## **Debugging Metview**

## Setup

In order to debug Metview, you must first ensure that it has been built with debugging enabled:

- for Metview 4.5 and above: supply -DCMAKE\_BUILD\_TYPE=Debug to the cmake command line see the Installation Guide
- for Metview 4.4, supply the --enable-debug flag when running the configure script see the Installation Guide for Metview 4.4 and below

## Starting the debugger in the correct environment

When debugging a Metview module, the debugger must be started from the correct environment.

- If you are running Metview from the directory where it was built (not installed), then you must start Metview with the command

   metview -xserv
- when you start Metview like this, you will see the XServ bar. There is a button here called debug which brings up an xterm with Metview's environment in it. Go to section "Debugging a module".
- If you are running Metview from its installation directory, you can create a debug window as follows:
  - a. Create a new Shell Script icon (from the New Icon desktop menu)
  - b. Edit the icon and type 'xterm' (without the quotes) in it, and save
  - c. Right-click | execute the Shell icon

## Debugging a module

From a terminal which has the correct environment as described above, you can go to Metview's executable directory:

cd \$METVIEW\_BIN

Now, you can either just start up your chosen module in this xterm to see more output messages from it, or else you can start the debugger with it. Generally, the command-line arguments -nofork and -debug should be supplied. In the case of the **Macro** module, -serve should also be supplied. Example with totalview:

```
tv8 ./ObsFilter -a -nofork -debug
```

If the module does not start, then it might be because it is already running - from a Metview desktop, choose **Process Monitor** from the **Tools** menu and stop the process. If debugging the **Display Window**, then the corresponding executable is called uPlot; running this will create an empty **Display Window** into which you can drop icons. Note that you may have to kill the existing instance of the module if it has already been run during this Metview session - use Metview's Process Monitor for this.

Example - getting more output from the Observation Filter.

./ObsFilter -nofork -debug

Now run your Observation Filter task and look at the output in the xterm.