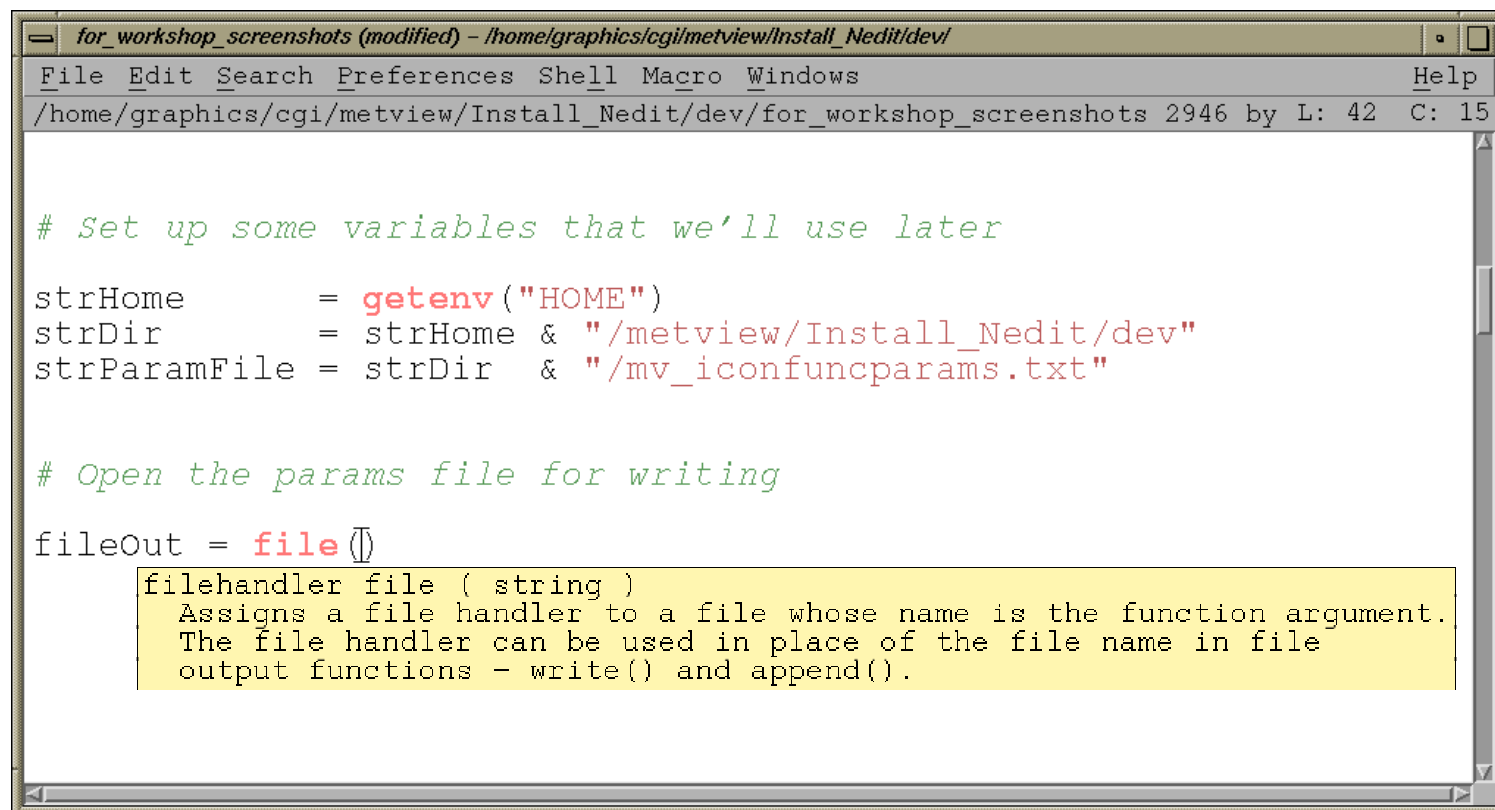
The screenshot shows a macro editor window with a menu bar (File, Edit, Search, Preferences, Shell, Macro, Windows, Help) and a title bar. The main text area contains the following code:

```
# Set up some variables that we'll use later  
strHome      = getenv("HOME")  
strDir       = strHome & "/metview/Install_Nedit/dev"  
strParamFile = strDir  & "/mv_iconfuncparams.txt"  
  
# Open the params file for writing  
fileOut = file {}
```

The code is color-coded: comments are green, function names like `getenv` and `file` are red, and strings are black. The window has a sidebar on the left with a file list and a 'Cancel' button on the right.

# Macro Editor - Calltips

- We are working on (II):
  - ◆ “Calltips” to describe function parameters and usage



The screenshot shows a macro editor window titled "for\_workshop\_screenshots (modified) - /home/graphics/cgi/metview/Install\_Nedit/dev/". The menu bar includes File, Edit, Search, Preferences, Shell, Macro, Windows, and Help. The status bar shows the file path, line number 2946, column 42, and character 15. The code in the editor is as follows:

```
# Set up some variables that we'll use later

strHome      = getenv("HOME")
strDir       = strHome & "/metview/Install_Nedit/dev"
strParamFile = strDir  & "/mv_iconfuncparams.txt"

# Open the params file for writing

fileOut = file()
```

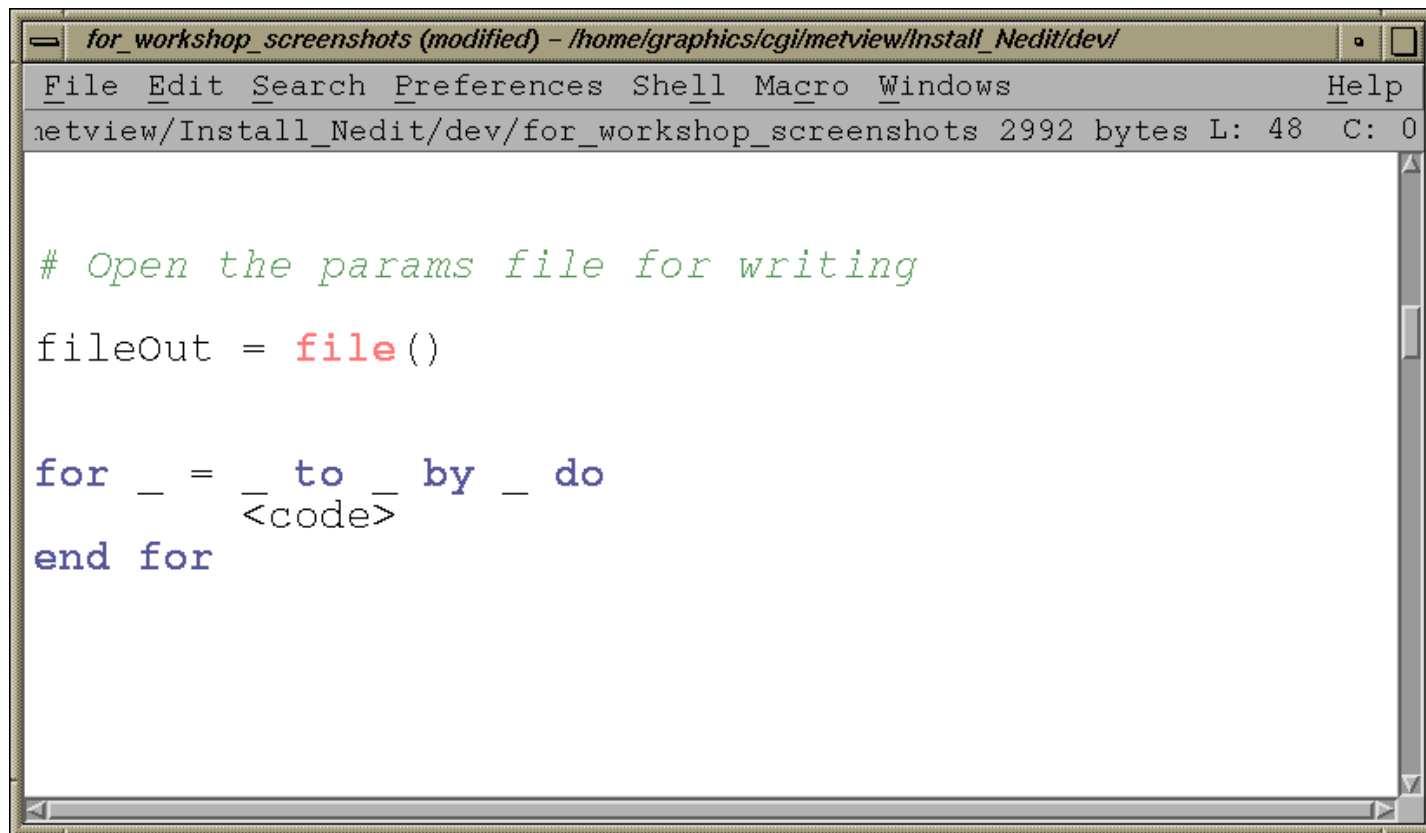
A calltip box is displayed over the `file()` function call, containing the following text:

```
filehandler file ( string )
Assigns a file handler to a file whose name is the function argument.
The file handler can be used in place of the file name in file
output functions - write() and append().
```

# Macro Editor – Code Templates

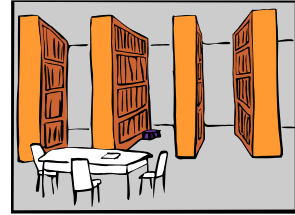
- We are working on (III):

- ◆ Code templates



```
for_workshop_screenshots (modified) - /home/graphics/cgi/metview/Install_Nedit/dev/  
File Edit Search Preferences Shell Macro Windows Help  
metview/Install_Nedit/dev/for_workshop_screenshots 2992 bytes L: 48 C: 0  
  
# Open the params file for writing  
fileOut = file()  
  
for _ = _ to _ by _ do  
    <code>  
end for
```

# Common Macro Library



- Why write a macro if somebody else has written it?
- Metview provides macro function library facilities
  - ◆ User level - private (Metview/Macros folder)
  - ◆ System level - common (shared directory)
- We are planning to expand our Common Macro Library
  - ◆ Full application macros
  - ◆ Macro functions
- New Enhanced Macro Editor will help accessing and using library macros
- Common Macro Library is distributed with Metview Export

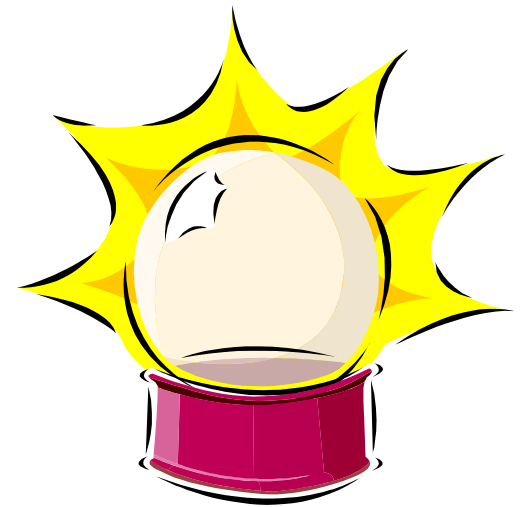
# Metview Availability

- Member State users can run Metview on *ecgate1*
- Metview can be installed and run locally
  - ◆ Installed in ~25 countries
  - ◆ Latest export version 3.5 released 1 October 2003
  - ◆ Runs under Unix operating system
- Platforms:
  - ◆ Linux, AIX/IBM, IRIX/SGI (used at ECMWF)
  - ◆ HPUX/HP, Solaris/Sun, OSF1/Alpha (tested at ECMWF)

# Future

---

- Finalise Macro editor and build up Macro library
- Keep up with changing environment
- Add new requested features
- “Compute-only” Metview for HPCF environment
- Use internally new Magics++
  
- Celebrate...





# Happy 10 Year Birthday !

- **Metview first release December 1993!**
- **PS: Metview demo on Thursday in Classroom**
- **PPS: Metview training course in March 2004**

